

 Download Skript

Informatik 7. Klasse - Schuljahr 2021/22

Lehrinhalte

- Lehrplaninhalte

Remote-Zugriff auf Schulserver

Kapitel

- 1) PHP
- 2) Netzwerke
- 3) Datenbanken
- 4) MS Office (Excel)
- 5) 3D Modellierung & 3D Druck
- 6) Bildbearbeitung
- 7) C++ Strukturen & Klassen
- 8) CMS Systeme

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

Stoff für die 1. Schularbeit in Informatik

06.12.2021 2+3.Stunde

- PHP

Stoff für die 2. Schularbeit in Informatik

???.???.2022 ???.Stunde

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



Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122

Last update: **2021/12/14 16:17**

Was wird in der 7. Klasse gemacht?

5. Semester

Sicherung der Nachhaltigkeit

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

Gesellschaftliche Aspekte der Informationstechnologie

Verantwortung, Datenschutz und Datensicherheit

- 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)

- Aktuelle Entwicklungen auf dem Hardwaresektor kennen
- Mit Hardware-Komponenten praktisch umgehen können

Virtualisierung

- Virtualisierungs-Software für Betriebssysteme und Server einsetzen können
- Betriebssysteme und Software virtualisieren können

Betriebssysteme (Linux)

- Die Arbeitsumgebung (Oberflächen, Tools, Script-Programmierung) von Linux einrichten können
- Anwenderprogramme unter Linux installieren und verwenden können
- Die Bedeutung von Linux als Betriebssystem für Internetdienst kennen
- Im LAN und im Internet mit Linux arbeiten können

Netzwerke

- Grundlagen der Vernetzung von Computern beschreiben und lokale und globale Computernetzwerke nutzen können.
- Netz-Topologien kennen
- Den Datenverkehr in Netzen simulieren und analysieren können
- Netzhardware kennen und praktisch damit umgehen können
- Protokolle und Adressierung kennen und anwenden können
- Sicherheitskonzepte für Netzwerke kennen
- Netztypen (Peer-to-Peer, Server-Client) kennen
- Peer-to-Peer und Server-Client-Netz installieren können
- Einfache Netzadministration durchführen können
- Programme für die Verwendung im Netz installieren können

Angewandte Informatik, Datenbanksysteme und Internet

Suche, Auswahl und Organisation von Information

- Informationsquellen erschließen, Inhalte systematisieren, strukturieren, bewerten, verarbeiten und unterschiedliche Informationsdarstellungen verwenden können.

Datenmodelle und Datenbanksysteme

- Datenbanken durch ein Programm ansteuern können
- Datenbanken benutzen und einfache Datenmodelle entwerfen können.
- Die Charakteristika von SQL kennen
- PHP-Scripts zur Anbindung einer Webseite an eine MySQL-Datenbank erstellen können

6. Semester

Sicherung der Nachhaltigkeit

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

Angewandte Informatik, Datenbanksysteme und Internet

Web-Techniken

- PHP-Scripts zur Anbindung einer Webseite an eine MySQL-Datenbank erstellen können
- MySQL-Datenbank mittels PHP verwalten können

Gesellschaftliche Aspekte der Informationstechnologie

Verantwortung, Datenschutz und Datensicherheit

- Kommunikation mit einem externen Server herstellen können
- Die Bedeutung der Absicherung extern verwalteter Daten einschätzen können

Algorithmik und Programmierung

Algorithmen und Datenstrukturen

- Algorithmen erklären, entwerfen, darstellen können.
- Klassen und Objekte kennen und verwenden können
- Files kennen und einsetzen können
- Datenstrukturen Zeiger und Listen kennen und visualisieren können
- Algorithmen mit Zeigern und Listen erstellen können

Programmierung (Objektorientierte visuelle Programmiersprache)

- Weitere Komponenten der visuellen Programmierung kennen und anwenden können
- Mit einer visuellen Programmiersprache auf fortgeschrittenem Niveau arbeiten können
- Mit Files arbeiten können
- Graphikelemente einsetzen können
- Algorithmen in der visuellen Programmiersprache implementieren können
- Algorithmen mit Zeigern und Listen programmieren können

7. Klasse (3 Stunden, je eine 2h Schularbeit pro Semester)

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



Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:0_lehrplaninhalte

Last update: **2021/09/16 08:33**

1. PHP

1.1) Was ist PHP?

- 1.1.1) Was ist PHP
- 1.1.2) How to use PHP

1.2) Grundlegende Sprachelemente

- 1.2.1) Kommentare
- 1.2.2) Ausgabe
- 1.2.3) ÜBUNG 1

1.3) Variablen und Operatoren

- 1.3.1) Variablen und Operatoren

1.4) Interaktive Webseiten

- 1.4.1) Formulare
- 1.4.2) ÜBUNG 2

1.5) Anführungszeichen

- 1.5.1) Verwenden von Anführungszeichen

1.6) Kontrollstrukturen

- 1.6.1) Vergleichsoperatoren
- 1.6.2) if-Anweisung
- 1.6.3) switch-Anweisung
- 1.6.4) ÜBUNG 3-7

1.7) Schleifen

- 1.7.1) for-Schleife
- 1.7.2) while-Schleife
- 1.7.3) do-while-Schleife
- 1.7.4) foreach-Schleife

- 1.7.5) ÜBUNG 8-11

1.8) Felder

- 1.8.1) Grundlagen zu Felder
- 1.8.2) Indizierte Felder
- 1.8.3) Assoziative Felder
- 1.8.4) Mehrdimensionale Felder
- 1.8.5) Weitere Informationen zu Feldern
- 1.8.6) ÜBUNG 12-18
- 1.8.7) KONTROLLFRAGEN

1.9) Formulare

- 1.9.1) Hidden-Feld
- 1.9.2) Textarea
- 1.9.3) Checkbox
- 1.9.4) Radio-Button
- 1.9.5) Auswahllisten
- 1.9.6) ÜBUNG 20

1.10) Externe Dateien

- 1.10.1) Theorie: Nutzung von externen Dateien
- 1.10.2) Dateien öffnen, lesen und schließen
- 1.10.3) Aus Dateien lesen
- 1.10.4) In Dateien schreiben
- 1.10.5) ÜBUNG 21 Gästebuch
- 1.10.6) ÜBUNG 22 Besucher zählen
- 1.10.7) ÜBUNG 23-25

1.11) Datum und Zeit

- 1.13.1) Datum und Zeit
- 1.13.2) ÜBUNG 26

1.12) Funktionen

- 1.12.1) Funktionen erstellen und aufrufen
- 1.12.2) Funktionen verwenden
- 1.12.3) Gültigkeitsbereich von Variablen
- 1.12.4) PHP-Dateien einbinden

- 1.12.5) Andere Dateitypen einbinden
- 1.12.6) ÜBUNG 27-29

1.13) Sessions

- 1.13.1) Grundlagen zu Sessions
- 1.13.2) Registrieren von Sessions
- 1.13.3) Session löschen
- 1.13.4) Session Beispiel
- 1.13.5) ÜBUNG 30

1.14) Cookies

- 1.14.1) Grundlagen
- 1.14.2) Cookies "backen"/setzen
- 1.14.3) Cookies auslesen
- 1.14.4) Cookies löschen
- 1.14.5) Sicherheit von Cookies
- 1.14.6) ÜBUNG 31-36

From:

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



Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1

Last update: **2021/11/17 08:57**

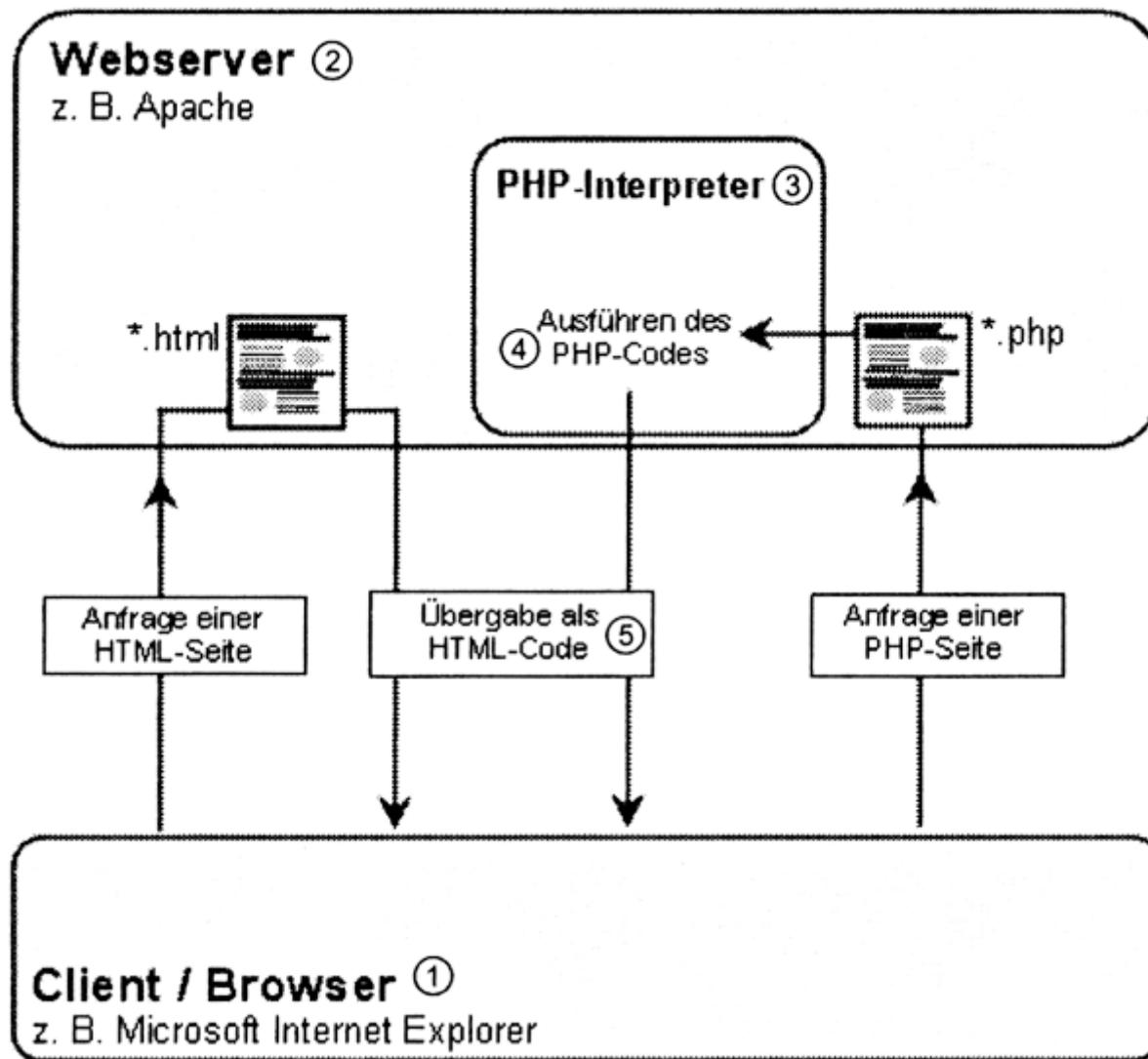
1.1.1) Was ist PHP?

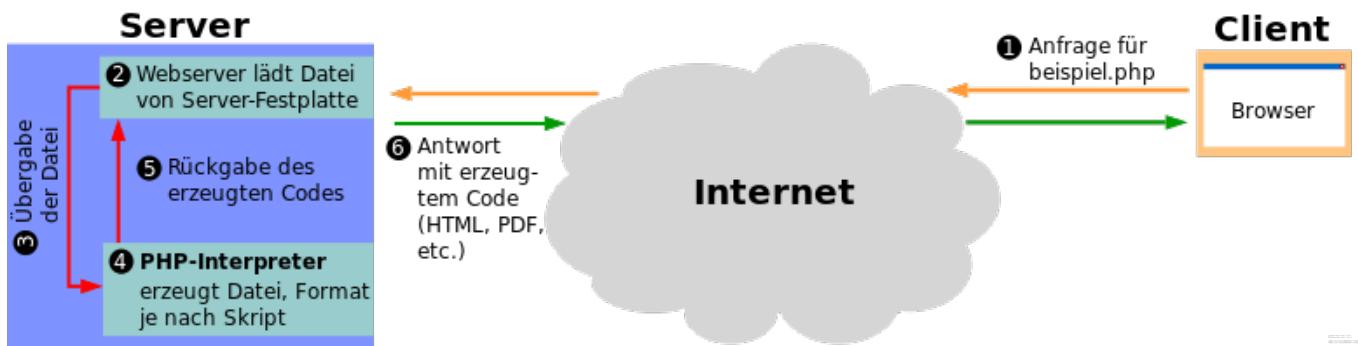
PHP (= PHP Hypertext Preprocessor) ... **Skriptsprache** für Internet, wird am **Webserver** ausgeführt.

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.



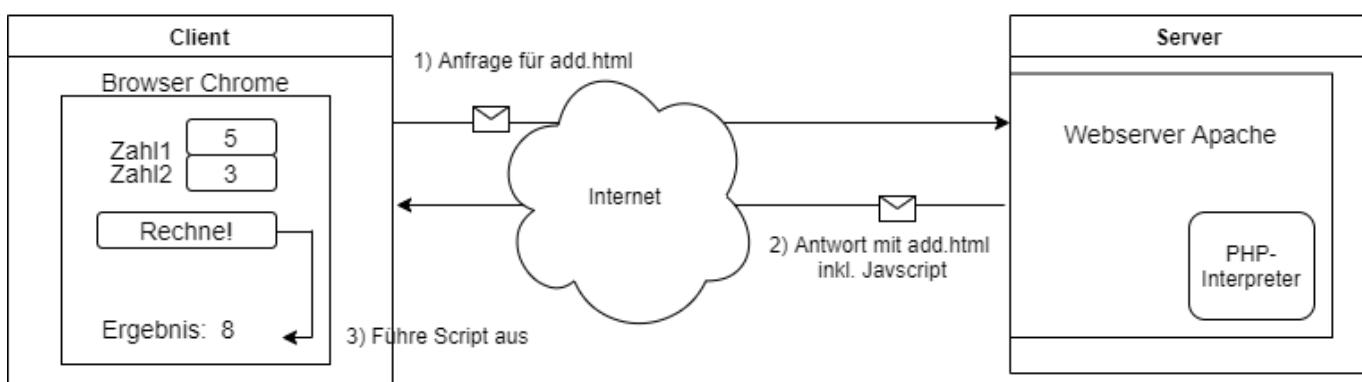


Um dynamische Webseiten zu erstellen und später auf Datenbanken zugreifen zu können, benötigt man neben PHP einen Webserver und einen Datenbankserver. Unter Windows kann man einen lokalen Webserver installieren. Als Entwicklungssoftware hat sich XAMPP sehr bewährt.

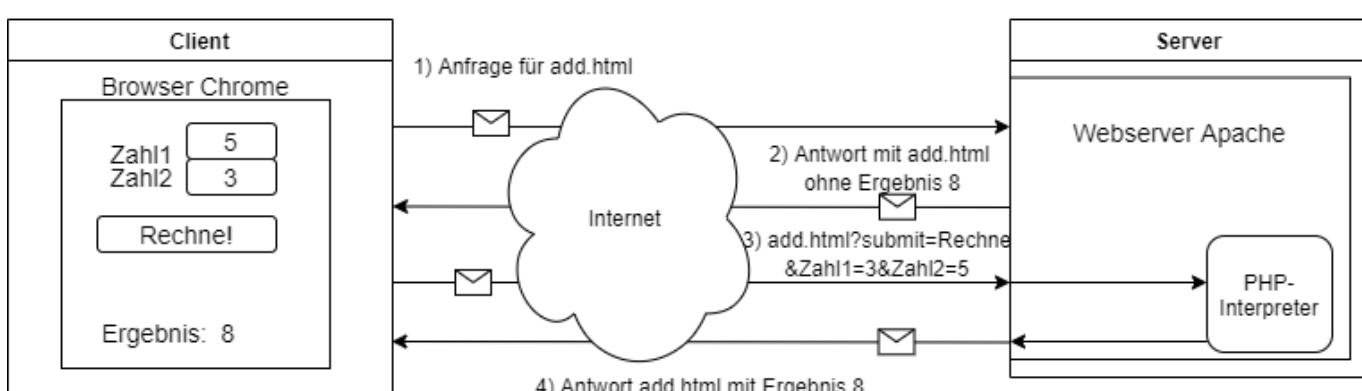
Unterschied zu Javascript

Im Gegensatz zu Javascript benötigt PHP einen Webserver, der eine Interaktion mit dem Benutzer ermöglicht und somit eine Webseite dynamisch erscheinen lässt.

Javascript



PHP



Server-seitiges vs. Client-seitiges Skripting

PHP ist eine serverseitige Skriptsprache. Das bedeutet, dass es auf dem Webserver läuft, im Gegensatz zu einem Client-Rechner.

Serverseitige Programmierung ist nützlich, um Benutzern dynamische Inhalte (typischerweise aus einer Datenbank) zu liefern, wie z.B. eine Willkommensnachricht („Hi, Claire!“), wenn sich ein Benutzer anmeldet.

Noch wichtiger ist, dass serverseitiges Scripting im E-Commerce eingesetzt wird. Zum Beispiel gibt es über 100 WooCommerce-Erweiterungen, die sich über APIs (Application Programming Interfaces) mit verschiedenen Zahlungsanbietern verbinden, um Transaktionen zu verarbeiten.

JavaScript ist eine clientseitige Sprache, also läuft es auf dem Laptop, dem Telefon oder dem Tablet des Nutzers.

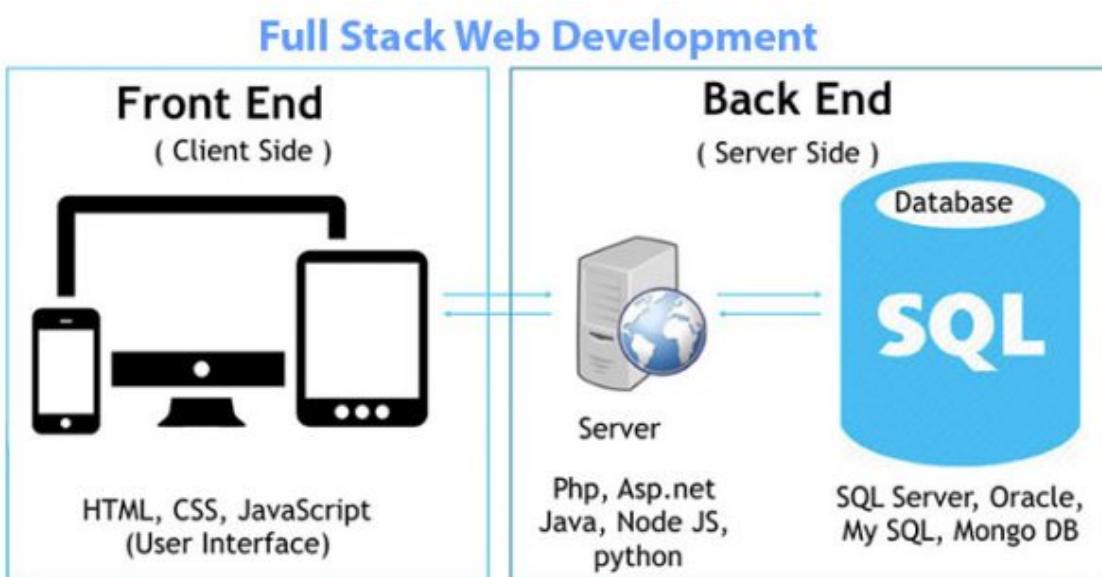
JavaScript kann das DOM manipulieren, was für Document Object Model steht und man kann es sich als eine baumartige Struktur vorstellen, die aus dem HTML einer Webseite gebildet wird.

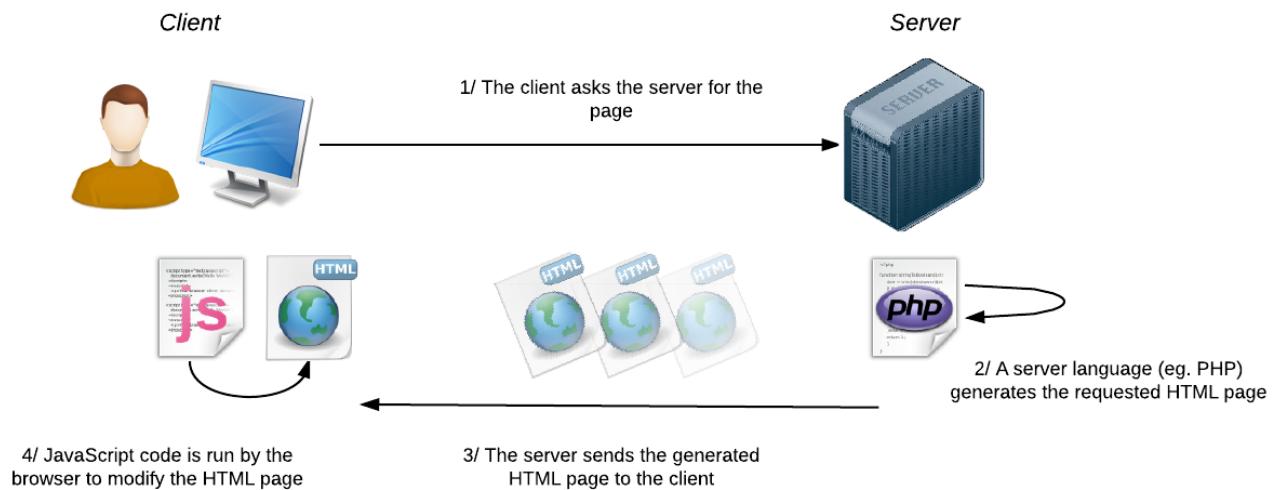
Wenn du jemals auf ein Akkordeon oder einen Toggle gestoßen bist, vielleicht als Teil eines FAQ Plugins, hast du clientseitiges JavaScript in Aktion gesehen. Wenn du auf eine Frage klickst oder tipps, schalten JavaScript-Event-Handler die CSS-Anzeige- oder Sichtbarkeits-Eigenschaften ein oder aus und zeigen oder verstecken die entsprechende Antwort.

Frontend vs Backend

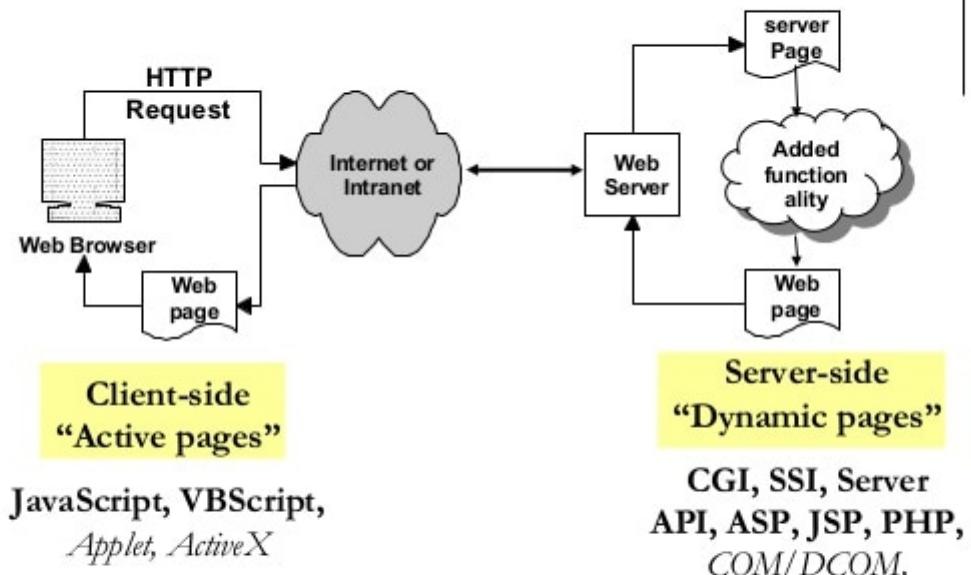
PHP läuft im Backend einer Webseite – dem Teil, den Besucher nicht sehen! In WordPress bedeutet das, dass PHP seine ganze Arbeit auf dem Webserver und im WordPress-Admin macht.

JavaScript lief traditionell im Frontend, aber das änderte sich 2009, als Node.js, eine Backend-Runtime, eingeführt wurde. Heute ist JavaScript wirklich eine Full-Stack-Sprache.





Where does PHP fit ?



Active and dynamic page technology can be used together – server-side program generates customized active pages.

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:inf7bi8bi_202122:1:1_01

Last update: **2021/12/04 13:29**

1.1.2) 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 Dokumentrumpf <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:inf7bi8bi_202122:1:1_02

Last update: **2021/09/16 08:05**

1.2.1) 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 Kommentar

    /*
        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:inf7bi8bi_202122:1:1_03



Last update: **2021/09/16 08:05**

1.2.2) 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.

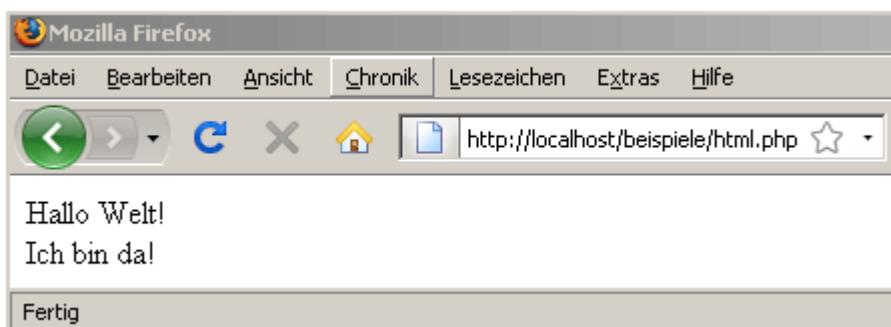
```
echo "Hallo Welt";
```

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.
) 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
 richtig interpretiert. Der Browser fügt eine Leerzeile ein. Der Kommentar wird natürlich nicht angezeigt!

Quelltext HTML



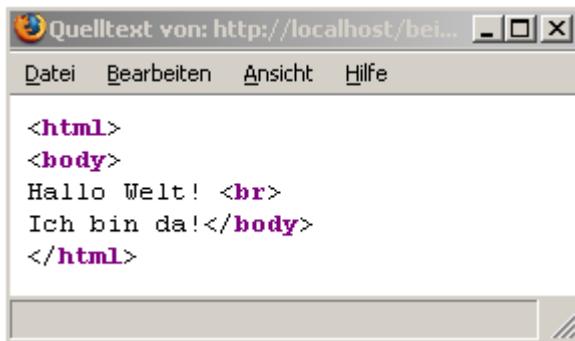
```
<html>
<body>
Hallo Welt! <br>Ich bin da!
</body>
</html>
```

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:



```
<html>
<body>
Hallo Welt! <br>
Ich bin da!
</body>
</html>
```

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:inf7bi8bi_202122:1:1_04



Last update: **2021/09/16 08:05**

1.2.3) 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:inf7bi8bi_202122:1:1_05

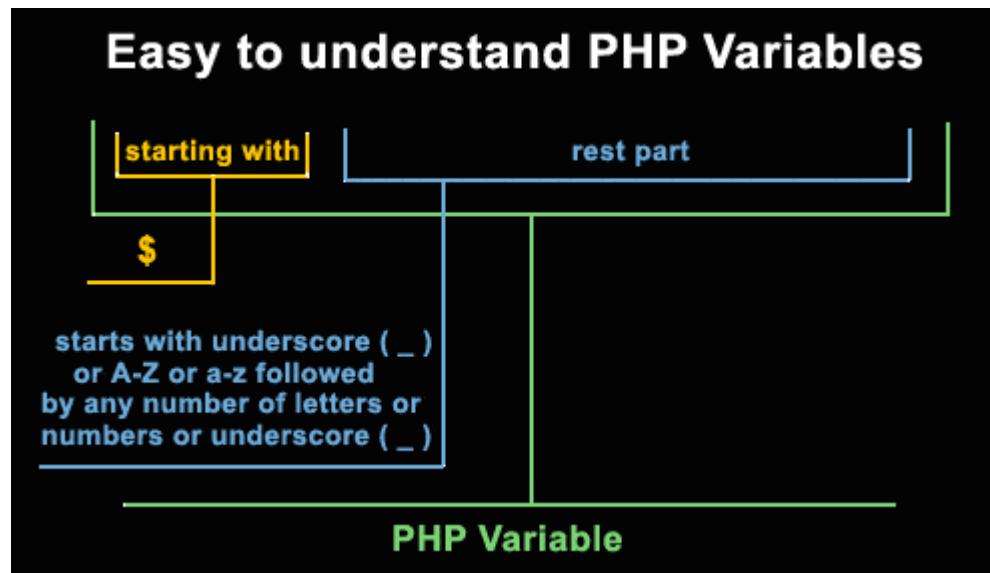


Last update: **2021/09/16 08:05**

1.3.1) 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
Ganze Zahlen	integer	1911 oder -1911
Dezimalzahl	double	19.11 oder -19.11
Zeichenketten	string	„Kette von Zeichen“ oder 'Kette von Zeichen'
Felder (ein- oder mehrdimensional)	array	(„Frankfurt“, „Berlin“, „Zürich“)
Objekte	object	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.

Typumwandlung in PHP funktioniert oft wie in C. Der Name des gewünschten Typs wird vor der umzuwandelnden Variablen in Klammern gesetzt, dies wird auch als Cast-Operation bezeichnet.

```
<?php
    echo (int)$z;
    echo "<br>";
    var_dump($z); //Gibt die Variable mit zugehörigen Datentyp aus!
    echo "<br>";
    echo $z;
?>
```

Der obige PHP-Code ergibt folgende Ausgabe:

```
10
float(10.5)
10.5
```

Folgende Umwandlungen sind möglich:

- (int), (integer) – nach integer
- (bool), (boolean) – nach boolean
- (float), (double), (real) – nach float
- (string) – nach string
- (array) – nach array
- (object) – Wandlung zum Objekt

Anstatt eine Variable in einen String umzuwandeln, können Sie die Variable auch in doppelte Anführungszeichen einschließen.

Beachten Sie, dass Tabulatoren und Leerzeichen innerhalb der Klammern erlaubt sind. Deshalb sind die folgenden Beispiele identisch:

The screenshot shows a Mozilla Firefox window displaying a PHP documentation page. The title of the page is "Type-Casting bei PHP". The page content discusses type casting in PHP, mentioning that variables can contain strings and numbers, and that different output types are used for different outputs. Below the text, there is a list of examples: 12.3, 12.3, 12, 17.5, 17.5, 12.35.2, and 12. At the bottom of the browser window, there is a search bar and a status bar indicating "Fertig".

From:

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



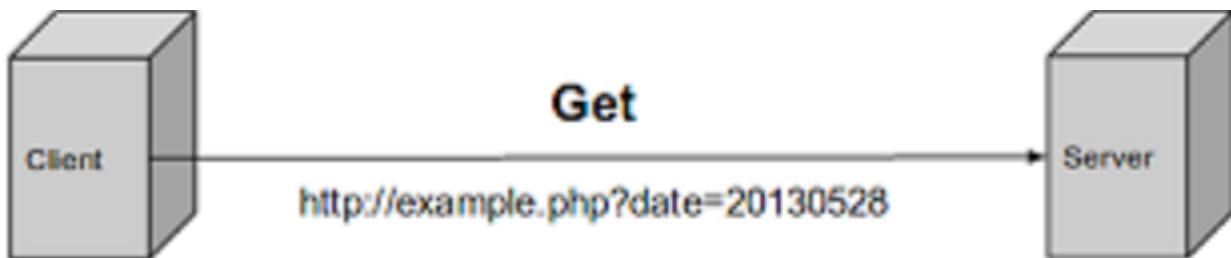
Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_06

Last update: **2021/09/16 08:06**

1.4.1) 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."</b> Euro</b>";  
}  
}  
  
?>
```

Überprüfen des mitgeschickten Arrays mittels POST

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

`print_r()` Zeigt den Inhalt der Variablen in lesbare 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(Variablen [, Variablen ,...]);` Das gesamte mittels POST übermittelte Array wird ausgegeben.

Beispiel Newsletter

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

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_07



Last update: **2021/09/16 08:06**

1.4.2) 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.

From:

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

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_08



Last update: **2021/09/16 08:07**

1.5.1) 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:inf7bi8bi_202122:1:1_09

Last update: **2021/09/16 08:07**

1.6.1) 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
<code>==</code>	Gleichheit	<code>\$a == \$b</code> ergibt TRUE, wenn \$a und \$b gleich sind.
<code>===</code>	Identisch	<code>\$a === \$b</code> ergibt TRUE, wenn \$a und \$b gleich und vom selben Datentyp sind.
<code>!=</code>	Ungleichheit	<code>\$a != \$b</code> ergibt TRUE, wenn \$a und \$b ungleich sind.
<code><</code>	Kleiner	<code>\$a < \$b</code> ergibt TRUE, wenn \$a kleiner \$b ist.
<code>></code>	Größer	<code>\$a > \$b</code> ergibt TRUE, wenn \$a größer \$b ist.
<code><=</code>	Kleiner gleich	<code>\$a <= \$b</code> ergibt TRUE, wenn \$a kleiner oder gleich \$b ist.
<code>>=</code>	Größer gleich	<code>\$a >= \$b</code> ergibt TRUE, wenn \$a größer oder gleich \$b ist.

Verknüpfung von Bedingungen

Operator	Name	Bedeutung
and bzw. <code>&&</code>	UND	<code>\$a and \$b</code> ergibt 1, wenn \$a und \$b 1 sind, ansonsten wird 0 zurückgegeben
or bzw. <code> </code>	ODER	<code>\$a or \$b</code> ergibt 1, wenn mindestens eine der beiden Variablen 1 ist.
<code>xor</code>	ausschließendes ODER	<code>\$a xor \$b</code> ergibt 1, wenn entweder \$a oder \$b 1 sind, aber nie beide gleichzeitig
!	NICHT	<code>!\$a</code> 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:inf7bi8bi_202122:1:1_10



Last update: **2021/09/16 08:07**

1.6.2) 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:inf7bi8bi_202122:1:1_11

Last update: **2021/09/16 08:08**

1.6.3) 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 angegebene 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:inf7bi8bi_202122:1:1_12

Last update: **2021/09/16 08:08**

1.6.4) Aufgaben 3-7

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 6 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 Apfelsorte 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:inf7bi8bi_202122:1:1_13



Last update: **2021/09/30 10:38**

1.7.1) for-Schleife

In der for-Schleife wird genau angegeben, wie oft die Schlaufe 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."  
\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:inf7bi8bi_202122:1:1_14

Last update: **2021/09/16 08:09**

1.7.4) Foreach-Schleife

Die foreach-Schleife ermöglicht es, auf einfache Weise ein Array zu durchlaufen. foreach funktioniert nur in Verbindung mit Arrays. Wenn Sie versuchen, foreach mit einer Variablen eines anderen Datentyps oder einer nicht initialisierten Variablen zu benutzen, gibt PHP einen Fehler aus.

Definition

```
foreach (array_expression as $value) {  
    Anweisung/en;  
}
```

Mithilfe von foreach wird das Durchlaufen eines Arrays wesentlich vereinfacht. Gegenüber der while-Schleife mit list und each ist die foreach-Schleife syntaktisch deutlich im Vorteil.

Beispiel while

```
<?php  
$zahlen = array (10, 20, 30, 40);  
while (list(), $value) = each ($zahlen)) {  
    echo "Wert: $value<br>\n";  
}  
?>
```

Beispiel foreach

```
<?php  
foreach ($zahlen as $value) {  
    echo "Wert: $value<br>\n";  
}  
?>
```

Das Array wird von Anfang bis Ende durchlaufen und bei jedem Schleifendurchlauf wird das aktuelle Element der Variablen array_expression zugewiesen. Jedes Arrayelement kann wiederum ein Array sein, dann könnte mit einer verschachtelten foreach-Schleife auch dieses Array ausgewertet werden.

Beispiel - mit assoziativem Array

```
$personen = array("Matthias", "Caroline", "Gülten");  
  
foreach ($personen as $person) {  
    echo "Name: $person<br>\n";  
}
```

```
// Ausgabe:  
// Name: Matthias  
// Name: Caroline  
// Name: Gütten
```

Typischerweise ist es erlaubt, einzelne Elemente eines solchen Arrays mit unterschiedlichen Datentypen zu belegen. Das können auch weitere Arrays sein.

Erweiterte Syntax

Sollten Sie Arrays mit Schlüssel-Wert-Paaren bauen, kann foreach mit einer erweiterten Syntax diese Paare direkt auslesen. Die grundlegende Syntax lautet:

Definition

```
foreach (array as $key => $value) {  
    Anweisung/en;  
}
```

Hier wird der Operator `=>` eingesetzt, der schon bei der Konstruktion des Arrays verwendet wurde.

Beispiel

```
<?php  
$person = array("Vorname" => "Caroline",  
    "Nachname" => "Kannengiesser",  
    "Alter" => 25,  
    "Ort" => Berlin);  
foreach ($person as $key => $val) {  
    echo "Feld $key hat den Wert: $val<br>";  
}  
?>
```

Ausgabe

```
Feld Vorname hat den Wert: Caroline  
Feld Nachname hat den Wert: Kannengiesser  
Feld Alter hat den Wert: 25  
Feld Ort hat den Wert: Berlin
```

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



Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_14_1

Last update: **2021/09/16 08:09**

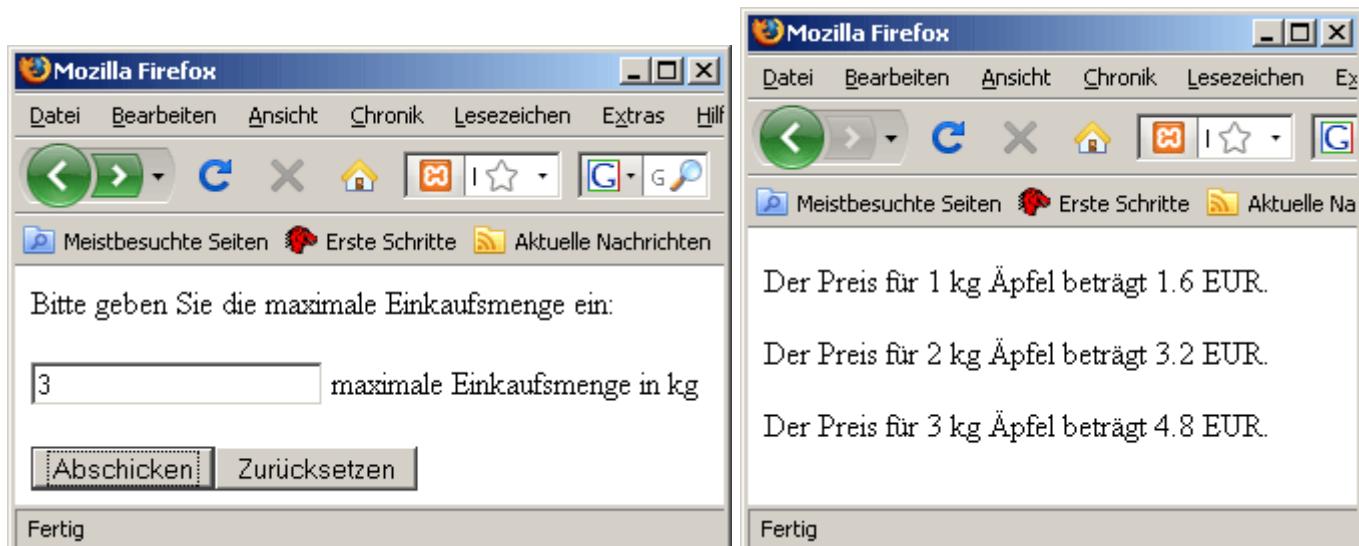
1.7.2) while-Schleife

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:inf7bi8bi_202122:1:1_15

Last update: **2021/09/16 08:09**



1.7.3) 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:inf7bi8bi_202122:1:1_16



Last update: **2021/09/16 08:09**

1.7.5) Aufgabe 8 bis 11

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

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



Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_17

Last update: **2021/09/16 08:10**

1.8.1) Was sind Felder?

Felder, auch Arrays oder Feldvariablen genannt, sind in der Lage, mehrere Variablen unter einem gemeinsamen Bezeichner zu speichern. Beispielsweise erhalten Sie in einer Berechnung 20 Ergebnisse. Die Ausgabe der 20 Werte soll erst erfolgen, wenn alle Berechnungen abgeschlossen wurden. Sie könnten 20 Variablen (\$ergebnis1, \$ergebnis2, \$ergebnis3, \$ergebnis4, ...) definieren, um alle Ergebnisse der Berechnungen zu speichern. Einfacher und schneller kann die Aufgabe gelöst werden, wenn die Ergebnisse der Berechnung in einem Feld abgelegt werden. Der Zugriff auf die einzelnen Ergebnisse erfolgt jeweils über ihre Position innerhalb des Feldes.



Felder in PHP

Felder unterscheiden sich in der Art, wie auf die Werte innerhalb des Feldes zugegriffen werden kann, und in ihrem Aufbau.

Auf Werte in Feldern zugreifen:

- Im numerisch indizierten Feld werden die einzelnen Werte (Value) innerhalb des Feldes über eine laufende Nummer, auch **Index oder Schlüssel** genannt, angesprochen. Der **Schlüssel** ist eine **laufende Nummer**, die automatisiert vergeben wird und bei **Null beginnt**.

```
$myArray = array();
$myArray[0] = "Susi";
$myArray[1] = "Maxi";

//oder

$myArray = array("Susi", "Maxi");
```

Schlüssel & Inhalt des Arrays \$myArray

key (index)	value (Name)
0	Susi
1	Maxi

- Im assoziativen Feld werden die einzelnen Werte (Value) innerhalb des Feldes über einen eindeutigen Schlüssel, Key genannt, angesprochen.

```
$myArray = array();
$myArray['key'] = "value";

//oder

$staedte = array('Guetersloh' => 98000,
                  'Bielefeld' => 326715);

echo $staedte["Guetersloh"]; //gibt 98000 aus
```

Schlüssel & Inhalt des Arrays \$staedte

key (Ort)	value (PLZ)
Guetersloh	98000
Bielefeld	326715

Aufbau von Feldern

Felder können ein- oder mehrdimensional sein:

- In einem eindimensionalen Feld können Sie beispielsweise eine Liste von Städten darstellen.
- Wenn Sie z. B. eine Tabelle mit verschiedenen Feldern abbilden möchten, können Sie dies mit einem zweidimensionalen Feld tun.
- Sie haben auch die Möglichkeit, Felder mit mehr als zwei Dimensionen zu nutzen.

Eigenschaften von Feldern

- In PHP müssen die Elemente von Feldern nicht vom selben Datentyp sein.
- Felder-Indizes bei numerisch indizierten Feldern beginnen mit 0. Das erste Element hat dementsprechend den Index 0, das zweite Element hat den Index 1, das dritte Element hat den Index 2 und so weiter. Der Index eines Feldes mit n Elementen reicht somit von 0 bis $n-1$. Der Index eines Feldes mit vier Elementen reicht somit von 0 - 3.
- Beim Anlegen eines Feldes muss nicht angegeben werden, wie viele Elemente in diesem Feld gespeichert werden. Sie können auch nach der Erstellung dem Feld zusätzliche Elemente hinzufügen.

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



Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_18

Last update: **2021/10/07 10:36**

1.8.2) Indizierte Felder

Indizierte Felder erstellen

Ein indiziertes Feld anlegen

PHP bietet Ihnen eine einfache und schnelle Möglichkeit, Felder mit verschiedenen Werten zu füllen. Dies wird über die array-Anweisung realisiert.

Syntax und Bedeutung der array() -Anweisung

```
$Feldvariable = array(Wert1, Wert2, Wert3, ...);
```

- Die array()-Anweisung beginnt mit dem reservierten Wort array.
- Der Variablen \$Feldvariable werden die einzelnen Werte übergeben, indem sie durch Kommata voneinander getrennt angegeben werden.
- Die Indizierung der einzelnen Elemente erfolgt automatisch in der Reihenfolge der Angabe.

Beispiel

```
$staedte = array("Frankfurt", "Berlin", "Bern");
```

Es befinden sich drei Werte in dem Feld \$staedte; Frankfurt, Berlin und Bern. Die Reihenfolge, in der die Werte zugewiesen worden sind, bestimmt die Indizes der Werte.

Index	0	1	2
Wert	Frankfurt	Berlin	Bern

Auf indizierte Felder zugreifen

Um auf einzelne Elemente in einem Feld zugreifen zu können, muss ihnen eine eindeutige Kennung zugewiesen werden. Bei numerisch indizierten Feldern entspricht der Kennung der automatisch erzeugte fortlaufende Index.

Auf einen bestimmten Wert eines Feldes greifen Sie zu, indem Sie zusätzlich zum Namen der Feldvariablen in den eckigen Klammern den Indexwert angeben.

