

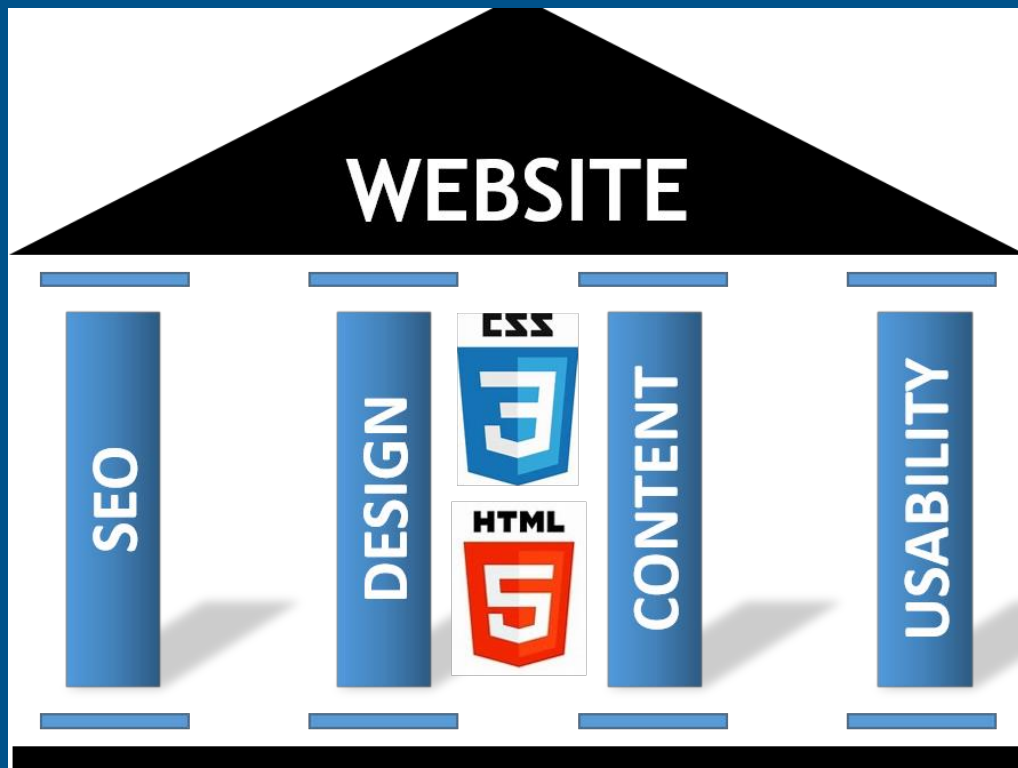


Das erwartet Sie:

- HTML und PHP - Zusammenspiel
- Mit Formulardaten DB-Zugriff realisieren

**Benutzerschnittstellen  
gestalten und  
entwickeln**





# Benutzerschnittstellen gestalten und entwickeln

HTML/CSS/PHP/SQL

## Lernziel

Gestalten und Entwickeln einer Benutzeroberfläche für softwarebasierte

- Arbeitsabläufe
- Geschäftsprozesse

# Themen



## HTML

Grundlagen  
Tabellen, Listen,  
Formular etc.



## CSS

Cascading  
Stylesheets



## Responsive Webdesign

Anpassen der Webseite  
für verschiedene  
Ausgabemedien



## SEO

Suchmaschinen-  
optimierung



## PHP-SQL

Einbinden von  
Datenbanken mittels  
PHP

# Statische Webseiten



HTML-Datei

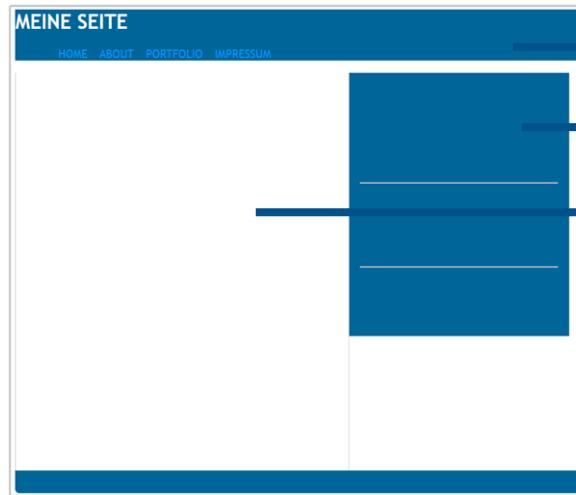


Auslieferung

# Statische Webseiten

- Eine Datei für jede Seite der Website
- Gleiche Ausgabe (Inhalt fest), wenn Seite angefordert wird
- Vorteile:
  - Erstellungsaufwand gering
  - Kosten gering
  - Einfacher zu hosten
- Nachteile:
  - Für Änderungen müssen HTML-Kenntnisse vorhanden sein
  - Änderungen des Webdesigns oder des Seitenaufbaus müssen an jedem Dokument vorgenommen werden (evtl. zeitintensiv)
  - Zusätzliche Software notwendig, zum Hochladen der geänderten Dateien

