

Grundlagen von PHP

Sommersemester 2022

Prof. Dr.-Ing. Thomas Barth
Anwendungsentwicklung und IT-Systemarchitekturen
Studiengang Wirtschaftsinformatik
RFH Köln
barth@rfh-koeln.de

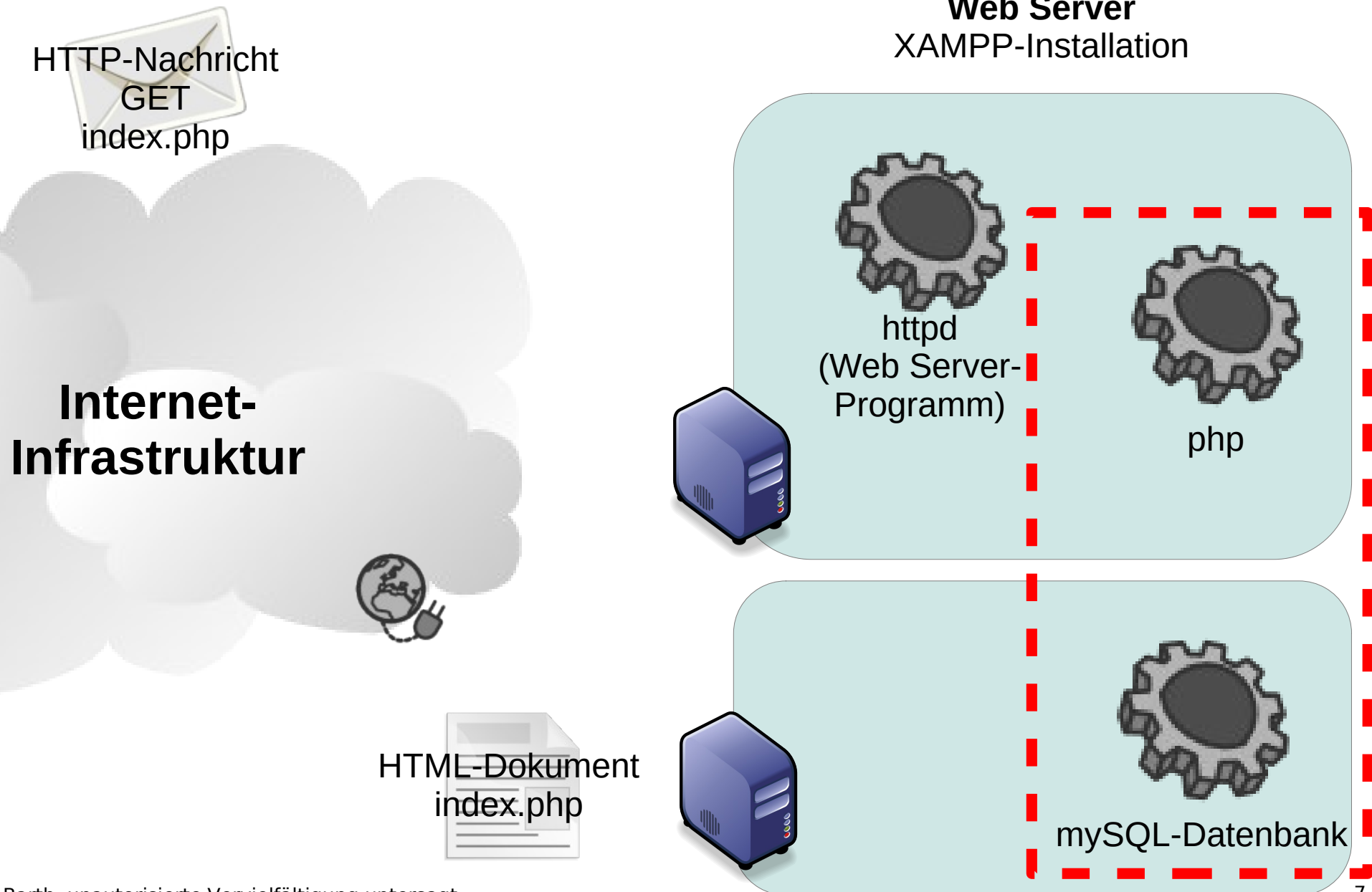
- ✓ Bedeutung von php im Kontext von Web Anwendungen kennen
- ✓ Eingabe-Daten aus einem Formular serverseitig verarbeiten können
- ✓ Funktionalität von PHP zum Zugriff auf Datenbanken kennen
- ✓ Grundzüge der Datenverwaltung kennen
- ✓ Daten/Informationen für eine Web-Anwendung identifizieren können
- ✓ Einfache Datenmodelle für die Verwaltung in einer Datenbank erstellen können
- ✓ Einfache Datenmanipulationen in einer Datenbank ausführen können
- ✓ Server-seitige Funktionalität in PHP formulieren können