```
$staedte[Index];
```

Beispiel

```
$stadt = $staedte[2];
```

Der Variablen \$stadt wird der dritte Wert des Feldes \$staedte, die Zeichenkette Bern, zugewiesen.

Mit indizierten Feldern arbeiten

Indizierte Felder ändern

Um einen Wert innerhalb einer Feldvariablen zu ändern, geben Sie bei der Wertzuweisung bei der Feldvariablen den entsprechenden Index des zu ändernden Wertes an und weisen Sie den neuen Wert zu.

```
$staedte[1] = "München";
```

In dem bisherigen Beispiel würde somit die Zeichenkette Berlin durch München ersetzt werden.

Indizierte Felder erweitern

Soll ein Feld um einen weiteren Wert am Ende des Feldes ergänzt werden, erzeugt PHP den entsprechenden Index automatisch.

Syntax und Bedeutung

```
$Feldvariable[] = Wert Zuweisung;
```

- Um einem Feld einen weiteren Wert (oder mehrere Werte) hinzuzufügen, brauchen Sie bei der Wertzuweisung keinen Index anzugeben. Das Feld wird am Ende um den Wert (bzw. die Werte) ergänzt und der Index automatisch erhöht.
- Sollte das Feld noch nicht bestehen, erstellt PHP automatisch ein Feld, sobald Sie der Variablen einen Wert zugewiesen haben.

Beispiel

```
$staedte = array("Frankfurt", "Berlin", "Bern");
$staedte[] = "Graz";
$staedte[] = "Rom";
```

ergibt:

Index	0	1	2	3	4
Wert	Frankfurt	Berlin	Bern	Graz	Rom

Anzahl der Feldelemente ermitteln

Der Befehl COUNT(\$bereich); liefert die Anzahl der Feldelemente des Arrays \$bereich.

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



Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_19

Last update: **2021/09/16 08:10**

1.8.3) Assoziative Felder

Assoziative Felder erstellen

Ein assoziatives Feld anlegen

Eine weitere Möglichkeit, Feldvariablen anzulegen, sind die assoziativen Felder. Bei ihnen werden für den Zugriff keine fortlaufenden Indizes benutzt, sondern Schlüssel, über deren Wert auf die einzelnen Werte zugegriffen werden kann, so genannte Schlüsselwörter.

Syntax der array()-Anweisung bei assoziativen Feldern

Auch die assoziativen Felder lassen sich über die array()-Funktion füllen.

```
$Feldvariable = array (Schlüssel1 => Wert1, Schlüssel2 => Wert2, ...);
```

Beispiel:

```
$hauptstaedte = array("Schweiz" => "Bern", "Frankreich" => "Paris");
```

- Die array ()-Anweisung beginnt mit dem reservierten Wort array.
- Der Variablen werden die einzelnen Wertpaare, bestehend aus Schlüsselwort (Key) und Wert (Value), übergeben, indem sie durch Kommata voneinander getrennt angegeben werden.
- Die Indizierung der einzelnen Werte erfolgt über die Angabe eines Schlüssels in Form einer beliebigen, eindeutigen Zeichenkette oder eindeutiger Integerwerte. Die Zuweisung erfolgt über =>.

Auf assoziative Felder zugreifen

Beim Zugriff auf einen Wert des Feldes kann der Schlüsselwert direkt angegeben oder als Variable übergeben werden:

	Ergebnis
\$stadt = \$hauptstaedte["Schweiz"];	Variable \$stadt hat den Wert „Bern“

oder:

	Ergebnis
\$k = "Schweiz"; \$stadt = \$hauptstaedte[\$k];	Variable \$stadt hat den Wert „Bern“

Beispiel: arr_assoz.php Hier wurde die Zuweisung der Städtenamen mit der array() -Anweisung realisiert.

```

<html>
<body>
<?php
$hauptstaedte = array("Schweiz" => "Bern",
                      "Frankreich" => "Paris",
                      "Deutschland" => "Berlin",
                      "Österreich" => "Wien");
$k = "Österreich";
echo "<p>Hauptstadt von $k: " . $hauptstaedte[$k] . "</p>";
echo "Die Hauptstadt von $k ist " . $hauptstaedte["Österreich"];
?>
</body>
</html>

```

- Das assoziative Feld wird über die `array()`-Funktion und die Zuweisung der Schlüssel sowie Feldinhalte angelegt.
- Der Variablen `$k` wird die Zeichenkette Österreich zugewiesen.
- Mithilfe der Variablen `$k` wird das entsprechende Element aus dem Feld ausgelesen und angezeigt.
- Hier wird das Element durch die direkte Angabe des Schlüssels herausgefiltert. Die Bezeichnung des Schlüssels wird als Zeichenkette in den eckigen Klammern angegeben.

Mit assoziativen Feldern arbeiten

Daten an Variablen übergeben

Mit einer `foreach()`-Schleife sind Sie in der Lage, die Werte der Feldelemente einzeln auszulesen. Dabei wird jedes Element in einer neuen Variable zwischengespeichert.

```

foreach($Feld as $wert) Anweisung;
foreach($Feld as $index => $wert) Anweisung;

```

- Die Funktion `foreach()` erwartet die Angabe des assoziativen Feldes, dessen Elemente durchlaufen werden sollen.
- In der ersten Variante werden mithilfe von `foreach` bei jedem Schleifendurchlauf der Variablen (`$wert`) nacheinander die Werte der Arrayelemente zugewiesen.
- Mithilfe der zweiten Variante können Sie zusätzlich den Index des Arrayelements auslesen und der Variablen `$index` zuweisen, was beispielsweise bei assoziativen Feldern hilfreich ist.

Beispiel: foreach.php

Verschiedene Länder der Welt und deren Hauptstädte werden in einem assoziativen Feld gespeichert. Zur Ausgabe sollen das Land (= Schlüssel) und die dazugehörige Hauptstadt (= Wert) in einer Tabelle angezeigt werden:

```

<?php
$hauptstaedte = array("Schweiz" => "Bern",

```

```

        "Frankreich" => "Paris",
        "Deutschland" => "Berlin");
$hauptstaedte["Polen"] = "Warschau";
$hauptstaedte["Italien"] = "Rom";
$hauptstaedte["Österreich"] = "Wien";

echo "<table border>";
echo "<tr><td><b>Land</b></td>" ;
echo "<td><b>Hauptstadt</b></td></tr>" ;
foreach($hauptstaedte as $land=>$stadt)
{echo "<tr><td>$land</td><td align='left'>$stadt</td></tr>" ;
}
echo "</table>";

?>

```

- Das assoziative Feld \$hauptstaedte wird über die array-Anweisung gefüllt.
- Dem assoziativen Feld \$hauptstaedte werden weitere Elemente hinzugefügt.
- Der Tabellenkopf wird erstellt.
- Mithilfe einer foreach-Schleife können Sie auf jedes Element des Feldes zugreifen. Die Schleife wird so lange durchgeführt, bis alle Elemente des Feldes angesprochen wurden. In jedem Schleifendurchlauf wird der Variablen \$land der jeweilige Schlüsselwert und der Variablen \$stadt der Wert des Feldes an der durch das Schlüsselwort festgelegten Position zugeordnet.
- Die Werte werden in die Tabelle eintragen und über den Ausgabebefehl echo am Bildschirm ausgegeben.



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



Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_20

Last update: **2021/09/16 08:11**

1.8.4) Mehrdimensionale Felder

Mehrdimensionale indizierte Felder erstellen

Grundlagen zu mehrdimensionalen Feldern

Im folgenden Beispiel sehen Sie mehrere Angaben (Vorname, Nationalität und Alter), die jeweils einen Eintrag in einer Feldvariablen bilden. Damit ist dieses Feld ein mehrdimensionales Feld. Bei jeder der Angaben sind Mehrfachwerte möglich. Mehrdimensionale Felder sind verschachtelte Felder, man spricht hier von äußeren und inneren Feldern.

Mehrdimensionale Felder können sowohl numerisch indiziert als auch assoziativ sein. Eine Mischform aus numerisch indizierten und assoziativen Feldern ist ebenfalls möglich. Im Beispiel bietet sich durch mögliche Mehrfachwerte z. B. ein indiziertes Feld an, das jeden Satz an Angaben mit einem eindeutigen numerischen Index versieht.

Index 1	Index 2		
	0	1	2
0	Oliver	spanisch	37 Jahre
1	Maria	deutsch	23 Jahre
2	Oliver	englisch	46 Jahre

In dieser Tabelle wird bereits die Arbeitsweise mehrdimensionaler Felder ersichtlich. Um beispielsweise an die Informationen von Maria zu gelangen, suchen Sie die Zeile mit der Angabe Maria und lesen die einzelnen Werte dieser Zeile aus: Maria, deutsch, 23 Jahre.

Mit mehrdimensionalen indizierten Feldern arbeiten

Ein mehrdimensionales indiziertes Feld hat statt eines Indexes - in Abhängigkeit von der Anzahl der Verschachtelungen - *mehrere* Indizes.

Syntax eines mehrdimensionalen indizierten Feldes

Die bereits erwähnten Informationen sollen im Folgenden als mehrdimensionales Feld realisiert werden.

Auf eine bestimmte Angabe in einem mehrdimensionalen indizierten Feld zugreifen

Ausschnitt aus der Beispieldatei „mehrdimensional.php“:

```
$person = array(array("Oliver", "spanisch", "37    Jahre"),
                array("Maria",      "deutsch",   "23 Jahre"),
```

```

        array("Oliver",      "englisch", "46     Jahre"));
$vorname1  = $person[l][0]      ;
$nationalitaetl  - $person[l][1]      ;
$alterl      = $person[l][2]      ;

```

Ergebnis:

```

$vorname1      = "Maria";
$nationalitaetl = "deutsch";
$alterl        = "23 Jahre".

```

Ein mehrdimensionales indiziertes Feld erweitern

```

$person[3][0] = "Johanna";
$person[3][1] = "schwedisch";
$person[3][2] = "19 Jahre";

```

oder alternativ:

```
$person[] = array("Johanna", "schwedisch", "19 Jahre");
```

Mit mehrdimensionalen assoziativen Feldern arbeiten

Syntax eines mehrdimensionalen assoziativen Feldes

Die assoziativen Felder besitzen zum Ansprechen der einzelnen Elemente einen sogenannten Schlüssel. Um ein mehrdimensionales Feld anzulegen, wird jedem Schlüssel ein weiteres Feld mit Schlüssel-Wert-Paaren übergeben (Sonderfall zweidimensional). Verwenden Sie aus Gründen der Übersichtlichkeit und Nachvollziehbarkeit der Programmierung nicht mehr als drei Dimensionen bzw. Verschachtelungen.

Beispiel: mehrdimensional.php

```

$land = array("Spanien"  => array("Hauptstadt"      =>      "Madrid",
                                         "Sprache"       =>      "Spanisch",
                                         "Waehrung"     =>      "Euro",
                                         "Flaeche"       =>      "504645 qkm"),
              "England"    => array("Hauptstadt"      =>      "London",
                                         "Sprache"       =>      "Englisch",
                                         "Waehrung"     =>      "Pfund Sterling",
                                         "Flaeche"       =>      "130395 qkm"),
              "Portugal"   => array("Hauptstadt"      =>      "Lissabon",
                                         "Sprache"       =>      "Portugiesisch",
                                         "Waehrung"     =>      "Euro",
                                         "Flaeche"       =>      "92345 qkm"));

```

Hiermit ist jeder Satz an Eintragungen mit entsprechenden Länderinformationen über einen Schlüssel mit aussagekräftiger Bezeichnung - Spanien, England oder Portugal - direkt ansprechbar.

Auf eine bestimmte Angabe in einem mehrdimensionalen assoziativen Feld zugreifen

Ausschnitt aus Beispieldatei „mehrdimensional.php“

```
$hauptstadt2 = $land["Portugal"]["Hauptstadt"];
$sprache2    = $land["Portugal"]["Sprache"];
$waehrung2   = $land["Portugal"]["Waehrung"];
$flaeche2    = $land["Portugal"]["Flaeche"];
```

Ergebnis:

```
$hauptstadt2 = "Lissabon";
$sprache2    = "Portugiesisch";
$waehrung2   = "Euro";
$flaeche2    = "92345 qkm";
```

Ein mehrdimensionales assoziatives Feld erweitern

```
$land["Ungarn"]["Hauptstadt"] = "Budapest";
$land["Ungarn"]["Sprache"]    = "Ungarisch";
$land["Ungarn"]["Waehrung"]   = "Forint";
$land["Ungarn"]["Flaeche"]    = "93036 qkm";
```

oder alternativ:

```
$land["Ungarn"] = array("Hauptstadt" => "Budapest",
                        "Sprache"    => "Ungarisch",
                        "Waehrung"   => "Forint",
                        "Flaeche"    => "93036 qkm");
```

Daten aus mehrdimensionalen Feldern extrahieren

Bei der Arbeit mit mehrdimensionalen Feldern kommt es häufig vor, dass Sie die kompletten Daten eines inneren Feldes benötigen. Mit einer foreach-Schleife können Sie alle Werte einer Feldvariablen auslesen. Durch die Verwendung einer foreach-Schleife müssen Sie keinen Index oder Schlüssel kennen. Bei jedem Schleifendurchlauf wird das aktuelle Element in einer von Ihnen angegebenen Variablen zwischengespeichert. Hierbei handelt es sich auch um eine Feldvariable mit Angaben zum aktuellen „Datensatz“.

Syntax:

```
foreach($feld as $wert);
```

Um die einzelnen Elemente auszulesen und jeden Wert verschiedenen Variablen zu übergeben, verwenden Sie innerhalb der Schleife die `list()`-Funktion.

Syntax

```
list($variable1, $variable2, ...) = $wert;
```

- Der Funktion `list()` wird das entsprechende Element übergeben, das in die angegebenen Variablen aufgeteilt werden soll.
- Die Variablen werden nacheinander, durch Kommata voneinander getrennt, in den runden Klammern der Funktion `list()` angegeben.
- Die Anzahl und die Position der Variablen richten sich nach der Anzahl der Elemente im Feld.
- Bei der an dieser Stelle verwendeten Variablen `$wert` handelt es sich um die Variable, die als letztes Argument in der `foreach()`-Schleife definiert wurde.

Beispiel: array_zdim_list.php

Verschiedene Angaben zu Ländern werden in einem mehrdimensionalen Feld gespeichert. Zur Ausgabe sollen der Name des Landes, die Hauptstadt, die Sprache und die Landeswährung in separaten Variablen abgelegt werden.

```
<body>





```

```

②      foreach($staedte as $key=>$ausgabe)
        {
            list($hauptstadt, $sprache, $waehrung) = $ausgabe;
            echo "<tr><td>" . $key . "</td>";
            echo "<td>" . $hauptstadt . "</td>";
            echo "<td>" . $sprache . "</td>";
            echo "<td>" . $waehrung . "</td></tr>";
        }
    ?>
</table>
</body>

```

(1) Das assoziative Feld `$staedte` wird neu definiert und mit Werten gefüllt. Die Daten enthalten Informationen zur Hauptstadt eines Landes, zur Sprache und zur im Land verwendeten Währung. Der Name des Landes selbst wird als eindeutiger Schlüssel verwendet.

(2) Mithilfe einer `foreach`-Schleife können Sie auf jedes Element des Feldes zugreifen, indem die Schleife so oft durchgeführt wird, bis auch das letzte Element angesprochen wurde. Da es sich in diesem Beispiel um mehrere Werte pro Element handelt, müssen die zusammenhängenden Werte eines Landes in dem neuen Feld `$ausgabe` zwischengespeichert werden. Im ersten Schleifendurchlauf enthält die Variable `$ausgabe` nur die Werte des zuerst definierten Landes Japan (Tokio, Japanisch, Yen), im zweiten Schleifendurchlauf die Werte der Niederlande etc.

(3) Über den Befehl `list()` werden die Daten zum gerade abgerufenen Land (Hauptstadt, Sprache und Währung) in die entsprechenden Variablen `$hauptstadt`, `$sprache` und `$waehrung` aufgeteilt.

(4) Über den Ausgabebefehl `echo` werden die Werte der Variablen am Bildschirm ausgegeben. Bei der Variablen `$key` handelt es sich um den Schlüssel (Landesnamen), der in (2) definiert wurde.



The screenshot shows a Microsoft Internet Explorer window displaying a table of country information. The table has four columns: Land, Hauptstadt, Sprache, and Währung. The data is as follows:

Land	Hauptstadt	Sprache	Währung
Japan	Tokio	Japanisch	Yen
Niederlande	Amsterdam	Niederländisch	Euro
Polen	Warschau	Polnisch	Zloty
Indien	Neu Delhi	Indisch	Rupie
Island	Reykjavík	Isländisch	Krone
Italien	Rom	Italienisch	Euro
Frankreich	Paris	Französisch	Euro
Spanien	Madrid	Spanisch	Euro
England	London	Englisch	Pfund Sterling

Beispiel aus SÜ

```
<?php
// Nicht assoziatives Array
$matrix = array (
    array(1,2,3,4),
    array(5,6,7,8),
    array(9,10,11,12),
    array(13,14,15,16),
    array(17,18,19,20)
);
// Ausgabe als Zeilen und Spalten
for ($i=0;$i<count($matrix);$i++)
{
    for ($j=0;$j<count($matrix[$i]);$j++)
    {
        echo $matrix[$i][$j].' ';
    }
    echo '<br>';
}

// Assoziatives Array
$user = array(
    'person'=>array('name'=>'Thomas', 'nickname'=>'Baumi'),
    'lang'
=>array('deutsch','mostviertlerisch','english','spanisch','französisch'),
    'kontakt'=>array('email'=>'thomas.baumgartner@bgamstetten.ac.at',
    'web' =>'www.baumi.net')
);

// Ausgabe
foreach ($user as $eigenschaft => $arrayinhalt)
{
    echo $eigenschaft.':';
    foreach ($arrayinhalt as $index =>$wert)
        echo ' '.$index.' ist '.$wert.' | ';
    echo '<br>';
}
?>
```

Verwenden des passenden Feldtyps

Der passende Feldtyp hängt von den Daten ab, die Sie in der Feldvariablen speichern wollen. Die folgende Tabelle soll Ihnen bei der Auswahl des Feldvariablentyps helfen:

Daten	Anzahl an Informationen pro Eintrag	Bevorzugter Feldtyp
Einfache Liste (z. B. Namen, Länder, Automarken etc.)	1	<input checked="" type="checkbox"/> eindimensional/indiziert
Wertpaare (z. B. Länder - Hauptstädte, Artikel - Preis etc.)	2	<input checked="" type="checkbox"/> eindimensional/assoziativ, sofern ein Wert eindeutig ist (= Schlüssel), sonst <input checked="" type="checkbox"/> mehrdimensional/indiziert oder assoziativ
Mehrere evtl. mehrfach verschachtelte Werte (z. B. Mitarbeiter und ihre Stammdaten, Artikelkategorien - Artikel - Artikeldetails etc.)	> 2	<input checked="" type="checkbox"/> mehrdimensional/indiziert oder assoziativ

From:

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



Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_21

Last update: **2021/12/03 21:08**

1.8.5) Weitere Informationen zu Feldern in PHP

Felder begegnen uns in PHP nicht nur, wenn wir selbst Feldvariablen definieren. PHP arbeitet intern automatisch mit Feldern, und zwar bei folgenden Aktionen:

- Die in einem HTML-Formular übermittelten Daten speichert PHP in einer Feldvariablen.
- Wenn man mit Sessions arbeitet, werden die zur Session gehörigen Daten automatisch in einem Feld abgelegt.
- Bei Abfragen aus Datenbanken befinden sich die Ergebnisse ebenfalls üblicherweise in einer Feldvariablen.

Weitere Feld-Funktionen

PHP kennt annähernd 100 Funktionen für den Umgang mit Feldern. Für nahezu jede Fragestellung liefert PHP eine passende Funktion. Folgende Aufgabengebiete werden abgedeckt:

- Auslesen
- Auswerten (u.a. Dopplungen finden)
- Sortieren
- Suchen
- Teilen und Zusammensetzen
- Verändern (u.a. Füllen, Hinzufügen, Löschen)

Eine Auswahl zusätzlicher wichtiger Feldfunktionen wird kurz in der nachfolgenden Tabelle vorgestellt:

Feldfunktion	Beschreibung
<code>array_flip(\$feld)</code>	Im Feld werden Indizes mit Werten vertauscht.
<code>array_key_exists(wert,\$feld)</code>	Prüfung, ob ein Schlüssel in einem Feld vorhanden ist.
<code>array_keys(\$feld)</code>	Liefert die Indizes des angegebenen Feldes zurück.
<code>array_merge(\$feld1,\$feld2...)</code>	Fügt die Elemente mehrerer Felder zu einem Feld zusammen.
<code>array_push(\$feld,werte)</code>	Das Feld wird um den oder die angegebenen Werte am Ende des Feldes erweitert.
<code>array_search(wert,\$feld)</code>	Das Feld wird nach dem angegebenen Wert durchsucht.
<code>array_sum(\$feld)</code>	Addiert die Werte des Feldes und liefert das Ergebnis zurück.
<code>array_unique(\$feld)</code>	Es wird ein neues Feld erstellt, aus dem doppelte Werte des angegebenen Feldes gelöscht wurden.
<code>array_values(\$feld)</code>	Alle Werte des Feldes werden zurückgeliefert.
<code>count(\$feld)</code>	Gibt die Anzahl der Elemente des Feldes zurück.
<code>sort(\$feld)</code> <code>rsort(\$feld)</code>	Sortiert die Werte des angegebenen Feldes aufsteigend bzw. absteigend.

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



Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_22

Last update: **2021/09/16 08:11**

1.8.6) Aufgaben 12 bis 18

Aufgabe 12 - Indizierter Array

1. Schreibe mittels PHP-Code ein „indiziertes Array“, das folgende Elemente auflistet:

- Englisch
- Informatik
- Deutsch
- Geografie
- Bewegung und Sport

2. Gib alle eingegebenen Fächer aus.

3. Schreibe folgenden Satz im PHP-Block und greife auf den Array zu, um ihn zu vervollständigen:
„**Mein Lieblingsfach ist [.....]**“

Aufgabe 13 - Assoziativer Array

1. Schreibe mittels PHP-Code ein „assoziatives Array“. Beziehe folgende Werte ein:

- Hauptfach - Englisch
- Nebenfach - Geschichte

2. Schreibe die folgenden Sätze im PHP-Block und greife auf den Array zu, um den Satz zu vervollständigen:

[Englisch] ist ein Hauptfach. [Geschichte] ist ein Nebenfach.

Aufgabe 14 - Jahreszeit

Erstelle ein Dokument namens jahreszeit.php. Hier sollen die vier Jahreszeiten in einer Werteliste namens

```
$jahreszeit[0]
```

abgebildet werden können. Nutze dafür die **Kurzform**, indem du alle Werte auf einmal zuweist. Gib den Herbst aus.

Aufgabe 15 - Monatsnamen

Erstelle ein Dokument namens monat.php. Speichere alle Monatsnamen in einem Array namens \$monat. Nutze dafür die **Langform**, indem du die Monate einzeln zuweist! Gib sodann folgenden

String aus:

„Ein Jahr hat 12* Monate. Der aktuelle Monat ist der Mai.“ (* Zählfunktion einsetzen)

Überlege dabei, welcher Arraytyp sinnvoll ist (indiziert oder assoziativ)!

Aufgabe 16 - Zufallszahlen

Ein Array soll mit einer vorgegebenen Anzahl von Zufallszahlen zwischen den Grenzen ug und og belegt werden. Dabei können sich die Zahlen wiederholen.

Zufallszahlen

Bitte geben Sie die gewünschte Anzahl der Zufallszahlen bzw. die obere und untere Grenze ein:

Anzahl der Zufallszahlen:	5
Untere Grenze:	10
Obere Grenze:	20

Folgende Zufallszahlen wurden ermittelt:

Untergrenze 10 Obergrenze 20

Zufallszahlen: 18 15 16 12 11

Hinweis: Wie man eine Zufallszahl zwischen den Grenzen ug und og zu erhalten kann, zeigt die folgende Zeile:

```
$zahl[$i]=rand($ug,$og);
```

1. Nach der Generierung der Zufallszahlen soll das Array ausgegeben werden.
2. Der Mittelwert aller Zahlen soll ausgegeben werden.
3. Die Standardabweichung soll ausgegeben werden.

Aufgabe 17 - Primzahloberechnung mit dem Sieb des Erathostenes

Zur Berechnung aller Primzahlen bis zu einer Obergrenze n gibt es ein Verfahren, das vom griechischen Mathematiker Erathostenes stammt. Es wird das *Sieb des Erathostenes* genannt und basiert auf der Idee, dass die Vielfachen einer Primzahl mit Sicherheit keine Primzahlen sind. Wenn wir z.B. wissen, dass 2 eine Primzahl ist, können wir alle Vielfache von 2 aus der Menge der Primzahlenkandidaten streichen. Die kleinste Zahl der dann verbleibenden Menge ist die nächste Primzahl. Wir eliminieren wiederum alle ihre Vielfache.

Implementieren lässt sich dieser Algorithmus sehr schön mit Hilfe eines Arrays, das zunächst lauter „p“ (Primzahl) in den Feldern stehen hat. Wird eine Zahl gestrichen, ersetzt man das „p“ durch ein „z“

(zusammengesetzte Zahl).

Die Primzahlen sind dann die Indizes jener Felder, in denen „p“ steht.

Aufgabe 18 - Leichtathletik

Erstelle eine neue Datei unter dem Namen uebung_mehrdim.php mit einem mehrdimensionales indiziertes Feld mit folgenden Inhalten und gib die Daten anschließend in Tabellenform (mit Überschrift) auf dem Bildschirm aus:

Beginn	Disziplin	Ort	Bemerkung
09:30 Uhr	Diskuswurf	Nebenplatz	Jugendmeisterschaften
10:00 Uhr	5-km-Lauf	Stadion - Laufbahn	Offener Lauf
11:00 Uhr	Halbmarathon	Waldgebiet	Teilnahme ab 18 Jahren
12:00 Uhr	Stabhochsprung	Stadion - Stabhochsprunganlage	Nur Frauen

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

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_23



Last update: **2021/09/16 08:12**

1.8.7) Fragen zu Feldern

- Erkläre die Begriffe „eindimensionales“ und „mehrdimensionales Feld“ und gib jeweils ein sinnvolles bzw. typisches Beispiel an.
- Erkläre die Begriffe „indiziertes“ und „assoziatives Feld“. Worin besteht der Unterschied.
- Wie kann man bei indizierten bzw. bei assoziativen Felder auf die Feldwerte zugreifen?
- Welche typischen Befehle für den Zugriff auf Feldern sind dir bekannt?

From:

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



Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_24

Last update: **2021/09/16 08:12**

1.9.1) HIDDEN-Feld

Beim Formularfeldtyp **hidden** handelt es sich um ein verstecktes input-Feld, das für den Betrachter nicht sichtbar ist. Dennoch kann in ihm eine Variable gespeichert und auch übergeben werden. Genau darin besteht die Aufgabe eines versteckten Feldes, nämlich im Hintergrund verborgen Daten zu übergeben.

Datei einkauf.html

```
<html>
<head>
  <title>Datenübergabe mittels HIDDEN-Feld</title>
</head>
<body>
  Heute kosten alle Artikel 5,99 Euro
  <form action="kaufen.php" method="POST">
    <input type="Hidden" name="preis" value="5.99">
    Artikel <input type="Text" name="artikel">
    Menge <input type="Text" name="menge">
    <input type="Submit" name="submit" value="Kaufen">
  </form>
</body>
</html>
```

Datei kaufen.php

```
<html>
<head>
  <title>Datenübergabe mittels HIDDEN-Feld</title>
</head>
<body>
  <?php
    echo "Sie haben den Artikel <b>" . $_POST["artikel"] . "</b>" .
    $_POST["menge"] . "
      mal zum Gesamtpreis von " . ($_POST["menge"] * $_POST["preis"]) . " Euro
    gekauft." ;
  ?>
</body>
</html>
```

Hidden-Input-Felder stellen eine geeignete Möglichkeit dar, Daten über mehrere Webseiten hinweg zu transportieren. Schließlich muss es ausreichen, wenn der Benutzer beispielsweise seinen Vornamen nur auf der ersten Seite eines fünfseitigen Fragebogens eingibt und wir ihn trotzdem auf der fünften Seite noch immer freundlich beim Vornamen ansprechen können.

Wird die Eingabe auf der ersten Seite mittels eines Texteingabefeldes getätigt, so steht sein Vorname auf der zweiten Seite des Fragebogens in `$_GET` bzw. `$_POST` zur Verfügung. Diesen Wert können wir dank PHP in ein Hidden-Input-Feld übernehmen und transportieren den Wert damit sicher zur nächsten Seite, sobald der Benutzer die aktuelle Seite durch Aufruf des Absenden-Buttons verlässt.

Diese Kette kann, solange auf der nächsten Seite ein Formular folgt, beliebig lang sein.

Hier zur Verdeutlichung die entsprechenden Skriptbeispiele:

Erste Seite

```
Name : <input type="Text" name="name">
```

Nutzung des Hidden Input Feldes in den folgenden Seiten

```
<input type="Hidden" name="name" value="<?php echo $_POST["name"]; ?>">
```

Zu beachten ist, dass dieses Beispiel nur funktioniert, wenn keine Anführungszeichen im Namen vorkommen!

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

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_25



Last update: **2021/10/19 16:54**

1.9.2) Textarea

Bei der Textarea handelt es sich um den großen Bruder des bereits mehrfach erwähnten Textfeldes. Die beiden Varianten unterscheiden sich nur dadurch, dass die Textarea aus mehreren Zeilen bestehen kann und somit für größere Eingaben besser geeignet ist, als ein einzeiliges Texteingabefeld. Im vorliegenden Beispiel wird beim Abschicken des Formulars die Variable \$bemerkung mit dem Inhalt des Textareafeldes als Wert übergeben. Will man der Textarea einen Wert vorgeben, so muss man diesen Wert zwischen Anfangs- und Endtag des Textfeldes schreiben.

```
<textarea name="bemerkung"></textarea>
```

so sieht es aus:

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

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_26



Last update: **2021/10/19 16:54**

1.9.3) Checkbox

Checkbox ohne Array und implode()

Wird ein Formular mit Checkboxen abgeschickt, werden in PHP nur die Einträge im Array `$_GET` bzw. `$_POST` erzeugt, deren Felder beim Ausfüllen ausgewählt werden. Wenn also nur das 2. und 3. Kästchen angeklickt wird, dann werden nur die Variablen `$_GET["artikel2"]` mit dem Wert Artikel 2 und `$_GET["artikel3"]` mit dem Wert Artikel 3 erzeugt.

Wenn eine oder mehrere Checkboxen standardmäßig den Zustand „Ausgewählt“ besitzen sollen, kann man innerhalb des input-Tags das Attribut `checked` verwenden.

```
<html>

<head>
    <title>Checkbox</title>
</head>
<body>

<?php
    echo "<b>Sie haben folgende Artikel bestellt:</b><br>";
    echo $_GET["artikel1"] . "<br>";
    echo $_GET["artikel2"] . "<br>";
    echo $_GET["artikel3"] . "<br>";
    echo $_GET["artikel4"] . "<br>";
?>

<form action="check.php" method="get">
    <input type="Hidden" name="preis" value="5.99">
    <input type="Checkbox" name="artikel1" value="Artikel 1">Artikel 1<br>
    <input type="Checkbox" name="artikel2" value="Artikel 2">Artikel 2<br>
    <input type="Checkbox" name="artikel3" value="Artikel 3">Artikel 3<br>
    <input type="Checkbox" name="artikel4" value="Artikel 4">Artikel 4<br>
    <input type="Submit" name="submit" value="Kaufen">
</form>
</body>
</html>
```

so sieht es aus:

<input type="checkbox"/> Artikel 1
<input type="checkbox"/> Artikel 2
<input type="checkbox"/> Artikel 3
<input type="checkbox"/> Artikel 4
<input type="button" value="Kaufen"/>

Checkbox mit Array und implode()

Checkbox lassen sich auch mittels einer Array - Anweisung erstellen. Wobei es bei der Ausgabe wichtig ist, nicht nur ein Element auszugeben, sondern es könnten möglicherweise alle Elemente, oder nur zwei sein.

Dazu dient die implode() - Anweisung

```
implode ( string $glue , arrayname )
Verbindet die Elemente eines Array mittels glue zu einer Zeichenkette
```

Nachdem er nur etwas im Array findet, wenn etwas angeklickt ist, sollte die Ausgabe nur erfolgen, wenn das Array nicht leer ist (eingebaute IF - Anweisung).

```
<html>
<head>
  <title>Checkbox</title>
</head>
<body>
<form action="check.php" method="POST">
  <input type="hidden" name="preis" value="5.99">
  <input type="checkbox" name="artikel[]" value="Artikel 1">Artikel 1<br>
  <input type="checkbox" name="artikel[]" value="Artikel 2">Artikel 2<br>
  <input type="checkbox" name="artikel[]" value="Artikel 3">Artikel 3<br>
  <input type="checkbox" name="artikel[]" value="Artikel 4">Artikel 4<br>
  <input type="submit" name="submit" value="Kaufen">
</form>

<?php
  if(isset($_POST["artikel"])){
    $var = $_POST["artikel"];
    $ausgabe = implode(", ", $var);
    echo "<b>Sie haben folgende Artikel bestellt: $ausgabe</b><br>";
  }
</body>
</html>
```

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

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_27



Last update: **2021/10/19 16:55**

1.9.4) Radio-Button

Radio-Buttons ohne Arrays erstellen

Während man bei Checkboxen immer beliebig viele Möglichkeiten auswählen kann, ist man bei Radio-Buttons auf nur eine einzige Antwortmöglichkeit beschränkt. Wie man sehen kann, haben bei Radio-Buttons im Gegensatz zu Checkboxen alle zusammengehörigen Auswahlmöglichkeiten denselben Namen. Daher kann auch nur ein Array-Eintrag erstellt werden, dem dann beim Abschicken der entsprechende Wert des angeklickten Feldes zugewiesen wird.

Ähnlich wie bei Checkboxen kann man hier für einen Radio-Button den Zustand „ausgewählt“ vorgeben, indem man im entsprechenden input-Tag das Attribut checked angibt. Zu beachten ist, dass die Angabe des Attributs nur dann sinnvoll ist, wenn sie sich auf genau einen Radio-Button beschränkt.

```
<html>
  <head>
    <title>Radio-Buttons</title>
  </head>
  <body>
    <?php
      if (isset($_POST["submit"]))
      {
        echo "Sie haben Artikel <b>" . $_POST["artikel"] . "</b> bestellt!";
      }
    ?>
    <form action="radiobutton.php" method="POST">
      <input type="Hidden" name="preis" value="5.99">
      <input type="Radio" name="artikel" value="1" checked>Artikel 1<br>
      <input type="Radio" name="artikel" value="2">Artikel 2<br>
      <input type="Radio" name="artikel" value="3">Artikel 3<br>
      <input type="Radio" name="artikel" value="4">Artikel 4<br>
      <input type="Radio" name="artikel" value="5">Artikel 5<br>
      <input type="Submit" name="submit" value="Kaufen">
    </form>
  </body>
</html>
```

so sieht es aus:

<input type="radio"/> Artikel 1
<input type="radio"/> Artikel 2
<input type="radio"/> Artikel 3
<input type="radio"/> Artikel 4
<input type="radio"/> Artikel 5
<input type="button" value="Kaufen"/>

Radio-Buttons mit Arrays erstellen

Radio Buttons lassen sich auch mittels einer Array - Anweisung erstellen. Wozu es einem PHP Code im HTML Teil benötigt.

Die Werte der RadioButtons werden in einem Array abgelegt und danach mittels forach ausgegeben. Dabei bekommt der Schüssel, der auch als value-Wert des Radio-Buttons dient, und der Ausgabewert denselben Inhalt. Nämlich jenen der im Array abgespeichert ist.

ACHTUNG!: Übermittelt wird nur der Indexwert, aber nicht der Inhalt des Arrays!

```
<html>
<head>
  <title>Radio-Buttons</title>
</head>
<body>
<?php
if (isset($_POST["submit"]))
{ $nummer=$_POST["artikel"]+1;
echo "Sie haben Artikel <b>" . $nummer . "</b> bestellt!";
}
?>
<form action="radiobutton.php" method="POST">
<input type="Hidden" name="preis" value="5.99">
<?php
$artikel = array ("Artikel1", "Artikel2", "Artikel3", "Artikel4",
"Artikel5");
foreach($artikel as $key => $artikelnr){
echo "<input type='radio' name='artikel' value='$key' />$artikelnr<br>"; }
?>
<input type="Submit" name="submit" value="Kaufen">
</form>
</body>
</html>
```

Für komplexere Ausgaben, lässt sich diese auch mit einer **switch-case Anweisung** lösen.

```
<?php
if (isset($_POST["submit"]))
{ echo "Sie haben ";
switch($_POST["artikel"])
{case "0":
echo "Artikel1";
break;
case "1":
echo "Artikel2";
break;
case "2":
echo "Artikel3";
break;
case "3":
```

```
echo "Artikel4";
break;
case "4":
echo "Artikel5";
break;
}
echo " bestellt";
?
?>
```

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

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_28



Last update: **2021/10/19 16:55**

1.9.5) Auswahlfelder

Als Alternative zu den Radiobuttons stehen noch die Select-Auswahlfelder zur Verfügung:

Select-Auswahlfelder:

```
<select name="artikel">
<option value="1">Auswahl 1</option>
<option value="2">Auswahl 2</option>
<option value="3">Auswahl 3</option>
<option value="4">Auswahl 4</option>
<option value="5">Auswahl 5</option>
</select>
```

so sieht es aus:

Auswahl 1 ▼

Hier wird analog zu den Radiobuttons nur ein Eintrag im Array `$_POST` erzeugt, dessen Name durch die Angabe im Element `select` vorgegeben ist. Der Wert des Eintrags ergibt sich aus der gewählten Option. Ähnlich wie bei einem Radio-Button kann man angeben, dass ein Eintrag einer Auswahlliste standardmäßig ausgewählt sein soll. Dazu muss man beim entsprechenden `option`-Tag das Attribut `selected` eintragen. Genau wie bei Radio-Buttons macht die Angabe natürlich nur Sinn, solange sie sich auf einen Eintrag der Liste beschränkt.

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

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_29



Last update: **2021/10/19 16:55**

1.9.6) Aufgabe 20

Erstelle das nachstehende Formular für die Preisberechnung auf der Website eines Winzerbetriebs:

Weinbestellung

Hier können Sie den Preis für Ihre Weinbestellung einfach berechnen:

- Flaschen (1l) Grüner Veltliner Kabinett 2006 (€ 2,20)
- Flaschen (0,75 l) Chardonnay 2009 (€ 5,00)
- Flaschen (0,75 l) Muskateller 2009 (€ 5,50)
- Flaschen (0,75 l) Merlot 2008 (€ 4,50)

Selbstabholung

Lieferung innerhalb des Bezirks (zzgl. € 25,00)

Ab einem Gesamtbestellwert (inkl. eventueller Liefergebühr) von insg. 150€ wird ein 4%iger Rabatt gewährt.

Ja, ich möchte den original Winzer-Newsletter erhalten!

Die Ausgabe soll auf der selben Seite unterhalb des Submit-Buttons erfolgen und etwa wie folgt aussehen:

Sie haben

- 5 Flaschen Grüner Veltliner (€ 11,00)
- 10 Flaschen Chardonnay (€ 50,00)
- 15 Flaschen Merlot (€ 67,50)

bestellt und wollen die Ware geliefert (€ 25,00) bekommen. Da der Gesamtbestellwert (€ 153,50) mind. € 150,00 beträgt, erhalten Sie einen 4%igen Rabatt und zahlen daher insgesamt nur **€ 147,36**.

Ja, Sie werden in Zukunft den original Winzer-Newsletter erhalten.

Herzlichen Dank für Ihre Bestellung!

From:

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



Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_30

Last update: **2021/10/19 16:56**

1.10.1) Nutzung von externen Dateien

Beispiel für die Nutzung von Daten aus externen Dateien

Wenn Sie Daten, die die Benutzer in ein Formular eingegeben haben, z. B. in einer Excel-Tabelle auswerten oder bearbeiten möchten, können Sie die Daten außerhalb des PHP-Programms in einer externen Datei speichern.

So können Sie beispielsweise auf diese Datei zugreifen, die Daten lesen, aber auch neue Daten hinzufügen bzw. die Daten ersetzen. Beispiele für die sinnvolle Nutzung von externen Dateien sind:

- Dynamische Daten in eine Webseite einbauen: Beispielsweise können Sie einen Hinweis auf das Angebot des Tages in einer externen Textdatei speichern. So brauchen Sie, um die Daten zu verändern, nur die externe Datei zu bearbeiten.
- Eingaben aus einem Formular speichern
- Logdateien schreiben: Um z. B. die Anzahl der Aufrufe einer Seite zu protokollieren, können Sie die Daten in einer eigenen Datei speichern.

Wichtig bei der Bearbeitung von externen Dateien ist die Art des Zugriffs auf die Datei. Hierbei kann PHP unterscheiden zwischen dem Zugriff auf sequenzielle und binäre Dateien.

Sequenzielle Dateien, z. B. Textdateien (*.txt) oder Dateien im Tabellenformat (*.csv), beinhalten Zeilen unterschiedlicher Länge. Sie werden sowohl beim Schreiben als auch beim Lesen sequenziell (also eine Zeile nach der anderen) bearbeitet.

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

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_31



Last update: **2021/10/19 16:56**

1.10.2) Dateien öffnen, lesen und schließen

Dateien mit fopen() öffnen

Bevor Sie auf die Daten in Dateien zugreifen können, müssen Sie die entsprechende Datei öffnen. Dazu können Sie die fopen()-Anweisung verwenden.

Syntax und Bedeutung der fopen() -Anweisung

```
fopen(Dateiname, Modus);  
  
$datei = fopen("user.txt", "r")
```

- Zusätzlich zur Angabe der zu öffnenden Datei wird der Modus, in dem die Datei geöffnet werden soll, angegeben:
- r ... Die Datei wird zum Lesen (r = read) geöffnet; Schreiben ist nicht erlaubt.
- r+ ... Die Datei wird zum Lesen und Schreiben geöffnet, und der Dateizeiger an den Anfang der Datei gesetzt.
- w ... Öffnet eine Datei zum Schreiben (w = write)
- w+ ... Hier handelt es sich um dieselbe Option wie die Option w, nur wird hier die Datei zum Lesen und Schreiben geöffnet. Der Dateizeiger wird auf den Anfang der Datei gesetzt sowie die Länge der Datei auf 0 Byte.
- a ... Die Datei wird zum Schreiben geöffnet, wobei die neuen Daten die vorhandenen Daten ergänzen (a = append).
- a+ ... Die Datei wird zusätzlich zum Lesen geöffnet.
- b ... Zusätzlich kann der Option Modus der Buchstabe 'b' hinzugefügt werden, der die Behandlung von Binär-Dateien erlaubt.

Dies ist auf Systemen sinnvoll, wenn diese zwischen Binär- und Text-Dateien unterscheiden (wie z. B. Windows).

- Der Rückgabewert der Funktion fopen() ist ein Zeiger auf die geöffnete Datei und wird für weitere Zugriffe auf die Datei genutzt. Im Beispiel wird der Dateizeiger in der Variablen \$datei gespeichert.

Wenn die angegebene Datei nicht gefunden wird, gibt die Funktion fopen Oden Wert FALSE zurück.

Dateien mit fgets() lesen

Mithilfe der Funktion fgets() können Sie den Inhalt einer Datei auslesen.

Syntax und Bedeutung der fgets() -Anweisung

```
fgets(Dateizeiger[, Modus]);  
  
$zeile = fgets($datei);
```

- Im ersten Parameter (Dateizeiger) finden Sie die Angabe zur Datei, aus der gelesen werden soll. Hierbei wird der Dateizeiger im Beispiel `$datei` aus der vorher geöffneten Datei verwendet.
- `fgets()` liest die angegebene Datei zeilenweise aus, wobei Sie optional im Parameter Modus die Länge angeben können. Lassen Sie diesen Parameter weg, ist automatisch eine Länge von maximal 1024 Zeichen für eine zu lesende Zeile festgelegt.
- `fgets()` liest bis zu dem ersten Auftreten eines Zeilenumbruchs bzw. der maximalen Zeilenlänge, dem Ende der Datei (EOF = End of File) oder bis zu der im Parameter Modus ausgegebenen Länge.
- Wenn Sie `fgets()` erneut aufrufen, wird die nächste Zeile gelesen.
- Der Rückgabewert der Funktion `fgets()` ist eine Zeichenkette. Im Beispiel wird der Rückgabewert in der Variablen `$zeile` gespeichert.

Dateien mit `fclose()` schließen

Nachdem Sie eine Datei geöffnet und die Daten ausgelesen haben, müssen Sie die Datei wieder schließen, um sie für andere Prozesse oder Benutzer nutzbar zu machen.

Syntax und Bedeutung der `fclose()`-Anweisung

```
fclose(Dateizeiger);  
  
fclose($datei);
```

- Mit der Funktion `fclose()` schließen Sie die Datei, auf die der Dateizeiger weist. Der Dateizeiger muss gültig, also mit der Funktion `fopen()` geöffnet worden sein.
- Wurde die Datei erfolgreich geschlossen, liefert die Funktion `fclose()` den Wert TRUE zurück, sonst FALSE.

Beispiel zu externe Datei öffnen, lesen und schließen: `fgets.php, user.txt`

```
<html>  
<body>  
<?php  
if (file_exists("Gedicht.txt"))  
    echo "Datei ist vorhanden!<br><hr>";  
  
$datei = fopen("Gedicht.txt", "r");  
if ($datei)  
{  
    echo "<p>Datei Gedicht.txt:</p>";  
    while (!feof($datei))  
    {  
        $zeile = fgets($datei);  
        echo $zeile."<br>";  
    }  
    fclose($datei);  
}
```

```
else
    echo " Es trat ein Fehler auf! " ;
?>
</body>
</html>
</html>
```

From:

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



Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_32

Last update: **2021/11/10 08:13**

1.10.3) Weitere Möglichkeiten zum Lesen von Dateien

Dateien mit `readfile()` oder `file()` lesen

Während die Funktion `fgets()` den Inhalt einer Datei zeilenweise einliest, können Sie mit den Befehlen `readfile()` und `file()` den gesamten Inhalt einer Datei auf einmal wiedergeben.

Der Unterschied zwischen den beiden Funktionen `readfile()` und `file()` liegt in der Ausgabe der Daten der Datei:

- `readfile()` liest vollständig den Inhalt der Datei aus und sendet das Ergebnis ohne weitere Bearbeitung direkt an den Browser.
- `file()` hingegen liest den vollständigen Inhalt der Datei zeilenweise in einen eindimensionales array. Hierbei wird jeweils der Inhalt einer Zeile zu einem Element des Feldes.

```
readfile(Datei);  
  
file(Datei);
```

Beispiel `readfile.php`

```
<html>  
<body>  
  <h4>Auslesen einer Textdatei</h4>  
  <?php readfile("user.txt"); ?>  
</body>  
</html>
```



Beispiel `file.php`

```
<html>  
<body>  
  <h4>Auslesen einer Textdatei</h4>  
  <?php  
    $feld = file("user.txt");
```

```

echo "<p>";
$i=1;
foreach($feld as $zeile)
{
    echo "Zeile ".$i++." :   ";
    echo $zeile."<br>";
}
?>
</body>
</html>

```



Inhalt einer Datei in eine Zeichenkette einlesen

Sie können die gesamte Datei in einen String einlesen. Hierbei verbinden Sie die einzelnen ausgelesenen Zeilen mithilfe der Zeichenkettenfunktion `join()`.

Sie können mit der Funktion `join()` ein Feld anhand eines Trennzeichens zu einem String zusammenfügen. Dabei werden die Elemente des Feldes nacheinander an den String angehängt.

```
join(Trennzeichen, Feld);
```

Beispiel join.php

```

<?php
if (file_exists("user.txt"))
{$inhalt = join (";", file("user.txt"));
echo $inhalt;
}
?>

```

Beispieldatei „join.php“

- Mithilfe der Funktion `file_exists()` wird überprüft, ob die Datei existiert. Die Funktion `file_exists()` erwartet als einzigen Parameter den Namen einer Datei auf dem Webserver. Dateien anderer Server (URL) können nicht auf ihre Existenz überprüft werden.
- Der Befehl `join()` verbindet die mithilfe der Funktion `file()` ausgelesenen Datei `user.txt` zu

einer Zeichenkette. Als Trennzeichen wird hierbei das Semikolon „;“ verwendet.



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



Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_33

Last update: **2021/11/24 09:29**

1.10.4) In Dateien schreiben

Dateien zum Schreiben öffnen

Wollen Sie Daten in eine Datei schreiben, müssen Sie die Datei zuerst öffnen. Beim Öffnen der Datei können Sie bestimmen, ob Sie die bestehenden Daten der Datei überschreiben wollen oder der bestehenden Datei weitere Daten hinzufügen möchten.

(1) Dateien überschreiben

Wenn Sie eine Datei mit der Funktion `fopen()` und dem Modus 'w' für „write“ öffnen, wird die Datei zum Schreiben geöffnet und der Dateizeiger auf den Anfang der Datei verschoben. Gleichzeitig wird die Länge der Datei auf 0 Byte gesetzt. Wenn die Datei nicht existiert, wird sie angelegt.

(2) Daten in Dateien hinzufügen

Um beispielsweise Daten, die in ein Formular eingegeben wurden, fortlaufend in eine Datei zu schreiben, verwenden Sie den Befehl `fopen()` mit dem Modus „a“. Die Datei wird zum Schreiben geöffnet und der Dateizeiger an das Ende der Datei gesetzt, sodass die neuen Daten hinzugefügt werden. Existiert sie nicht, legt PHP sie an.

Daten in Dateien schreiben

Um eine Zeichenkette in eine geöffnete Datei zu schreiben, verwenden Sie die Funktion `fputs()`. Syntax und Bedeutung der `fputs`-Anweisung

```
fputs (Dateizeiger, Zeichenkette, [Länge])
```

- Mit der Funktion `fputs()` können Sie eine beliebige Zeichenkette in eine Datei schreiben.
- Die entsprechende Datei wird über den mit `fopen()` festgelegten Dateizeiger angesprochen.
- Ist der optionale Parameter Länge angegeben, wird das Schreiben nach der angegebenen Anzahl Bytes beendet. Geben Sie ihn nicht an, wird die gesamte Zeichenkette geschrieben.
- `fputs()` gibt bei Erfolg die Anzahl der geschriebenen Bytes zurück, andernfalls FALSE.

Auf Systemen, die zwischen Binär- und Textdateien (z. B. Windows) unterscheiden, muss die Datei mit der Option 'b' in der `fopen()`-Funktion geöffnet werden.

Nachdem Daten in die Datei geschrieben wurden, muss die Datei geschlossen werden.

Beispiel: Dateidaten ergänzen `bestellformular.html`, `bestellung.php`, `bestellung_daten.csv`

Die Daten, die der Benutzer in das Bestellformular für Äpfel eingibt, sollen in einer *.csv-Datei an das Ende der schon vorhandenen Daten gehängt werden.



```

<html>
<body>
<?php
$datei = fopen("bestellung_daten.csv", "a");
if ($datei == false)
{ echo "<p>Datei konnte nicht zum Schreiben geöffnet werden</p>";
exit ;
}
$name    = $_POST["name"];
$strasse = $_POST["strasse"];
$ort     = $_POST["ort"];
$sorte   = $_POST["sorte"];
$menge   = $_POST["menge"];
fputs ($datei, "$name;$strasse;$ort;$sorte;$menge;n");
echo "<p>Folgende Angaben wurden gespeichert</p>";
echo "$name<br>"; echo "$strasse<br>";
echo "$ort<br>"; echo "$menge kg $sorte<br>";
fclose($datei);
?>
</body>
</html>

```

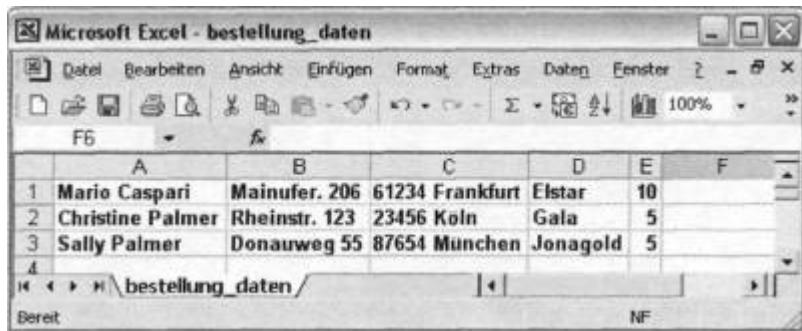
Die Datei bestellungdaten.csv wird mit fopen() und dem Modus 'a' geöffnet, d.h. zum Schreiben geöffnet, und der Dateizeiger wird an das Ende der Datei gesetzt, sodass die neuen Daten hinzugefügt werden.

- Es wird geprüft, ob das Öffnen der Datei erfolgreich war.
- Wenn die Datei nicht erfolgreich geöffnet werden kann, wird mit exit das Programm gestoppt.
Mit exit beenden Sie - ohne Möglichkeit der Rückkehr - die Skriptausführung.
- Die Variablen \$name, \$strasse etc. werden mit den Daten, die aus dem Formular übergeben wurden, gefüllt.
- Über die Funktion fputs() werden die Werte der Variablen in die geöffnete Datei geschrieben.
Hierbei werden die Daten jeweils durch ein Semikolon voneinander getrennt, und am Ende einer

Zeile wird mit Hilfe von „\n“ ein Zeilenvorschub erzeugt, um die neuen Daten in eine eigene Zeile zu schreiben.

- Die geöffnete Datei wird geschlossen.

*.csv können Sie mit Programmen, wie z. B. Microsoft Excel, öffnen und bearbeiten.



	A	B	C	D	E	F
1	Mario Caspari	Mainufer. 206	61234 Frankfurt	Elstar	10	
2	Christine Palmer	Rheinstr. 123	23456 Köln	Gala	5	
3	Sally Palmer	Donauweg 55	87654 München	Jonagold	5	

From:

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



Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_34

Last update: **2021/11/10 08:15**

1.10.5) Gästebuch

In der hier vorgestellten Variante eines Gästebuches werden nur der Name des Gastes und ein Text in der zugehörigen Textdatei `gastbuch.txt` gespeichert. Diese Textdatei muss Lese- und Schreibrechte für alle aufweisen.

- Die Funktion `file_exists()` testet, ob die genannte Datei im angegebenen Verzeichnis vorliegt.
- Die Funktion `feof()` testet, ob das Ende der Textdatei erreicht wurde. Der Ausdruck `!feof($fp)` liefert also so lange true, so lange das Dateiende nicht erreicht wurde.
- Die Funktion `fgets()` liest aus der mit der Ressource-ID festgelegten Datei einen so langen Datenstrom, wie das zweite Argument (im Beispiel 1000) angibt.
- Die Funktion `is_writeable()` testet, ob Schreibrechte für die angegebene Datei gewährt wurden.
- Die Funktion `fputs()` schreibt die angegebene Zeichenkette bzw. die Werte der angegebenen Variablen in die Datei, die mit der Ressource-ID identifiziert wurde.

Zusatz-Aufgabe

- Zusätzliche Informationen - etwa über Homepage, Datum, IP-Adresse, der Mitteilung, etc. sollen in das Gästebuch eingetragen werden. Informiere dich bezüglich des Einfügens des aktuellen Datums im Internet.

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



Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_35

Last update: **2021/11/10 08:23**

1.10.6) Beispiel: Besucherzähler

Eine häufige Anwendung in Verbindung mit Dateizugriffen ist die Einbettung eines Counters (Zugriffszählers) in eine Webseite. Ein Counter zählt die Anzahl der Webseiten-Zugriffe. Dabei wird bei jedem Zugriff die Zahl der bisherigen Besucher um eins erhöht. Das heißt, es wird auf eine bestehende Datei zugegriffen und mit ihrem Inhalt gerechnet. Das Ergebnis der Berechnung überschreibt dann den Inhalt der Datei.

Hinweise zum Beispiel:

- In den HTML-Text wird die Funktion `counter()` zur Anzeige der Besucherzahl eingebunden.
- Die Funktion `counter()` wird angelegt.
- Der Name der Datei, in der die Anzahl der Besucher abgelegt werden soll, lautet `counter.txt` und wird der Variablen `$name` zugewiesen. Die Variable `$count` erhält den Ausgangswert 0.
- Die Datei `counter.txt` wird über die Funktion `fopen()` zum Lesen und Schreiben geöffnet. Der Zugriff auf diese Datei erfolgt über den Dateizeiger `$datei`. Der Dateizeiger wird mit dem Modus `r+` an den Anfang der Datei gesetzt. War der Zugriff auf die Datei nicht erfolgreich, wird das Skript mit einer Meldung verlassen.
- Damit während der Arbeit mit der Datei kein weiterer Zugriff und somit ein gleichzeitiges Überschreiben der Daten möglich ist, wird die Datei exklusiv (`LOCK_EX`) gesperrt. `LOCK_EX` steht für eine exklusive, nur schreibende Verriegelung einer Datei (vgl. folgenden Abschnitt).
- Mit `fgets()` werden die ersten 10 Zeichen der Datei ausgelesen. Der Wert 10 wurde gewählt, da hiermit ein Zählen der Besucherzahl mit 10 Stellen (bis 9999999999) gewährleistet ist.
- Damit der neue Wert den alten Wert in der Datei überschreiben kann, wird der Dateizeiger mittels `fseek()` und der Angabe 0 an den Anfang der Datei gesetzt.
- Nach dem Einlesen des Wertes wird er um eins erhöht und am Bildschirm fett formatiert ausgegeben. Über die Funktion `fwrite()` wird der neue Wert `$count` in die Datei geschrieben.
- `fclose()` schließt die Datei `counter.txt` und beendet zugleich die Zugriffssperre, die über `flock()` veranlasst wurde.



Mit dem Befehl `flock()` können Sie eine Datei in der Zeit, in der das Skript in eine Datei schreibt, zusätzlich für weitere Zugriffe sperren.

Syntax und Bedeutung der `flock()-Anweisung`

```
flock (Dateizeiger,Modus);
```

- `flock()` sperrt den Zugriff auf die Datei, auf die der Parameter Dateizeiger verweist. Der Modus ermöglicht verschiedene Sperrzustände der Datei.
- Darf während Ihres Zugriffs die Datei von anderen nur gelesen werden, setzen Sie den Modus auf `LOCK_SH` (Shared Lock, Lesezugriff). Soll kein anderer Nutzer zeitgleich die Datei nutzen dürfen, setzen Sie die Option `LOCK_EX` (Exclusive Lock), `flock()` wartet, bis die Datei wie angegeben benutzt werden kann. Geben Sie `locknb` (No Block) an, kehrt die Funktion mit dem Wert `FALSE` zurück, wenn die Datei bereits von einem anderen Programm gesperrt ist. Möchten Sie die Verriegelung wieder freigeben, geben Sie `LOCK_UN` (Unlock) an.
- Die Funktion liefert bei Erfolg den Wert `TRUE` zurück bzw. `FALSE`, wenn ein Fehler auftrat. Ob die Zugriffssicherung funktioniert, erfahren Sie, indem Sie den Rückgabewert überprüfen.
- Die Sperre wird über die Funktion `fclose()` oder am Ende des Skripts automatisch aufgehoben.
- Bei manchen Betriebssystemen ist die Funktion `flock()` auf Prozess-Ebene (Teil einer Applikation) implementiert. Hierbei können Sie sich nicht auf `flock()` verlassen, um Dateien vor dem Zugriff von anderen PHP-Skripten zu schützen. Zudem wird `flock()` nicht von allen Dateisystemen unterstützt und gibt deshalb in solchen Umgebungen immer `FALSE` zurück.

Den Dateizeiger mit "fseek()" an eine bestimmte Position setzen

Im vorher gezeigten Beispiel ist es notwendig, den Dateizeiger an den Anfang der Datei zu setzen, um den alten Wert des Zählers mit dem neuen Wert überschreiben zu können. Den Dateizeiger können Sie über `fseek()` positionieren.

```
fseek (Dateizeiger,Stelle[,Wie]);
```

- Hierbei geben Sie für den Parameter Stelle die Anzahl in Bytes, bezogen auf den Dateianfang, der durch den Dateizeiger festgelegt wird, an.
- Der optionale Parameter `[,Wie]` legt bestimmte Bezugspunkte für die Ermittlung der Position fest:
 - Der Parameter `SEEK_CUR` ermöglicht die Verschiebung von der aktuellen Position des Dateizeigers plus Stelle, `SEEK_END` vom Dateiende und `SEEK_SET` vom Dateianfang.
- Fehlt der Parameter Wie, ist standardmäßig die Option `SEEK_SET` gewählt.

From:

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

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_36



Last update: **2021/11/10 08:24**

1.10.7) Übungsaufgaben

Aufgabe 23: Textdatei einlesen und zeilenweise ausgeben

Gedicht.txt

```
Wer will was Lebendig's erkennen und beschreiben,  
Sucht erst den Geist heraus zu treiben,  
Dann hat er die Teile in seiner Hand,  
Fehlt leider! nur das geistige Band
```

Aufgabe 24: Apfel-Bestellungen aufgeben und csv-Datei ausgeben

[bestellung_daten.csv](#)

Aufgabe 25: Umfrage mit Gewinnspiel

a) Du möchtest eine Umfrage durchführen. Hierfür erstellst Du ein Formular in das Daten eingegeben werden. Um die Daten auswerten zu können, schreibe ein PHP-Skript, das die Daten an eine Tabelle (*.csv) übergibt und dort speichert.

Das Umfrageformular, das auch das PHP-Script beinhalten soll, speichere unter dem Namen `umfrage.php` und die übergebenen Daten werden in der Datei `umfrage_daten.csv` gespeichert. Hinweis: Die gewählten Optionen werden als Zahlen in die Datendatei geschrieben (1 für 1. Option, 2 für 2. Option, ...).

Firefox ➔

Übung Kapitel 9

localhost/herdt_php54_grundlagen/uebungen_ergebnisse/kap09/formular_umfrage.html

Google

Umfrage mit Gewinnspiel

Wir freuen uns, dass Sie an unserer kleinen Umfrage zu unserer Webseite teilnehmen! Unter allen Teilnehmern verlosen wir drei Präsentkörbe.

Vorname und Nachname

Straße

PLZ und Ort

Bitte wählen Sie die Antwort aus, die für Sie am ehesten zutrifft:

Wie gefällt Ihnen unser Internetangebot?

sehr gut
 gut
 nicht so gut
 gar nicht

Wie beurteilen Sie den Informationsgehalt unserer Webseite?

sehr informativ
 die eine oder andere Information fehlt
 es fehlen sehr viele wichtige Informationen

Wie kommen Sie mit dem Bestellsystem zurecht?

sehr gut
 gut
 nicht besonders gut
 gar nicht

Möchten Sie uns noch etwas mitteilen?

Absenden

b) Baue (wie bei Aufgabe 2 aus der Schulübung) die Datei `umfrage.php` zur einem Verwaltungstool aus, dass es ermöglicht, neben der Eingabe von neuen Daten auch die Datei `umfrage_daten.csv` übersichtlich am Schirm auszugeben.

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

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_37



Last update: **2021/11/10 08:27**

1.11.1) Datum und Zeit

getdate()

- \$datum = `getdate()` liefert Informationen zur aktuellen Zeit, zum Wochentag, Monat und vielem mehr. Diese Funktion liefert das Ergebnis in einem assoziativen Feld zurück.
- Der Anweisung `getdate()` kann man den Parameter `Zeitstempel` hinzufügen, mit dem man das auszulesende Datum angibt. Der Parameter `Zeitstempel` erwartet als Wert die Sekundenanzahl seit dem 01.01.1970. Sämtliche Zeitangaben beruhen auf diesem Datum, das auch als Beginn der UNIX-Epoche bezeichnet wird.
- `print_r($datum)` veranschaulicht den Rückgabewert der Variablen `$datum`.

Übung 1

Versuche mittels `$datum = getdate()` folgende Ausgabe des aktuellen Datums durchzuführen:

Stunde: 12

Minute: 36

Sekunde: 20

Tag der Woche: 3 = Wednesday

Tag des Monats: 17

Tag des Jahres: 321

Monat: 11 = November

Jahr: 2010

Lösung

```
<?php
$datum=getdate();
echo "Sekunden ".$datum[seconds] . "<br>";
echo "Minuten ".$datum[minutes] . "<br>";
echo "Stunden ".$datum[hours] . "<br>";
echo "Tag des Monats ".$datum[mday] . "<br>";
echo "Tag der Woche ".$datum[wday] . "<br>";
echo "Monat ".$datum[mon] . "<br>";
echo "Jahr ".$datum[year] . "<br>";
echo "Tag des Jahres ".$datum[yday] . "<br>";
echo "Wochentag ".$datum[weekday] . "<br>";
echo "Monat ".$datum[month] . "<br><br>";
?>
```

Datum und Zeit formatieren

Die `date()`-Anweisung erwartet als Parameter die Formatanweisungen, welche Datum-Segmente angezeigt werden sollen.

Bsp: echo **date("l d")**." of ". **date("F Y h:i:s A")**; liefert:
 Wednesday 17 of November 2010 12:36:16 PM

Übung 2

Versuche mittels **date()** folgende Ausgabe zu erstellen:

17.11.10 um 12:36:16
 17.11.10 = 321. Tag des Jahres 2010
 17-11-2010
 2010-11-17

Lösung

```
<?php
echo date("d.m.y"). " um ".date("H:i:s")."<br>";
echo date("d.m.y"). " = ".date("z").". Tag des Jahres ".date("Y")."<br>";
echo date("d-m-Y")."<br>";
echo date("Y-m-d")."<br>";
?>
```

Mit **date()** kann man eine Zeitangabe formatieren oder auswerten. Die Zeitangabe übergeben Sie im Parameter **timestamp**. Lassen Sie diesen Parameter leer, nimmt die Funktion die aktuelle Zeit. Der Parameter **format** ist ein String, der festlegt, welche Informationen über die Zeitangabe Sie benötigen. In diesem String sind folgende Platzhalter möglich (*: Ausgabe mit führenden Nullen):

- a - „am“ oder „pm“
- A - „AM“ oder „PM“
- B - Tage bis Jahresende
- d - Tag des Monats *(01 - 31)
- D - Tag der Woche (Wed - 3stellig)
- F - Monatsangabe (December - ganzes Wort)
- g - Stunde im 12-Stunden-Format (1-12)
- G - Stunde im 24-Stunden-Format (0-23)
- h - Stunde im 12-Stunden-Format *(01-12)
- H - Stunde im 24-Stunden-Format *(00-23)
- i - Minuten *(00-59)
- l - (großes i) 1 bei Sommerzeit, 0 bei Winterzeit
- j - Tag des Monats (1-31)
- l - (kleines L) ausgeschriebener Wochentag (Monday)
- L - Schaltjahr = 1, kein Schaltjahr = 0
- m - Monat *(01-12)
- n - Monat (1-12)
- M - Monatsangabe (Feb - 3stellig)
- O - Zeitunterschied gegenüber Greenwich (GMT) in Stunden (z.B.: +0100)
- r - Formatiertes Datum (z.B.: Tue, 6 Jul 2004 22:58:15 +0200)
- s - Sekunden *(00 - 59)
- S - Englische Aufzählung (th für 2(second))
- t - Anzahl der Tage des Monats (28 - 31)
- T - Zeitzoneneinstellung des Rechners (z.B. CEST)

- U - Sekunden seit Beginn der UNIX-Epoche (1.1.1970)
- w - Wochentag (0(Sonntag) bis 6(Samstag))
- W - Wochenummer des Jahres (z.B.: 28)
- Y - Jahreszahl, vierstellig (2001)
- y - Jahreszahl, zweistellig (01)
- z - Tag des Jahres (z.B. 148 (entspricht 29.05.2001))
- Z - Offset der Zeitzone gegenüber GMT (-43200 – 43200) in Minuten

Länder- und Spracheinstellungen ändern

Mit folgendem Code wird die deutsche Sprache für alle Werte eingestellt.

```
<?php
    setlocale(LC_ALL, "german");
?>
```

strftime(Format [,Zeitstempel])

- Die Anweisung erwartet als Parameter die Formatanweisungen
- Ohne die Angabe des zweiten Parameters Zeitstempel liefert die Funktion das aktuelle Datum zurück analog der date()-Funktion.

Zeitfunktionen

time()

- Diese Funktion ist ohne Parameter zu verwenden und gibt die Zeit, die seit dem 1. 1. 1970 um 00:00:00 Uhr vergangen ist, in Sekunden zurück.

strtotime()

- Die Funktion liefert die Anzahl der Sekunden zwischen dem 1.1.1970 und dem angegebenen Datum zurück.

```
<?php
    echo strtotime("07.05.2019 10:00:00");
?>
```

mktime([Stunde[, Minute[, Sekunde[, Monat[, Tag[, Jahr]]]]]);

- Die Funktion liefert die Anzahl der Sekunden zwischen dem 1. 1. 1970 und dem angegebenen Datum zurück.

```
<?php
    $tag    = 24;
```

```
$monat = 8;
$jahr = 1978;

$geburt = mktime(0,0,0,$monat,$tag,$jahr);
$differenz = time() - $geburt;

echo ((integer)($differenz/86400))." Tage liegen zwischen dem heutigen
Datum, ";
echo date("j.n.Y")." und dem ".date("j.n.Y", $geburt).".";
?>
```

Datumsangaben überprüfen

checkdate(Monat, Tag, Jahr)

- Als Rückgabewert liefert die Funktion den Wert TRUE, falls das Datum existiert, sonst FALSE.

Zeit mittels microtime

`microtime(true)` liefert den aktuellen UNIX-Zeitstempel mit Mikrosekunden

```
$zeitstempel=microtime(true);
```

Beispiel:

```
<?php
$time_start = microtime(true);

// Sleep for a while
usleep(100);

$time_end = microtime(true);
$time = $time_end - $time_start;

echo "Did nothing in $time seconds\n";
?>
```

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

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_38



Last update: **2021/11/10 08:28**

1.11.2) ÜBUNG 26 - GÄSTEBUCH ERWEITERUNG

Erweitere das Gästebuch aus der ÜBUNG 21 dahingehend, dass du nicht nur den Namen und den Text speicherst, sondern auch das Datum des Eintrages.

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



Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_39

Last update: **2021/11/10 08:28**

1.12.1) Funktionen erstellen und aufrufen

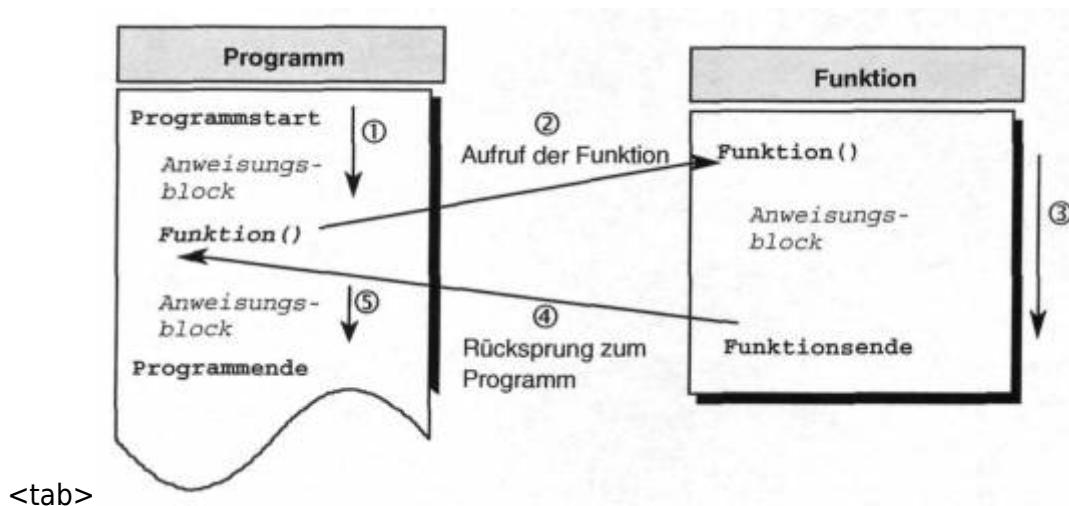
Was sind Funktionen?

Funktionen sind eigenständige Programmteile, die vom Skript beliebig oft aufgerufen und abgearbeitet werden können. Funktionen beinhalten Anweisungen, die innerhalb des Programms mehrmals benötigt werden. Anstatt die Anweisungen mehrfach im Programm zu codieren, wird die entsprechende Funktion aufgerufen, die die gewünschten Anweisungen durchführt.

PHP bietet eine Reihe vordefinierter Funktionen, um bestimmte Standardaufgaben zu lösen. Funktionen, die Sie selbst zur Lösung Ihrer Aufgaben erstellen, werden benutzerdefinierte Funktionen genannt.

Vorteile von Funktionen

- Immer wiederkehrende Abläufe werden nur einmal beschrieben und können danach beliebig oft ausgeführt werden.
- Der Programm-Code wird mithilfe von Funktionen strukturiert und ist dadurch einfacher zu pflegen.
- Änderungen am Programm, die zu einem späteren Zeitpunkt auftreten, lassen sich schneller durchführen, da sich eine Änderung in der Funktion auch auf den Quellcode auswirkt, der diese Funktion verwendet.
- Die Struktur des Programms lässt sich leichter nachvollziehen, da der Quellcode übersichtlicher ist.



Eine Funktion wird erst ausgeführt, wenn sie im Programm aufgerufen wird. Dies geschieht über den Namen der Funktion.

Das PHP-Programm wird bis zum Aufruf einer Funktion abgearbeitet (1).

Der Funktionsaufruf (2) erzwingt einen Sprung in die angegebene Funktion.

Jetzt werden die Anweisungen der Funktion abgearbeitet (3).

Mit dem Verlassen der Funktion wird zurück zum Programm gesprungen (4).

Das Programm wird weiter ausgeführt (5).

Eine Funktion erstellen

Syntax und Beschreibung der function-Anweisung

```
function Name([Parameter])
{Anweisungsblock;
}
```

- Die function-Anweisung leitet eine Funktion ein.
- Name ist die Bezeichnung der Funktion und sollte einen Bezug zum Inhalt haben, z. B. können Sie eine Funktion, die das Quadrat einer Zahl berechnet, *BerechneQuadratzahl* oder *Quadratzahl* nennen.
- Für den Namen einer Funktion gelten folgende Regeln:
 - Das erste Zeichen muss entweder ein Buchstabe oder ein Unterstrich sein.
 - Der Name darf nur aus Buchstaben und Ziffern bestehen, aber keine Umlaute oder „ß“ und keine Sonderzeichen außer dem Unterstrich „_“ enthalten.
 - Der Name kann Groß- oder Kleinbuchstaben enthalten, wobei zwischen Groß- und Kleinschreibung unterschieden wird: `function Quadratzahl()` ist nicht gleich `function quadratzahl()`.
 - Der Name darf nicht identisch sein mit einem so genannten reservierten Wort (z. B. Befehl aus PHP).
- In den runden Klammern werden die Bezeichnungen der einzelnen Parameter angegeben, für die beim Aufruf der Funktion Werte übergeben werden können.
- Eine Funktion kann keinen, einen oder mehrere Parameter besitzen.

Eine Funktion mit return-Anweisung erstellen

Syntax und Beschreibung der return-Anweisung

- Mit der return-Anweisung kann ein Wert zurückgeliefert werden. Sobald die return-Anweisung ausgeführt wird, wird eine Funktion verlassen und an die aufrufende Stelle zurückgekehrt.
- Die return-Anweisung kann ohne Rückgabewert auch ausschließlich zum Verlassen einer Funktion verwendet werden.

```
function Name([Parameter])
{Anweisungsblock;
 return [$Wert];
}
```

Eine Funktion aufrufen

Syntax und Beschreibung eines Funktionsaufrufs

- Eine Funktion können Sie aus einer beliebigen Stelle des PHP-Codes heraus aufrufen.
- Sie können eine Funktion auch aus einer anderen Funktion heraus aufrufen.
- Eine Funktion wird mit ihrem Namen aufgerufen.

- Die runden Klammern nach dem Funktionsnamen können, falls benötigt, Parameter zur Übergabe von Werten enthalten. Falls keine Werte übergeben werden, bleiben die Klammern leer.

```
<?php
Anweisungsblock
Funktionsname( );
Anweisungsblock
?>
```

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



Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_40

Last update: **2021/11/10 08:29**

1.12.2) Mit Funktionen arbeiten

Je nach Verwendungszweck einer Funktion kann eine Funktion mit oder ohne Parameter bzw. mit oder ohne Rückgabewert erforderlich sein.

Funktionen ohne Parameter

Bei einer Funktion ohne Parameter werden bei jedem Aufruf dieselben Anweisungen ausgeführt.

Beispiel: funktionen1.php

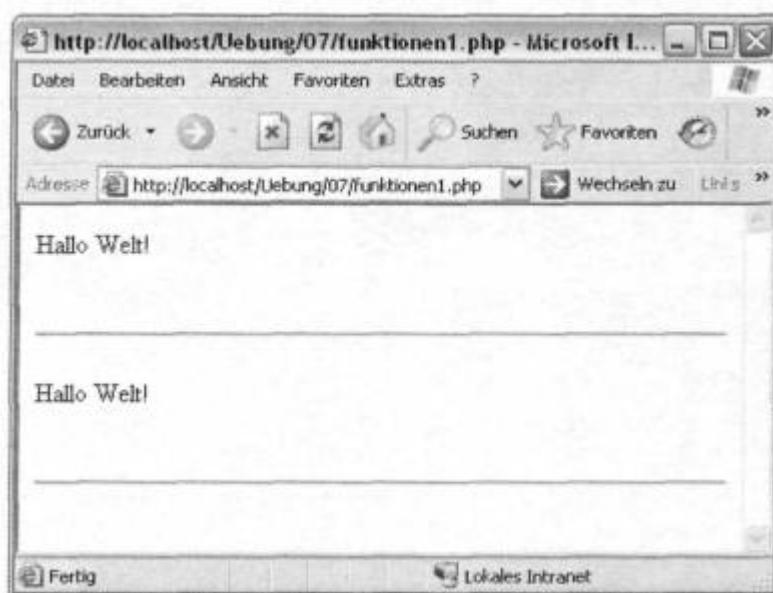
```
<?php

function Text()
{
    echo "<p>Hallo Welt!</p>";
}

function Linie()
{
    echo "<p><hr></p>";
}

Text();
Linie();
Text();
Linie();

?>
```



Funktionen mit einem oder mehreren Parametern

Bei einer Funktion mit einem oder mehreren Parametern werden bei jedem Aufruf der Funktion in Abhängigkeit von den Parametern ähnliche Aufgaben ausgeführt. Hierbei kann ein Parameter eine Konstante oder eine Variable sein.

Wenn Sie eine Funktion mit mehreren Parametern aufrufen, müssen die Anzahl und die Reihenfolge der Parameter übereinstimmen.

Beispiel: funktionen2.php

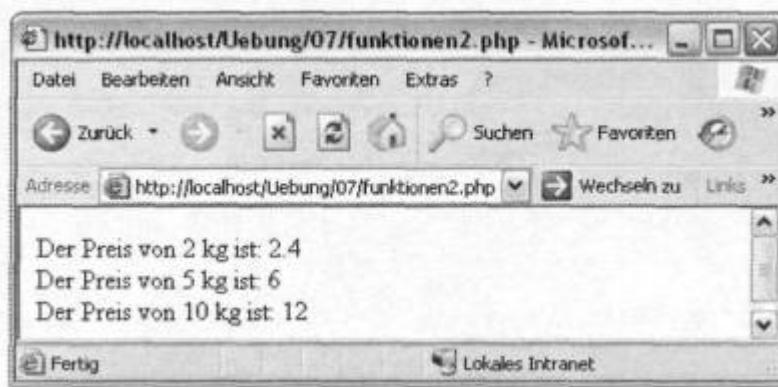
Im folgenden Skript werden die Preise pro kg mithilfe einer Funktion mit einem Parameter berechnet.

```
<?php

function Preis($menge)
{
    $betrag = 1.2*$menge;
    echo "Der Preis von $menge kg ist: $betrag<br>";
}

Preis(2);
Preis(5);
Preis(10);
$einkaufsmenge=20;
Preis($einkaufsmenge);
?>
```

Die Funktion `Preis()` wird dreimal aufgerufen. Der übergebene Parameter (der Wert in der Klammer) ist bei jedem Aufruf ein anderer. Der Wert in der Klammer wird der Variablen `$menge` übergeben. Mithilfe der Variablen `$menge` wird beim Aufruf der Funktion die Variable `$betrag` berechnet und am Bildschirm ausgegeben.



Beispiel: funktionen3.php

Wird eine Funktion mit mehreren Parametern aufgerufen, wird der erste Wert an den ersten Parameter, der zweite Wert an den zweiten Parameter usw. übergeben.

```
<?php
```

```

function Preis($sorte,$preis,$menge)
{
    $betrag = $preis * $menge;
    echo "Der Preis von $menge kg $sorte ist: $betrag<br>";
}

Preis("Jonagold",1.5,5);
Preis("Gala",1.65,10);
Preis("Elstar",2.00,4);

?>

```

Die Funktion Preis() wurde mit 3 Parametern versehen: \$sorte, \$preis, \$menge.
Beim Aufruf der Funktion werden durch Kommata getrennt drei Parameter übergeben.



Veränderung von Parametern innerhalb einer Funktion

Wenn Sie eine Variable als Parameter an eine Funktion übergeben, dann erhält die Funktion eine Kopie der übergebenen Variablen. Die Übergabe der Parameter als Kopie wird auch als **call-by-value** bezeichnet. Um den Wert einer übergebenen Variablen zu ändern, verändern Sie nur ihre lokale Kopie. Eine Veränderung der Kopie hat keinerlei Rückwirkung auf das Original.

Manchmal ist aber eine Veränderung der übergebenen Variablen erwünscht, beispielsweise um den Preis einer Bestellung abhängig zu machen von der vorgegebenen Menge. Um dies zu erreichen, übergeben Sie den Parameter als Referenz auf das Original. Sie übergeben also nicht, wie bei dem call-by-value-Verfahren, den Inhalt bzw. den Wert einer Variablen, sondern einen Verweis auf die Variable. Diese Vorgehensweise wird **call-by-reference** genannt. Eine Veränderung hat, anders als bei der Übergabe als Kopie, Rückwirkung auf das Original. Eine Referenz auf das Original erhalten Sie, indem Sie vor den Variablennamen ein & setzen, z. B. &\$preis.

Beispiel: byreference.php

```

<?php

function Preis($sorte, &$preis, $menge)
{
    if ($menge > 10)
    {
        $preis = 1;
    }
}

```

```

    $betrag = $preis * $menge;
    echo "Der Preis von $menge kg $sorte ist: $betrag EUR <br>";
}

$sorte = "Elstar";
$preis = 2;
$menge = 8;
Preis($sorte, $preis, $menge);

$sorte = "Elstar";
$preis = 2;
$menge = 15;
Preis($sorte, $preis, $menge);
?>

```

- Die Funktion Preis() wurde mit 3 Parametern versehen: \$sorte, \$preis, \$menge. Die Variable \$preis wird mithilfe des vorangestellten & als Referenz gekennzeichnet.
- Mithilfe einer if-Schleife wird festgelegt, dass der Wert der Variablen \$preis = 1, wenn \$menge > 10 ist. Beim Aufruf der Funktion werden drei Parameter durch Kommata getrennt übergeben. Da der Wert der Variablen \$menge <= 10 ist, wird zur Preisberechnung der ursprüngliche Wert der Variablen \$preis = 2 verwendet. Wenn aber der Wert der Variablen \$menge > 10 ist, wird der Wert der Variablen \$preis aufgrund der Bedingung in der if-Schleife auf 1 verändert.

Funktionen mit Rückgabewerten

Wenn Sie innerhalb einer Funktion ein Ergebnis ermitteln und dies an den Aufrufer zurückliefern möchten, verwenden Sie die return-Anweisung.

Allerdings ist hier zu beachten, dass eine return-Anweisung nicht nur den Rückgabewert festlegt, sondern auch die Ausführung der Funktion sofort beendet. Alle Anweisungen nach einer return-Anweisung werden nicht mehr ausgeführt, und der vorhandene Rückgabewert wird an das Programm ausgegeben.

Beispiel: funktionen4.php

```

<?php
function Summe ($zahl1, $zahl2)
{
    $ergebnis = $zahl1 + $zahl2;
    return $ergebnis;
}

$summel = Summe(10,20);
echo "Summe1: $summel<br>";
$zahl3 = 0.11;
$summe2 = Summe (19,$zahl3);
echo "Summe2: $summe2<br>";
?>

```

- Die Funktion Summe() hat die beiden Parameter \$zahl1 und \$zahl2.

- Die beiden Parameter werden innerhalb der Funktion addiert und in der Variablen \$ergebnis gespeichert.
- Der Wert der Variablen \$ergebnis wird mithilfe der Anweisung return an die aufrufende Stelle zurückgeliefert.
- Der Funktionsaufruf Summe(10,20) liefert in \$ergebnis den Rückgabewert zurück. Dieser Wert wird der Variablen \$summe zugewiesen.
- Parameter für Funktionen können sowohl als konstante Werte, als auch über Variablen übergeben werden.

PHP kann nur einen Wert mit `return()` zurückgeben. Wenn Sie mehrere Werte aus einer Funktion zurückgeben möchten, können Sie diese Werte in einem Array zusammenfassen.

Um Variablen die Elemente des zurückgelieferten Arrays zuzuweisen, verwenden Sie die `list()`-Anweisung.