- ✓ PHP ist ein rekursives Akronym: „PHP: Hypertext Preprocessor“
- ✓ „Präprozessor“ bedeutet hier, dass die Hypertext-Seiten (sprich: HTML-Dokumente, Web-Seiten) **vor** dem Verschicken an den Client **verarbeitet (d.h. erstellt)** werden
- ✓ PHP-Entwicklung wurde 1995 begonnen, wird derzeit verbreitet verwendet (Stand 1.6.2021: auf 79,2% der von www.w3techs.com analysierten Server)
- ✓ PHP ist ein Open Source Framework (Download unter www.php.net)
- ✓ Vorkonfigurierte Web Server (z.B. von Hosting-Anbietern) enthalten meist PHP
- ✓ PHP wird auch für die Implementierung verbreitet eingesetzter Content Management Systeme (CMS) verwendet (z.B. Joomla!, Drupal, TYPO3)
- ✓ PHP wird für die Implementierung von „kleinen“ (privaten) **und** großen (kommerziellen) Systemen verwendet Blogs, Shop-Systeme, Foren, Bildergalerien usw. usf.

- ✓ PHP ist eine **Programmiersprache, die auf der Server-Seite einer Web-Anwendung ausgeführt wird** (zur Erinnerung: JavaScript wird auf der Client-Seite ausgeführt!)
- ✓ PHP ist auf „allen“ Plattformen verfügbar (z.B. Windows, Linux, MacOS, Unix)
- ✓ PHP und JavaScript ergänzen sich: Prüfungen von Benutzereingaben können ...
 - ... durch JavaScript auf dem Client (z.B. „Ist die Mailadresse syntaktisch korrekt?“)
 - ... danach von PHP auf dem Server (z.B. „Ist unter dieser Mailadresse schon jemand registriert?“)durchgeführt werden
- ✓ Mit PHP wird die Erzeugung von **dynamischen Web-Seiten** ermöglicht, d.h. von Web-Seiten, die für jede Anfrage neu erzeugt werden
- ✓ Eine PHP-Datei (Endung .php) kann enthalten:
 - HTML,
 - Eingebettete JavaScript-Programmteile und
 - **PHP-Code**

- ✓ PHP kann mit vielen Datenbanken zusammen genutzt werden, z.B. auch mit MySQL (enthalten in XAMPP)
- ✓ PHP kann mit vielen Web-Servern genutzt werden (Apache httpd (enthalten in XAMPP), Microsoft Internet Information Server/IIS)
- ✓ PHP kann ...
 - **... dynamisch (d.h. zum Zeitpunkt der Anfrage an den Web-Server) HTML-Seiten erzeugen** (genauer: beliebigen Inhalt, auch Bilder, Flash-Filme usw.)
 - ... Dateien auf dem Web Server erzeugen/lesen/ändern/schreiben
 - **... Formulardaten verarbeiten**
 - ... Cookies senden und empfangen
 - **... Benutzer-Sitzungen („sessions“) verwalten**
 - **... Daten in einer Datenbank erzeugen/lesen/ändern/schreiben**
 - ... den Zugriff auf Web-Seiten regeln
 - .. Daten verschlüsseln
- ✓ Beim Zugriff auf Web-Seiten mit der Endung .php werden diese auf dem Web Server automatisch ausgeführt ... wenn PHP installiert ist ...