```
<?php

function zahlen()
{
    $ret = array(2,4,6);
    return $ret;
}

list($var1,$var2,$var3) = zahlen();
echo "Die zurückgelieferten Variablen sind: $var1, $var2, $var3";
?>
```

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

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_41



Last update: **2021/11/17 11:41**

1.12.3) Der Gültigkeitsbereich von Variablen

Auf Variablen, die Sie außerhalb einer Funktion im PHP-Programm einsetzen, können Sie in Funktionen nicht ohne weiteres zugreifen und mit ihnen arbeiten. Ebenso gibt es innerhalb einer Funktion Variablen, auf die Sie außerhalb der Funktion nicht zugreifen können.

Variablen haben, je nachdem, wo sie definiert sind, unterschiedliche Gültigkeitsbereiche.

- **Lokale Variablen** sind nur innerhalb der Funktion, in der sie definiert wurden, gültig.
- **Globale Variablen**, die außerhalb von Funktionen definiert wurden, haben standardmäßig nur außerhalb der Funktionen ihren Gültigkeitsbereich. Die globalen Variablen stehen innerhalb einer Funktion nicht zur Verfügung. Sollen die globalen Variablen innerhalb einer Funktion gültig sein, müssen sie dort mit dem Schlüsselwort `global` bekannt gemacht werden, z. B. `global $zahl`.
- **Superglobale Variablen** stehen sowohl innerhalb als auch außerhalb von Funktionen zur Verfügung, beispielsweise das assoziative Feld `$_POST`.

Im folgenden Beispiel werden globale und superglobale Variablen gegenübergestellt. Hierzu wird ein Bestell-Formular für Äpfel erstellt. Anhand des PHP-Codes im Auswertungsprogramm wird gezeigt, wann welche Variablen gültig sind.

Beispiel: formular_funktion.html, funktion_var.php

```
<html>
<body>
<h2>Apfelkauf</h2>
Bitte geben Sie die gewünschte Menge ein und wählen Sie eine Apfelsorte:
<form action="funktion_var.php" method="post">
Menge: <input name="menge"><br><br>
Apfelsorte:<br>
<input type="radio" name="sorte" value="Jonagold">Jonagold
<input type="radio" name="sorte" value="Gala">Gala
<input type="radio" name="sorte" value="Elstar">Elstar<br><br>
<input type="Submit" value="Abschicken">
<input type="Reset" value="Zurücksetzen">
</form>
</body>
</html>
```

- Mithilfe der Methode POST werden nach Absendung des Formulars folgende Variablen an das Auswertungsprogramm `funktion_var.php` weitergegeben:
- die Variable `$menge`, die vom Benutzer eingetragen wird, und
- die Variable `$sorte`, die der Benutzer mithilfe eines Optionsfeldes auswählt

Apfelkauf

Bitte geben Sie die gewünschte Menge ein und wählen Sie eine Apfelsorte:

Menge:

Apfelsorte:

Jonagold Gala Elstar

Fertig Lokales Intranet

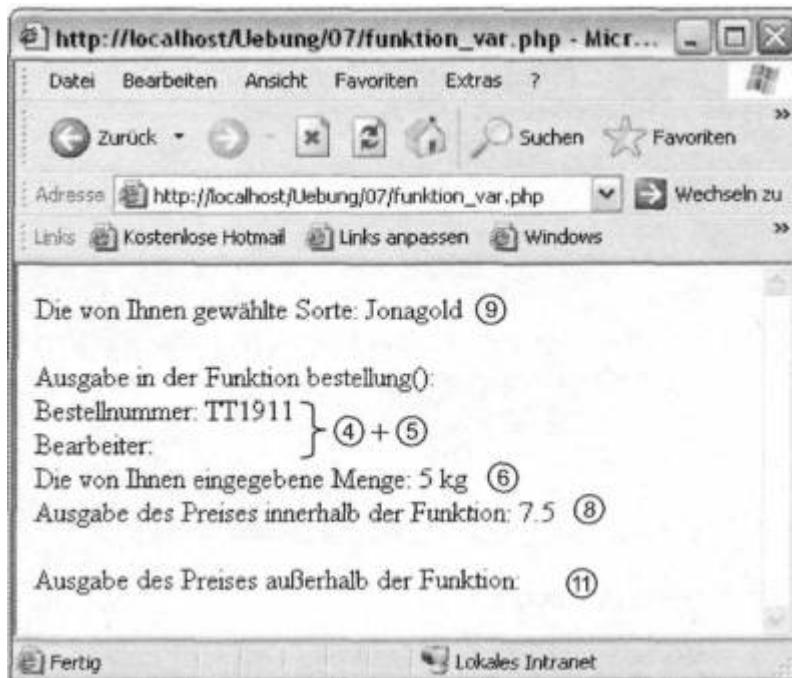
```
<?php
$bestellnummer = "TT1911";
$bearbeiter = "Mario Caspari";

function bestellung()
{
    global $bestellnummer;      // globale Variable
    echo "Bestellnummer: " . $bestellnummer . "<br>";
    echo "Bearbeiter: " . $bearbeiter . "<br>";
    echo "Die von Ihnen eingegebene Menge: " . $_POST["menge"] . " kg<br>";
    switch($_POST["sorte"])
    {
        case "Jonagold": $preis=$_POST["menge"]*1.50; // lokale Variable
                          break;
        case "Gala":     $preis=$_POST["menge"]*1.65; // lokale Variable
                          break;
        case "Elstar":   $preis=$_POST["menge"]*2.00; // lokale Variable
                          break;
    }
    echo "Ausgabe des Preises innerhalb der Funktion: $preis<br>";
}

echo "Die von Ihnen gewählte Sorte: " . $_POST["sorte"] . "<br>";
bestellung();
echo "Ausgabe des Preises außerhalb der Funktion: $preis<br>";
?>
```

- Die Variablen \$bestellnummer und \$bearbeiter werden als **globale** Variablen im Programm definiert.
- Die **globale** Variable \$bestellnummer wird innerhalb der Funktion bestellung() mithilfe des Schlüsselwortes global bekannt gemacht.
- Die Variablen \$bestellnummer und \$bearbeiter sollen innerhalb der Funktion ausgegeben werden. Da die Variable \$bearbeiter innerhalb der Funktion nicht mithilfe des Schlüsselwortes global bekannt gemacht wurde, wird sie bei der Ausgabe nicht angezeigt.

- Der Wert des **superglobalen** Feldes `$_POST["menge"]` wird innerhalb der Funktion `bestellung()` ausgegeben.
- Innerhalb der Funktion `bestellung()` wird der Wert der **lokalen** Variablen `$preis` berechnet.
- Die **lokale** Variable `'$preis'` wird innerhalb der Funktion ausgegeben.
- Der Wert des **superglobalen** Feldes `$_POST["sorte"]` wird außerhalb der Funktion `bestellung()` ausgegeben.
- Die Funktion `bestellung()` wird aufgerufen.
- Die **lokale** Variable `$preis` soll auch außerhalb der Funktion ausgegeben werden.



From:

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

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_42



Last update: **2021/11/10 08:30**

1.12.4) PHP-Dateien einbinden mit include und require

Manchmal ist es sinnvoll, wenn Sie selbst definierte Funktionen in eine separate PHP-Datei auslagern, um sie in mehreren Skripten nutzen zu können.

Mithilfe der `include`- bzw. `require`-Anweisung können Sie Dateien, z.B. Funktionen, die Sie benötigen, in Ihr PHP-Programm einbinden.

Mit `include` und `require` arbeiten

Die `include()`- bzw. die `require()`-Anweisung binden eine bestimmte Datei, deren Pfad als Argument übergeben wird, in den aktuellen Programmcode ein.

Unterschiede zwischen `include` und `require`

Der Unterschied zwischen den beiden Anweisungen zeigt sich, wenn eine Datei fehlerhaft eingebunden wurde:

- Wenn beispielsweise die angegebene Datei nicht vorhanden ist, erzeugt `include()` eine Warnung. Das Skript wird weiter ausgeführt.
- Der Befehl `require()` beendet bei einer fehlenden Datei das Skript sofort mit einer Fehlermeldung.

Syntax und Bedeutung der `include()`- und `require()`-Anweisungen

- Die `include`- und `require`-Befehle erwarten als Parameter die einzubindende Datei. Befindet sich die Datei nicht im selben Ordner, muss der Pfad angegeben werden.
- Beinhaltet die eingebundene Datei keinen PHP-Code, wird der Inhalt unverändert an den Browser weitergegeben und somit am Bildschirm dargestellt.
- Ist in der Konfigurationsdatei `php.ini` im Bereich `fopen wrappers` die Option `allow_url_fopen = on` gesetzt, so können Sie auch einen URL als Parameter angeben.
- Eingebundene Dateien, die eine Funktion mit Rückgabewert enthalten, werden ebenso behandelt wie Funktionen mit Rückgabewert.

So kann z. B. der Rückgabewert in einer Variablen gespeichert werden mit: `$variable = include(Datei)`.

Innerhalb der eingebundenen Dateien muss der PHP-Code von gültigen PHP-Tags `<?php ... ?>` umschlossen sein. Befindet sich kein PHP-Code in den Dateien, können die Tags weggelassen werden.

Um Skripte, die später in einen PHP-Code eingebunden werden sollen, zu kennzeichnen, ist es empfehlenswert, die Dateierweiterung `.inc.php` zu vergeben. Die Bezeichnung `.inc.php` steht hierbei für die Abkürzung von `include`.

Datei mit include() einbinden

Beispiel: include.php

Zur Veranschaulichung der Arbeitsweise der include()-Anweisung erstellen Sie ein Loginverfahren. Die Prüfung, ob der eingegebene Nickname und das Kennwort korrekt sind, wird in die Datei prueflogin.inc codiert. Vom Loginformular formular_login.html wird die Datei include.php aufgerufen.

Datei: include.php

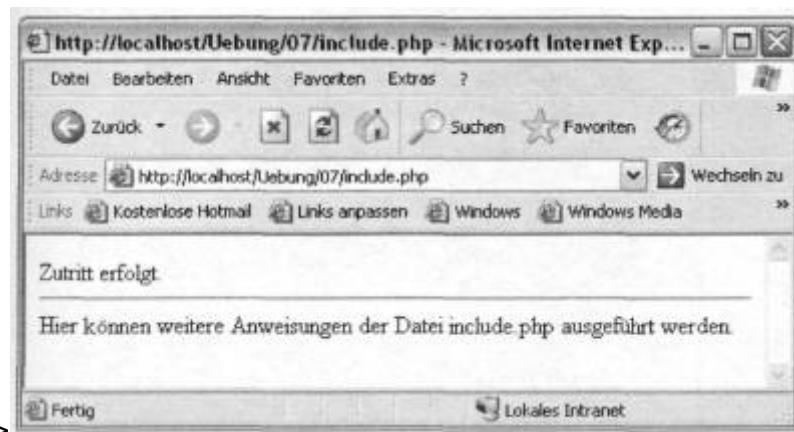
```
<html>
<body>
<h4> Loginergebnis über include </h4>
<?php
echo "Start<br>Nun wird ein PHP-Skript eingebunden:<hr>";
echo "Die Rückgabe der Datei ergibt:<br>";
include ("prueflogin.inc");
echo "<hr>Hier können weitere Anweisungen der Datei include.php ausgeführt werden.";
?>
</body>
</html>
```

Die include()-Anweisung bindet die Datei prueflogin.inc ein.

Datei: prueflogin.inc

Innerhalb der Datei prueflogin.inc wird die Kennwortprüfung vorgenommen.

```
<?php
if ($_POST["nickname"] == "Mario" && $_POST["kennwort"] == "geheim")
    echo "Zutritt erfolgt";
else
    echo "Zutritt erfolgt nicht";
?>
```



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



Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_43

Last update: **2021/11/10 08:31**

1.12.5) Andere Dateitypen einbinden

Sie können auch andere Dateien, z. B. reine Text- oder HTML-Dateien, in Ihr Programm einbinden.

Innerhalb eines PHP-Skripts soll beispielsweise eine einfache Textdatei eingebunden werden. Der Inhalt der Textdatei wird sofort am Bildschirm ausgegeben. Zuvor wird eine Textdatei mit folgendem Inhalt gespeichert.

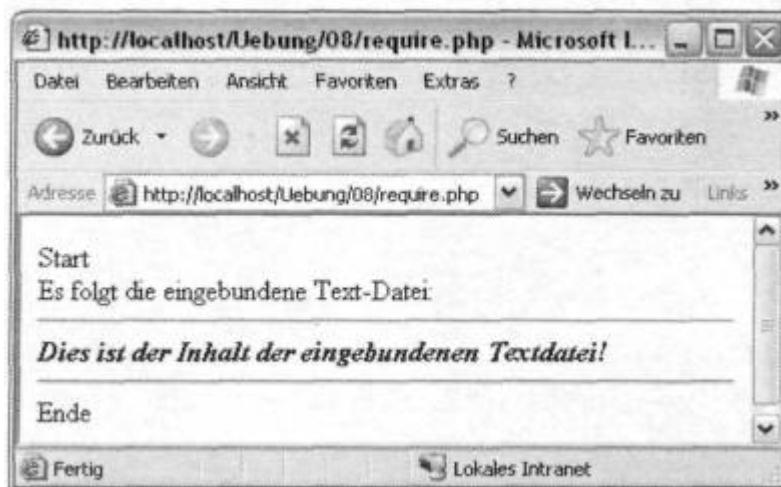
Datei: einfueg.txt

```
<b><i>Dies ist der Inhalt der eingebundenen Textdatei!</i></b>
```

Beispiel: require.php

```
<html>
<body>
<h4>Eine Textdatei über den Befehl require einbinden</h4>
<?php
echo "Start<br>Es folgt die eingebundene Text-Datei:<hr>";
require("einfueg.txt") ;
echo "<hr>Ende";
?>
</body>
</html>
```

Inmitten des PHP-Codes wird über den Befehl require () eine Textdatei eingebunden. Da diese keinen weiteren PHP-Code, sondern nur Text mit HTML-Befehlen beinhaltet, wird diese von PHP direkt an den Browser weitergeleitet.



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



Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_44

Last update: **2021/11/10 08:31**

1.12.6) Übungen zu Funktionen

Aufgabe 27 - Maximum

Erstelle eine PHP-Seite (mit einer Funktion) welches folgende Aufgabe realisiert:

1. Zahl: -Benutzereingabe-
 2. Zahl: -Benutzereingabe-
- größte Zahl: -gibt die größere der beiden Zahlen aus

Aufgabe 28 - Bruch

Schreibe eine PHP-Seite, in der ein Bruch soweit als möglich gekürzt wird. Dazu gibt der Benutzer einen Zähler und Nenner ein. Mit Hilfe einer Funktion, die den ggT der beiden Zahlen mit dem euklidischen Algorithmus berechnet, soll der Bruch gekürzt und wieder ausgegeben werden.

Aufgabe 29 - Quader

Schreibe eine PHP-Seite, in der alle möglichen Berechnungen für einen Quader durchgeführt werden.

- Lese dazu vom Benutzer die Länge, Breite und Höhe ein.
- Überprüfe ob alle Werte eingegeben wurden
- Überprüfe mit `is_numeric()` ob die eingegebenen Werte Zahlen sind
- Berechne in einer Funktion die Grundfläche
- Berechne in einer Funktion das Volumen
- Berechne in einer Funktion die Oberfläche
- Berechne in einer Funktion die Raumdiagonale
- Speichere bei jedem Programmaufruf, die Ergebnisse wie folgt in eine CSV-Datei.

Bispielhafte CSV-Datei:

```
Laenge;Breite;Hoehe;Grundflaeche;Volumen;Oberflaeche;Raumdiagonale
3;5;3;15;45;78;6.557;
3;2;2;6;12;32;4.123;
```

- Gib am Ende jeweils die ganze CSV-Datei als Tabelle aus!

Laenge	Breite	Hoehe	Grundflaeche	Volumen	Oberflaeche	Raumdiagonale
3	5	3	15	45	78	6.557
3	2	2	6	12	32	4.123
...						

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



Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_45

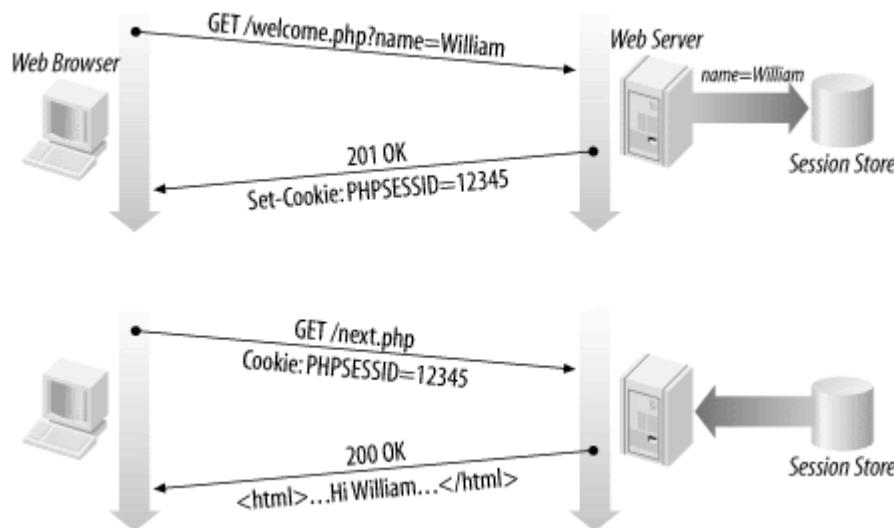
Last update: **2021/11/10 08:32**

1.13.1) Grundlagen zu Sessions

Daten, die ein Benutzer beispielsweise in ein Formular eingegeben hat, werden in Variablen nur bis zum auswertenden PHP-Programm weitergegeben. Oft ist es wünschenswert, auf solche Variablen auch zugreifen zu können, wenn der Benutzer weitere Seiten der Website besucht.

Gerade bei so genannten personalisierten Seiten, die auf den Benutzer „reagieren“, ist es notwendig, bestimmte Daten von Seite zu Seite mitzuführen bzw. weiterzugeben.

PHP unterstützt diese Anforderungen mithilfe von Sessions (Sitzungen) und der Session-ID.



Mit Session-IDs arbeiten

Damit Informationen zu einem Besucher innerhalb einer Website seitenübergreifend verwaltet werden können, erhält ein Besucher beim ersten Zugriff auf die Website eine zufällige, eindeutige 32-stellige Zeichenkette, die sogenannte Session-ID.

Für die Dauer eines Besuches auf der Website bzw. die Zeit, bis der Benutzer den Browser wieder schließt, werden die Daten in einer sogenannten Session-Datei als Cookie auf dem Server gespeichert.

In manchen Systemen wird die Session-ID als GET-Parameter über die URL übergeben und ist somit in der Adressleiste des Browsers sichtbar.

Die Zeit, in der sich der Besucher auf der Website aufhält, wird Session genannt. Der Vorteil einer Session liegt vor allen Dingen darin, dass während einer Session die globalen Variablen der Website übernommen und somit verwendet werden können.

Auf dem Server ist dann für jede Session-ID lokal ein Speicher eingerichtet, der beliebige Variablen für den Besucher beinhalten kann. So können wir in der Session für einen Besucher beispielsweise abspeichern, mit welchem Benutzernamen sich dieser eingeloggt hat, welche Waren in seinem Warenkorb liegen usw.

Der Benutzer hat keine Möglichkeiten, die Variablen in seiner Session zu sehen oder zu manipulieren. Er besitzt nur die Information, welche Session ID ihm vom Webserver zugewiesen wurde.

Das sogenannte Durchschleifen der Session-ID ist relativ einfach zu realisieren, da diese Funktionalität intern von PHP verwaltet wird. Dazu werden die Daten der Session temporär auf dem Webserver gespeichert.

Sicherheit von Sessions

Sessions bieten zwar keine 100%ige Sicherheit, dennoch sind sie relativ sicher. Bei jedem Seitenaufruf teilt der Browser dem Webserver mit, welche Session-ID dieser besitzt. Nun kann ein böswilliger Besucher seine Session-ID einfach manipulieren und könnte die Session-ID von einem anderen Besucher angeben. Der Dieb könnte sich so als jemand anderes in einer Community ausgeben.

Wie bekommt der Angreifer denn die Session ID heraus?

Wie eingangs erwähnt werden Session-IDs entweder über die URL mittels dem Parameter ?PHPSESSID übergeben oder als Cookie beim Besucher gespeichert. Sollten die Session-IDs per URL übertragen werden, dann kann es passieren dass der Benutzer diese URL kopiert und beispielsweise diese per E-Mail oder Facebook an Freunde und Bekannte weiterversendet. Rufen die Freunde dann den Link auf, so benutzen diese die Session-ID weiter und sind mit einem anderen Account eingeloggt. Dies ist gerade problematisch, wenn man den Link auf einer öffentlichen Seite postet.

Werden stattdessen Cookies verwendet, was der Standard ist und bei allen Besuchern passiert die nicht explizit Cookies deaktiviert haben, muss der Angreifer an diesen Cookie gelangen um die Session zu stehlen. Dies geht beispielsweise über einen Trojaner auf dem Rechner oder über das Abhören der Leitung.

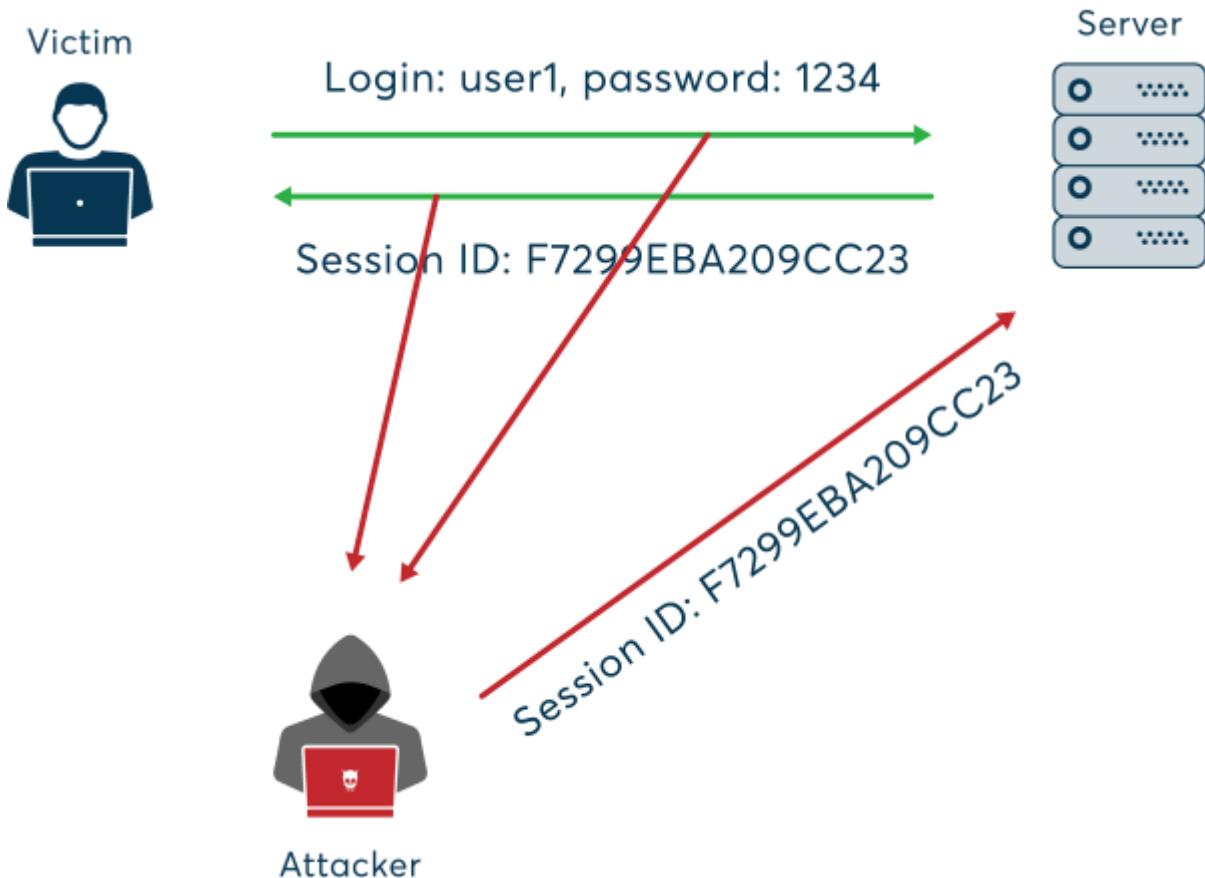
Session ID durch Zufall erraten

Natürlich kann der Dieb auch eine Session-ID erraten, allerdings ist dies sehr unwahrscheinlich. Es ist wahrscheinlicher, dass im Lotto mehrmals hintereinander die gleichen Zahlen gezogen werden als eine Session ID per Zufall zu erraten. Darum sollte wir uns eigentlich keine Sorgen machen.

Zusammenfassend ist zu sagen, dass Sessions schon recht sicher sind und wir uns um den Diebstahl einer Session-ID meistens keine Sorgen machen müssen. Sehr kritische Operationen, wie z.B. das Löschen des Accounts oder ähnliches, sollten aber mit einer zusätzlichen Passwortabfrage nochmals gesichert sein.



Gelingt es dem Dieb doch die Session-ID zu stehlen, so spricht man vom [Session-Hijacking](#).



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



Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_50

Last update: **2021/11/11 08:36**

Sessions registrieren

Möchten wir in einem Script auf Sessions zurückgreifen, müssen wir, bevor wir irgendeine Ausgabe machen, den Befehl `session_start()` aufrufen:

```
<?php
    session_start();
?>
```

Es empfiehlt sich, diesen Code immer ganz oben der Scripts stehen zu haben. Falls ihr die Fehlermeldung *Cannot send session cookie - headers already sent by* erhaltet, dann gab es irgendwo vor `session_start()` eine Ausgabe. Eine leere Zeile oder sogar ein Leerzeichen vor `<?php` reichen bereits aus.

Möchten wir einen Wert / eine Variable über mehrere Seitenaufrufe hinweg in der Session speichern, dann geht dies wie folgt:

```
<?php
    $_SESSION[ 'name' ] = "wert";
?>
```

Diesen Wert können wir später, auch auf anderen Seiten, wie folgt ausgeben:

```
<?php
    $name = $_SESSION[ 'name' ];
    echo $name;
?>
```

Wichtig, immer wenn ihr irgendwo mit Sessions arbeitet, muss zuvor `session_start()` ausgeführt worden sein. Probiert es aus, speichert in `seite1.php` einen Wert in eine Session und gebt dann in `seite2.php` diesen Wert wieder aus.

Ihr könnt dabei die Session-Variable wie jede andere Variable in PHP verwenden. Ihr könnt darin Zahlen, Zeichenketten oder sogar Arrays abspeichern.

Überprüfen ob Session-Variable registriert ist

Oft empfiehlt es sich zu überprüfen, ob eine gewisse Session-Variable bereits registriert wurde. Dies geht mittels der Funktion `isset($variable)`

```
<?php
    session_start();

    if (!isset($_SESSION['visited'])) {
        echo "Du hast diese Seite noch nicht besucht";
        $_SESSION['visited'] = true;
    }
    else {
```

```
        echo "Du hast diese Seite zuvor schon aufgerufen";
    }
?>
```

Hier wird zuerst überprüft ob es die Session-Variable schon gibt. Falls es sie nicht gibt, wird sie registriert.

Würden wir anstatt !isset nur isset schreiben, könnten wir überprüfen, ob die Session registriert wurde. Dies lässt sich bei Logins verwenden:

```
<?php
    session_start();

    if (isset($_SESSION['username'])) {
        echo "Herzlich Willkommen ".$_SESSION['username'];
    }
    else {
        echo "Bitte erst einloggen";
    }
?>
```

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

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_51



Last update: **2021/11/17 08:17**

Sessions löschen

Um alle Session-Daten der Sitzung zu löschen verwendet man `session_destroy()` :

```
<?php
    session_destroy();
?>
```

Dieser Befehl löscht alle Daten der Session und kann nützlich sein wenn sich der Benutzer z.B. aus eurem System ausloggen möchte. Denkt dran, ihr müsst zuvor `session_start()` ausführen bevor ihr `session_destroy()` ausführen könnt.

Um eine einzelne Session-Variablen zu löschen könnt ihr `unset($variable)` verwenden:

```
<?php
    unset($_SESSION['name']);
?>
```

Dieser Befehl löscht die Werte für die Session-Variable `name`. Hier ein kleines Beispiel:

```
<?php
    session_start();

    if(isset($_SESSION['besucht'])) {
        echo "Du hast die Seite zuvor besucht";
        unset($_SESSION['besucht']);
    }
    else {
        echo "Du hast die Seite zuvor NICHT besucht";
        $_SESSION['besucht'] = true;
    }
?>
```

Dieses Script springt bei jedem Aufruf zwischen den zwei Varianten hin und her.

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



Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_52

Last update: **2021/11/17 08:20**

Längeres Beispielscript mit Sessions

Um das gesamte Konzept von Sessions etwas verständlicher zu machen, folgt zum Abschluss noch ein kleines Beispiel. In einem Formular wird der Besucher seinen Namen eingeben können, es wird einen internen Bereich mit mehreren Seiten geben und eine Seite für den Logout.

formular.html

```
<form action="login.php" method="post">
    Dein Name: <br />
    <input type="Text" name="name" />
    <input type="Submit" />
</form>
```

login.php

```
<?php
    session_start(); //Nicht vergessen
    $name = $_POST['name'];

    if(!isset($name) OR empty($name)) {
        $name = "Gast";
    }

    //Session registrieren
    $_SESSION['username'] = $name;

    //Text ausgeben
    echo "Hallo $name <br />
    <a href=\"seite2.php\">Seite 2</a><br />
    <a href=\"logout.php\">Logout</a>";
?>
```

seite2.php

```
<?php
    session_start(); //Ganz wichtig

    if(!isset($_SESSION['username'])) {
        die("Bitte erst einloggen"); //Mit die beenden wir den weiteren
        Scriptablauf
    }

    //In $name den Wert der Session speichern
    $name = $_SESSION['username'];
```

```
//Text ausgeben
echo "Du heißt immer noch: $name
<a href=\"logout.php\">Logout</a>";
?>
```

logout.php

```
<?php
    session_start();
    session_destroy();
    echo "Logout erfolgreich";
?>
```

Dies ist ein ganz simples Beispiel, zeigt aber gleich die Stärke von Sessions. Bei dem Formular geben wir unseren Namen an, diese Daten werden dann an *login.php* gesendet. Dort werden sie abgefragt und der *name* aus dem Formular wird in der Session *username* gespeichert. Wenn wir dann auf den Link klicken und auf *seite2.php* gelangen, so können wir dort weiter den Namen des Besuchers (der Name aus dem Formular) ausgeben. Dies könnte man über beliebig viele Seiten weiterführen, mit unbegrenzt vielen Session-Variablen.

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

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_53



Last update: **2021/11/17 08:25**

Übung 30

Erstelle ein Bestellformular zur Bestellung von Weinen:

session1.html

- Der Benutzer gibt an, welche Weinsorte er bestellen möchte sowie die Anzahl der Flaschen.
- Auswahl mittels Radio-Buttons bzw. Checkboxen sowie Inputfeld:
 - Grüner Veltliner, Riesling bzw. Muskateller (verschiedene Preise auszeichnen)

session2.php

- In einer Session werden die ausgewählte Weinsorte und Menge des vorigen Formulars gespeichert.
- Weiters muss der Benutzer nun seine persönlichen Daten angeben um die Bestellung weiterzuführen:
 - Name, Adresse, Email, ...

session3.php

- Die Benutzerdaten werden ebenfalls in der Session gespeichert.
- Ausgabe der Session-ID sowie der Bestelldaten
 - Weinsorte, bestellte Menge, Name, ...
- Button, ob die Bestellung endgültig abgesendet werden soll (Angabe Gesamtpreis) → weiter zu session4.php
 - Änderungsmöglichkeit der Daten: Link zurück zu session1.html

session4.php

- Endgültige Speicherung der Session-Daten in die Datei bestellungen.csv, um alle Bestellungen zu sammeln.
- Löschen der Session-Variablen und der Session.
- Meldung, dass die Bestellung erfolgreich durchgeführt wurde.

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

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_54



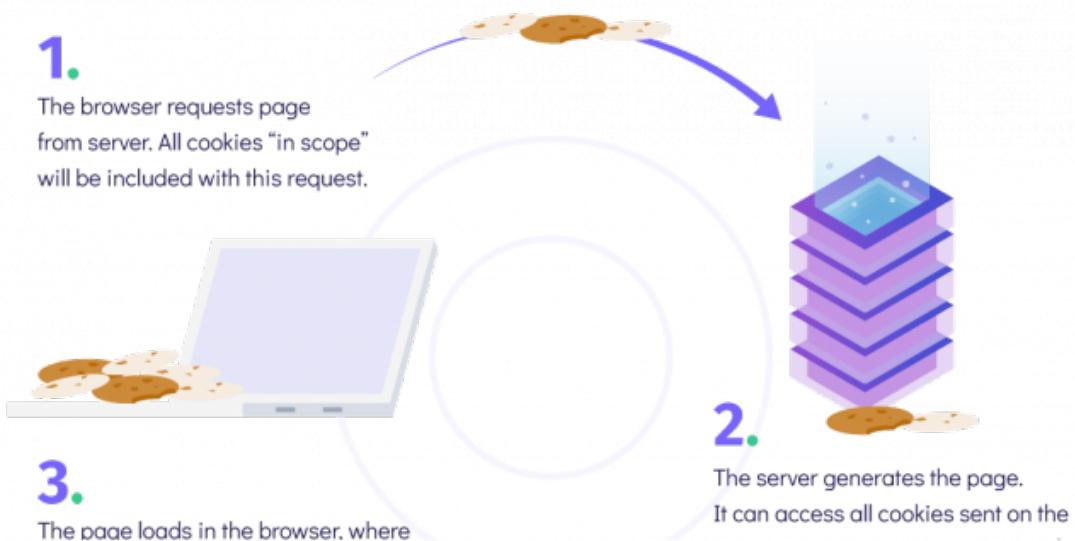
Last update: **2021/11/17 08:29**

Grundlagen zu Cookies



Mit Cookies hat man genau so die Möglichkeit, wie bei Sessions, bestimmte Daten während einer Folge von Aufrufen einer Website festzuhalten. Allerdings kann man mit Cookies auch Daten über die Sitzung hinaus speichern (z.B. 100 Tage nach der Speicherung), denn Cookies werden als Datei auf dem Computer des Besuchers gespeichert. Allerdings kann der Besucher uns natürlich verbieten, auf seinem Computer einen Cookie zu speichern, darum verwendet man nur für die Speicherung von Daten während einer Folge von Aufrufen einer Website lieber Sessions.

Ein Cookie benutzt man z.B. um Besucher auch nach Ablauf der Session, also i.d.R. nach dem Verlassen der Homepage, eindeutig zu identifizieren. Dies kann man z.B. bei Umfragen benutzt werden, oder als „Wiederkehrfunktion“ bei einem Forum.



Third Party Cookies

Third Party Cookies werden von Werbetreibenden genutzt, die über ihre Werbeschaltungen auf anderen Seiten mit den Cookies Nutzerinformationen sammeln. Es handelt sich dabei um Datensätze, die im Browser des Nutzers hinterlegt werden, wenn er eine Seite mit der Werbung besucht. Besucht er erneut eine Seite mit Werbung des gleichen Anbieters, wird er wiedererkannt.

Da sich mit Third Party Cookies die Wege des Nutzers im Internet verfolgen lassen, kann der Werbetreibende bei einem erneuten Besuch auf das "Profil" des Nutzers zugeschnittene Werbung ausspielen.

First vs. Third Party Cookies

Das Wort "Party" in First/Third Party Cookies bezieht sich auf die Domain, von der der Cookie stammt. Während ein **First Party Cookie in der Regel vom Websitebetreiber selbst** stammt, lassen sich **Third Party Cookies auf eine andere - eben "dritte" - Person** zurückführen, die ihre Cookies auf einer anderen Seite platzieren.

Es ist also möglich, dass auf Ihrem Rechner nach dem Besuch einer Website sowohl Third Party Cookies als auch First Party Cookies gespeichert werden – von der Seite selbst und von einer dritten Partei, die Werbung auf dieser Seite platziert hat. Bei einem First Party Cookie kann der Nutzer nur von der Seite wiedererkannt werden, von der der Cookie stammt, nicht aber über mehrere Domains hinweg. Die Daten werden dadurch auch nicht an Dritte weitergegeben.

Third Party Cookies sammeln für Werbetreibende viele nützliche Informationen der Nutzer, u.a.:

- Verweildauer
- Seitenaufrufe
- Bewegung des Nutzers über Links

Die Gesamtheit dieser Informationen liefert ein gutes Bild zu den Interessen des Nutzers, denn hier kann nicht nur nachverfolgt werden, wofür er sich innerhalb einer Domain interessiert, sondern über mehrere Domains hinweg. Third Party Cookies erlauben damit die Erstellung von umfangreichen Nutzerprofilen, sodass Werbetreibende genau dazu passende Werbung ausliefern können.

Man kann in Browsern einstellen, ob Third Party Cookies zugelassen sind oder nicht.

Do not track



In aktuellen Browsern kann man der Seite auch eine Do not track - Anfrage schicken, d.h. die Seite

wird darüber informiert, dass man nicht verfolgt werden möchte. Jedoch ist dies lediglich ein Wunsch des Users, der vom Seitenbetreiber erfüllt oder nicht erfüllt werden kann.

Third Party Cookies - Alternativen

Third Party Cookies werden häufig durch Browser oder Sicherheitseinstellungen in der Default-Einstellung blockiert, was natürlich nachteilig für Werbetreibende ist. Langfristig gesehen werden Cookies jedoch an Bedeutung verlieren – durch die sinkende Akzeptanz, die verstärkte Nutzung von Mobilgeräten und die Default-Einstellungen von Browsern und Betriebseinstellung. Das heißt aber nicht, dass bald keine Daten mehr über die Nutzer erhältlich sind, denn es gibt eine Reihe von Alternativen, die einerseits die Privatsphäre besser schützen und andererseits für mobile Geräte nutzbar sind. Dazu gehört u.a. das Fingerprinting, bei dem ein Gerät (und entsprechend der Nutzer) anhand bestimmter, individueller Merkmale wiedererkannt wird.

Es gibt bereits eine Third Party Cookie Alternative: **Google AdID** bzw. von Apple die **IDFA** soll Werbetreibenden ermöglichen, anonym Daten zu sammeln. Nutzer hingegen sollen die Möglichkeit erhalten, einen privaten Modus einzuschalten, bei dem gar keine Informationen mehr übermittelt werden können.

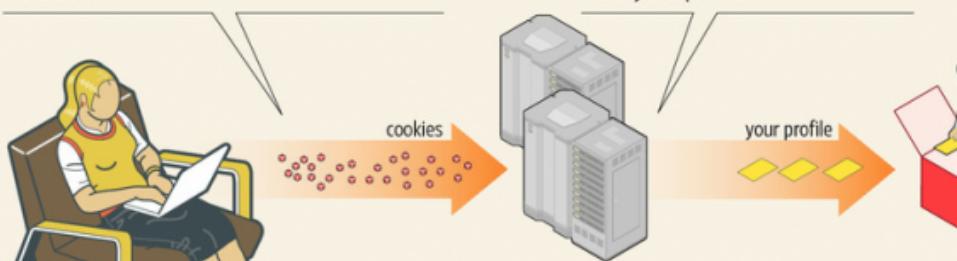


Video

Cookie Wars | A battle is brewing to control online tracking

HOW COOKIES TRACK

Cookies are tiny files that are invisible to the user but track their browsing habits, such as articles read and ads clicked on.



The information is sent to third-party data brokers who help major advertisers place ads and buy user profiles.

Advertisers have long been frustrated with the cookies' technological limitation. They don't sync across browsers, so much of what they collect is inaccurate.

HOW IT COULD CHANGE

Instead of cookies, Google and Microsoft do the tracking by assigning each user a number.



could track across Gmail, Android phones and the Chrome browser.



can track behavior across apps.



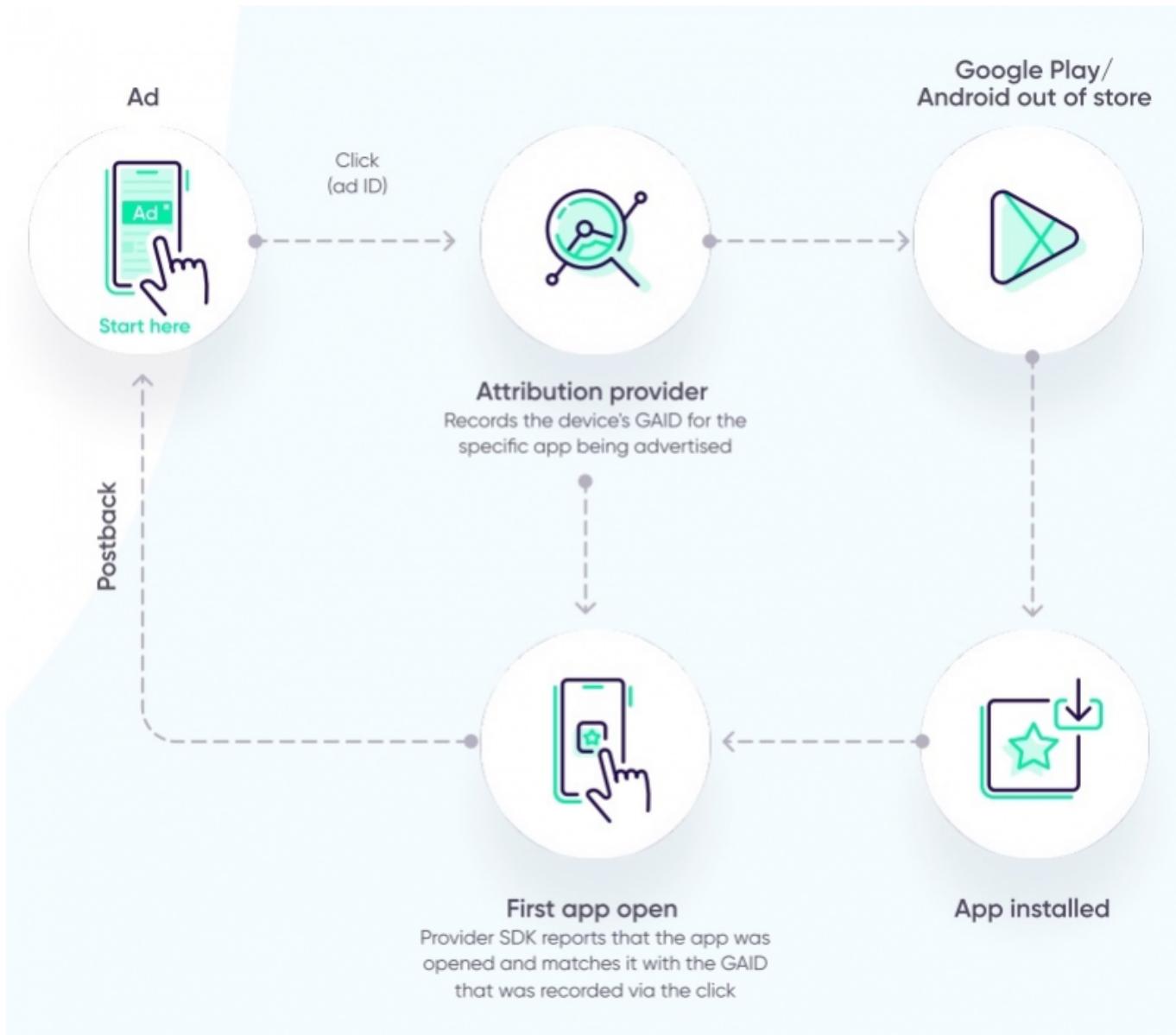
your profile

If the companies could control the data, then advertisers might need to go through them to place ads and buy user profiles.



The Wall Street Journal

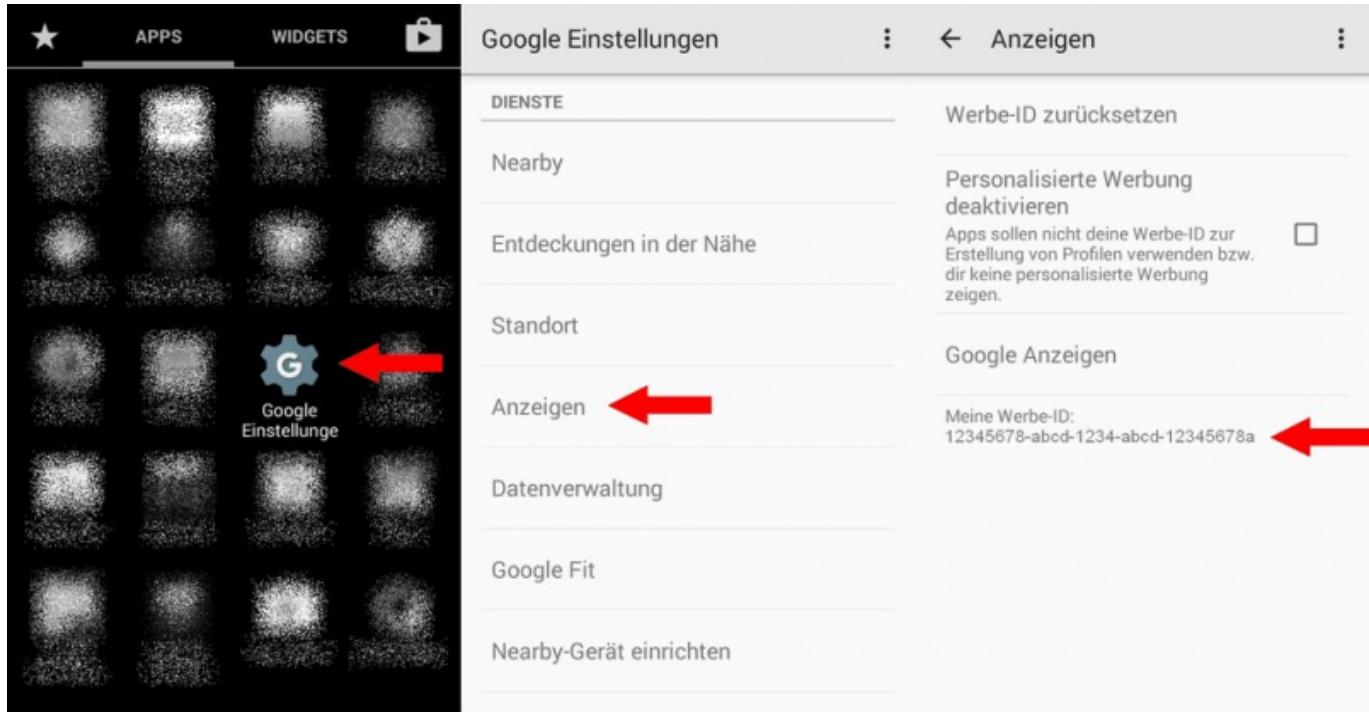
Sources: WSJ staff reports; the companies; industry experts



Mehr Infos

Die Google Werbe-ID (GAID) kann man am Handy leicht herausfinden:

Einstellungen ⇒ Google ⇒ Anzeigen



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

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_56



Last update: **2021/11/25 10:15**

Cookies setzen

Nur ganz oben bevor etwas mit oder ohne PHP ausgegeben wird können wir ein Cookie setzen:

```
<?php  
    setcookie("username", "Max", 0);  
?>
```

Die erste Angabe (username) ist der Name des Cookies, über die der Cookie später immer erreichbar ist. Die zweite Angabe ist der Wert, der nachher im Cookie gespeichert wird. In diesem Fall Max. Die dritte Stelle ist eine Zeitangabe, wie lange der Cookie gültig, d.h. auf dem Computer des Anwenders gespeichert wird. In diesem Fall hab ich aber keine Zeitangabe sondern die Zahl 0 verwendet. Null heißt, dass der Cookie bis zum Ende der Sitzung hält, also solange wie der Benutzer den Browser geöffnet hat. Um einen Zeitpunkt anzugeben, muss man die Sekunden seit dem 1.1.1970 angeben. Da man meistens aber keinen **exakten** Zeitpunkt angibt, z.B. den 31.08.2015, speichert man eigentlich nur eine Zeitspanne, nach wie vielen Tagen der Cookie verfällt. Um dies zu realisieren, verwendet man einen kleinen Trick. Man ermittelt den aktuellen Timestamp, also den Zeitpunkt der Cookiesetzung, und rechnet zu dem Wert dann noch die entsprechenden Sekunden für die Lebensspanne des Cookies:

```
<?php  
    setcookie("username", "Max", time()+(1*60*60*24));  
?>
```

Dieser Cookie würde jetzt $1*60*60*24$ Sekunden halten und das entspricht 24 Stunden (1 Tag).

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

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_57



Last update: **2021/11/25 10:11**

Cookies auslesen

Cookies können ähnlich wie Sessions ausgelesen werden:

```
<?php  
$cookie = $_COOKIE["username"];  
echo "Der Inhalt des Cookies: $cookie";  
?>
```

Bei dem Auslesen von Cookies ist außerdem noch zu beachten das sie nur in dem Verzeichnis wo ein Cookie gesetzt wurde, der Cookie auch ausgelesen werden kann. In einem anderen Verzeichnis ist er nicht auslesbar.

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

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_58



Last update: **2021/11/17 08:47**

Cookies Löschen

Um ein Cookie zu löschen, muss man einen Zeitpunkt in der Vergangenheit angeben. Dadurch erkennt der Browser, dass dieser Cookie abgelaufen ist, und löscht diesen:

```
<?php  
    setcookie("username", "", time() - 3600);  
?>
```

From:

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

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_59



Last update: **2021/11/17 08:48**

Sicherheit von Cookies

Im Gegensatz zu Sessions werden Cookies auf dem Computer des Besuchers und nicht auf dem Server der Webseite gespeichert, deshalb können sie gefälscht werden. Zum Beispiel beim Browser Firefox gibt es einen Datei, die so aussehen könnte:

```
# HTTP Cookie File
# http://www.netscape.com/newsref/std/cookie_spec.html
# This is a generated file! Do not edit.
# To delete cookies, use the Cookie Manager.
forum.php-einfach.de FALSE / FALSE 0 username Max
```

Der erste Wert gibt an, von welcher Website dieser Cookie stammt, die 0 ist die Lebensdauer des Cookies. Danach folgt der Cookie Name und der Cookiewert.

Natürlich kann jeder den Inhalt auslesen und auch verändern, darum sollte man einem Cookie nicht trauen. Falls ein Angreifer z.B. den Cookie kopiert, dann kann er sich damit als der Besitzer ausgeben.

Angemeldet bleiben

Cookies werden oft benutzt, um die beliebte Funktion Angemeldet bleiben zu implementieren. Dazu wird auf dem Rechner des Besuchers ein entsprechender, geheimer Wert hinterlegt. Sobald der Besucher die Webseite erneut besucht, wird dieser Wert im Cookie ausgelesen und sofern alles stimmt, wird der Benutzer eingeloggt. Die Umsetzung dieser Funktion ist nicht ganz einfach, deswegen haben wir diesem Thema einen extra Artikel gewidmet: Angemeldet bleiben.

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



Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_60

Last update: **2021/11/17 08:50**

- [Beispiel 1: Loginsystem](#)
- [Beispiel 2](#)
- [Beispiel 3](#)
- [Beispiel 4](#)
- [Beispiel 5](#)
- [Beispiel 6](#)
- [Beispiel 7](#)

From:

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



Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_61

Last update: **2021/12/02 08:46**

Cookies - Anwendungsbeispiel

Cookies bieten die Möglichkeit, nutzerspezifische Daten am Rechner des Benutzers zu speichern. (Im Gegensatz zu Sessions, die am Webserver gespeichert werden.) Dabei können auch Cookies und Sessions gemeinsam verwendet werden, wie z.B. im folgenden Beispiel.

Beispiel: Loginsystem

Verwendung von Cookies für ein Loginsystem, um eine Session nach dem schließen des Browsers wieder aufnehmen zu können.