Komponenten des Word Wide Web „in Aktion“

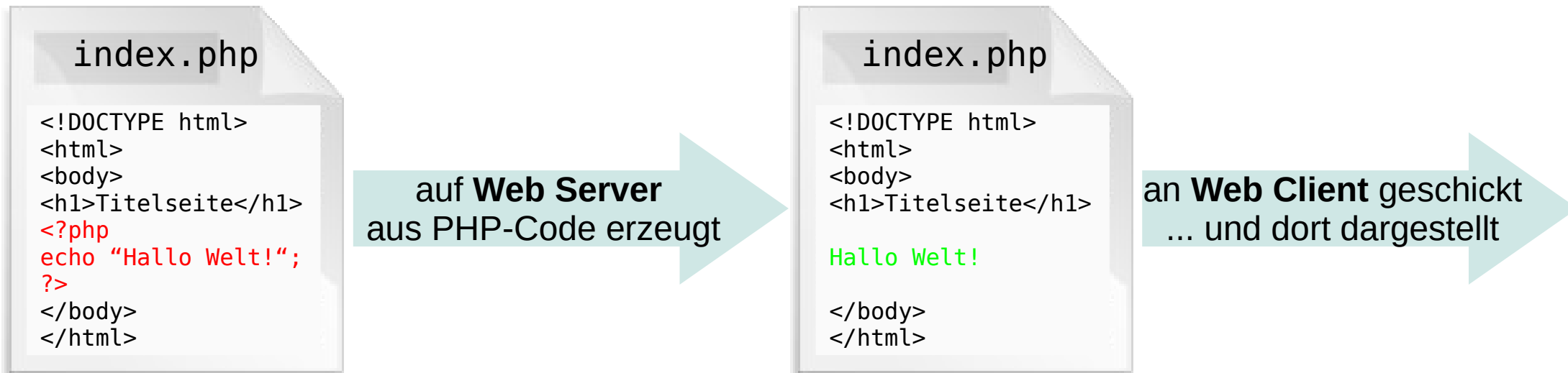


Grundlagen von PHP

- ✓ Abschnitte mit PHP-Code können beliebig in eine HTML-Seite integriert sein, die von einem Web-Server verwaltet wird
- ✓ PHP-Code wird in spitzen Klammern mit dem Schlüsselwort `?php` an beliebiger Stelle in der HTML-Seite notiert:

```
<?php  
// beliebiger Abschnitt PHP-Code  
?>
```

- ✓ Aus den HTML-Elementen zusammen mit dem ausgeführten PHP-Code wird das HTML-Dokument erzeugt, das dann an den Web-Server zurückgegeben



- ✓ In einer PHP-Datei wird also ...
 - ... ein **statisches Grundgerüst der HTML-Seite** in HTML definiert und
 - ... die **dynamischen Abschnitte**, die erst zur Laufzeit bekannt sein können (z.B. der Name des Benutzers, die einzelnen Treffer einer Suche), durch PHP **beschrieben**
- ✓ Das PHP-Programm im Web-Server übernimmt die HTML-Teile der PHP-Seite und **ersetzt die PHP-Abschnitte durch deren Ergebnis** (als HTML-Dokument)
- ✓ **Die PHP-Abschnitte definieren also den Inhalt der späteren Seite durch die Angabe, wie diese Inhalte ermittelt werden können**
- ✓ Dazu beinhaltet PHP alle gängigen Programmiersprachen-Konstrukte
 - **Variablen** und **Felder**
 - **Schleifen** zur Wiederholung von Code-Abschnitten
 - **Entscheidungen** zur Auswahl zwischen Code-Abschnitten
 - **Anweisungen** und **Funktionen**, z.B.
 - Zum Verarbeiten von Daten (z.B. Zeichenketten)
 - Zum Zugriff auf Datenbanken
 - Zum Ausgeben in eine HTML-Datei (z.B. Funktion `print(. .)`, PHP-Anweisung `echo`)

✓ Einige grundsätzliche Syntax-Regeln

- Jede Zeile PHP-Code muss mit einem **Semikolon** enden
- Ein einzeliger **Kommentar** – bei Ausführung ignoriert – im PHP-Code beginnt mit „//“
- Mehrzeilige Kommentare beginnen mit '/*' und enden mit '*/'

➤ Beispiele:

```
<?php
// hier folgt das unvermeidliche allererste Programm:
echo "Hallo Welt!";
/* and now for something completely different:
echo "Hello World!" */
?>
```

- ✓ **Variablen** können beliebige Werte (Zahlen, Zeichenketten ...) aufnehmen, die Ergebnis einer Berechnung oder einer anderen Bearbeitung sind, z.B. `$x=$x+1;`
- ✓ **Variablennamen** beginnen mit '\$', dann folgt eine beliebige Zeichenkette, die mit Buchstaben oder dem sog. „underscore“ ('_') beginnt und danach ausschließlich aus alphanumerischen Zeichen (a-z, A-Z, 0-9) oder '_' besteht, z.B.: `$x`, `$_x`, `$x_`, `$anzahlTrefferInDerListe`, `$anzahl_Personen`
- ✓ **Unzulässige Namen** sind z.B. `$12xy`, `$%x`, `$x%`, `$!anzahlStudis`, `$a b`

- ✓ PHP ist „case-sensitive“, d.h. Groß-/Kleinschreibung wird beachtet!
 - Die Variable `$vornameAusFormular` entspricht **nicht** der Variable `$vornameausformular`
- ✓ Dies gilt für Variablennamen **und** die PHP-Anweisungen ('echo' vs. 'Echo')
- ✓ PHP ist – übrigens wie Javascript, aber nicht wie Java – eine **schwach typisierte Sprache**, der Typ von Variablen orientiert sich nach dem aktuellen Inhalt
- ✓ Beispiel:
`$x=5; // x ist eine ganze Zahl`
`$txt="Hallo Welt!"; // txt ist eine Zeichenkette`
`$x="Hello World!!"; // jetzt ist x auch eine Zeichenkette`
- ✓ In PHP existiert eine Vielzahl vordefinierter Funktionen (z.B. zur Ausgabe in HTML-Dateien, zum Zugriff auf Datenbanken), deren Aufruf durch Angabe des Namens und möglicher Parameter (in runden Klammern) erfolgt:
 - `print("Hallo Welt!"); // gibt Text in HTML-Dokument aus`
 - `$connect=mysqli_connect("localhost","root",""); // erstellt DB-Verbindung`
- ✓ Rückgabewert eines Funktionsaufrufs (das Ergebnis) kann in Variablen gespeichert werden (siehe `$connect` im letzten Beispiel)

✓ **Q1:**

Schreiben Sie eine PHP-Seite `hallowelt.php`, die eine HTML-Seite mit „Hallo Welt!“ (z.B. fettgedruckt oder in einer h1-Überschrift) als Inhalt zurückgeliefert.

Ergänzen Sie dieses „Programm“ um eine Variable, der Sie einen Wert zuweisen, ihn verändern und ausgeben. Ändern Sie auch den „Typ“ der Variablenwerte in Ihrem PHP-Programm und experimentieren Sie mit Rechenoperationen auf Zeichenketten.

✓ Einige nützliche Funktionen sind in PHP vordefiniert:

- `isset(x)`: prüft, ob die angegebene Variable `x` überhaupt im PHP-Programm definiert ist (Rückgabe `true`), bspw. um Fehlermeldung und Programmabsturz zu vermeiden
Beispiel:
 - `isset($name)` liefert `true` zurück, wenn im PHP-Programm diese Variable existiert, sonst `false`
 - `isset($einFeld["einSchlüssel"])` liefert `true` zurück, wenn das Feld namens `einFeld` ein Element unter dem Schlüssel `einSchlüssel` enthält, sonst `false`
- `date()`: gibt das aktuelle Datum und die Uhrzeit (Datum/Zeit **auf dem Server!**) aus
- `date(format)`: `format` ist eine Zeichenkette die angibt, in welcher Form Datum und Uhrzeit angegeben werden sollen, z.B.
 - `date("d.m.Y G:i:s")` ergibt `"08.10.2013 14:35:07"`
 - `date("F M Y G:i:s")` ergibt `"8 Oct 2013 14:35:07"`
- `strlen(s)`: ermittelt die Länge der Zeichenkette `s`
- `strpos(satz,suchbegriff)`: ermittelt, an welcher Position `suchbegriff` in `satz` vorkommt; liefert `false`, falls `suchbegriff` nicht enthalten ist, z.B.
 - `strpos("onkel@otto.de","@")` ergibt `5`
 - `strpos("onkel@ottode",".")` ergibt `false`

Vordefinierte Funktionen PHP (2)

- `substr(satz, von, n)`: liefert `n` Zeichen der Zeichenkette `satz` ab Position `von` zurück.
Beispiel:
 - `substr("onkelotto", 5, 4)` liefert `otto` zurück
- `str_replace(auswahl, ersatz, satz)`: ersetzt in `satz` jedes Vorkommen von `auswahl` durch `ersatz`
 - `str_replace("ü", "ue", "Müller")` liefert `Mueller` zurück
- `explode(trenner, satz)`: spaltet `satz` an jedem Vorkommen von `trenner` auf und liefert die Bestandteile als Array zurück
 - `$feld=explode(" ", "Dies ist ein langer Satz.")` ergibt
`$feld[0]="Dies"`, `$feld[1]="ist"` usw. usf.
- `print(x)`: gibt den Inhalt der Variable `x` aus.
- `print_r(x)`: gibt den Inhalt der Variable `x` in lesbarer („readable“) aus; z.B. wird ein Array elementweise jeweils mit Index und Inhalt ausgegeben
- Verkettung von Zeichenketten kann in php mit dem „.-Operator, z. B. bei Ausgaben mit `print()` oder `echo` erfolgen: `print "Hallo ".$var;`