```
<?php
//Dieser Teil muss direkt am Anfang der Seite stehen, weil die Funktionen
//session_start() und setcookie() vor jeder Ausgabe im Browser aufgerufen
//werden müssen
if (isset($_COOKIE["session"]))
{
    session_id($_COOKIE["session"]); //Die Session-ID wird auf den Wert
    //der ID, die im Cookie gespeichert ist, geändert. Dies muss gemacht werden,
    //bevor session_start() aufgerufen wird!
}

session_start();

if (isset($_POST["user"]) && isset($_POST["password"]))
{
    $_SESSION["user"] = htmlspecialchars($_POST["user"]);
    $_SESSION["password"] = htmlspecialchars($_POST["password"]);

    setcookie("session", session_id(), time() + 172800); //Setzen des
    //Cookies mit der Session-ID, Gültigkeit: Aktuelle Zeit + 2 Tage
}

if ($_GET["do"]=="logout")
{
    session_destroy();
    setcookie("session", "", 1); //Setzen eines Cookies der keinen Inhalt
    //hat und sich sofort wieder löscht (Es gibt keine Cookie-Löschen-Funktion)
}

?>
<!DOCTYPE html PUBLIC
"-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="de" lang="de">
    <head>
```

```
<meta http-equiv="content-type" content="text/html; charset=UTF-8"
/>
<meta http-equiv="Content-Type" content="text/css" />
<meta http-equiv="content-language" content="de" />
<meta name="author" content="Dominik Prüller" />
<title>Passwortabfrage mit Cookies</title>
</head>

<body>
<h1>Passwortabfrage mit Cookies</h1>

<?php
    if (isset($_SESSION["user"]) && isset($_SESSION["password"]) &&
$_GET["do"] != "logout")
    {
        if ($_SESSION["user"] == "user" && $_SESSION["password"] ==
"schule")
        {
            echo "<p>Login erfolgreich.</p>";
            echo "<a href='./passwortabfrage_cookies.php?do=logout'>Logout</a>";
        }
        else
        {
            echo "<p>Login fehlgeschlagen.</p>";
            session_destroy();
        }
    }
    else
    {
?>
        <form action="passwortabfrage_cookies.php" method="post">
            <fieldset>
                <legend>Loginformular</legend>
                <label for="user">Benutzername: </label><input
type="text" name="user" id="user" size="20" /><br />
                <label for="password">Passwort: </label><input
type="password" name="password" id="password" size="20" /><br />
                <input type="submit" value="Daten absenden" />
            </fieldset>
        </form>
<?php
    }
?>
    </body>
</html>
```

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

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_61:1_61_1 

Last update: **2021/12/02 08:35**

Cookies: Beispiel 2

Das Projekt besteht aus 4 Dateien, die einzelnen Funktionen wurden ausgelagert um besser den Überblick behalten zu können.

Die Applikation bietet dem User die Möglichkeit kurze Notizen abzuspeichern, die als Cookies am PC des Users gelagert werden. Weiters ist es möglich die Notizen zu löschen.

Download der Dateien [hier](#)

Die Index-Datei

```
<?php
// Je nach GET-Parameter wird eine andere Funktion aufgerufen
switch (@$_GET['func']) {
    case 'create': $title = 'Notiz erstellen'; $file = 'create.php'; break;
    case 'delete': $title = 'Notiz löschen'; $file = 'delete.php'; break;
    default: $title = 'Notizen ansehen'; $file = 'show.php'; break;
}
?>

<?php echo '<?xml version="1.0" encoding="UTF-8"?>' ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"
/>
        <title>Notizen<?php if (!empty($title)) { echo ' - ' . $title; } ?></title>
    </head>
    <body>
        <h1>Notizen<?php if (!empty($title)) { echo ' - ' . $title; } ?></h1>
        <h2>Navigation</h2>
        <ul>
            <li><a href="index.php?func=show">Notizen lesen</a></li>
            <li><a href="index.php?func=create">Notiz erstellen</a></li>
            <li><a href="index.php?func=delete">Notiz löschen</a></li>
        </ul>
        <?php
            include_once($file);
        ?>
    </body>
</html>
```

show.php

```

<h2>Notizen abrufen</h2>

<h3>Index</h3>
<ul>
<?php

// Die Notizen werden iterativ per foreach gelesen und verlinkt

foreach ($_COOKIE['note'] as $key => $value) {
    echo '<li><a href="index.php?func=show&note=' . $key . '">' . $key .
'</a></li>';
}

?>
</ul>

<?php
// Falls der User eine Notiz lesen will, wird der Text einfach über die
superglobale Variable $_COOKIE ausgelesen
if (@$_GET['note']) {
    $key = $_GET['note'];
    $value = $_COOKIE['note'][$key];
    echo '<h3>Anzeige: ' . $key . '</h3>';
    echo '<p>' . $value . '</p>';
}

?>

```

create.php

```

<h2>Notiz erstellen</h2>

<form action="index.php?func=create" method="post">
    <table>
        <tr><td>
            <label for="titel">Titel der Notiz:</label>
        </td><td>
            <input type="text" size="20" name="titel" id="titel" />
        </td></tr><tr><td>
            <label for="text">Text:</label>
        </td><td>
            <textarea cols="50" rows="10" name="text"
id="text"></textarea>
        </td></tr><tr><td>
            <input type="submit" name="submit" />
        </td><td>

```

```
        <input type="reset" />
    </td></tr>
</table>
</form>
```

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

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_61:1_61_2

Last update: **2021/12/02 08:38**

Beispiel

Erklärung

Dieses Programm ermöglicht es dem Benutzer, sich mit seinem Namen anzumelden.

Dabei wird auch die Anzahl der Aufrufe mitgezählt und ausgegeben.

Beide Daten, sowohl Name als auch Zähler, werden als Cookie für ca. 5 Monate gespeichert. Bei jedem Aufruf wird die „Lebensdauer“ des Cookie um dieselbe Zeit verlängert. Es besteht weiters die Möglichkeit, sich wieder abzumelden, um sich mit einem anderen Namen erneut anzumelden. Natürlich wird bei der Eingabe eines neuen Namens auch der Zähler zurückgesetzt.

- **mit seinem Namen anmelden**
- **Anzahl der Aufrufe werden mitgezählt**
- **Beide Daten werden als Cookie für ca. 5 Monate gespeichert**
- **„Lebenszeit“ wird bei jedem Aufruf um dieselbe Zeit verlängert**
- **Benutzer kann sich auch abmelden**
- **bei der Anmeldung mit einem anderem Namen wird auch der Zähler zurückgesetzt**

Code

```
<?php
$angemeldet=false;

if(( $_POST["submit"]=="Absenden" AND isset($_COOKIE["name"]) ) AND
$_COOKIE["name"]==$_POST["name"] ) OR
( $_POST["submit"]!="Absenden" AND isset($_COOKIE["name"])) )
{
    $name=$_COOKIE["name"];
    setcookie("name", $name, time()+525600);
    $count=$_COOKIE["count"]+1;
    setcookie("count", $count, time()+525600);
    $angemeldet=true;
}

else if(( $_POST["submit"]=="Absenden" AND isset($_COOKIE["name"]) ) AND
$_COOKIE["name"]!=$_POST["name"] ) OR
( $_POST["submit"]=="Absenden" AND isset($_COOKIE["name"])==false))
{
    $name=$_POST["name"];
    setcookie("name", $name, time()+525600);
```

```

$count=1;
setcookie("count", $count, time()+525600);
$angemeldet=true;
}

if($_GET["logout"]==true)
{
$angemeldet=false;
$count=$_COOKIE["count"];
setcookie("count", $count, time()+525600);
}
?>

<html>
<head>
<title></title>
<meta name="author" content="Arti">
</head>
<body text="#000000" bgcolor="#FFFFFF" link="#FF0000" alink="#FF0000"
vlink="#FF0000">

<?php
if($angemeldet==true)
{
echo "Hallo $name!<br><br>\nDu bist das $count. Mal hier!<br><br>\n";
echo "<a href=\"beispiel.php?logout=true\">Logout</a>";
}
else
{
?>
<form action="beispiel.php" method="POST">
Bitte gib deinen Namen ein: <input type="Text" name="name"><br><br>
<input type="Submit" name="submit" value="Absenden">
</form>
<?php
}
?>

</body>
</html>

```

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

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_61:1_61_3

Last update: **2021/12/02 08:44**

Beispiel 4:

Wiederaufruf vom Namen

In diesem Programm wird der Benutzer aufgefordert seinen Namen einzugeben. Nachdem er „Name absenden“ gedrückt hat, wird der Name als Cookie gespeichert und steht somit bis zum eingegebenen Zeitlimit (in diesem Fall 4 Wochen lang) zu Verfügung.

name.html

```
<html>
<head>
<title>Namenseingabe</title>
</head>

Geben Sie bitte Ihren Namen ein! <br><br>

<form action="cookies.php" method="POST">
  Name:<input type="text" name="username"><br/>      //Eingabe von Username
  <input type="Submit" value="Name absenden">
</form>

</html>
```

cookies.php

```
<?php

$cookie_name = $_POST["username"];
$cookie_max_time = time() + 60*60*24*28;      //Gültigkeit für 4 Wochen
setcookie("username", $cookie_name, $cookie_max_time);

?>

<html>
<head>
<title>Cookies</title>
<meta name="author" content="user">
</head>
<body text="#000000" bgcolor="#FFFFFF" link="#FF0000" alink="#FF0000"
vlink="#FF0000">

<?php

$username = $_COOKIE['username'];
```

```
echo "Hallo " . $cookie_name . "!" ;      //Ausgabe von Username  
?  
</body>  
</html>
```

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

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_61:1_61_4

Last update: **2021/12/02 08:44**

Cookies

Anwendungsbeispiel

```
<?php
    if (isset($_COOKIE["cookietest"]) AND isset($_POST["info"])) ) //bei
späteren besuchen der seite
        setcookie("cookietest",$_POST["info"],time()+3024000); //Cookie wird
für 5 wochen gespeichert

    else if(isset($_POST["submit"])) //beim ersten mal
        setcookie("cookietest",$_POST["info"],time()+3024000); //Cookie wird
für 5 wochen gespeichert
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Cookies</title>
<meta name="author" content="Jobrot">
<meta name="editor" content="html-editor phase 5">
</head>
<body text="#000000" bgcolor="#FFFFFF" link="#FF0000" alink="#FF0000"
vlink="#FF0000">
<?php
    if (!isset($_COOKIE["cookietest"]))
    {
        echo "Worüber wollen sie informiert werden, wenn sie das nächste Mal auf
unsere Seite kommen?";
        echo '<form action="Cookies.php" method="post">';
        echo '<input type="radio" name="info" value="Sport">Sport <br>';
        echo '<input type="radio" name="info" value="Technik">Technik <br>';
        echo '<input type="radio" name="info" value="Literatur">Literatur <br>';
        echo '<input type="radio" name="info" value="Musik">Musik <br>';
        echo '<input type="radio" name="info" value="Moebel">Möbel <br>';
        echo '<input type="submit" name="submit" >';
        echo '</form>';
    }
    else
    { echo"af";
        if ($_COOKIE["cookietest"]=="Sport") //mittels $_COOKIE wird das
cookie abgerufen
            { echo "<h2>Sport</h2>";
                echo '<a href="http://sport.orf.at/"> Orf-Sport </a><br>';
                echo '<a
href="http://www.intersport.at/eshop?adword=SEM_GOOGLE/Brand/Brand&sVoucherN
r=71737&fnc=nk_adddirectvoucher"> Intersport </a><br>';
                echo '<a
href="http://www.amazon.de/sport-freizeit-sportartikel/b?ie=UTF8&node=164350
"
```

```

51"> Amazon-Sport </a><br>' ;
    }
else if ($_COOKIE["cookietest"]=="Technik")
{ echo "<h2>Technik</h2>" ;
  echo '<a
href="http://www.amazon.de/Elektronik-Foto/b?ie=UTF8&node=562066"> Amazon-
Elektronik </a><br>' ;
  echo '<a href="http://science.orf.at/"> Science-Orf </a><br>';
  echo '<a href="http://reichelt-elektronik.at/?PROVID=2597">
Reichelt-Elektronik-Shop </a><br>' ;
}
else if ($_COOKIE["cookietest"]=="Literatur")
{ echo "<h2>Literatur</h2>" ;
  echo '<a href="http://www.thalia.at/shop/ama_homestartseite/">
Thalia </a><br>';
  echo '<a href="http://www.weltbild.at/"> Weltbild </a><br>';
  echo '<a href="http://www.literature.at "> Online-Bibliothek
</a><br>';
}
else if ($_COOKIE["cookietest"]=="Musik")
{ echo "<h2>Musik</h2>" ;
  echo '<a href="http://www.musicload.at/"> Musicload </a><br>';
  echo '<a href="http://www.musiknews.de/"> News und Charts </a><br>';
  echo '<a href="http://www.weltbild-downloads.at/"> Weltbild-Musik
</a><br>';
}
else if ($_COOKIE["cookietest"]=="Moebel")
{ echo "<h2>Möbel</h2>" ;
  echo '<a href="http://www.xxlutz.at/"> XXX-Lutz </a><br>';
  echo '<a href="http://www.moemax.at/"> Mömax </a><br>';
  echo '<a href="http://www.moobelix.at/"> Möbelix </a><br>';
}

echo "<br><br>Oder wollen sie das nächste mal über etwas anderes informiert
werden?" ;
echo '<form action="Cookies.php" method="post">' ;
echo '<input type="radio" name="info" value="Sport" checked>Sport <br>';
echo '<input type="radio" name="info" value="Technik">Technik <br>';
echo '<input type="radio" name="info" value="Literatur">Literatur <br>';
echo '<input type="radio" name="info" value="Musik">Musik <br>';
echo '<input type="radio" name="info" value="Moebel">Möbel <br>';
echo '<input type="submit">' ;
echo '</form>' ;
}
?>
</body>
</html>

```

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

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_61:1_61_5 

Last update: **2021/12/02 08:45**

Cookies

Erzeugung von Cookies

Die erste Möglichkeit

Zutaten für eine Portion

370 g Butter

220 g Zucker

80 g Zucker, braun

2 Ei(er)

2 Pck. Zucker, (Orangenzucker)

440 g Mehl

1 TL Backpulver

250 g Nüsse, gehackte

200 g Schokolade, gehackte

Zubereitung Die Butter mit Zucker, braunem Zucker, Eiern und Orangenzucker cremig rühren. Mehl, Backpulver, gehackte Nüsse und Schokostückchen untermischen.

Mit einem EL Teigportionen abstechen, mit Abstand auf mit Backpapier ausgelegte Bleche setzen und bei 180 Grad goldbraune Cookies backen.

Zubereitungszeit: ca. 20 Min.

Schwierigkeitsgrad: simpel

Brennwert p. P.: keine Angabe

Zweite Möglichkeit

```
<?php
if(isset($_COOKIE["count"])==false){
    $login=false;
    $count=0;
}
if($login==false){
    $count=$_COOKIE["count"]+1;
    setcookie("count", $count, time()+60);
    $login=true;
?>
<html>
<head>
<p>Willkommen auf meiner Seite
<form action="counter2.php" method="POST" target="">
<input type="Submit" name="login" value="login">
</form> </p>
<?php
}
```

```
if($_POST["login"]){
if ($login==true) {
echo "Sie sind der ".$count."te Nutzer";
}
?>
</body>
</html>
```

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

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_61:1_61_6 

Last update: **2021/12/02 08:46**

To-Do-Liste mittels Cookies

Files zum Download

[cookies-todo-liste.rar](#)

name.html

```
<html>
<head>
<title>Namenseingabe</title>
</head>

Geben Sie bitte Ihren Namen ein! <br><br>

<form action="cookies.php" method="POST">
  Name:<input type="text" name="username"><br><br>
  <input type="Submit" value="Name absenden">
</form>
```

cookies.php

```
<?php

$cookie_name = $_POST["username"];
$cookie_max_time = time() + 60*60*24*28;      //Gültigkeit für 4 Wochen
setcookie("username", $cookie_name, $cookie_max_time);

if(!isset($_COOKIE["count"])){
  $count=1;
  setcookie("count", $count, time()+60);
}
else {
  $count=$_COOKIE["count"]+1;
  setcookie("count", $count, time()+60);
}

?>

<html>
<head>
<title>Cookies</title>
<meta name="author" content="user">
</head>
<body text="#000000" bgcolor="#FFFFFF" link="#FF0000" alink="#FF0000"
```

```
vlink="#FF0000">
<h2>Sie sind angemeldet!</h2>
<?php

$username = $_COOKIE["username"];
echo "Hallo " . $cookie_name . "!<br>" ; //Ausgabe von Username

echo "Sie besuchen diese Seite zum ". $count ." Mal.<br>";
echo "Hier können sie T0-Do-Listen erstellen.<br> Sie muessen einfach auf den nachfolgenden Link klicken!<br><br>";

?>
<h2>Navigation</h2>
<li><a href="create.php">Erstellen einer To-Do-Liste...</a></li>
<li><a href="show.php">Anzeige...</a></li>
<li><a href="delete.php">Loeschen...</a></li>

</body>
</html>
```

index1.php

```
<?php

switch (@$_GET['func'])
{
    case 'create': $title = ' erstellen'; $file = 'create.php'; break;
    case 'delete': $title = ' löschen'; $file = 'delete.php'; break;
    default: $title = 'ansehen'; $file = 'show.php'; break;
}
?>

<?php echo '<?xml version="1.0" encoding="UTF-8"?>' ?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"
/>

        <title>To-Do-Liste
        <?php if (!empty($title))
        {
            echo ' - ' . $title;
        } ?>
        </title>

    </head>
```

```

<body>
    <h1>To-Do-List
    <?php if (!empty($title))
    {
        echo ' - ' . $title;
    } ?>
    </h1>

    <h2>Navigation</h2>
    <ul>
        <li><a href="index1.php?func=show"> lesen</a></li>
        <li><a href="index1.ph
        p?func=create">erstellen</a></li>
        <li><a href="index1.php?func=delete">löschen</a></li>
    </ul>
    <?php
        include_once($file);
    ?>
</body>
</html>

```

create.php

```

<h2>To-Do-Liste erstellen</h2>

<form action="index1.php?func=create" method="post">
    <table>
        <tr><td>
            <label for="titel">Titel der To-Do-Liste:</label>
        </td><td>
            <input type="text" size="20" name="titel" id="titel" />
        </td></tr><tr><td>
            <label for="text">To-Do-Liste:</label>
        </td><td>
            <textarea cols="50" rows="10" name="text"
id="text"></textarea>
        </td></tr><tr><td>
            <input type="submit" name="submit" />
        </td><td>
            <input type="reset" />
        </td></tr>
    </table>
</form>

<?php

if (@$_POST['submit'])
{
    if (setcookie('note[' . $_POST['titel'] . ']', $_POST['text'],
time() + 60 * 60 * 24 * 30))

```

```

    {
        echo 'To-Do-Liste "' . $_POST['titel'] . '" erstellt';
    }
} else
{
    echo 'Fehler beim Erstellen';
}
}

?>

```

show.php

```

<h2>To-Do-Liste abrufen</h2>

<ul>
<?php

foreach ($_COOKIE['note'] as $key => $value)
{
    echo '<li><a href="index1.php?func=show&note=' . $key . '">' . $key . '</a></li>';
}

?>
</ul>

<?php
// Falls der User eine Notiz lesen will, wird $_COOKIE ausgelesen
if (@$_GET['note'])
{
    $key = @$_GET['note'];
    $value = $_COOKIE['note'][$key];
    echo '<h3>Anzeige: ' . $key . '</h3>';
    echo '<p>' . $value . '</p>';
}

?>

```

delete.php

```

<h2>To-Do-Liste löschen</h2>

<h3>Index</h3>
<ul>
<?php

```

```
foreach ($_COOKIE['note'] as $key => $value)
{
    echo '<li><a href="index1.php?func=delete&note=' . $key . '">' . $key .
'</a></li>';
}

?>
</ul>

<?php
if (@$_GET['note'])
{
    $key = $_GET['note'];
    if (setcookie('note['.$key.']', '', time() - 3600)) // setzte das Cookie
mit dem Namen der Notiz und einem Ablaufdatum in der Vergangenheit
        echo 'Eingabe "' . $key . '" gelöscht';
    else
        echo 'Fehler beim Löschen';
}
?>
```

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

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:1:1_61:1_61_7

Last update: **2021/12/02 08:47**

NETZWERKE

Allgemeine Netzwerkgrundlagen

- Grundlagen
- Topologien
- Übertragungsmedien
- Ethernet + Zugriffsverfahren CSMA/CD
- Schichtenmodell
- Netzwerkgeräte
- Adressierung
 - Adressierung Übungen
- Netzwerksimulation mit FILIUS
- Routing
- Netzwerkbefehle
- Protokolle
- Netzwerkanalyse mit Wireshark
 - Wireshark Übungen

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

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:2

Last update: **2021/12/14 16:19**

4.1) Netzwerk-Grundlagen

Was ist ein Netzwerk?

Ein Netzwerk ist die **physikalische und logische Verbindung von Computersystemen**. Ein einfaches Netzwerk besteht aus zwei Computersystemen. Sie sind über ein Kabel miteinander verbunden und somit in der Lage ihre Ressourcen gemeinsam zu nutzen. Wie zum Beispiel Rechenleistung, Speicher, Programme, Daten, Drucker und andere Peripherie-Geräte. Ein netzwerkfähiges Betriebssystem stellt den Benutzern auf der Anwendungsebene diese Ressourcen zur Verfügung.

Notwendigkeit für ein Netzwerk

Als es die ersten Computer gab, waren diese sehr teuer. Peripherie-Geräte und Speicher waren fast unbezahlbar. Zudem war es erforderlich zwischen mehreren Computern Daten auszutauschen. Aus diesen Gründen wurden Computer miteinander verbunden bzw. vernetzt.

Daraus ergaben sich einige Vorteile gegenüber unvernetzten Computern:

- zentrale Steuerung von Programmen und Daten
- Nutzung gemeinsamer Datenbeständen
- erhöhter Datenschutz und Datensicherheit
- größere Leistungsfähigkeit
- gemeinsame Nutzung der Ressourcen

Die erste Möglichkeit, Peripherie-Geräte gemeinsam zu nutzen, waren manuelle Umschaltboxen. So konnte man von mehreren Computern aus einen Drucker nutzen. An welchem Computer der Drucker angeschlossen war, wurde über die Umschaltbox bestimmt. Leider haben Umschaltboxen den Nachteil, dass Computer und Peripherie beieinander stehen müssen, weil die Kabellänge begrenzt ist.

Größenordnung von Netzwerken

Jedes Netzwerk basiert auf Übertragungstechniken, Protokollen und Systemen, die eine Kommunikation zwischen den Netzwerk-Teilnehmern ermöglichen. Bestimmte Netzwerktechniken unterliegen dabei Beschränkungen, die insbesondere deren Reichweite und Ausdehnung begrenzt. Hierbei haben sich verschiedene Netzwerk-Dimensionen durchgesetzt für die es unterschiedliche Netzwerktechniken gibt.

- PAN - Personal Area Network: personenbezogenes Netz, z. B. Bluetooth
- LAN - Local Area Network: lokales Netz, z. B. Ethernet
- MAN - Metropolitan Area Network: regionales Netz
- WAN - Wide Area Network: öffentliches Netz, z. B. ISDN
- GAN - Global Area Network: globales Netz, z. B. das Internet

Einteilung nach Ausdehnung



© 2010-2015 - Think a bit - www.thinkabit.ch

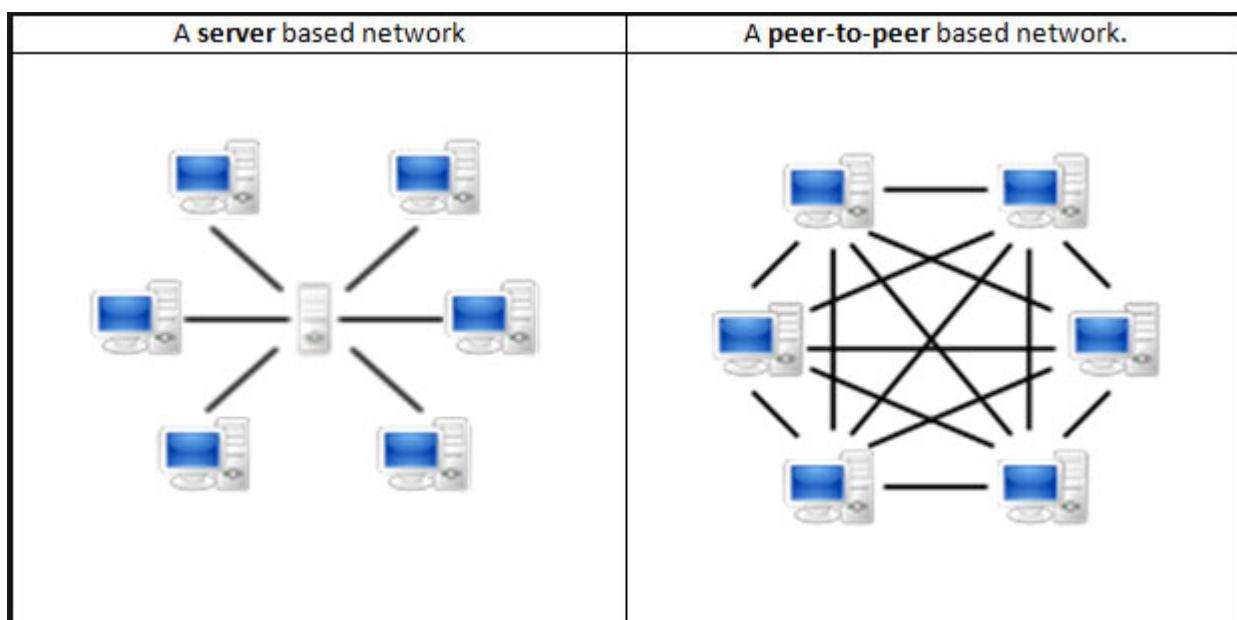
81



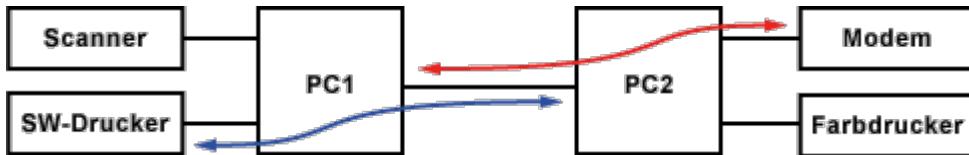
In der Regel findet ein Austausch zwischen den Netzen statt. Das heißt, dass Netzwerk-Teilnehmer eines LANs auch ein Teilnehmer eines WANs oder eines GANs ist. Eine 100%ig klare Abgrenzung zwischen diesen Dimensionen ist nicht immer möglich, weshalb man meist nur eine grobe Einteilung vornimmt. So unterscheidet man in der Regel zwischen LAN und WAN, wobei es auch Techniken und Protokolle gibt, die sowohl im LAN, als auch im WAN zum Einsatz kommen.

Peer-to-Peer-Netze und Client-Server-Architekturen

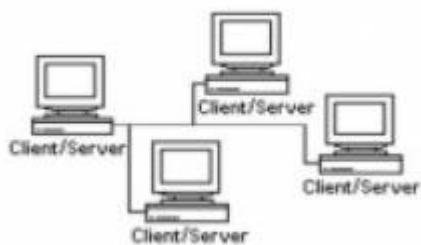
Man unterscheidet zwei „Philosophien“:



Peer-to-Peer-Netzwerke



In einem Peer-to-Peer-Netzwerk ist jeder angeschlossene Computer zu den anderen gleichberechtigt. Jeder Computer stellt den anderen Computern seine Ressourcen zur Verfügung. Ein Peer-to-Peer-Netzwerk eignet sich für bis zu 10 Stationen. Bei mehr Stationen wird es schnell unübersichtlich. Diese Art von Netzwerk ist relativ schnell und kostengünstig aufgebaut. Die Teilnehmer sollten möglichst dicht beieinander stehen. Einen Netzwerk-Verwalter gibt es nicht. Jeder Netzwerk-Teilnehmer ist für seinen Computer selber verantwortlich. Deshalb muss jeder Netzwerk-Teilnehmer selber bestimmen, welche Ressourcen er freigeben will. Auch die Datensicherung muss von jedem Netzwerk-Teilnehmer selber vorgenommen werden.

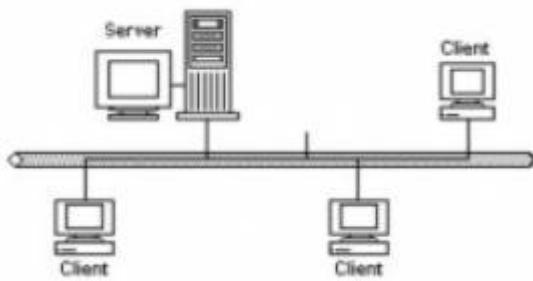


Peer-to-Peer-Netze brauchen keinen eigenen Server-Rechner, da jeder PC Server-Funktionen übernehmen kann.

Client/Server-Architekturen



In einem serverbasierten Netzwerk werden die Daten auf einem zentralen Computer gespeichert und verwaltet. Man spricht von einem dedizierten Server, auf dem keine Anwendungsprogramme ausgeführt werden, sondern nur eine Server-Software und Dienste ausgeführt werden. Diese Architektur unterscheidet zwischen der Anwender- bzw. Benutzerseite und der Anbieter- bzw. Dienstleisterseite. Der Anwender betreibt auf seinem Computer Anwendungsprogramme (Client), die die Ressourcen des Servers auf der Anbieterseite zugreifen. Hier werden die Ressourcen zentral verwaltet, aufgeteilt und zur Verfügung gestellt. Für den Zugriff auf den Server (Anfrage/Antwort) ist ein Protokoll verantwortlich, dass sich eine geregelte Abfolge der Kommunikation zwischen Client und Server kümmert. Die Client-Server-Architektur ist die Basis für viele Internet-Protokolle, wie HTTP für das World Wide Web oder SMTP/POP3 für E-Mail. Der Client stellt eine Anfrage. Der Server wertet die Anfrage aus und liefert eine Antwort bzw. die Daten zurück.



From:

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

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:2:2_00



Last update: **2021/12/14 16:19**

2.2) Netzwerk-Topologien

Die Struktur eines Netzwerks bezeichnet man als Topologie. Wie wichtig die Struktur eines Netzwerks ist, merkt man bei einem Leitungsausfall: ein gutes Netzwerk findet bei einem Leitungsausfall selbstständig einen neuen Pfad zum Empfänger.

Physikalische und logische Topologie

Interessant ist, dass sich die **sichtbare Topologie** (also die physische Verkabelungsstruktur) vom tatsächlichen Datenfluss unterscheiden kann. Deshalb verwendet man für die hardwaremäßige Realisierung den Begriff **physikalische Topologie** während man für den tatsächlichen Datenfluss den Begriff „logische Topologie“ verwendet.

Die wichtigsten Netzwerktopologien sind:

Bus-Topologie



ALLE GERÄTE nutzen DASSELBE KABEL

Bei einem Bussystem sind alle Rechner hintereinander geschaltet und über Abzweige (T-Stücke) an das Netzwerkkabel angeschlossen. Problem: Eine Verbindungsunterbrechung betrifft den ganzen Bus!

Vorteile

- Relativ niedrige Kosten, da geringe Kabelkosten
- Ausfall einer Station führt zu keinem Netzausfall

Nachteile

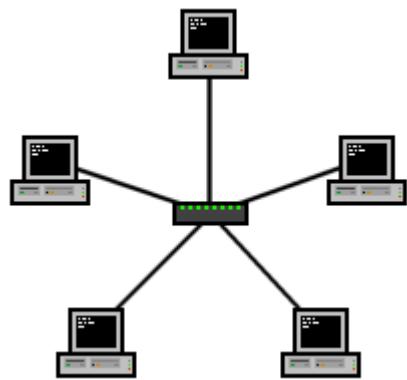
- Alle Daten über ein Kabel
- Nur eine Station kann senden. Alle anderen sind blockiert.
- Eine Störung an einer Stelle (z.B.: Defektes Kabel) führt zu einem Netzausfall (⇒ aufwendige Fehlersuche)
- Unverschlüsselter Netzwerkverkehr kann direkt am Bus (=Kabel) mitgelesen werden

Einsatzgebiet

Früher aufgrund der niedrigen Kosten häufig verwendet, heute spielt die Bus-Topologie keine Rolle mehr und wurde von der Stern-Topologie verdrängt.

[W Bus-Topologie - Details](#)

Stern-Topologie



Jedes Gerät nutzt ein Kabel.

Damit ist es zu einem Verteiler verbunden. Es existiert eine Punkt-zu-Punkt Verbindung zwischen Verteiler und Gerät. Als Verteiler kann ein HUB oder ein SWITCH dienen.

Vorteile

- Ausfall einer Station oder eines Defekts an einem Kabel führt zu keinem Netzausfall
- Aktive Verteiler (Switch, Hub) dienen gleichzeitig als Signalverstärker
- Bei richtiger Konfiguration können zwei Stationen die volle Bandbreite des Übertragungsmediums für ihre Kommunikation nutzen, ohne andere Stationen dabei zu behindern. Diese physikalische Topologie erlaubt somit sehr hohe Datendurchsatzraten.
- Weitere Stationen oder Verteiler können einfach hinzugefügt werden. Sehr leicht skalierbar.

Nachteile

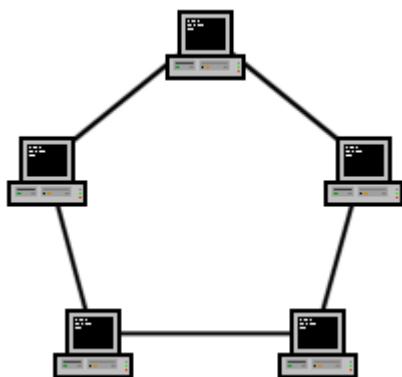
- Große Kabelmengen
- Beim Ausfall des Verteilers ist kein Netzverkehr mehr möglich

Einsatzgebiet

Im praktischen Einsatz bei lokalen Netzwerken findet die Stern-Topologie Verwendung. Häufigste Form der Verkabelung.

[W Stern-Topologie - Details](#)

Ring-Topologie



JEDES GERÄT ist mit ZWEI NACHBARN verbunden.

Die Ring-Topologie ist eine geschlossene Form, es gibt keinen Kabelanfang und kein Kabelende. Es handelt sich jeweils um eine Punkt-zu-Punkt Verbindung zwischen den Rechnern. Jede Station hat genau einen Vorgänger und einen Nachfolger. Datenverkehr findet immer nur in eine Richtung statt.

Vorteile

- Vorgänger und Nachfolger sind festgelegt
- Alle Stationen verstärken das Signal
- Alle Stationen haben gleiche Zugriffsmöglichkeit
- Leicht umsetzbar

Nachteile

- Ausfall einer Station oder eines Kabelteils führt zu einem Netzausfall
- Hoher Aufwand bei der Verkabelung (Jede Station braucht 2 Netzwerkkarten)
- Leicht abhörbar
- langsame Datenübertragung bei vielen Stationen

Einsatzgebiet

Physikalische Anwendung gibt es heute keine mehr.
Logische Anwendung findet sie im Token Ring.

[W Ring-Topologie - Details](#)

Mischformen

Sind zumeist **Kombinationen aus Bus, Stern und Ring**.

Backbone

Unter einem Backbone („Rückgrat“) wird die physikalische Verbindung zwischen einzelner Teilnetze verstanden. Es wird auch oft als Hintergrundnetz betitelt und verbindet z.B. mehrere Gebäude.

Stern-Bus-Netz

Ein Stern-Bus-Netz entsteht, wenn mehrere Verteiler über einen Bus miteinander verbunden sind. Häufig sind so mehrere Stockwerke miteinander verbunden.



Stern-Stern-Netz

Ein Stern-Stern-Netz entsteht, wenn mehrere Verteiler wiederum über einen Verteiler verbunden sind. Häufigste Anwendung ist wiederum das Verbinden von mehreren Subnetzen (z.B. Netze in verschiedenen Stockwerken).

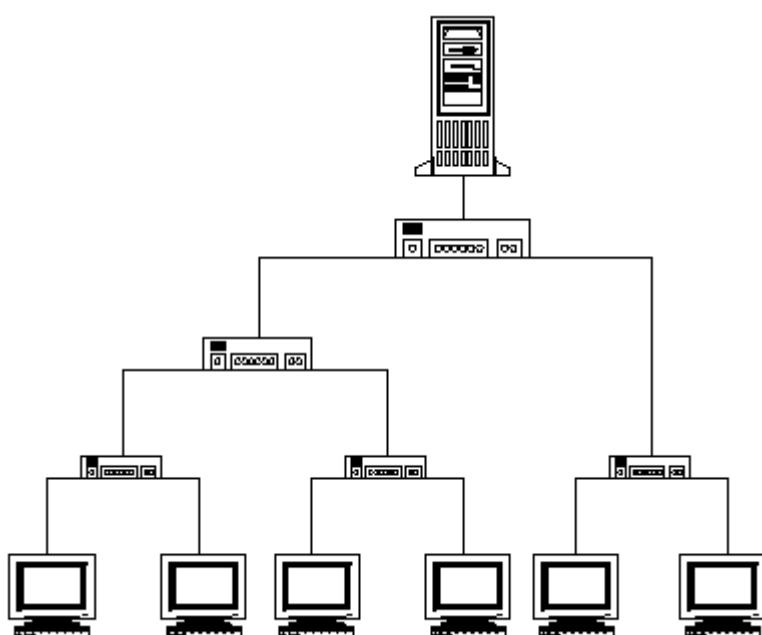
Fällt der Hauptverteiler aus, so kann zwischen den Stockwerken nicht mehr kommuniziert werden. Jedoch kann man auch die Hauptverteiler redundant auslegen.



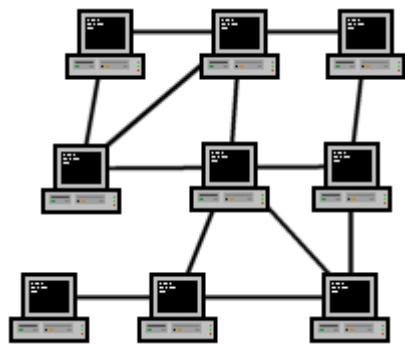
Baum

Eine Baum-Topologie wird so aufgebaut, dass, ausgehend von der Wurzel, eine Menge von Verzweigungen zu weiteren Verteilungsstellen existiert.

Es handelt sich somit um eine Erweiterung der Stern-Stern-Topologie auf mehrere Ebenen.



Maschen-Topologie



Vorherrschende Netzstruktur in großflächigen Netzen (z.B. öffentliche Telekommunikationsnetze).

[W Maschen-Topologie - Details](#)

Zell-Topologie

Die Zell-Topologie kommt hauptsächlich bei drahtlosen Netzen zum Einsatz. Eine Zelle ist der Bereich um eine Basisstation (z.B. Wireless Access Point), in dem eine Kommunikation zwischen den Endgeräten und der Basisstation möglich ist.

[W Zell-Topologie - Details](#)

Zusammenfassung

Topologie	Vorteile	Nachteile
Bus-Topologie	<ul style="list-style-type: none"> • einfach installierbar • kurze Leitungen 	<ul style="list-style-type: none"> • Netzausdehnung begrenzt • bei Kabelbruch fällt Netz aus • aufwändige Zugriffsmethoden
Ring-Topologie	<ul style="list-style-type: none"> • verteilte Steuerung • große Netzausdehnung 	<ul style="list-style-type: none"> • aufwendige Fehlersuche • bei Störungen Netzausfall • hoher Verkabelungsaufwand
Stern-Topologie	<ul style="list-style-type: none"> • einfache Vernetzung • einfache Erweiterung • hohe Ausfallsicherheit 	<ul style="list-style-type: none"> • hoher Verkabelungsaufwand • Netzausfall bei Ausfall oder Überlastung des Hubs
Maschen-Topologie	<ul style="list-style-type: none"> • dezentrale Steuerung • unendliche Netzausdehnung • hohe Ausfallsicherheit 	<ul style="list-style-type: none"> • aufwendige Administration • teure und hochwertige Vernetzung



Netztopologien

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

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:2:2_01

Last update: **2021/12/14 16:20**

2.3) Übertragungsmedien

Die Übertragungsmedien sind die Straßen der Daten. Der Aufbau dieser Straßen muss sehr gut geplant werden, um alle aktuellen Anforderungen bzw. eventuelle zukünftige Anforderungen ohne großartige Veränderungen zu erfüllen.

Als **Maßeinheit für die Übertragungsgeschwindigkeit** werden die Werte in **bit/s, b/s bps, -> also Bit pro Sekunde** angegeben. Achtung: Nicht zu verwechseln mit **Byte/s -> Byte pro Sekunde!!**

$$C = D/t \text{ (bits)/(s)}$$

1) Rechenbeispiel:

Es werden 100MB in 10s übertragen. Wie hoch ist die Übertragungsgeschwindigkeit?

- $D=100\text{MByte}$
- $t=10\text{s}$

Rechenschritt	Berechnung
D umwandeln in bits	$100*1024*1024*8 = 838860800 \text{ ``bits``}$
C berechnen	$C=D/t=(838\ 860\ 800)/10=83886080 \text{ `(bit)/(s)`}$
C umwandeln in Mbit/s	$(83886080)/(1024)/(1024)=80 \text{ `(Mbit)/(s)`}$

2) Rechenbeispiel:

Max hat eine Datentransferrate von 10Mbit/s Download und 2Mbit/s Upload.

a) Wie lange braucht er, um 10MB runterzuladen?

- $D=10\text{MB}$
- $C=10\text{Mbit/s}$

Rechenschritt	Berechnung
D umwandeln in bits	$10*1024*1024*8 = 83886080 \text{ ``bits``}$
C umwandeln in bit/s	$10*1024*1024=10485760 \text{ `(bit)/(s)`}$
Formel umformeln	$t=(D)/(C) \text{ `(bits)/((bit)/s)`}$
In Formel einsetzen	$t=83886080/10485760=8 \text{ `s`}$

b) Wie lange braucht er, um 10MB hochzuladen?

Rechenschritt	Berechnung
D umwandeln in bits	$10*1024*1024*8 = 83886080 \text{ ``bits``}$
C umwandeln in bit/s	$2*1024*1024=2097152 \text{ `(bit)/(s)`}$
Formel umformeln	$t=(D)/(C) \text{ `(bits)/((bit)/s)`}$
In Formel einsetzen	$t=83886080/2097152=40 \text{ `s`}$

Leitergebundene Übertragung

Bei einer leitergebundenen Übertragung werden Medien in Form von Kabeln benötigt (Metallische Leiter, Glasfaser).

Ein Kabel besteht zumindest aus einer Ader (=Faser).

Mehrere Adern sind durch entsprechende Isolationsschichten getrennt.

Alle Adern wiederum werden von einem Mantel als Schutz umgeben.

Die Übertragung selbst erfolgt durch elektromagnetische Schwingungen.

Koaxialkabel

Das früher verwendete Koaxialkabel ist in modernen Netzen praktisch vollständig von Twisted Pair-Kabeln (TP) und Lichtwellenleiter (LWL) abgelöst worden.



INFOTIP

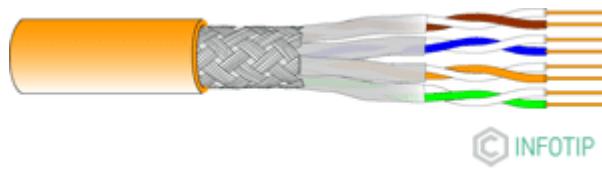
Es besteht aus

- einem Innenleiter (Kupfer, Stahlkupfer)
- einer Isolation (Dielektrikum)
- einer Abschirmung (Metallgeflecht schützt vor magnetischen Störungen -> Rauschen & Übersprechen)
- einem Mantel

Es waren bis zu 10Mbps möglich:

- Thicknet (10Base5)
- Thinnet (10Base2) - Heute noch bei Satellitenempfang im Einsatz

Twisted-Pair Kabel



INFOTIP

Twisted Pair ist ein vier-, acht- oder mehradriges Kupferkabel, bei dem jeweils zwei Adern miteinander verdrillt sind. Durch die Verdrillung kompensieren sich Leitungskapazität und -induktivität. Dadurch steigt die Übertragungsbandbreite und die mögliche Übertragungsreichweite wird praktisch nur durch die Dämpfung des Wirkwiderstandes begrenzt. Die Verwendung von symmetrischen Signalen (Differentialspannungen) erhöht die Festigkeit gegen elektromagnetische Störstrahlung.



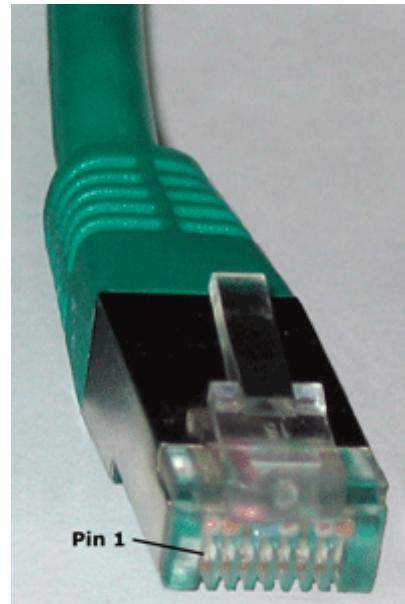
Twisted Pair-Kabel gibt es in zahlreichen Varianten. Twisted Pair-Verbindungen werden außer in der Kommunikationstechnik (Netzwerkkabel, Telefonkabel) auch bei HDMI-, DVI- und LVDS-(in LCD/Plasma-TV zwischen Signalprozessor und Display) Verbindungen eingesetzt. Die Anzahl der Leiterpaare im Kabel hängt dabei von der benötigten Datenübertragungsrate ab. In Netzwerken wird für jede Übertragungsrichtung (senden, empfangen) wird jeweils ein Adernpaar (bei 100BaseT4 und 1000BaseT jeweils zwei) genutzt. Die Übertragungsreichweite ist abhängig vom Aufbau des Kabels, von der Dämpfung (=Länge) des Kabels und von den externen Störeinflüssen. Twisted Pair-Kabel für Netzwerke gibt es in zahlreichen Varianten:

Bild	Benennung	Beschreibung
U/UTP (UTP)	U/UTP-Kabel	Unshielded/Unshielded Twisted Pair sind nicht abgeschirmte verdrillte Leitungen und gehörten früher typischerweise der CAT3 an. UTP-Kabel sollten im industriellen Bereich oder in der Datentechnik mit hohen Datenraten nicht verwendet werden.
S/UTP (S/UTP)	S/UTP-Kabel	Screened/Unshielded Twisted Pair haben einen Gesamtschirm aus einem Kupfergeflecht zur Reduktion der äußeren Störeinflüsse
F/UTP (F/UTP)	F/UTP-Kabel	Foilshielded/Unshielded Twisted Pair besitzen zur Abschirmung einen Gesamtschirm, zumeist aus einer alukaschierten Kunststofffolie
U/FTP (FTP)	U/FTP-Kabel	Unshielded/Foilshielded Twisted Pair auch genannt CAT5 oder CAT5e . Die Leitungssader sind paarweise mit Folie abgeschirmt
S/FTP (S/FTP)	S/FTP-Kabel	Screened/ Foilshielded Twisted Pair auch genannt CAT6 sollten in Bereichen mit hoher Störstrahlung (z.B. Büros mit mehreren PCs) eingesetzt werden.
SF/FTP (SF/FTP)	SF/FTP-Kabel	Screened Foilshielded/Foilshielded Twisted Pair auch genannt CAT6e oder CAT7 besitzen eine Abschirmung für jedes Kabelpaar sowie eine doppelte Gesamtschirmung. Hierdurch kann eine optimale Störleistungsunterdrückung erreicht werden. Auch das Übersprechen zwischen den einzelnen Aderpaaren wird so wirksam unterdrückt



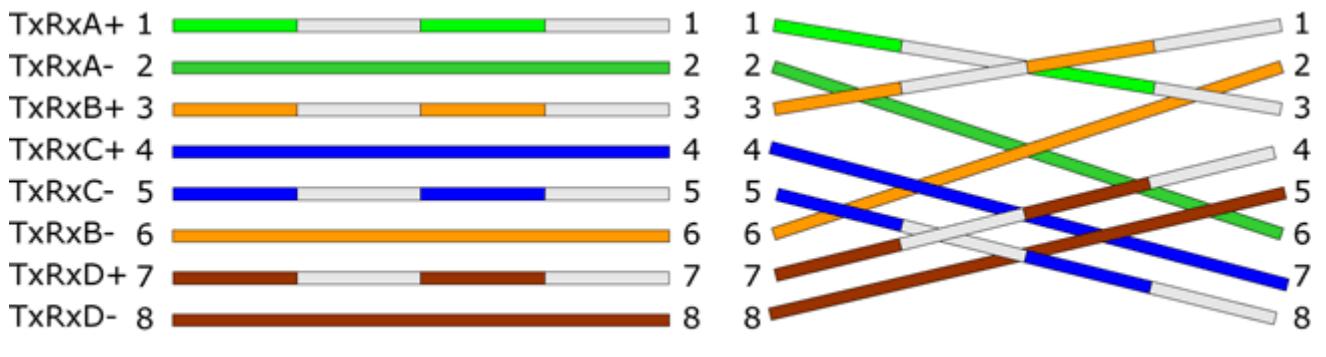
Die Preisunterschiede zwischen CAT-5e- Kabeln und CAT-7-Kabeln ist so gering, dass es sich bei Neuinstallation auf jeden Fall empfiehlt, CAT-7-Kabel einzusetzen. Dieses ist als einziges Kupfermedium in der Lage mit dem kommenden 10GBit-LAN verwendet zu werden.

Verbinder - RJ45



Der typische Standardverbinder für die Twisted-Pair-Verkabelung eines kupfergebundenen Ethernet-Netzwerkes ist der **8polige Western-Modularstecker RJ-45** (8P8C), auch RJ-48 oder RJ-49 genannt. RJ-45 Steckverbindungen können auf zwei Arten belegt sein, wobei die Belegung nach T568B am weitesten verbreitet zu sein scheint:

Belegung nach EIA/T-T568A		Belegung nach EIA/T-T568B	
Pin	Farbe	Pin	Farbe
1	weiß-grün	1	weiß-orange
2	grün	2	orange
3	weiß-orange	3	weiß-grün
4	blau	4	blau
5	weiß-blau	5	weiß-blau
6	orange	6	grün
7	weiß-braun	7	weiß-braun
8	braun	8	braun



Bei 1:1-Verbindungen sind beide Beschaltungen elektrisch zueinander kompatibel. Nur bei Erweiterungen von fest verdrahteten Netzen ist festzustellen, welche Belegung bereits vorgegeben ist. Normale Verbindungskabel („Patchkabel“) mit RJ-45-Steckern sind 1:1 verschaltet, d.h. Pin 1 des einen Steckers geht auf den Pin 1 des anderen Steckers usw. Nur in besonderen Fällen, wenn z.B.

zwei Netzwerkkarten direkt miteinander verbunden werden sollen oder wenn Netzwerkkomponenten (z.B. Hubs älterer Bauart) über keinen dedizierten Uplink-Port verfügen, kann der Einsatz von Crossover-Kabeln notwendig werden.

RJ45-Stecker crimpeln



LichtWellenLeiter (LWL)

Sind mit der Netzwerkverkabelung weite Strecken zu überwinden, z.B. zwischen einzelnen Gebäuden auf einem Fabrikgelände („Campusbereich“), sind sehr hohe Datenübertragungsraten (z.Zt. bis zu 170Gb/s) gefordert oder wenn sich die Datenübertragung per Kupferkabel aus technischen Gründen (z.B. bei extremer Störstrahlung) oder aus Gründen der Sicherheit verbietet, werden Lichtwellenleiter (LWL, Glasfasern) als Übertragungsmedium eingesetzt. Die Lieferprogramme der Hersteller erlauben mittlerweile die Übertragungsstrecken bis zum Einzelplatz komplett auf der Basis von LWL auszuführen.



Funktionsweise

Aufbau und Prinzip

In einem LWL werden die Informationen nicht, wie in einem Kupferkabel, elektrisch übertragen, sondern mit **Licht**.

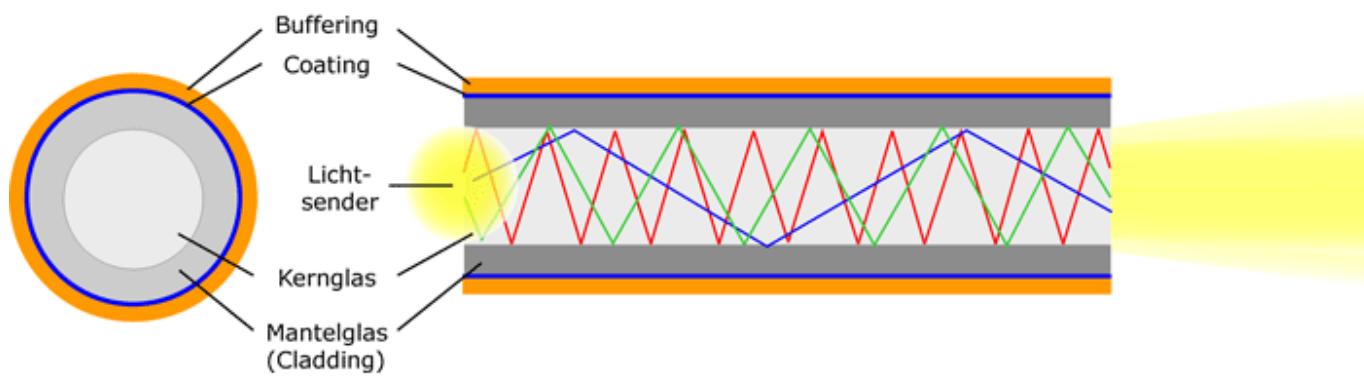
Der eigentliche LWL ist eine Faser aus Glas oder Kunststoff. Jede Faser besteht aus zwei Schichten. Der konzentrische Kern besteht aus einem optischen Material mit einem hohen Brechungsindex, das Mantelglas („Cladding“) aus einem Material mit niedrigem. Licht, das in einem bestimmten Winkelbereich auf den Übergang von Kern zum Mantel trifft wird dort vollständig reflektiert. Über solche fortlaufenden Totalreflexionen pflanzt sich das Licht durch den LWL bis zum Ende der Faser fort.

Je steiler der Einfallswinkel des Lichts bei der Einspeisung in den LWL ist, desto häufiger wird die Lichtwelle reflektiert. Mit jeder Reflektion der Lichtwelle wird der Weg, des sogenannten Modes, länger.

Licht, das wenig häufig reflektiert wird, hat einen kürzeren Weg und durchläuft die Faser schneller. Es ist Licht niedrigen Modes.

Licht, das sehr häufig reflektiert wird, hat eine niedrige Ausbreitungsgeschwindigkeit in der Faser. Es ist Licht hohen Modes.

Erzeugt die Lichtquelle des Senders ein nicht-kohärentes Licht, tritt das Licht mit einer Vielzahl unterschiedlicher Winkel in die Faser ein. Dadurch entstehen natürlich durch die unterschiedlichen Moden Laufzeitunterschiede zwischen den Signalanteilen. Ein Eingangsimpuls mit steilen Flanken wird dadurch verschliffen und in seiner Breite gedehnt. Je länger ein Kabel ist, desto höher wird auch diese sog. Dispersion (Einheit: ns/km). Die Dispersion beeinflusst direkt die Übertragungsbandbreite der Glasfaserverbindung.



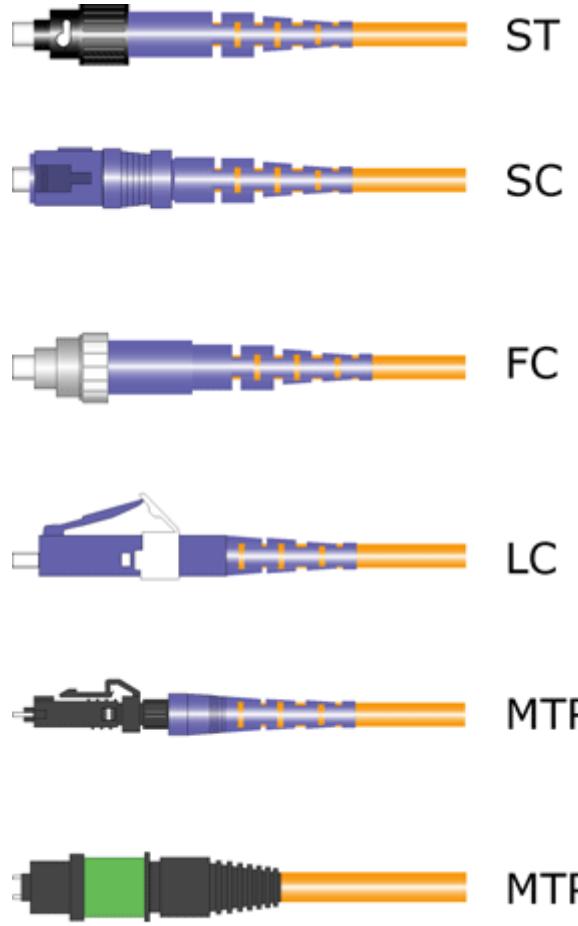
INFOTIP



Da die Fasern sehr dünn und empfindlich sind, werden sie zum mechanischen Schutz mit einer Kunststoffbeschichtung („Coating“) und einem Schutzüberzug versehen. In einem LWL-Kabel können mehrere Fasern, sogar in mehreren Bündeln, zusammen gefasst sein.

Verbinder

Die Hersteller von Netzwerkzubehör bieten konfektionierte Verbindungs- und Patchkabel mit einer Vielzahl von verschiedenen Steckerformen an. Meist sind die Kabel paarweise angelegt um beide Datenflussrichtungen (TX und RX) gleichzeitig herstellen zu können.



INFOTIP

Vorteile

- hohe Reichweite
- hohe Übertragungsbandbreite
- Potentialfrei, daher auch für explosionsgefährdete Bereiche geeignet
- hohe Störfestigkeit, LWL können sogar zu Energieversorgungskabeln parallel verlegt werden
- hohe Abhörsicherheit

Nachteile

- Material für die Verkabelung ist teuer
- teure Verbindungstechnik
- Die Montagekosten sind wegen des höheren technischen Aufwandes höher
- komplexe und teure Messtechnik
- zusätzliche Kosten für Medienkonverter auf Kupfer-Ethernet

Leiterunggebundene Übertragung

Als leiterunggebundene Übertragung bezeichnet man eine Übertragung per

- Funk
- Ultraschall

- Infrarot
- Laser
- Licht

Drahtlose Übertragung (WLAN)

Die Übertragung von Informationen ohne Kabel ist mittlerweile in vielen Lebensbereichen als praktische Alternative eingezogen. Von der Fernbedienung eines Fernsehers über drahtlose Lautsprecher bis zum Smartphone gibt es viele Beispiele für die Umsetzung dieser Technik. Dabei werden Funksignale in frei verfügbaren Frequenzbändern anstelle von Kabeln für die Datenübertragung verwendet.

Vorteile

- Es sind keine baulichen Maßnahmen innerhalb eines Gebäudes nötig.
- Die baulichen Maßnahmen zwischen verschiedenen Gebäuden sind geringer als bei einer Verkabelung.
- Höhere Mobilität, da theoretisch jeder Punkt eines Firmengeländes drahtlos erreichbar ist.

Nachteile

- Oft geringere Datenübertragungsraten als bei Kabeln, die abhängig von Hindernissen sind.
- Anfällig für Störeinflüsse und Abhören durch Unbefugte.
- Probleme mit Ausleuchtung und Reflexionen.
- Bei vielen gleichzeitigen Nutzern an einem WLAN-Zugang bricht die Übertragungsrate ein (Shared Media).

Sicherheit als kritischer Bereich

Gerade im Bereich Sicherheit gibt es bei WLANs einige Punkte zu beachten, die sich auch in einem etwas größeren Konfigurationsaufwand äußern. Die sogenannte **SSID (Service Set Identifier)** kann eine eindeutige Identifikation (Firmenname etc.) enthalten, damit bei Problemen eine Kontaktaufnahme mit dem Betreiber möglich ist. Ein Verbergen bringt nicht viel, da die SSID in jedem Paket mitgeschickt wird und es Programme gibt, die auch verborgene SSIDs auslesen können. Eine versehentliche Verbindung durch Unbefugte ist bei verschlüsselten Zugängen nicht zu erwarten. Es gibt auch Empfehlungen, als SSID eine zufällige Zeichenfolge einzugeben, damit die SSID keine Rückschlüsse auf den Betreiber zulässt.

Letztendlich sollte die Übertragung im WLAN nur **verschlüsselt** erfolgen, wobei die Verschlüsselung mit **WEP (Wired Equivalent Privacy)** unsicher ist. Besser ist der Einsatz von **WPA (Wi-Fi Protected Access)** bzw. der Nachfolgetechnologie **WPA2**, da hier deutlich stärkere Verschlüsselungsmechanismen mit **AES (Advanced Encryption Standard)** verwendet werden. Zusammen mit **TKIP (Temporal Key Integrity Protocol)** sind allerdings nur max. 54 Mbit/s möglich! Höhere Raten erreicht z. B. WPA2 zusammen mit CCMP (Counter-Mode/CBC-Mac Protocol)

Grundlegende Beschreibung

Kommunikation über WLAN erfolgt entweder als Punkt-zu-Punkt- oder als Mehrpunkt-Kommunikation. Die erste Variante dient z. B. der Überwindung größerer Distanzen durch den Einsatz zweier Richtantennen.

Bei der Mehrpunkt-Kommunikation werden ein oder mehrere sogenannte Access Points eingesetzt, die im Prinzip jeweils wie Zentralen (Verteiler) fungieren und die Datenströme mehrerer Clients koordinieren.

Diese Access Points können bei größeren Installationen über Kabel und geeignete Managed Switches eine Verbindung zu einem sogenannten RADIUS (Remote Authentication Dial-In User Service)-Server erhalten, um zwischen berechtigten und nicht berechtigten Sendern zu unterscheiden (Authentifizierung).

Eine Unterscheidung über die MAC-Adresse sollte nur den zum Zugang berechtigten Geräten vorbehalten bleiben, die sich nicht per RADIUS authentifizieren können. Dabei wird in einer Zugangsliste (Access Control Table) vom Managed Switch die MAC-Adresse eingetragen. Auf einem Access-Point bringt dies keine höhere Sicherheit, da per Funk eine MAC-Adresse unverschlüsselt verschickt wird und somit gefälscht werden kann. Die kleinste Einheit ist eine sogenannte Funkzelle, womit der Bereich gemeint ist, der von einem Sender (=Access Point) abgedeckt werden kann. Er umfasst ca. 30m im Gebäude und bis zu 300m im Freien. Mit speziellen Antennen können auch mehrere Kilometer überbrückt werden.

ISM-Frequenzbänder

In den meisten Fällen wird von den Herstellern ein sogenanntes **ISM-Band (Industrial, Scientific and Medical)** verwendet. Manchmal finden Sie auch die Abkürzung ISMO, wobei der letzte Buchstabe für den Begriff „Office“ (Büro) steht. Der Einsatz dieser Frequenzbänder bietet zwei Vorteile. Sie sind

- **gebührenfrei**
- **genehmigungsfrei**

Hierin liegt aber auch gleichzeitig der Nachteil. Sie werden von sehr vielen Herstellern für die unterschiedlichsten Zwecke genutzt, wie z. B. drahtlose Lautsprecher, elektronische Türöffnung bei Autos oder Garagen, und so ist die Gefahr, dass sich Geräte gegenseitig stören, relativ hoch.

Die für WLAN wichtigsten ISM-Bänder sind

- das **2,4-GHz-Band** (2,3995 bis 2,4845 GHz) mit max. **13 überlappenden Kanälen** von 20 MHz Bandbreite; auch 40 MHz Bandbreite ist möglich, dann aber mit weit weniger nutzbaren Kanälen. Es sind Geschwindigkeiten bis zu 300MBit/s möglich.
- das **5-GHz-Band** (5,150 bis 5,350 GHz für Kanalnummer 36-64 und 5,470 bis 5,725 GHz für Kanalnummer 100-140) mit Kanälen von 20, 40, 80 oder 160 MHz Bandbreite, wobei **max. 19 Kanäle** bei 20 MHz Bandbreite nicht überlappend nutzbar sind. Beim Funken mit 40 MHz Bandbreite sind 2 dieser Kanäle gebündelt erforderlich, mit 80 MHz 4 Kanäle usw. Die **Reichweite ist geringer** als im 2,4 GHz-Band. Es sind Geschwindigkeiten bis zu 600MBit/s möglich.
- zukünftig das 60-GHz-Band (57 bis 66 GHz) mit vier 2000 MHz breiten Funkkanälen für kurze Distanzen

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

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:2:2_02 

Last update: **2021/12/14 16:20**

2.4) Ethernet

Erklärung

Ethernet ist eine Technik, die **Software (Protokolle usw.) und Hardware (Kabel, Verteiler, Netzwerkarten usw.) für kabelgebundene Datennetze spezifiziert**, welche ursprünglich für lokale Datennetze (LANs) gedacht war und daher auch als LAN-Technik bezeichnet wird. Sie ermöglicht den Datenaustausch in Form von Datenframes zwischen den in einem lokalen Netz (LAN) angeschlossenen Geräten (Computer, Drucker und dergleichen). Derzeit sind Übertragungsraten von 1, 10, 100 Megabit/s (Fast Ethernet), 1000 Megabit/s (Gigabit-Ethernet), 2,5, 5, 10, 40, 50, 100, 200 und 400 Gigabit/s spezifiziert. In seiner ursprünglichen Form erstreckt sich das LAN dabei nur über ein Gebäude; Ethernet-Varianten über Glasfaser haben eine Reichweite von bis zu 70 km.

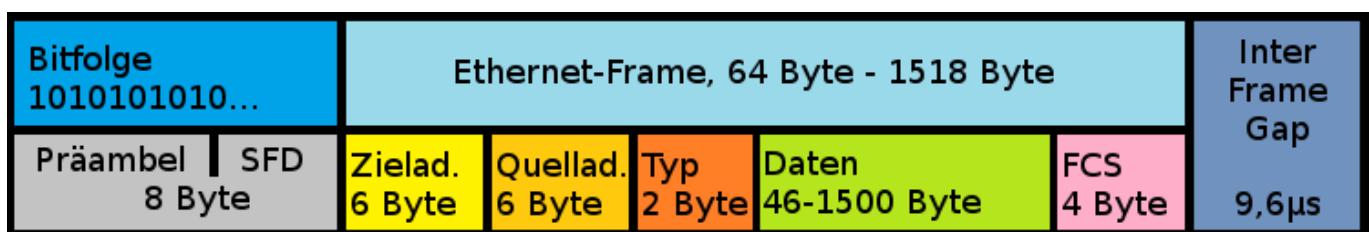
Die Ethernet-Protokolle umfassen Festlegungen für Kabeltypen und Stecker sowie für Übertragungsformen (Signale auf der Bitübertragungsschicht, Paketformate). Im OSI-Modell ist mit Ethernet sowohl die **physische Schicht (OSI Layer 1)** als auch die **Data-Link-Schicht (OSI Layer 2)** festgelegt.

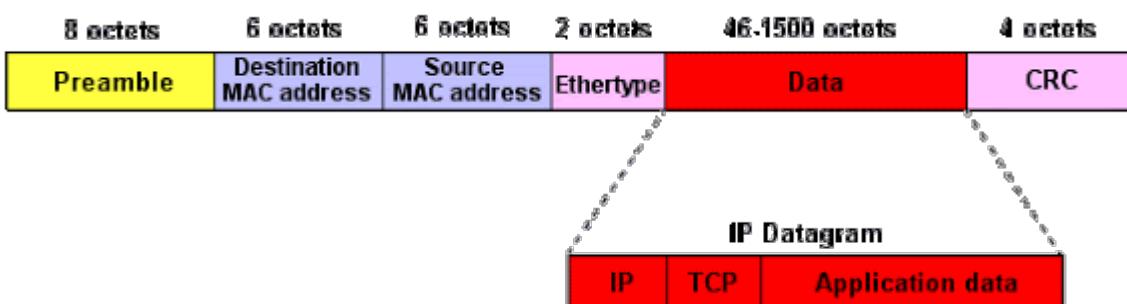
Ethernet basiert auf der Idee, dass die Teilnehmer eines LANs Nachrichten durch Hochfrequenz übertragen, allerdings **nur innerhalb eines gemeinsamen Leitungsnetzes**. Jede Netzwerkschnittstelle hat **einen global eindeutigen 48-Bit-Schlüssel**, der als **MAC-Adresse** (=Media-Access-Control-Adress) bezeichnet wird. Das stellt sicher, dass alle Systeme in einem Ethernet unterschiedliche Adressen haben.

Ethernet Frame

Bei der Übertragung von Daten über Ethernet ist das Ethernet-Frame hauptverantwortlich für die korrekte Regelsetzung und erfolgreiche Übermittlung von Datenpaketen. Versendete Daten über Ethernet werden vom Frame sozusagen getragen. Ein Ethernet-Frame ist zwischen 64 Byte und 1518 Byte groß, abhängig von der Größe der zu transportierenden Daten.

Im OSI-Modell befindet sich der Frame auf der Sicherungsschicht, die für die fehlerfreie Übertragung verantwortlich ist und trennt den Bitdatenstrom in Blöcke bzw. Frames auf.





Ein Ethernet-Frame muss standardmäßig mindestens 64 Byte groß sein, damit die Kollisionserkennung funktioniert, und kann maximal 1.518 Byte groß sein. Das Paket beginnt immer mit einer Präambel, die die Synchronisation zwischen Sender und Empfänger regelt und einem „Start Frame Delimiter“ (SFD), der das Frame definiert. Beide Informationen sind eine Bitfolge im Format 10101010... Im eigentlichen Frame finden sich Informationen zu Ziel- und Quelladressen (MAC-Format) und Steuerinformationen (im Fall von Ethernet II das Type-Field, später eine Längenangabe), dann folgt der zu übermittelnde Datensatz. Eine „Frame Check Sequence“ (FCS) schließt als Prüfsumme das gesamte Frame (ausgenommen Präambel und SFD). Das Paket wird von einem „Inter Frame Gap“ abgeschlossen, der eine 9,6 µs lange Sendepause festlegt.

Ethernet II benutzt die klassische Framestruktur, die das sogenannte Type-Field („Typ“) beinhaltet, womit verschiedene Protokolle der Vermittlungsschicht definiert werden. Im OSI-Modell ist die Vermittlungsschicht (auch „Network Layer“) wichtig für die Schaltung von Verbindungen und die Bereitstellung von Netzwerkadressen. Das Type-Field wurde in späteren Frame-Formaten durch eine Längenangabe ersetzt.

CSMA/CD

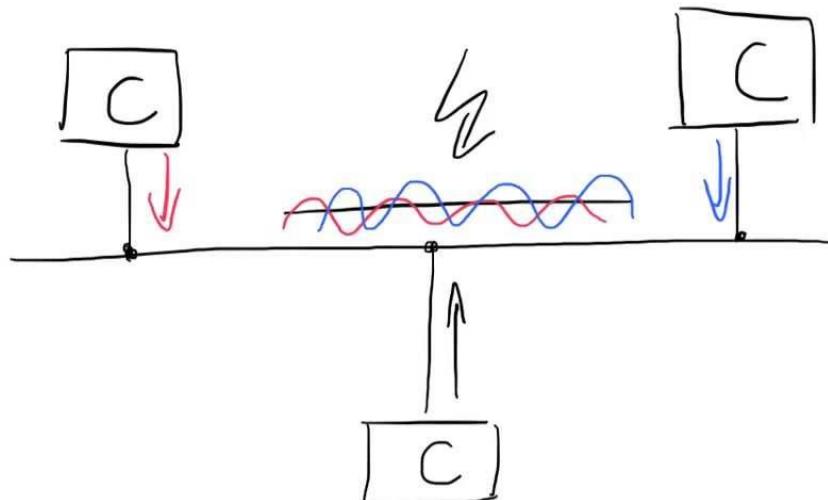
CSMA/CD (Carrier Sense Multiple Access with Collision Detection) ist das Zugriffsverfahren des Ethernets. Die Grundidee dabei ist, dass jede Station zu senden beginnen kann, wann sie will. Die einzelnen Stationen haben jederzeit und konkurrierend Zugang (Multiple Access) zum gemeinsamen Übertragungsmedium. Das grundlegende Motto könnte damit lauten:

Jeder darf, wann er will

Eingesetzt wird dieses Verfahren bei logischen Bus-Topologien. Dabei ist egal, ob physikalisch eine Bus- oder eine Stern-Topologie vorliegt.

Vorgehen

Durch Regelungen wird versucht, das Risiko zu minimieren, dass zwei Stationen ungünstigerweise gleichzeitig zu senden beginnen und somit die Signale auf dem Übertragungsweg zerstört/gestört werden (**=Kollision**).



1) Leitung prüfen (Carrier Sense)