Arrays (Felder) in PHP

- ✓ In **Feldern** können „beliebig“ viele Werte (evtl. unterschiedlichen Datentyps!) gespeichert werden
- ✓ Auf die einzelnen Elemente kann zugegriffen werden über ...
 - ... den **Index** (0, 1, 2, ... (Anzahl Elemente - 1)) oder
 - ... einen **Schlüssel** (im Falle eines sog. „**assoziativen Felds**“ → JavaScript)... der hinter dem Feldnamen in eckigen Klammern „[...]“ angegeben wird
- ✓ Ein Feld name kann definiert werden durch `$namen=array(“Jack“,“Jill“);`
- ✓ Mit `count($namen)` kann die Anzahl Elemente im Feld ermittelt werden
- ✓ Bei **assoziativen Feldern** kann ein „Name“ (=Schlüssel) für jedes Element vergeben werden; das Element ist dann mit dem Schlüssel eindeutig assoziiert
- ✓ Wichtige Anwendung: Der serverseitige Zugriff auf übergebene Formular-Daten von einer HTML-Seite erfolgt über assoziative Felder, auf die mit dem HTML-Namen als Schlüssel zugegriffen wird

Arrays (Felder) in PHP

✓ Beispiel: „Telefonbuch“

```
// erste Variante über die PHP-Funktion array(..) und den Operator =>  
$telefonbuch = array("Jack"=>"0171/1234567","Jill"=>"0151/987654");  
// zweite Variante über die direkte Zuweisung  
$telefonbuch["John"]="0163/11223344";
```

```
// Zugriff über den Schlüssel als Feldindex  
$nummer=$telefonbuch["Jack"];  
// ergibt "0171/1234567" in Variable nummer
```

Formulardaten verarbeiten in PHP

- ✓ Eine Funktionalität von PHP ist, Eingaben in einem HTML-Formular zu verarbeiten
- ✓ Beispiel:
 - Benutzer meldet sich am Client mit Name und Passwort an (in einem HTML-Formular)
 - Web-Browser überträgt (s. Attribute method und action) die Eingaben an eine PHP-Seite
 - Auf dieser PHP-Seite steht der Code, der prüft, ob der Benutzer registriert und sein Passwort korrekt ist (z.B. durch Suche in einer Datenbank mit den Nutzerdaten)

HTML kann Prüfung nicht durchführen, da es keine Programmiersprache ist!

- ✓ Auf die Formulareingaben (die vom Web-Browser übertragen wurden) kann in PHP (z.B. in der Datei **check_login.php**) wie folgt zugegriffen werden:
 - Bei Methode **POST** (im HTML-Formular ausgewählt): `$name=$_POST['pwd'];`
 - Bei Methode GET (im HTML-Formular ausgewählt): `$name=$_GET['pwd'];`
 - Danach ist in der PHP-Variable \$pwd der Wert verfügbar, der in das HTML-Eingabefeld mit Namen pwd eingegeben wurde; zur Erinnerung:

```
<form action="check_login.php"
  Username: <input type="password"
eingegeben" />
```


...

```
method="post"
name="pwd" > value="Passwort
```


Bedingungen und Verzweigungen in PHP

- ✓ An bestimmten Stellen in einem Programmablauf kann es erforderlich sein, anhand einer **Bedingung** zu unterscheiden, was als nächstes aufgeführt wird
- ✓ Eine Bedingung kann ein Vergleich oder die logische Verknüpfung von Vergleichen sein
 - Vergleiche
 - <, >, <=, >=
 - !=, ==, ===
 - Logische Verknüpfung
 - UND ('&')
 - ODER ('|')
 - NICHT ('!')
- ✓ Beispiel:
 - „Hat x den Wert 4711?“ ... `$x==47`
 - „Hat x einen Wert kleiner gleich 0 oder größer als 5?“ ... `($x<=0 | $x>5)`
 - „Hat x den gleichen Wert wie y und ist z größer als 42?“ ... `($x==$y) & ($z>42)`

✓ **Q1:**

Schreiben Sie ein HTML-Dokument login.html, in dem Name und Passwort (in einem Formular) eingegeben werden können und senden Sie diese Daten an eine PHP-Datei check_login.php. Diese Seite soll anhand des Namens (OHNE Passwort-Überprüfung) die/den Benutzer/in begrüßen.

Vollziehen Sie den Zusammenhang zwischen HTML-Formular und php-Code und -Variablen nach.