Der erste Teil der Abkürzung steht für Kollisionsverhinderung. Dabei wird vor einer geplanten Sendung das Übertragungsmedium abgehört, ob dieses frei ist. Wenn das Medium frei ist, wird gesendet. -> **LISTEN BEFORE TALKING**

2) Erkennen von Kollisionen (Collision Detection) Kommt es trotzdem zu einer Kollision, weil zwei Stationen gleichzeitig zu senden beginnen (**Multiple Access**), dann muss diese Kollision erkannt (**Collision Detection**) und reagiert werden. Dabei müssen alle Stationen immer am Medium horchen, ob eine Kollision auftritt. Ist dies der Fall, so sendet die erste Station, die eine Kollision erkennt, ein sogenanntes JAM-Signal aus. Jede Station, die das JAM-Signal registriert, stoppt unmittelbar das Senden von Daten. Nach einer zufälligen Zeitspanne, wird das Medium wieder überprüft und anschließend wieder begonnen zu senden.



Video

Frage: Wie kann bei einer physikalischen Stern-Topologie eine Kollision auftreten?

Vorteile

- Jeder kann zu jeder Zeit senden

Nachteile

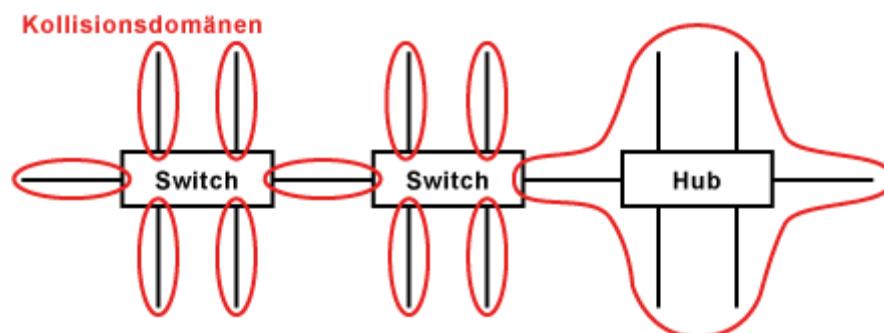
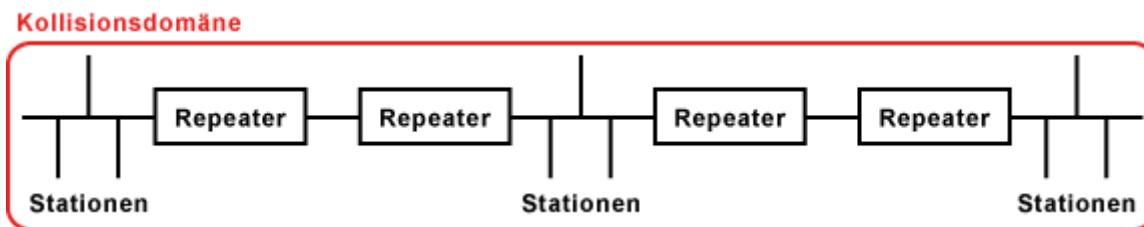
- Je mehr Stationen in einer Kollisionsdomäne, desto häufiger treten Kollisionen auf
- Der Zeitpunkt einer Sendung ist zufällig
- Das Verfahren ist ungeeignet für zeitkritischen Anwendungen

Kollisionsdomäne

Mit dem Begriff Kollisionsdomäne wird in einem Computernetz ein Teilbereich aus Teilnehmerstationen in derselben OSI-Modell-Schicht 1 bezeichnet. Eine Kollisionsdomäne umfasst alle Netzwerkgeräte, die um den Zugriff auf ein gemeinsames Übertragungsmedium konkurrieren. Das Übertragungsmedium ist daher eine zwischen allen Netzstationen geteilte Ressource. Grundlegende Vorstellung dabei ist, dass alle Netzwerkeinheiten die Chance zur gleichberechtigten Nutzung des Netzwerkes besitzen.

Bei einem gemeinsamen Medium kann zu einer bestimmten Zeit nur jeweils eine Station Informationen übertragen, die an alle anderen Stationen übertragen bzw. von diesen empfangen wird. Fangt in einem derartigen gemeinsamen Schicht-1-Segment zwei Stationen gleichzeitig an zu senden, kommt es zu Kollisionen.

Beispiele für Kollisionsdomänen:



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

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:2:2_03 

Last update: **2021/12/14 16:20**

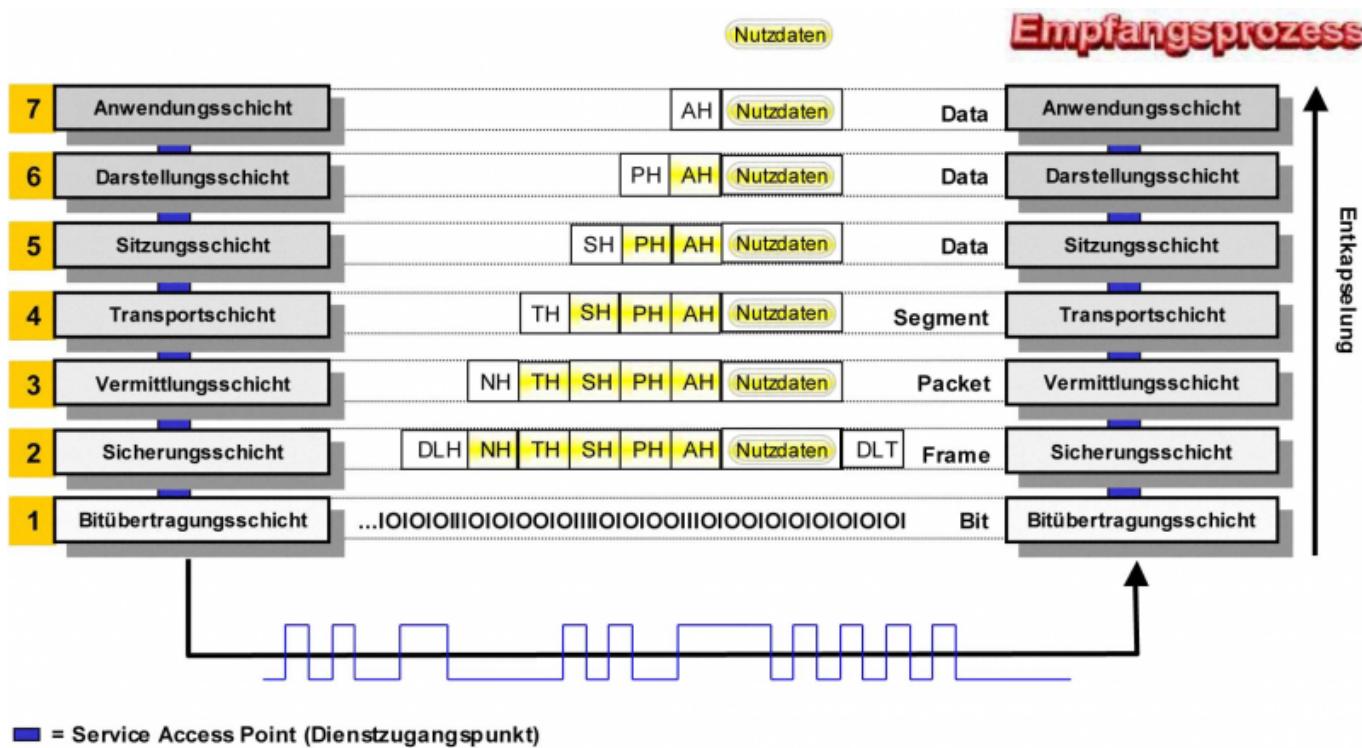
2.5) OSI - Schichtenmodell

Das OSI-7-Schichtenmodell ist ein Referenzmodell für herstellerunabhängige Kommunikationssysteme bzw. eine Design-Grundlage für Kommunikationsprotokolle und Computernetze. OSI steht für Open System Interconnection (Offenes System für Kommunikationsverbindungen) und wurde von der ISO (International Organization for Standardization), das ist die Internationale Organisation für Normung, als Grundlage für die Bildung von offenen Kommunikationsstandards entworfen. Bei allen ISO-Standards handelt es sich um Handlungsempfehlungen. Die Einhaltung einer ISO-Norm ist freiwillig. In der Regel wird die Einhaltung der ISO-Standards von verschiedenen Seiten, zum Beispiel Kooperationspartnern, Herstellern und Kunden, gefordert.



Das Modell

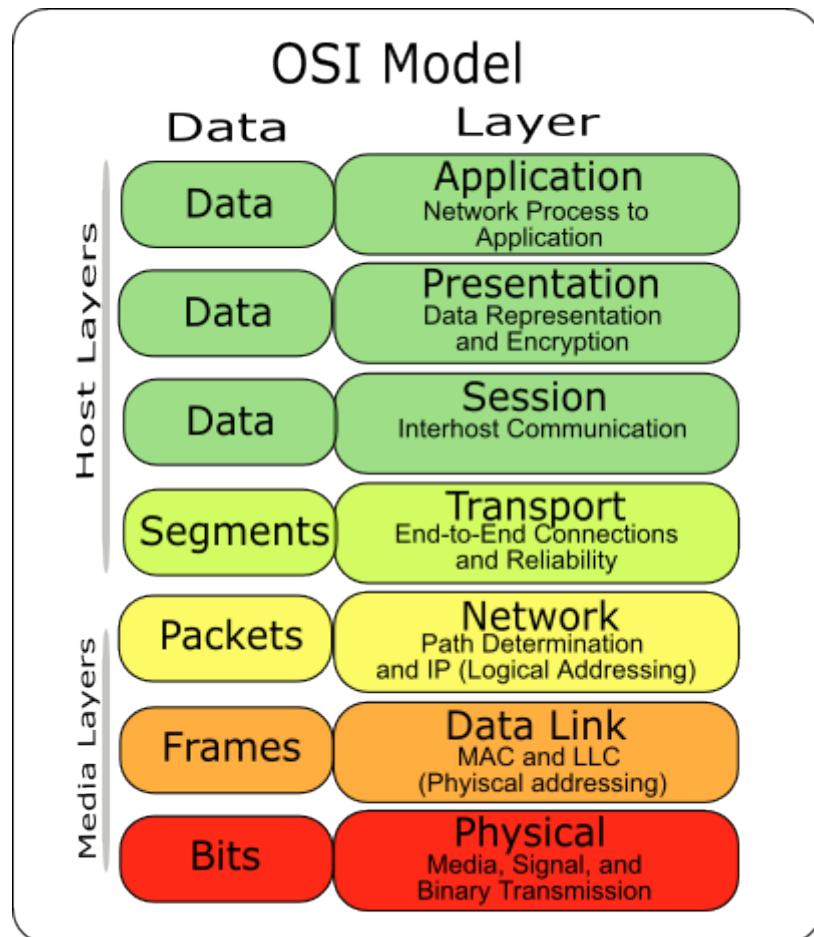
(Offenes System für Kommunikationsverbindungen) und wurde von der ISO (International Organization for Standardization), das ist die Internationale Organisation für Normung, als Grundlage für die Bildung von offenen Kommunikationsstandards entworfen. Bei allen ISO-Standards handelt es sich um Handlungsempfehlungen. Die Einhaltung einer ISO-Norm ist freiwillig. In der Regel wird die Einhaltung der ISO-Standards von verschiedenen Seiten, zum Beispiel Kooperationspartnern, Herstellern und Kunden, gefordert.



Die Schichten im Detail

Application Layer (Anwendungsschicht)	Benutzerschnittstelle, Dienste, Anwendungen und Netzmanagement
Schicht 7	Die Anwendungsschicht stellt Funktionen für die Anwendungen zur Verfügung. Diese Schicht stellt die Verbindung zu den unteren Schichten her. Auf dieser Ebene findet auch die Dateneingabe und -ausgabe statt.
Presentation Layer (Darstellungsschicht)	Übersetzung, Verschlüsselung , Kompression in Standardformate
Schicht 6	Die Darstellungsschicht setzt die Daten der Anwendungsebene in ein Zwischenformat um. Diese Schicht ist auch für Sicherheitsfragen zuständig. Durch sie werden Dienste zur Verschlüsselung von Daten bereitgestellt und gegebenenfalls Daten komprimiert.
Session Layer (Sitzungsschicht)	Erstellung einer Verbindung, Freigabe von Verbindungen, Dialogsteuerung
Schicht 5	Diese Schicht ermöglicht zwei Anwendungen auf verschiedenen Computern, eine gemeinsame Sitzung aufzubauen, damit zu arbeiten und sie zu beenden. Sie übernimmt ebenfalls die Dialogsteuerung zwischen den beiden Computern einer Sitzung und regelt, welcher der beiden wann und wie lange Daten überträgt.
Transport Layer (Transportschicht)	Logische Ende-zu-Ende-Verbindungen (Transportkontrolle, Paketbildung)
Schicht 4	Die Transportschicht stellt die zuverlässige Auslieferung der Nachrichten sicher und erkennt sowie behebt allfällige Fehler. Sie ordnet bei Bedarf auch die Nachrichten in Paketen neu, indem sie lange Nachrichten zur Datenübertragung in kleinere Pakete aufteilt. Am Ende des Weges stellt sie die kleinen Pakete wieder zur ursprünglichen Nachricht zusammen. Die empfangene Transportebene sendet auch eine Empfangsbestätigung.
Network Layer (Vermittlungsschicht)	Routing (Internet), Datenflusskontrolle, Adressierung

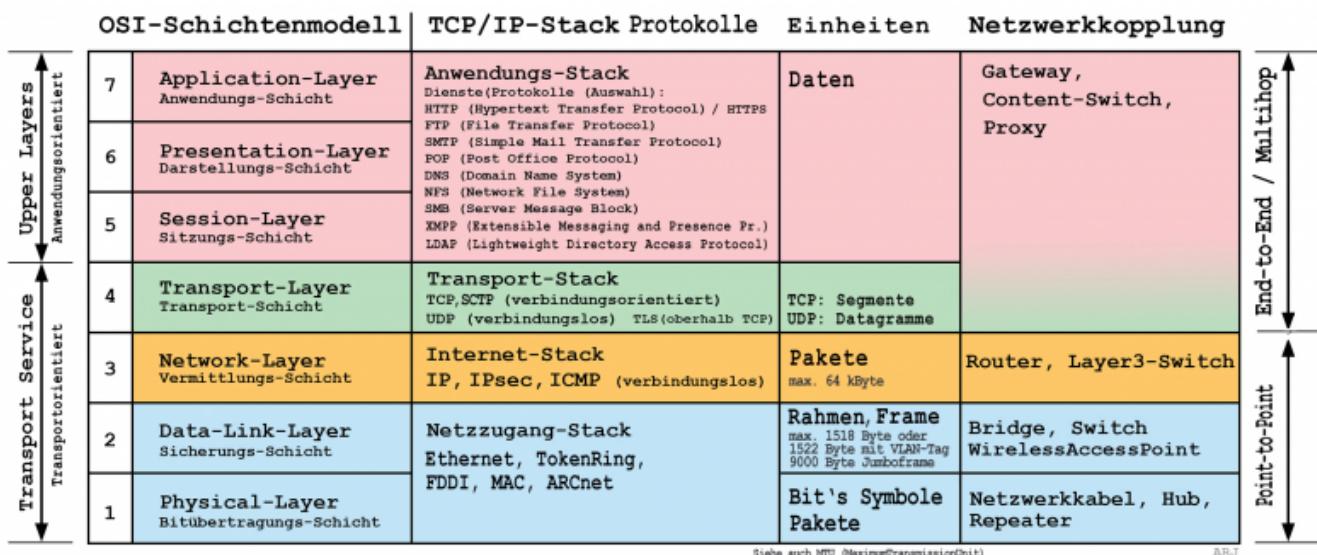
Application Layer (Anwendungsschicht)	Benutzerschnittstelle, Dienste, Anwendungen und Netzmanagement
Schicht 3	Die Vermittlungsschicht steuert die zeitliche und logische getrennte Kommunikation zwischen den Endgeräten, unabhängig vom Übertragungsmedium und der Topologie. Auf dieser Schicht erfolgt erstmals die logische Adressierung der Endgeräte. Die Adressierung ist eng mit dem Routing (Wegfindung vom Sender zum Empfänger) verbunden.
Data Link Layer (Sicherungsschicht)	Logische Verbindungen mit Datenpaketen und elementare Fehlererkennungsmechanismen
Schicht 2	Die Sicherungsschicht sorgt für eine zuverlässige und funktionierende Verbindung zwischen Endgerät und Übertragungsmedium. Zur Vermeidung von Übertragungsfehlern und Datenverlust enthält diese Schicht Funktionen zur Fehlererkennung, Fehlerbehebung und Datenflusskontrolle. Auf dieser Schicht findet auch die physikalische Adressierung von Datenpaketen statt.
Physical Layer (Physikalische Schicht)	Maßnahmen und Verfahren zur Übertragung von Bitfolgen
Schicht 1	Die Bitübertragungsschicht definiert die elektrische, mechanische und funktionale Schnittstelle zum Übertragungsmedium. Die Protokolle dieser Schicht unterscheiden sich nur nach dem eingesetzten Übertragungsmedium und -verfahren. Das Übertragungsmedium ist jedoch kein Bestandteil der Schicht 1.



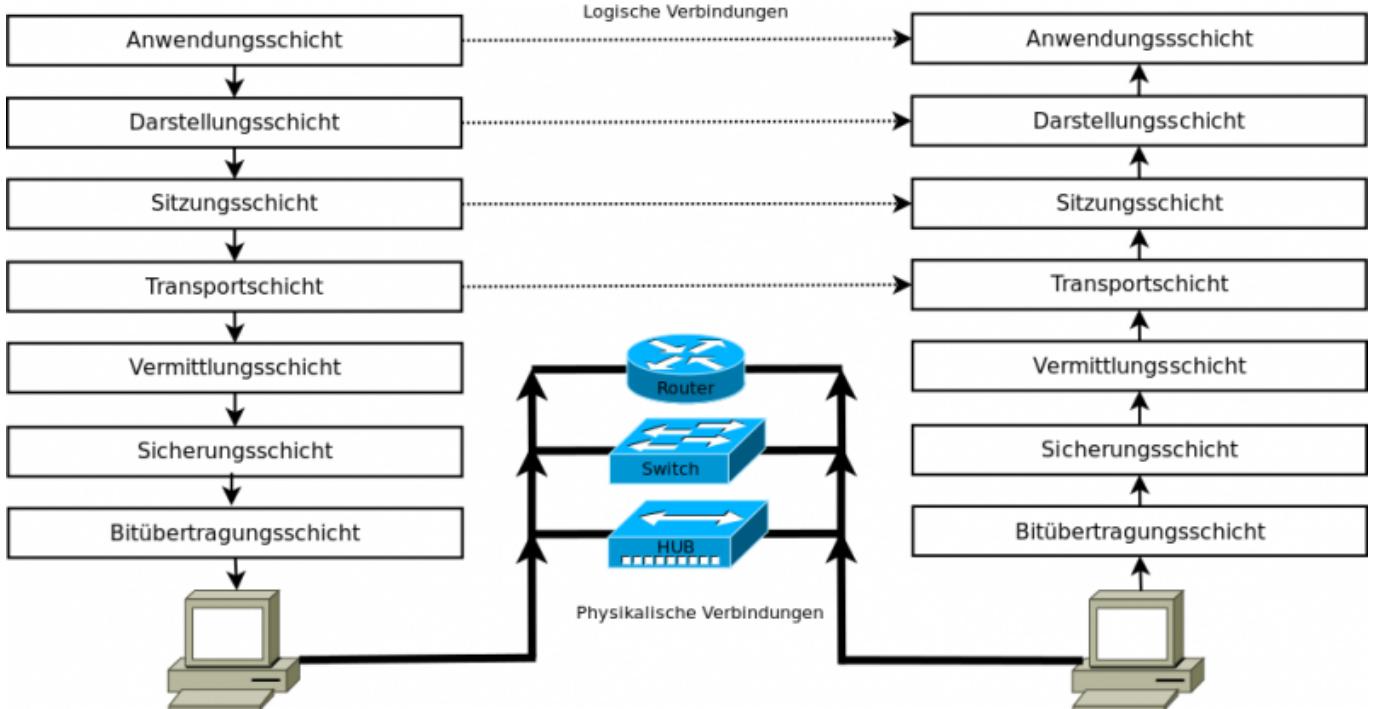
Protokolle

Protokolle sind eine **Sammlung von Regeln zur Kommunikation** auf einer bestimmten Schicht des OSI-Schichtenmodells. Die Protokolle einer Schicht sind zu den Protokollen der über- und untergeordneten Schichten weitestgehend transparent, so dass die Verhaltensweise eines Protokolls sich wie bei einer direkten Kommunikation mit dem Gegenstück auf der Gegenseite darstellt.

Die **Übergänge zwischen den Schichten sind Schnittstellen**, die von den Protokollen verstanden werden müssen. Weil manche Protokolle für ganz bestimmte Anwendungen entwickelt wurden, kommt es auch vor, dass sich **Protokolle über mehrere Schichten erstrecken** und mehrere Aufgaben abdecken. Dabei kommt es vor, dass in manchen Verbindungen einzelne Aufgaben in mehreren Schichten und somit mehrfach ausgeführt werden.



Anmerkung zur Skizze: End-zu-End-Verbindungen finden laut Definition erst ab Schicht 4 statt.



Das wichtigste Protokoll im Netzwerkverkehr, ist das **TCP & IP Protokoll** (Transmission Control Protocol & Internet Protocol), welche sich heute als Standard durchgesetzt haben.

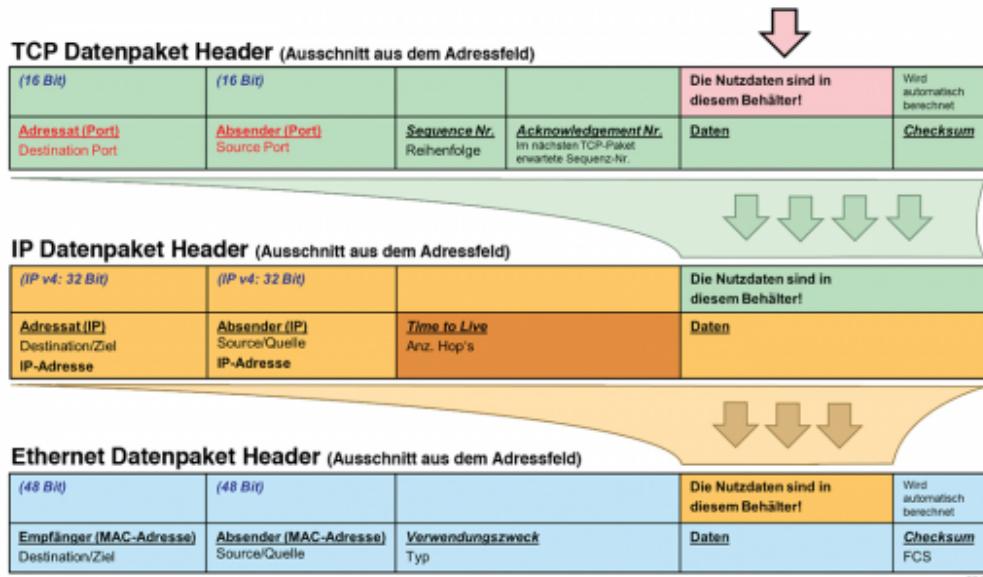
Neben TCP und IP gibt es natürlich noch viele weitere Protokolle, wie in der obigen Abbildung zu sehen ist.

Die folgende Tabelle listet einige dieser Protokolle auf und ordnet sie ins obige Modell ein:

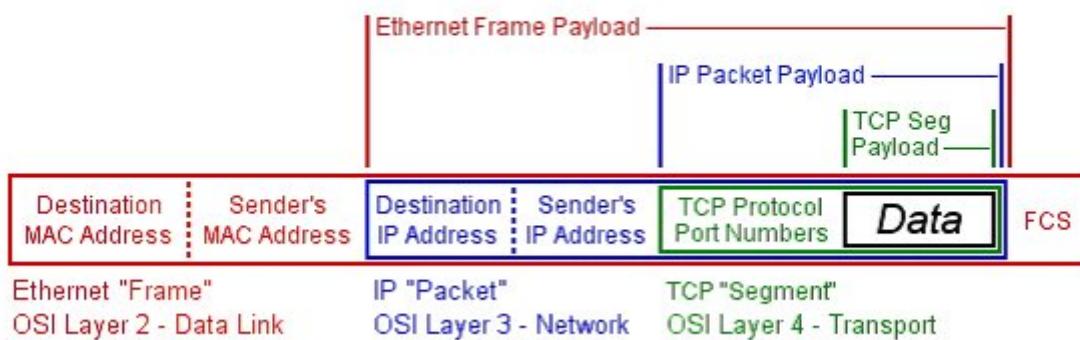
Protokoll	Schicht	Name	Beschreibung
FTP	5	File Transfer Protocol	Datenaustausch zwischen Rechnern
Telnet	5	Telecommunication Network Protocol	Terminalemulation zur Host-Kommunikation
SMTP	5	Simple Mail Transfer Protocol	Versenden von E-Mails
HTTP	5	Hypertext Transfer Protocol	Übertragen von HTML-Seiten
POP	5	Post Office Protocol	Abrufen von E-Mails
TCP	4	Transmission Control Protocol	Aufbau logischer Verbindungen zwischen Applikationen
UDP	4	User Datagram Protocol	Verbindungsloses Übertragungsprotokoll. Es ist nicht so gesichert wie TCP dafür aber schneller
IP	3	Internet Protocol	Verbindungsloses Protokoll zur Paketlenkung und Paketvermittlung über IP-Adressen
IPSec	3	IP Secure	Erweitert das reguläre IP-Protokoll um ein Bündel von Sicherheitsmechanismen
ARP	3	Address Resolution Protocol	Dient dazu logische IP-Adressen physikalischen MAC-Adressen zuzuordnen

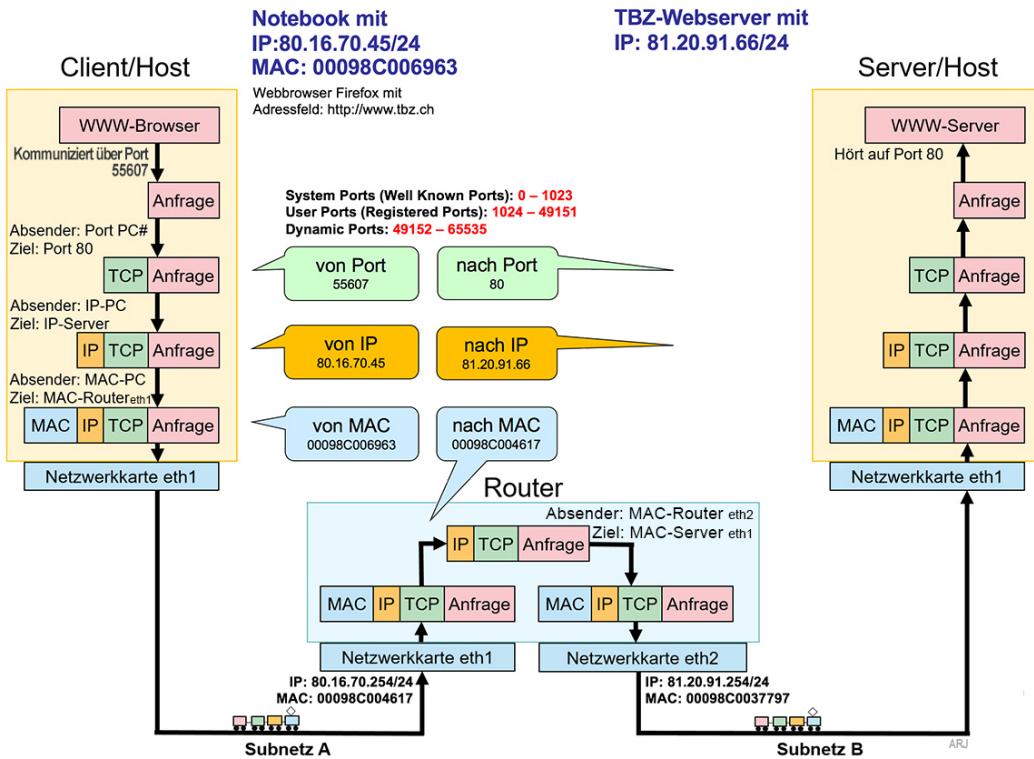
Datenkapselung

Unter der Datenkapselung versteht man den Prozess im OSI-Modell und im TCP/IP-Referenzmodell, der die zu versendenden Daten im Header (und ggf. Trailer) der jeweiligen Schichten ergänzt. Im OSI-Modell betrifft dies die Datagramme der Schichten 2 bis 4, die gekapselt (verpackt) werden.



Encapsulation Payloads





OSI-Schichtenmodell

TCP/IP-Referenzmodell

https://edu.juergarnold.ch/fach_it/netzwerktheorie/article1.html?q=osi#nw_3_19

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

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:2:2_04

Last update: **2021/12/14 16:20**

Netzwerkkomponenten

In der Netzwerktechnik unterscheidet man zwischen aktiven und passiven Netzwerk-Komponenten. Während **aktive Netzwerk-Komponenten eine eigene Logik haben**, zählen die passiven Netzwerk-Komponenten zur fest installierten Netzwerk-Infrastruktur. In der Regel dienen Netzwerk-Komponenten zur Kopplung der Netzwerk-Stationen. Man spricht deshalb auch von Kopplungselementen.

Passive Netzwerk-Komponenten

- Patchkabel und Installationskabel
- Anschlussdose
- Steckverbinder
- Patchfeld / Patchpanel
- Netzwerk-Schrank / Patch-Schrank

Hinweis: Zu den passiven Netzwerk-Komponenten zählen die Bestandteile der Verkabelung. Diese ist im OSI-Schichtenmodell nicht definiert.

Aktive Netzwerk-Komponenten

In kleinen privaten Netzwerken, haben Netzwerk-Komponenten noch klare Bezeichnung, wie Switch oder Router. In großen Unternehmensnetzwerken ist die Benennung der Kopplungselemente nicht immer eindeutig.

- Netzwerkkarte
- Repeater
- Hub
- Bridge
- Switch
- Router
- Gateway
- Server

Netzwerkkarte

Eine Netzwerkkarte wird auch als Netzwerkadapter bezeichnet. Die englische Bezeichnung ist Network Interface Card (NIC). Eine Netzwerkkarte ermöglicht es, auf ein Netzwerk zuzugreifen und arbeitet auf der Bitübertragungsschicht (Schicht 1) und der Datensicherungsschicht (Schicht 2) des OSI-Schichtenmodells. Jede Netzwerkkarte hat eine Hardware-Adresse (Format: XX-XX-XX-XX-XX-XX), die es auf der Welt nur einmal gibt. Anhand dieser Adresse lässt sich eine Station auf der Bitübertragungsschicht adressieren.

Im Falle von Ethernet-Netzen besteht die **MAC-Adresse aus 48 Bit (sechs Bytes)**. Die Adressen werden in der Regel **hexadezimal** geschrieben. Üblich ist dabei eine **byteweise Schreibweise**,

wobei die einzelnen Bytes durch Bindestriche oder Doppelpunkte voneinander getrennt werden, z. B. 00-80-41-ae-fd-7e oder 00:80:41:ae:fd:7e. Seltener zu finden sind Angaben wie 008041aefd7e oder 0080.41ae.fd7.

In den **ersten 24 Bits** (Bit 3 bis 24) wird eine von der IEEE vergebene **Herstellerkennung** (auch OUI – Organizationally Unique Identifier genannt) beschrieben, die weitgehend in einer Datenbank einsehbar sind[6]. Die **verbleibenden 24 Bits** (Bit 25 bis 48) werden **vom jeweiligen Hersteller** für jede Schnittstelle individuell **festgelegt**.

Repeater

Ein Repeater ist ein Kopplungselement, um die Übertragungsstrecke innerhalb von Netzwerken, zum Beispiel Ethernet, zu verlängern. Ein Repeater empfängt ein Signal und bereitet es neu auf. Danach sendet er es weiter. Auf diese Weise verlängert der Repeater die Übertragungsstrecke und räumliche Ausdehnung des Netzwerks. Im einfachsten Fall hat ein Repeater zwei Ports, die wechselseitig als Ein- und Ausgang funktionieren (bidirektional).

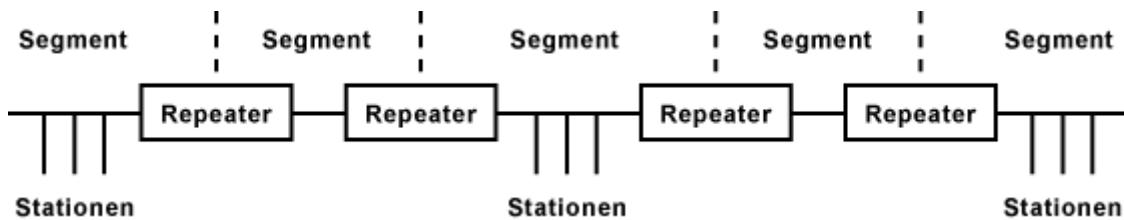
Repeater versteht man in der Regel als Verstärker von Übertragungsstrecken. Die weitere Beschreibung bezieht sich auf Repeater in kabelgebundenen Netzwerken, speziell in Ethernet-Netzwerken.

Ein Repeater arbeitet auf der Schicht 1, der Bitübertragungsschicht des OSI-Schichtenmodells. Der Repeater übernimmt keinerlei regulierende Funktion in einem Netzwerk. Er kann nur Signale empfangen und weiterleiten. Für angeschlossene Geräte ist nicht erkennbar, ob sie an einem Repeater angeschlossen sind. Er verhält sich völlig transparent.

Ein Repeater erweitert somit eine Kollisionsdomäne!!

Ein Repeater mit mehreren Ports wird auch als Hub (Multiport-Repeater) bezeichnet. Er kann mehrere Netzwerk-Segmente miteinander verbinden.

Die Repeater-Regel (5-4-3)



Um ein großes Netzwerk mit einer möglichst großen Reichweite aufzubauen, können mehrere Repeater hintereinandergeschaltet werden. Allerdings, nicht in beliebiger Anzahl. Der Grund liegt im Laufzeitverhalten und der Phasenverschiebung zwischen den Signalen an den Enden des Netzwerks. Deshalb gilt folgende Repeater-Regel:

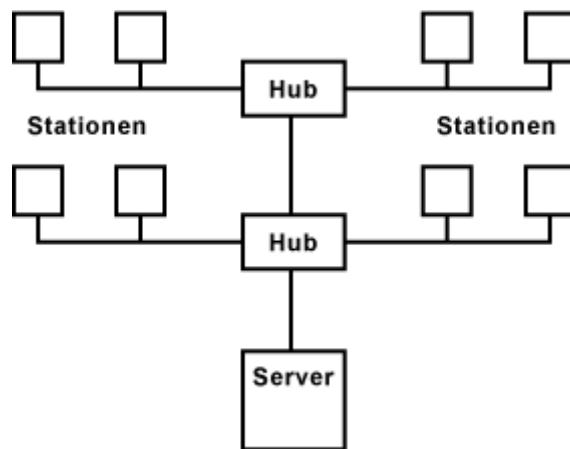
Es dürfen nicht mehr als fünf (5) Kabelsegmente verbunden werden. Dafür werden vier (4) Repeater eingesetzt. An nur drei (3) Segmenten dürfen Endstationen angeschlossen werden.

Diese **Repeater-Regel hat nur in den Ethernet-Netzwerken 10Base2 und 10BASE5** eine

Bedeutung. In Netzwerken, die mit Switches und Router aufgebaut sind, hat diese Repeater-Regel keine Bedeutung. Um die Nachteile von Repeatern in Ethernet-Netzwerken zu umgehen, werden generell Switches zur Kopplung der Hosts eingesetzt. In großen Netzwerken, insbesondere über unterschiedliche Übertragungssysteme hinweg, werden zusätzlich Router eingesetzt.

Hub

Ein Hub ist ein Kopplungselement, das mehrere Hosts in einem Netzwerk miteinander verbindet. In einem Ethernet-Netzwerk, das auf der Stern-Topologie basiert dient ein Hub als Verteiler für die Datenpakete. Hubs arbeiten auf der Bitübertragungsschicht (Schicht 1) des OSI-Schichtenmodells und sind damit auf die **reine Verteilfunktion** beschränkt. **Hubs erweitern somit die Kollisionsdomäne**. Ein Hub nimmt **ein Datenpaket entgegen und sendet es an alle anderen Ports** weiter. Das bedeutet, er **broadcastet**. Dadurch sind nicht nur **alle Ports**, sondern **auch alle Hosts belegt**. Sie bekommen alle Datenpakete zugeschickt, auch wenn sie nicht die Empfänger sind. Für die Hosts bedeutet das auch, dass sie nur dann senden können, wenn der Hub gerade keine Datenpakete sendet. Sonst kommt es zu Kollisionen.



Wenn die Anzahl der Anschlüsse an einem Hub für die Anzahl der Hosts nicht ausreicht, dann benötigt man noch einen zweiten Hub. Zwei Hubs werden über einen Uplink-Port eines der beiden Hubs oder mit einem Crossover-Kabel (Sende- und Empfangsleitungen sind gekreuzt) verbunden. Es gibt auch spezielle „stackable“ Hubs, die sich herstellerspezifisch mit Buskabeln kaskadieren lassen. Durch die Verbindung mehrerer Hubs lässt sich die Anzahl der möglichen Hosts im Netzwerk erhöhen. Allerdings ist die Anzahl der anschließbaren Hosts begrenzt. Hier gilt die Repeater-Regel.

Nachteile

- ineffizient
- unsicher (Jeder bekommt jede Nachricht)

Vorteile

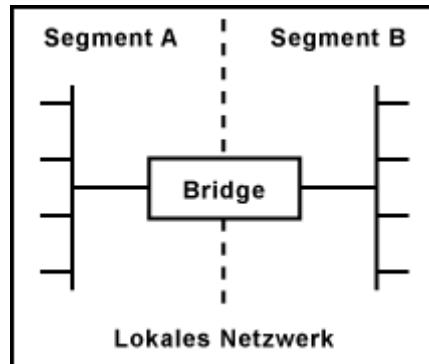
- zentraler Verteiler

Bridge

Eine Bridge ist ein Kopplungselement, das ein lokales Netzwerk in zwei Segmente aufteilt. Dabei

werden die Nachteile von Ethernet, die besonders bei großen Netzwerken auftreten ausgeglichen. Als Kopplungselement ist die Bridge eher untypisch. Man vermeidet die Einschränkungen durch Ethernet heute eher durch Switches.

Eine Bridge teilt eine Kollisionsdomäne!!

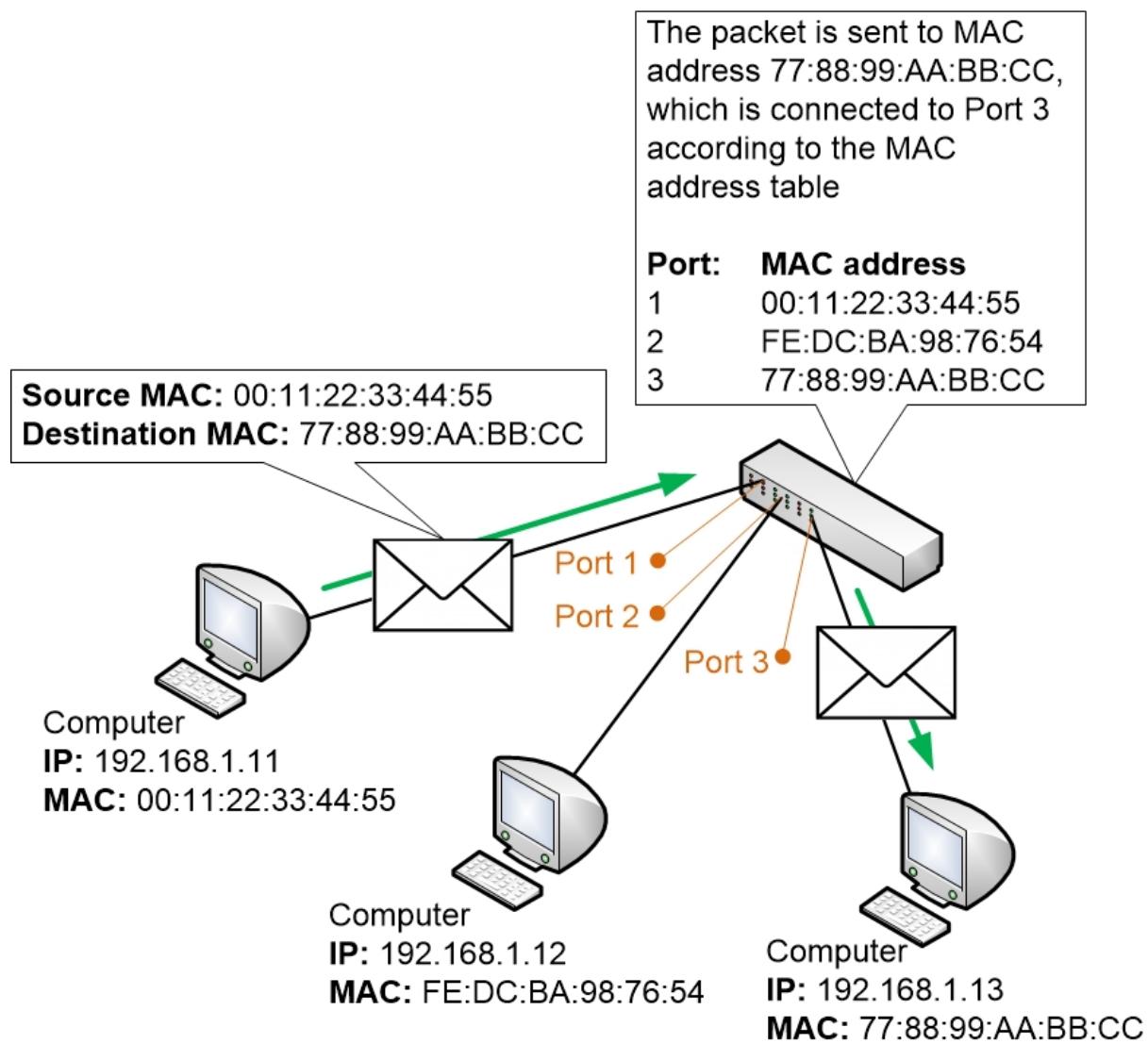


Switch

Ein Switch ist ein Kopplungselement, das mehrere Stationen in einem Netzwerk miteinander verbindet. In einem Ethernet-Netzwerk, das auf der Stern-Topologie basiert dient ein Switch als Verteiler für die Datenübertragung.

Die Funktion ist ähnlich einem Hub, mit dem Unterschied, dass ein Switch direkte Verbindungen zwischen den angeschlossenen Geräten schalten kann, sofern ihm die Ports der Datenpaket-Empfänger bekannt sind. Somit kommunizieren wirklich nur jene miteinander, die auch miteinander reduzieren wollen. Wenn nicht, dann broadcastet der Switch die Datenpakete an alle Ports. Wenn die Antwortpakete von den Empfängern zurück kommen, dann merkt sich der Switch die MAC-Adressen der Datenpakete und den dazugehörigen Port und sendet die Datenpakete dann nur noch dorthin. Er baut also eine sogenannte MAC-Adressen-Tabelle auf:

Beispiel:



Während ein Hub die Bandbreite des Netzwerks limitiert, steht der Verbindung zwischen zwei Hosts, die volle Bandbreite der Ende-zu-Ende-Netzwerk-Verbindung zur Verfügung.

Ein Switch arbeitet auf der Sicherungsschicht (Schicht 2) des OSI-Modells und arbeitet ähnlich wie eine Bridge. Daher haben sich bei den Herstellern auch solche Begriffe durchgesetzt, wie z. B. Bridging Switch oder Switching Bridge. Die verwendet man heute allerdings nicht mehr.

Switches unterscheidet man hinsichtlich ihrer Leistungsfähigkeit mit folgenden Eigenschaften:

- Anzahl der speicherbaren MAC-Adressen für die Quell- und Zielports
- Verfahren, wann ein empfangenes Datenpaket weitervermittelt wird (Switching-Verfahren)
- Latenz (Verzögerungszeit) der vermittelten Datenpakete

Ein Switch ist im Prinzip nichts anderes als ein intelligenter Hub, der sich merkt, über welchen Port welcher Host erreichbar ist. Teure Switches können zusätzlich auf der Schicht 3, der Vermittlungsschicht, des OSI-Schichtenmodells arbeiten (Layer-3-Switch oder Schicht-3-Switch). Sie sind in der Lage, die Datenpakete anhand der IP-Adresse an die Ziel-Ports weiterzuleiten. Im Gegensatz zu normalen Switches lassen sich auch ohne Router logische Abgrenzungen erreichen.

Switching-Verfahren

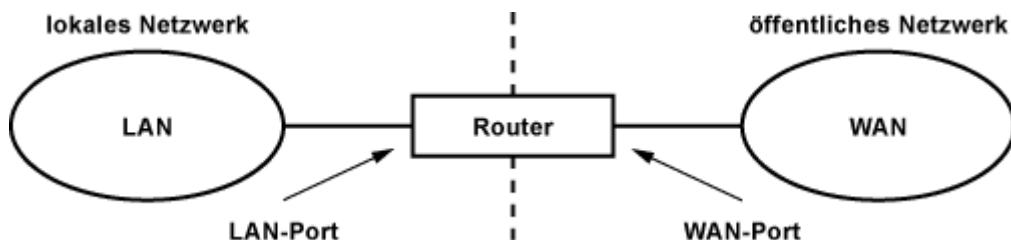
Switching-Verfahren	Beschreibung	Vorteile	Nachteile
Cut-Through	Der Switch leitet das Datenpaket sofort weiter, wenn er die Adresse des Ziels erhalten hat.	Die Latenz, die Verzögerungszeit, zwischen Empfangen und Weiterleiten ist äußerst gering.	Fehlerhafte Datenpakete werden nicht erkannt und trotzdem an den Empfänger weitergeleitet.
Store-and-Forward	Der Switch nimmt das gesamte Datenpaket in Empfang und speichert es in einem Puffer. Dort wird dann das Paket mit verschiedenen Filtern geprüft und bearbeitet. Erst danach wird das Paket an den Ziel-Port weitergeleitet.	Fehlerhafte Datenpakete können so im voraus aussortiert werden.	Die Speicherung und Prüfung der Datenpakete verursacht eine Verzögerung, abhängig von der Größe des Datenpaketes.
Fragment-Free	Der Switch empfängt die ersten 64 Byte des Daten-Paketes. Ist dieser Teil fehlerlos werden die Daten weitergeleitet. Die meisten Fehler und Kollisionen treten während den ersten 64 Byte auf. Dieses Verfahren wird trotz seiner effektiven Arbeitsweise selten genutzt.	sehr effizient	selten genutzt

Router

Ein Router verbindet mehrere Netzwerke mit unterschiedlichen Protokollen und Architekturen. Ein Router befindet sich häufig an den Außengrenzen eines Netzwerks, um es mit dem Internet oder einem anderen, größeren Netzwerk zu verbinden. Über die Routing-Tabelle entscheidet ein Router, welchen Weg ein Datenpaket nimmt. Es handelt sich dabei um ein dynamisches Verfahren, das Ausfälle und Engpässe ohne den Eingriff eines Administrators berücksichtigen kann. Ein Router hat mindestens zwei Netzwerkanschlüssen. Er arbeitet auf der Vermittlungsschicht (Schicht 3) des OSI-Schichtenmodells.

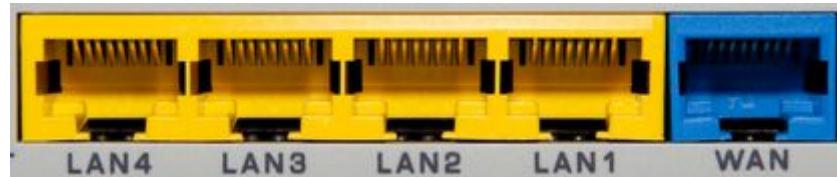
Die Aufgabe eines Routers ist ein komplexer Vorgang, der sich in 4 Schritte einteilen lässt:

- Ermittlung der verfügbaren Routen
- Auswahl der geeignetsten Route unter Berücksichtigung verschiedener Kriterien
- Herstellen einer physikalischen Verbindung zu anderen Netzwerken
- Anpassen der Datenpakete an die Übertragungstechnik (Fragmentierung)



Ein Router hat in der Regel zwei Anschlüsse. Einen für die LAN-Seite und einen für die WAN-Seite.

Häufig sind die Ports mit der Bezeichnung LAN und WAN gekennzeichnet. Manchmal gibt es Port-Beschriftungen, bei denen nicht immer eindeutig ist, um was es sich handelt. Mit LAN ist immer das lokale Netzwerk mit privaten IP-Adressen gemeint, während die WAN-Seite das öffentliche Netzwerk kennzeichnet.



Zusammenfassung



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

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:2:2_05

Last update: **2021/12/14 16:21**

IP-Adressen

Damit die Wegewahl in Netzwerken und im Internet zur Übermittlung von Datenpaketen vom Senden zum Empfänger funktioniert, wird jedem Knotenpunkt im **Internet eine weltweit eindeutige (und einmalige) Adresse (IP-Adresse)** zugeordnet.

Für Rechner die ohne Router direkt mit dem Internet verbunden sind, heißt das, dass deren IP-Adressen weltweit eindeutig sind.

Standardmäßig haben diese IP-Adressen eine Länge von **32 Bit (4 x 8Bit)**.

Diese werden aus Gründen der **Übersichtlichkeit in 4 Zahlen zu je 1 Byte** aufgeteilt. Meist werden die einzelnen Bytes durch einen Punkt getrennt und dezimal dargestellt.

Öffentliche IP-Adressen

Wenn man von öffentlichen Adressen spricht, meint man die IP-Adressen, die im Internet erreichbar sind. Die Zuweisung einer öffentlichen IP-Adresse erfolgt in der Regel durch einen Provider (z.B. A1, Telekom, ...) welche diese wiederum in Europa durch die Organisation [RIPE-NCC](#), die wiederum Adressen von der [IANA](#) zugewiesen bekommt.



Private IP-Adressen

Private IP-Adressen (abgekürzt Private IP) sind IP-Adressen, die von der [IANA](#) nicht im Internet vergeben sind. Sie wurden für die **private Nutzung aus dem öffentlichen Adressraum ausgespart**, damit sie ohne administrativen Mehraufwand (Registrierung der IP-Adressen) in lokalen Netzwerken genutzt werden können. Als die **IP-Adressen des Internet Protokolls v4 knapp** wurden und dadurch eine **bewusste Einsparung öffentlicher IP-Adressen** notwendig wurde, war es umso wichtiger, private IP-Adressen in lokalen Netzwerken zur Verfügung zu haben, die **beliebig oft bzw. in beliebigen Netzwerken genutzt** werden können.

Netzklassse: Anzahl Netze (ohne Subnetting)	Netzadressbereich	Subnetzmaske	CIDR-Notation	Anzahl Adressen
Klasse A: 1 privates Netz mit 16.777.216 Adressen	10.0.0.0 bis 10.255.255.255	255.0.0.0	10.0.0.0/8	$2^{24} = 16.777.216$
Klasse B: 16 private Netze mit jeweils 65.536 Adressen	172.16.0.0 bis 172.31.255.255	255.240.0.0	172.16.0.0/12	$2^{20} = 1.048.576$
Klasse C: 256 private Netze mit jeweils 256 Adressen	192.168.0.0 bis 192.168.255.255	255.255.0.0	192.168.0.0/16	$2^{16} = 65.536$

Loopback Adresse

Die Class-A-Netzwerkadresse 127 ist weltweit reserviert für das sogenannte local loopback; sie dient zu Testzwecken der Netzwerkschnittstelle des eigenen Rechners. Die IP-Adresse **127.0.0.1** ist standardmäßig dem Loopback-Interface jedes Rechners zugeordnet.

Alle an diese Adresse geschickten Datenpakete werden nicht nach außen ins Netzwerk gesendet, sondern an der Netzwerkschnittstelle reflektiert. Die Datenpakete erscheinen, als kämen sie aus einem angeschlossenem Netzwerk.

Vergleich IP-Adresse vs. Telefonnummer

Ähnlich wie bei einer Telefonnummer setzt sich eine IP-Adresse aus mehreren Segmenten zusammen. Eine Telefonnummer besteht aus einer Vorwahl und einer Teilnehmernummer. Führen Sie ein Ortsgespräch, so muss die Vorwahl nicht angegeben werden. Ähnlich ist es bei IP-Adressen.

Diese bestehen auch aus zwei Teilen, wobei die ID für Identifikation steht:

- **Netzwerk-ID** im vorderen linken Teil entspricht der Vorwahl und gibt das entsprechende IP-Subnetz an.
- **Host-ID** im hinteren rechten Teil kennzeichnet eine einzelne Netzwerkkarte und entspricht dem Teilnehmernummer im Ortsnetz.

Entsprechend können Rechner im selben Subnetz direkt miteinander kommunizieren. Dagegen erfordert Kommunikation zwischen Subnetzen eine Vermittlungsstelle, einen Router (Standardgateway), wo alle nicht im selben Netz adressierten Pakete hingeschickt werden.

Um zu erkennen, wo die Netzwerk-ID endet und die Host-ID beginnt, muss zusätzlich zur IP-Adresse zwingend eine sogenannte Subnetzmaske mit angegeben werden.

Subnetzmaske

Eine Subnetzmaske ist ein Bitmuster, das (von links nach rechts) Teile der IP-Adresse „maskiert“, um den Übergang zwischen Netz-ID und Host-ID zu kennzeichnen. Binär betrachtet besteht eine Subnetzmaske aus einer Folge von Einsen, die ab einer bestimmten Stelle umschlägt in eine Folge von Nullen. Dieser Umschlagpunkt gibt an, wie viele Bits zur Netzwerk-ID (Einsen) und zur Host-ID (Nullen) gehören.

Ein Auszug aus den möglichen Subnetzmasken für ein Klasse C-Netz

Maske	Maske (kurze Schreibweise)	Anzahl Hosts pro Netz	Netze	Beispiel Klasse C-Netz (192.168.1.0)
255.255.255.0	/24	256	1	192.168.1.0 - 192.168.1.255
255.255.255.128	/25	128	2	192.168.1.0 - 192.168.1.127 192.168.1.128 - 192.168.1.255

Maske	Maske (kurze Schreibweise)	Anzahl Hosts pro Netz	Netze	Beispiel Klasse C-Netz (192.168.1.0)
255.255.255.192	/26	64	4	192.168.1.0 - 192.168.1.63 192.168.1.64 - 192.168.1.127 192.168.1.128 - 192.168.1.191 192.168.1.192 - 192.168.1.255
255.255.255.224	/27	32	8	192.168.1.0 - 192.168.1.31 192.168.1.32 - 192.168.1.63 ... 192.168.192 - 192.168.1.223 192.168.1.224 - 192.168.1.255
255.255.255.240	/28	16	16	192.168.1.0 - 192.168.1.15 192.168.1.16 - 192.168.1.31 ... 192.168.1.224 - 192.168.1.239 192.168.1.240 - 192.168.1.255
255.255.255.248	/29	8	32	192.168.1.0 - 192.168.1.7 192.168.1.8 - 192.168.1.15 ... 192.168.1.240 - 192.168.1.247 192.168.1.248 - 192.168.1.255
255.255.255.252	/30	4	64	192.168.1.0 - 192.168.1.3 192.168.1.4 - 192.168.1.7 ... 192.168.1.248 - 192.168.1.251 192.168.1.252 - 192.168.1.255
255.255.255.254	/31	2	128	192.168.1.0 - 192.168.1.1 192.168.1.2 - 192.168.1.3 ... 192.168.1.252 - 192.168.1.253 192.168.1.254 - 192.168.1.255
255.255.255.255	/32	1	256	192.168.1.0 192.168.1.1 ... 192.168.1.254 192.168.1.255



Erklärung

Subnetting

Die Aufteilung eines zusammenhängenden Adressraums von IP-Adressen in mehrere kleinere Adressräume nennt man Subnetting. Ein Subnet, Subnetz bzw. Teilnetz ist ein physikalisches Segment eines Netzwerks, in dem IP-Adressen mit der gleichen Netzwerkadresse benutzt werden. Diese Teilnetze können über Routern miteinander verbunden werden und bilden dann ein großes zusammenhängendes Netzwerk.

Beispiel 1:

IP-Adresse: 192.168.128.17

Dezimale Darstellung:	192 .	168 .	128 .	17
Bit-Darstellung:	1100 0000	1010 1000	1000 0000	0001 0001
Hex-Darstellung:	C0	A8	80	11

Subnetzmaske: 255.255.255.0

Dezimale Darstellung:	255 .	255 .	255 .	0
Bit-Darstellung:	1111 1111	1111 1111	1111 1111	0000 0000
Hex-Darstellung:	FF	FF	FF	00

Verknüpft man nun die binäre IP-Adresse mit der Subnetzmaske mit einem Logischen AND, so bekommt man die Netz-ID (=Netzadresse).

<----- NETZ-ID -----><- HOST-ID ->				
1100 0000	1010 1000	1000 0000	0001 0001	(IP-Adresse)
AND 1111 1111	1111 1111	1111 1111	0000 0000	(Subnetzmaske)
<hr/>				
1100 0000	1010 1000	1000 0000	0000 0000	(Netz-ID)
Netz-ID in Dezimaldarstellung:				
192.168.128				

Verknüpft man nun die binäre IP-Adresse mit der negierten Subnetzmaske mit einem Logischen AND, so bekommt man die Host-ID.

<----- NETZ-ID -----><- HOST-ID ->				
1100 0000	1010 1000	1000 0000	0001 0001	(IP-Adresse)
AND 0000 0000	0000 0000	0000 0000	1111 1111	(negierte Subnetzmaske)
<hr/>				
0000 0000	0000 0000	0000 0000	0001 0001	(Host-ID)
Host-ID in Dezimaldarstellung:				
17				

Das heißt im Netzwerk 192.168.128.0 stehen theoretisch 256 (0-255) Adressen zur Adressierung von Netzwerkgeräten zur Verfügung. Praktisch sind es aber nur 254 Adressen, da zwei Adressen

- die **Netzadresse** (= 1. Adresse im Netz -> 192.168.128.0)
- die **Broadcastadresse** (= letzte Adresse im Netz -> 192.168.128.255)

reserviert sind.

Beispiel 2:

IP-Adresse 130.94.122.195/27

Berechnung	Dezimal	Binär	
IP Adresse	130.094.122.195	10000010 01011110 01111010 11000011	
ip-adresse			
Netzmaske	255.255.255.224	11111111 11111111 11111111 11100000	AND
netzmaske			

Netzwerkteil	130.094.122.192	10000010 01011110 01111010 11000000	=
netzwerkanteil			

IP Adresse	130.094.122.195	10000010 01011110 01111010 11000011	
ip-adresse			
Netzmaske	255.255.255.224	00000000 00000000 00000000 00011111	AND(NOT
netzmaske)			

Geräteteil	3	00000000 00000000 00000000 00000011	=
geräteteil			

Bei einer Netzmaske mit 27 gesetzten Bits ergibt sich ein Netzwerkteil von 130.94.122.192. Es verbleiben 5 Bits und damit $2^5=32$ Adressen für den Geräteteil. Hiervon werden noch je 1 Adresse für das Netzwerk selbst und für den Broadcast benötigt, sodass 30 Adressen für Geräte zur Verfügung stehen.

Beispiel 3:

IP-Adresse 130.94.122.117/28

Berechnung	Dezimal	Binär	
IP Adresse	130.094.122.117	10000010 01011110 01111010 01110101	
ip-adresse			
Netzmaske	255.255.255.240	11111111 11111111 11111111 11110000	AND
netzmaske			

Netzwerkteil	130.094.122.112	10000010 01011110 01111010 01110000	=
netzwerkanteil			

IP Adresse	130.094.122.117	10000010 01011110 01111010 01110101
ip-adresse		
Netzmaske	255.255.255.240	00000000 00000000 00000000 00001111 AND (NOT netzmaske)
<hr/>		
<hr/>		
Geräteteil	0. 0. 0. 5	00000000 00000000 00000000 00000101 = geräteteil

Bei einer Netzmaske mit 28 gesetzten Bits ergibt sich ein Netzwerkteil von 130.94.122.112. Es verbleiben 4 Bits und damit $2^4=16$ Adressen für den Geräteteil. Hiervon werden noch je 1 Adresse für das Netzwerk selbst und für den Broadcast benötigt, sodass 14 Adressen für Geräte zur Verfügung stehen.

- [Zusammenfassung zu Subnetzmasken](#)

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

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:2:2_06 

Last update: **2021/12/14 16:21**

Übungen IP-Adressierung

Bsp 1

Befinden sich 192.168.0.93/27 und 192.168.0.97/27 im gleichen Netzwerk-Segment?

[Lösung](#)

- 1) NW-ID: 192.168.0.64
 - 2) NW-ID: 192.168.0.96
- => Nein

Bsp 2

Wie lauten die Netzwerkadressen des ersten/letzten Hosts des Netzwerkes 192.168.0.96/27 und die Broadcastadresse?

[Lösung](#)

- 1. Host: 192.168.0.97
- letzter Host: 192.168.0.126
- BC: 192.168.0.127

Bsp 3

Das Netz 195.1.31.0 /24 soll in weitere 30 Subnetze aufgeteilt werden. Bestimme die Subnetzmaske, und die Anzahl Adressen pro Subnet!

[Lösung](#)

- Subnetzmaske: 255.255.255.248
- Adressen: 6

Bsp 4

Das Netz 10.0.0.0 /16 soll in mindestens 200 Subnetze weiter unterteilt werden. Bestimme die Subnetzmaske, und die Anzahl Adressen pro Subnet!

[Lösung](#)

- Subnetzmaske: 255.255.255.0

Adressen: 256

Bsp 5

Das Netz 192.168.1.0 /24 soll in Subnetze mit je 18 Host-Adressen aufgeteilt werden. Bestimme die Subnetzmaske, und die Anzahl Subnetze!

[Lösung](#)

Subnetzmaske: 255.255.255.224
Subnetze: 8

Bsp 6

Das Netz 192.168.100.0 /24 soll in Subnetze mit je 5 Host-Adressen aufgeteilt werden. Bestimmen Sie die Subnetzmaske, und die Anzahl Subnetze!

[Lösung](#)

Subnetzmaske: 255.255.255.248
Subnetze: 32

Bsp 7

Bestimme Netz- und Broadcastadresse des Subnets, in dem die Adresse 195.1.31.135 mit Netzmaske 255.255.255.128 liegt!

[Lösung](#)

NW-ID: 195.1.31.128
BC: 195.1.31.255

Bsp 8

Bestimme Netz- und Broadcastadresse des Subnets in dem die Adresse 195.1.31.135 mit Netzmaske 255.255.255.192 liegt!

[Lösung](#)

NW-ID: 195.1.31.128/26
BC: 195.1.31.191

Bsp 9

Bestimme Netz- und Broadcastadresse des Subnets in dem die Adresse 15.3.128.222 mit Netzmaske 255.255.255.240 liegt!

[Lösung](#)

```
NW-ID: 15.3.128.208  
Broadcast: 15.3.128.223
```

Bsp 10

Gib den Bereich der Rechneradressen an, den das Teilnetz Nummer drei (132.45.96.0/19) hat.

[Lösung](#)

```
132.45.96.1  
bis  
132.45.127.254
```

Bsp 11

Berechne Netz-, Broadcast und Hostadressen des 0.,1.,15. und 31 Subnetz des Netzes 192.168.1.0/29

[Lösung](#)

```
0. Netz:  
* NW-ID: 192.168.1.0  
* BC: 192.168.1.1  
* Hosts: 192.168.1.1-6
```

```
1. Netz:  
* NW-ID: 192.168.1.8  
* BC: 192.168.1.15  
* Hosts: 192.168.1.9-14
```

```
15. Netz:  
* NW-ID: 192.168.1.120  
* BC: 192.168.1.127  
* Hosts: 192.168.1.121-126
```

```
31. Netz:  
* NW-ID: 192.168.1.248  
* BC: 192.168.1.255  
* Hosts: 192.168.1.249-254
```

Bsp 12

Vervollständige folgende Tabelle:

IP-Adresse	10.1.1.1	Netzwerkadresse des Subnet	...
Netmask	255.255.0.0	Broadcastadresse des Subnet	...
Anzahl Host pro Subnet		Anzahl Subnets in diesem Netz	

[Lösung](#)

IP-Adresse	10.1.1.1	Netzwerkadresse des Subnet	10.1.0.0
Netmask	255.255.0.0	Broadcastadresse des Subnet	10.1.255.255
Anzahl Host pro Subnet	65536	Anzahl Subnets in diesem Netz	256

Bsp 13

Vervollständige folgende Tabelle:

IP-Adresse	10.1.1.1/24	Netzwerkadresse des Subnet	...
Netmask	255.255.255.0	Broadcastadresse des Subnet	...
Anzahl Host pro Subnet		Anzahl Subnets in diesem Netz	

[Lösung](#)

IP-Adresse	10.1.1.1/24	Netzwerkadresse des Subnet	10.1.1.0
Netmask	255.255.255.0	Broadcastadresse des Subnet	10.1.1.255
Anzahl Host pro Subnet	254	Anzahl Subnets in diesem Netz	65536

From:

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

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:2:2_06:2_06_1



Last update: **2021/12/14 16:21**

IP-Routing

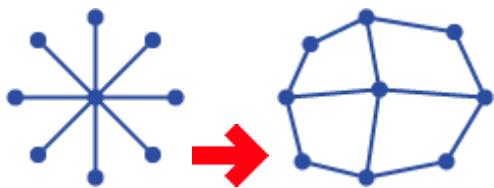
Das **Internet Protocol (IP)** ist ein **routingfähiges Protokoll** und sorgt dafür, dass **Datenpakete über Netzgrenzen** hinweg einen Weg zu anderen Hosts finden. Es kann die Daten über jede Art von physikalischer Verbindung oder Übertragungssystem vermitteln. Der hohen Flexibilität steht ein hohes Maß an Komplexität bei der Wegfindung vom Sender zum Empfänger gegenüber. Der **Vorgang der Wegfindung wird Routing** genannt

Wozu Routing?

Das grundlegende Verbindungselement in einem Ethernet-Netzwerk ist der Hub oder Switch. Daran sind alle Netzwerk-Teilnehmer angeschlossen. Wenn ein Host Datenpakete verschickt, dann werden die Pakete im Hub an alle Stationen verschickt und von diesen angenommen. Jedoch verarbeitet nur der adressierte Host die Pakete weiter. Das bedeutet aber auch, dass sich alle Hosts die Gesamtbandbreite dieses Hubs teilen müssen.

Um die Nachteile von Ethernet in Verbindung mit CSMA/CD auszuschließen, wählt man als Kopplungselement einen Switch und nutzt Fast Ethernet (kein CSMA/CD mehr). Der Switch merkt sich die Hardware-Adressen (MAC-Adressen) der Stationen und leitet die Ethernet-Pakete nur an den Port, hinter dem sich die Station befindet. Ist einem Switch die Hardware-Adresse nicht bekannt, leitet er das Datenpaket an alle seine Ports weiter (Broadcast) und funktioniert in diesem Augenblick wie ein Hub. Neben der begrenzten Speichergröße des Switches machen sich viele unbekannte Hardware-Adressen negativ auf die Performance eines Netzwerks bemerkbar.

Daher eignet sich zum **Verbinden großer Netzwerke** weder ein Hub noch ein Switch. Aus diesem Grund wird ein Netzwerk **durch Router** und IP-Adressen in **logische Segmente bzw. Subnetze** unterteilt. Die Adressierung durch das Internet Protocol ist so konzipiert, dass der Netzwerkverkehr innerhalb der Subnetze bleibt und erst dann das Netzwerk verlässt, wenn das Ziel in einem anderen Netzwerk liegt.



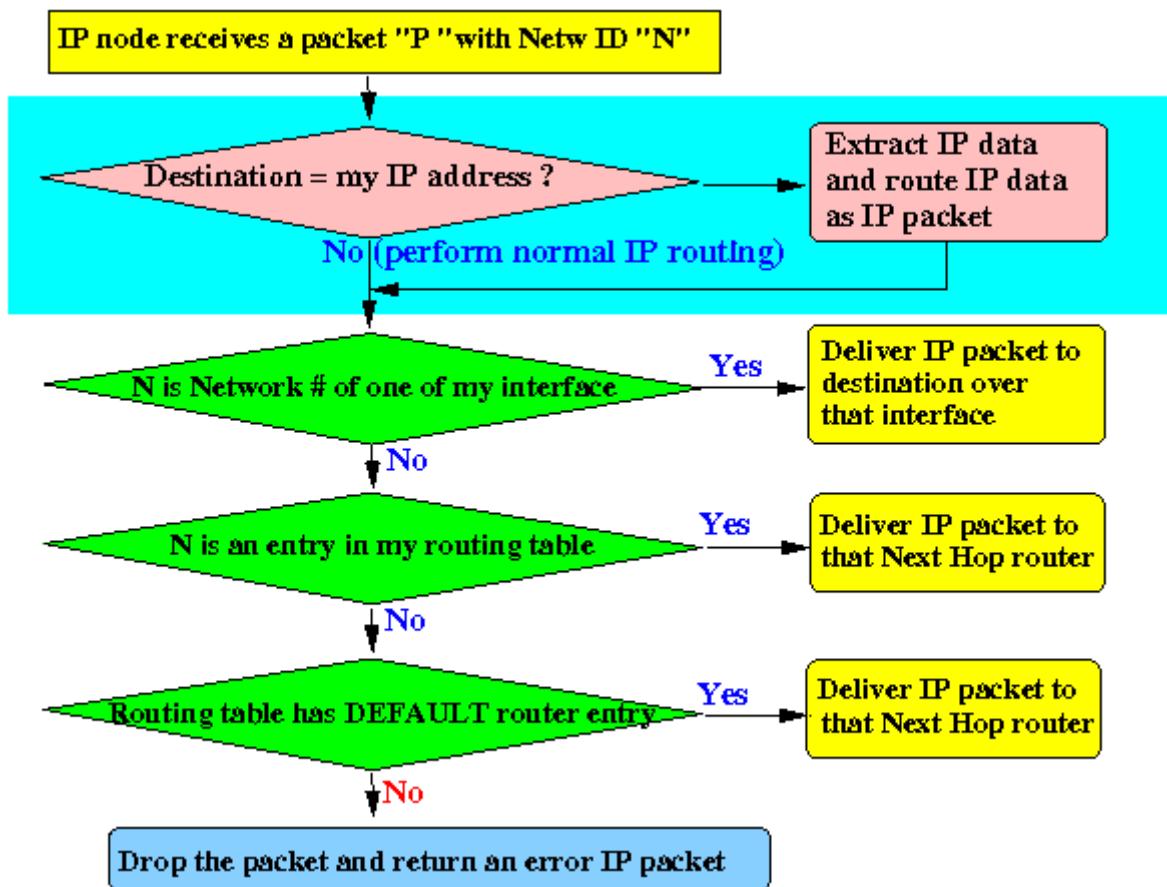
Insbesondere folgende **Probleme** in einem Ethernet-Netzwerk machen **IP-Routing notwendig**:

- **Vermeidung von Kollisionen und Broadcasts durch Begrenzung der Kollisions- und Broadcastdomäne**
- **Routing über unterschiedliche Netzarchitekturen und Übertragungssysteme**
- **Paket-Filter und Firewall**
- **Routing über Backup-Verbindungen bei Netzausfall**

IP-Routing-Algorithmus

Der IP-Routing-Algorithmus gilt nicht nur für IP-Router, sondern für alle Host, die IP-Datenpakete

empfangen können. Die empfangenen Datenpakete durchlaufen diesen Algorithmus bis das Datenpaket zugeordnet oder weitergeleitet werden kann.



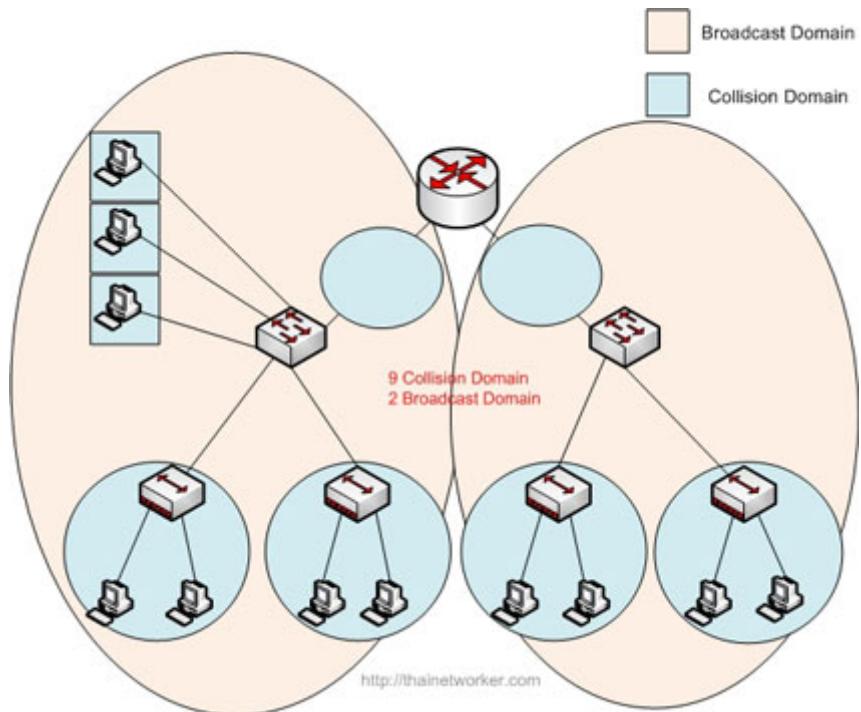
- Prüfe, ob das Datenpaket mir gehört?
 - Wenn ja, dann hat das Datenpaket sein Ziel erreicht und kann verarbeitet werden.
 - Wenn nein, prüfe ob das Datenpaket in mein Subnetz gehört?
 - Wenn ja, schick es in das eigene Subnetz weiter!
 - Wenn nein, prüfe ob die Route zum Empfänger bekannt ist?
 - Wenn ja, schicke es über die bekannte Route zum nächsten Router!
 - Wenn nein, prüfe ob es ein Standard-Gateway gibt?
 - Wenn ja, schick das Paket zum Standard-Gateway!
 - Wenn nein, schreibe eine Fehlermeldung und verwirf das Datenpaket!

Broadcastdomäne

Eine Broadcast-Domäne ist ein logischer Verbund von Netzwerkgeräten in einem lokalen Netzwerk, der sich dadurch auszeichnet, dass ein **Broadcast alle Domänenteilnehmer erreicht**.

Ein lokales Netzwerk auf der 2. Schicht des OSI-Modells (Sicherungsschicht) besteht durch seine Hubs, Switches und/oder Bridges aus einer Broadcast-Domäne. Erst durch die Unterteilung in VLANs oder durch den Einsatz von Routern, die auf Schicht 3 arbeiten, wird die Broadcast-Domäne aufgeteilt.

Eine Broadcast-Domäne besteht aus einer oder mehreren Kollisionsdomänen.



From:

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

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:2:2_07



Last update: **2021/12/14 16:21**

Netzwerbefehle

Windows-Kommandos

Wichtige Netzwerbefehle unter Windows XP

- ipconfig
 - /all ... detaillierte Informationen über Netzwerk-Konfiguration
 - /renew ... erneuert IP-Adressen
 - /release ... gibt IP-Adressen frei
- ping (sendet Datenpakete zu Rechner)
 - ping IP-Adresse
 - ping hostname
 - -n Anzahl ... Anzahl der Pakete
- tracert (Route zu Rechner)
- nslookup
 - nslookup (DNS-Abfragen)
 - nslookup IP-Adresse
 - nslookup Domain-Name

Linux-Kommandos

- ifconfig

Zeigt Informationen zu Netzwerk-Interfaces

- ping

```
ping -c 4 www.example.com
```

Pingt 4 mal www.example.com

- tracepath

```
tracepath www.example.com
```

Zeigt Hops zum Host an (benötigt eventuell SU-Rechte)

```
sudo tracepath www.example.com
```

- nslookup

```
nslookup www.example.com
```

Überprüft DNS-Eintrag

weitere hilfreiche Befehle:

- nmap (scannet („mappt“) das Netzwerk, führt Portscans aus und findet die Software eines fremden PCs heraus)
- mtr (kombiniert tracert (ohne su-Rechte) und ping, anschauliche Darstellung)

Fragen - Aufgaben - Arbeitsaufträge

Arbeitsaufträge:

1. Gib deiner/m Nachbarn/in die IP-Adresse deines Computers und notiere dir die IP-Adresse seines/ihres Computers.
2. Versuche deinen Nachbarcomputer anzupingen. Wie lange brauchte das Datenpaket zu diesem Rechner?
3. Gib deine IP-Adresse frei und versuche den Nachbarn anzupingen bzw. dich anpingen zu lassen. Erneuere anschließend deine IP-Adresse und überprüfe, wie sie nun lautet.
4. Versuche deinen Rechner (mit deiner IP-Adresse und der Loopback-Adresse) anzupingen.
5. Finde heraus, welche IP-Adresse der Domain-Name mail.bgamstetten.ac.at hat.
6. Finde einen Rechner im Internet, bis zu welchem ein Datenpaket sehr lange dauert und schicke zu diesem Rechner 20 Datenpakete hintereinander.
7. Finde heraus, welcher Domain-Name zu folgender IP-Adresse gehört 131.130.250.250
8. Lasse dir die Route zu www.yahoo.com anzeigen. Wie viele Rechner liegen zwischen dir und dem Webserver von www.yahoo.com ?
9. Versuche mit dem Internetexplorer die Website von www.google.at aufzurufen, indem du die IP-Adresse von Google in der Browserzeile eingibst.
10. Suche im Internet die Homepage von der australischen Regierung. Welche IP-Adresse hat der Rechner, auf dem die Homepage liegt? Wie viele Rechner liegen zwischen dir und diesem Rechner?
11. Finde einen Rechner im Internet, bis zu dem möglichst viele Hops dazwischen sind (Wer findet die meisten?).

From:

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

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:2:2_08



Last update: **2021/12/14 16:22**

Protokolle

Protokolle sind eine **Sammlung von Regeln zur Kommunikation** auf einer bestimmten Schicht des OSI-Schichtenmodells. Die Protokolle einer Schicht sind zu den Protokollen der über- und untergeordneten Schichten weitestgehend transparent, so dass die Verhaltensweise eines Protokolls sich wie bei einer direkten Kommunikation mit dem Gegenstück auf der Gegenseite darstellt.

Die **Übergänge zwischen den Schichten sind Schnittstellen**, die von den Protokollen verstanden werden müssen. Weil manche Protokolle für ganz bestimmte Anwendungen entwickelt wurden, kommt es auch vor, dass sich **Protokolle über mehrere Schichten erstrecken** und mehrere Aufgaben abdecken. Dabei kommt es vor, dass in manchen Verbindungen einzelne Aufgaben in mehreren Schichten und somit mehrfach ausgeführt werden.

Protokoll-Stack

Da sich ein einzelnes Protokoll immer nur um eine Teilaufgabe im Rahmen der Kommunikation kümmert, werden mehrere Protokolle zu Protokollsammlungen oder Protokoffamilien, den sogenannten Protokoll-Stacks, zusammengefasst. Die wichtigsten Einzel-Protokolle werden dann oft stellvertretend als Bezeichnung des gesamten Protokoll-Stapels genutzt.

Kommunikation zwischen Netzwerkkomponenten funktioniert nur dann, wenn sie denselben Protokoll-Stack benutzen oder wenn Geräte eingesetzt werden, die zwischen verschiedenen Stacks vermitteln können.

Portnummern (ab Layer 5)

Um die einzelnen Dienste (Protokolle), die bei einem Rechner über dieselbe IP-Adresse ausgeführt werden, voneinander zu differenzieren, wurden Portnummern eingeführt, um bei einer Anfrage deutlich zu machen, welcher Dienst gemeint ist.

Diese Portnummern, die im TCP- oder UDP Header angegeben werden, sind weltweit eindeutig festgelegt und können z.B. auf der Homepage der [IANA](#) eingesehen werden.

Allgemein lässt sich also sagen, dass die **IP-Adresse** den Rechner, **und** Die **Port-Nummer** den Dienst auf dem jeweiligen Rechner angibt. Diese beiden Informationen zusammen werden als **Socket** bezeichnet.

Wichtige Protokolle

OSI-LAYER	PROTOKOLL (Port)
5-7	DHCP (67+68/UDP)
	DNS (53/UDP)
	HTTP (80/TCP)
	HTTPS (443/TCP)
	FTP (20+21/TCP)
	SSH (22/TCP)
	SMTP (465 und 587 oder 25/TCP)
	POP3 (995 oder 110/TCP)
	IMAP (993 oder 143/TCP)
4	TCP
	UDP
3	IPv4
	IPv6
	NAT
	ICMP
2	Ethernet 802.3 , Wireless 802.x, MAC, ISDN,...
1	

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

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:2:2_09



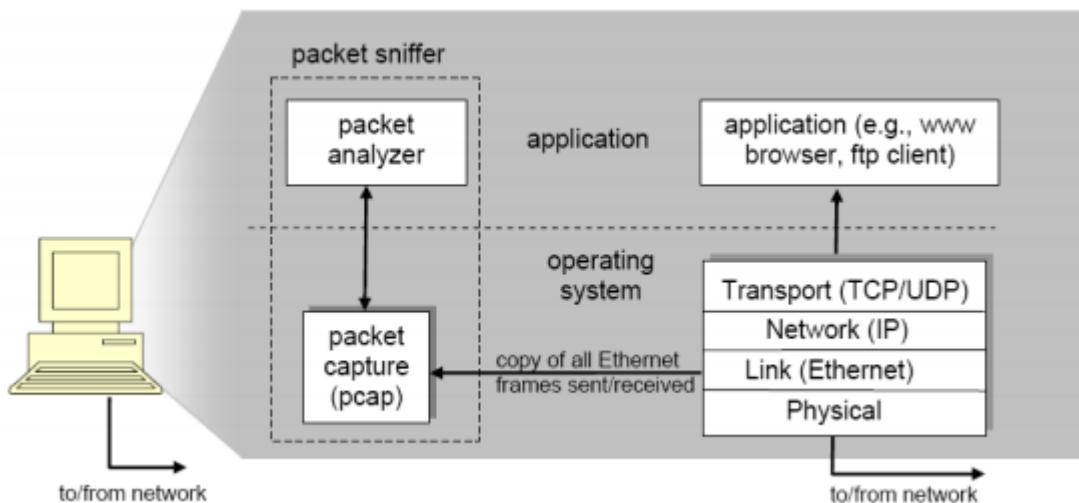
Last update: **2021/12/14 16:24**

Netzwerkanalyse mit Wireshark

Das Monitoring inklusive der notwendigen Detailanalyse des Datenverkehrs im Netzwerk ist ohne ein leistungsfähiges Analysesystem unmöglich. Eines der **wichtigsten Netzwerktools** für jeden **Administrator** ist **Wireshark** – ein Open Source-Netzwerkanalysator, mit dem Sie alle **Pakete im Netzwerk aufzeichnen** und die Paketinhalte detailliert analysieren.

Packet Sniffer

Das grundlegende Werkzeug für die Beobachtung von Daten zwischen Rechnern wird als **packet sniffer** bezeichnet. Wie der Name schon sagt, **fängt** dieses Werkzeug **empfangene/gesendete Daten** Ihres Rechners **ab**. Ein Sniffer ist immer **passiv**, das heißt er **verschickt keine Daten**, sondern **speichert Kopien der Daten** der Kommunikation auf Ihrem Rechner.



Das Bild 1 zeigt die Struktur eines „Sniffers“. Rechts unten im Bild sind die beteiligten Protokolle abgebildet, es werden also Protokolle der Layer 1 bis 4:

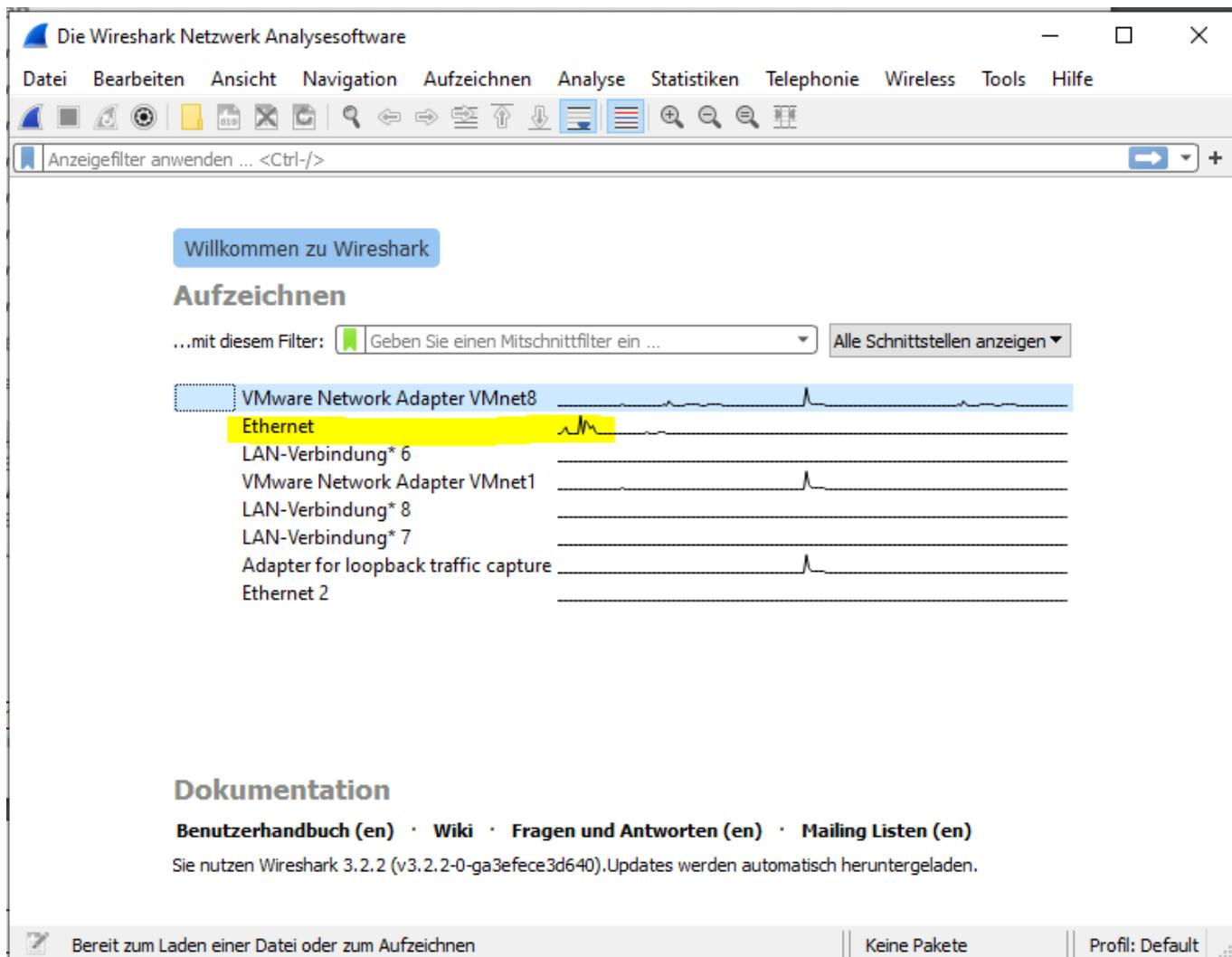
- oben ist die Applikation zu sehen, in unserem Fall ist dies der Webbrowser (Firefox, IE).
- die Blöcke die in den gestrichelten Linien eingerahmt sind gehören zu unserem „Sniffer“.

Am einfachsten lässt sich ein Netzwerk sniffen wenn Hubs als Verbindung der Netzwerkssegmente zwischengeschaltet sind. Bei anderen Verbindungen wie z.B. bei einem Switch bekommt man im Normalfall Probleme den Netzwerkverkehr abzuhören, da der Switch die Daten vom Sender nur an den tatsächlichen Empfänger weiterleitet. Somit würde man nur die eigenen Daten, sowie unwichtigeren Netzwerkverkehr wie z.B. Broadcasts aufzeichnen. Für diesen Fall haben viele Hersteller dieser Komponenten **Switches** bzw. **Router** mit einer **Monitorfunktion** in Ihrem Portfolio. Damit ist es möglich Netzwerddaten eines gewünschten Ports auf einen anderen zu spiegeln. Auf diesen gespiegelten Port kann man nun direkt zugreifen.

Einführung in Wireshark

Startet man das Programm, muss man zuerst eine Schnittstelle (z.B.: Ethernet) wählen, die man

abhören möchte:



Der Bildschirm, in dem die aufgezeichneten Daten bearbeitet und analysiert werden ist in 3 Bereiche aufgeteilt:

1) Paketliste

In der Paketliste, sieht man alle aufgezeichneten Frames. Folgende Spalten werden standardmäßig angezeigt:

1. No. = ist eine fortlaufende Nummerierung der Frames 2. Time = zeigt den Zeitabschnitt der Aufzeichnung an 3. Source = zeigt den Absender eines Frames an (meist die IP) 4. Destination = zeigt den Empfänger des Frames an (meist die IP) 5. Protocol = zeigt das verwendete Protokoll des Frame an 6. Info = gibt zusätzliche Informationen zum Frame bekannt

Aufzeichnen von Ethernet

Datei Bearbeiten Ansicht Navigation Aufzeichnen Analyse Statistiken Telefonie Wireless Tools Hilfe

No.	Time	Source	Destination	Protocol	Length	Info
1158	214.841234	192.168.1.29	192.168.1.10	TCP	54	63336 → 443 [ACK] Seq=580
1159	215.590861	192.168.1.29	192.168.1.10	TLSv1.3	995	Application Data
1160	215.590942	192.168.1.29	192.168.1.10	TLSv1.3	155	Application Data
1161	215.591186	192.168.1.10	192.168.1.29	TCP	60	443 → 63336 [ACK] Seq=861
1162	215.984084	74.125.133.189	192.168.1.29	UDP	82	443 → 65264 Len=40
1163	216.010090	192.168.1.29	74.125.133.189	UDP	70	65264 → 443 Len=28
1164	216.202940	52.109.88.122	192.168.1.29	TLSv1.2	99	Application Data
1165	216.243318	192.168.1.29	52.109.88.122	TCP	54	63005 → 443 [ACK] Seq=71
1166	217.636990	192.168.1.29	185.199.109.153	TCP	55	[TCP Keep-Alive] 63228 →
1167	217.646662	185.199.109.153	192.168.1.29	TCP	66	[TCP Keep-Alive ACK] 443
1168	217.879712	108.177.15.189	192.168.1.29	UDP	82	443 → 59210 Len=40
1169	217.906999	192.168.1.29	108.177.15.189	UDP	70	59210 → 443 Len=28
1170	219.255630	192.168.1.20	239.255.255.250	SSDP	167	M-SEARCH * HTTP/1.1
1171	219.782986	192.168.1.29	52.113.194.132	TCP	55	[TCP Keep-Alive] 63339 →
1172	219.794214	52.113.194.132	192.168.1.29	TCP	66	[TCP Keep-Alive ACK] 443
1173	220.790910	192.168.1.29	173.194.76.189	UDP	65	59222 → 443 Len=23
1174	220.830983	173.194.76.189	192.168.1.29	UDP	63	443 → 59222 Len=21

2) Paketdetails

In den Paketdetails werden die OSI-Layer (Schichten) des Datenframes angezeigt. Durch anklicken des Pfeil-Symbols kann der gewählte Layer erweitert werden.

```
> Frame 1384: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{428B0
> Ethernet II, Src: Giga-Byt_4f:3a:d2 (00:1a:4d:4f:3a:d2), Dst: HewlettP_0b:2f:71 (24:be:05:0b:2f:71)
> Internet Protocol Version 4, Src: 192.168.1.10, Dst: 192.168.1.29
▼ Transmission Control Protocol, Src Port: 443, Dst Port: 63336, Seq: 10545, Ack: 8934, Len: 0
  Source Port: 443
  Destination Port: 63336
  [Stream index: 26]
  [TCP Segment Len: 0]
  Sequence number: 10545 (relative sequence number)
  Sequence number (raw): 1749144258
  [Next sequence number: 10545 (relative sequence number)]
  Acknowledgment number: 8934 (relative ack number)
  Acknowledgment number (raw): 2511940191
  0101 .... = Header Length: 20 bytes (5)
> Flags: 0x010 (ACK)
  Window size value: 434
  [Calculated window size: 55552]
  [Window size scaling factor: 128]
  Checksum: 0x366a [unverified]
```

Natürlich variieren die Protokolldetails von Protokoll zu Protokoll. Nur die ersten 2 Schichten sind immer vorhanden.

3) Paketdaten (Hexadezimal)

Hier sind die Daten einmal im Hexadezimalsystem (links) und nebendran im Klartext (rechts) bzw. in entschlüsselter Form angezeigt.

0000	24 be 05 0b 2f 71 00 1a 4d 4f 3a d2 08 00 45 00	\$.../q... MO:...E...
0010	00 28 8b 6c 40 00 40 06 2b ec c0 a8 01 0a c0 a8	.(1@... +.....
0020	01 1d 01 bb f7 68 68 41 d2 c2 95 b9 2a 5f 50 10hhA*_P...
0030	01 b2 36 6a 00 00 00 00 00 00 00 00 00 00 00 00	6j.....

Filter

Nachdem praktisch ständig Netzwerkpakete gesendet und empfangen werden, ist es wichtig, dass es Möglichkeit gibt, um den Netzwerktraffic zu filtern. In dem in der Abbildung gezeigten Textfeld können beliebige Filter eingestellt werden.

No.	Time	Source	Destination	Protocol	Length	Info
9256	704.600432	192.168.1.29	194.232.104.150	HTTP	904	GET / HTTP/1.1
9262	704.621447	194.232.104.150	192.168.1.29	HTTP	550	HTTP/1.1 301 Moved Permanent

```

> Frame 9256: 904 bytes on wire (7232 bits), 904 bytes captured (7232 bits) on interface \Device\NPF_{4
> Ethernet II, Src: HewlettP_0b:2f:71 (24:be:05:0b:2f:71), Dst: Netgear_3e:59:a9 (b0:7f:b9:3e:59:a9)
> Internet Protocol Version 4, Src: 192.168.1.29, Dst: 194.232.104.150
> Transmission Control Protocol, Src Port: 63465, Dst Port: 80, Seq: 1, Ack: 1, Len: 850
< Hypertext Transfer Protocol
  > GET / HTTP/1.1\r\n
    Host: www.orf.at\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.150 Safari/537.36
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=1
    Accept-Encoding: gzip, deflate\r\n
  
```

Im obigen Beispiel wurden nur http-Pakete mit der IP-Adresse 194.232.104.150 angezeigt. Wie man in der Abbildung erkennen kann, ist/war dies die IP-Adresse der Internetseite www.orf.at. Da http natürlich das TCP-Protokoll, IP-Protokoll & Ethernet-Protokoll nutzt, werden auch diese Daten angezeigt.

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

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:2:2_10

Last update: **2021/12/14 16:24**



Wireshark Übungen

1. Übung

1. Starte Wireshark
2. Starte einen Webbrower mit einer beliebigen Seite
3. Filtere nach dem Protokoll http.
4. Öffnen die Seite <http://elearn.bgamstetten.ac.at>
5. Starte die Aufzeichnung nochmals neu und aktualisiere die Seite mit STRG+F5
6. Dokumentiere mittels Screenshot, wie viele gesendete und empfangen Pakete über die Schnittstelle mitgesnifft wurden.
7. Dokumentiere, welche IP-Adresse sich hinter dem FQDN elearn.bgamstetten.ac.at steckt.
8. Dokumentiere, welche Protokolle bei der ersten Antwort (text/html) zum Einsatz kamen.
9. Dokumentiere, wie groß die Antwort (text/html) in Bytes war.
10. Dokumentiere, wie groß
 - der Header des Ethernet 2 – Frames
 - der Header des IP-Paket-Pakets
 - der Header des TCP-Segments
 - und die Daten des http-Protokolls (gzip) der Antwort (text/html) waren.
11. Überprüfe, ob die die Größen zusammen die Gesamtgröße von zuvor ergeben.
12. Finde im Bereich Paketdetails den HTML-Quellcode und eine im Quellcode versteckte Botschaft.
Der Name des input-Tags lautet STRENGGEHEIM und die Botschaft verbirgt sich value-Attribut.
13. Kontrolliere deine eigene MAC-Adresse (ipconfig /all) und finde die Ziel-MAC-Adresse heraus.
14. Finde heraus, ob IPv4 oder IPv6 als Protokoll auf der Netzwerkschicht verwendet wurde.
15. Finde heraus, welcher Port auf deinem PC bzw. am ELEARN-Server für die Netzwerkverbindung verwendet wurde.
16. Finde heraus, warum beim ersten Paket im TCP-Header die Sequence Number=1 ist und die Sequence Number beim übernächsten Paket nicht 2 sondern z.B.: 603 ist.
17. Finde heraus, welche HTTP-Version 1.0 oder 1.1 verwendet wurde.
18. Finde heraus, welcher User-Agent verwendet wurde.
19. Finde heraus, welche Sprachen vom Browser akzeptiert werden.
20. Bei der Antwort (text/html) wird ein HTTP-Status-Code mitgeschickt. Finde ihn heraus und recherchiere, was dieser Status Code bedeutet.
21. Finde heraus, welches Betriebssystem bzw. welcher Webserver verwendet wird.
22. Melde dich von der elearn-Plattform ab und anschließend wieder an. Finde heraus, ob und wie deine Login-Informationen mitgeschickt werden (Tipp Info=POST & do=login).

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

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf7bi8bi_202122:2:2_10:2_10_01



Last update: **2021/12/14 16:25**