# Dynamische Webseiten



HTML-Datei

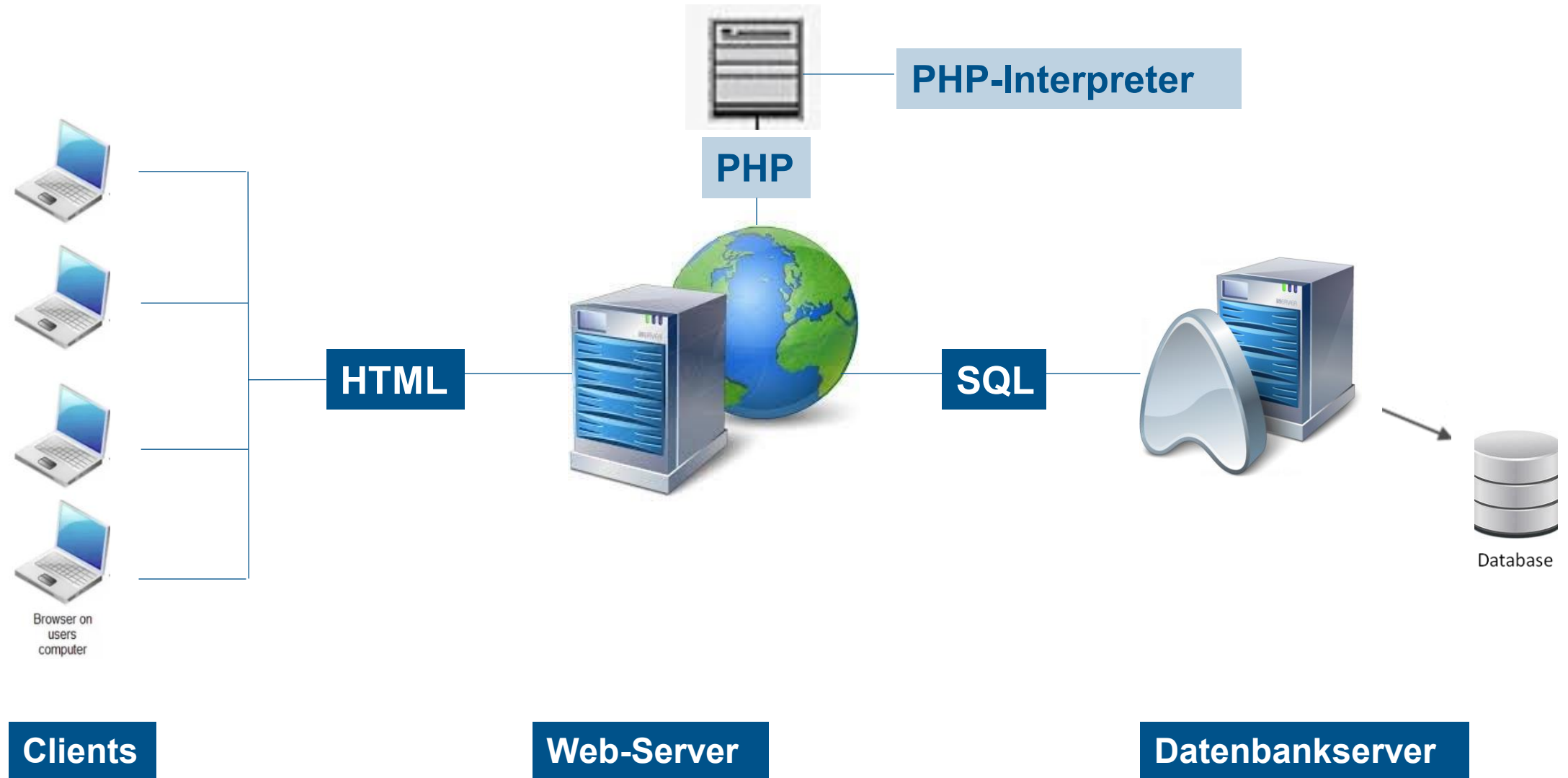


Auslieferung

# Dynamische Webseiten

- Serverseitig oder clientseitig erzeugte Dokumente
- Kommunikation mit dem Server  
beispielsweise Kontaktformulare, Besucherzähler, aktuelles Datum etc.
- Besucher fordert Seite an, Dokument fordert vom Server dynamische Inhalte an, diese werden zur Verfügung gestellt und anschließend angezeigt
- Vorteile:
  - Änderungen der Inhalte ohne spezielle Kenntnisse und Software online möglich wie z. B. bei Content Management Systeme (CMS), Onlineshops
  - Inhalte und Design können voneinander getrennt werden
- Nachteile:
  - PHP-Kenntnisse für die Entwicklung erforderlich
  - Möglicherweise auch eine Datenbank erforderlich

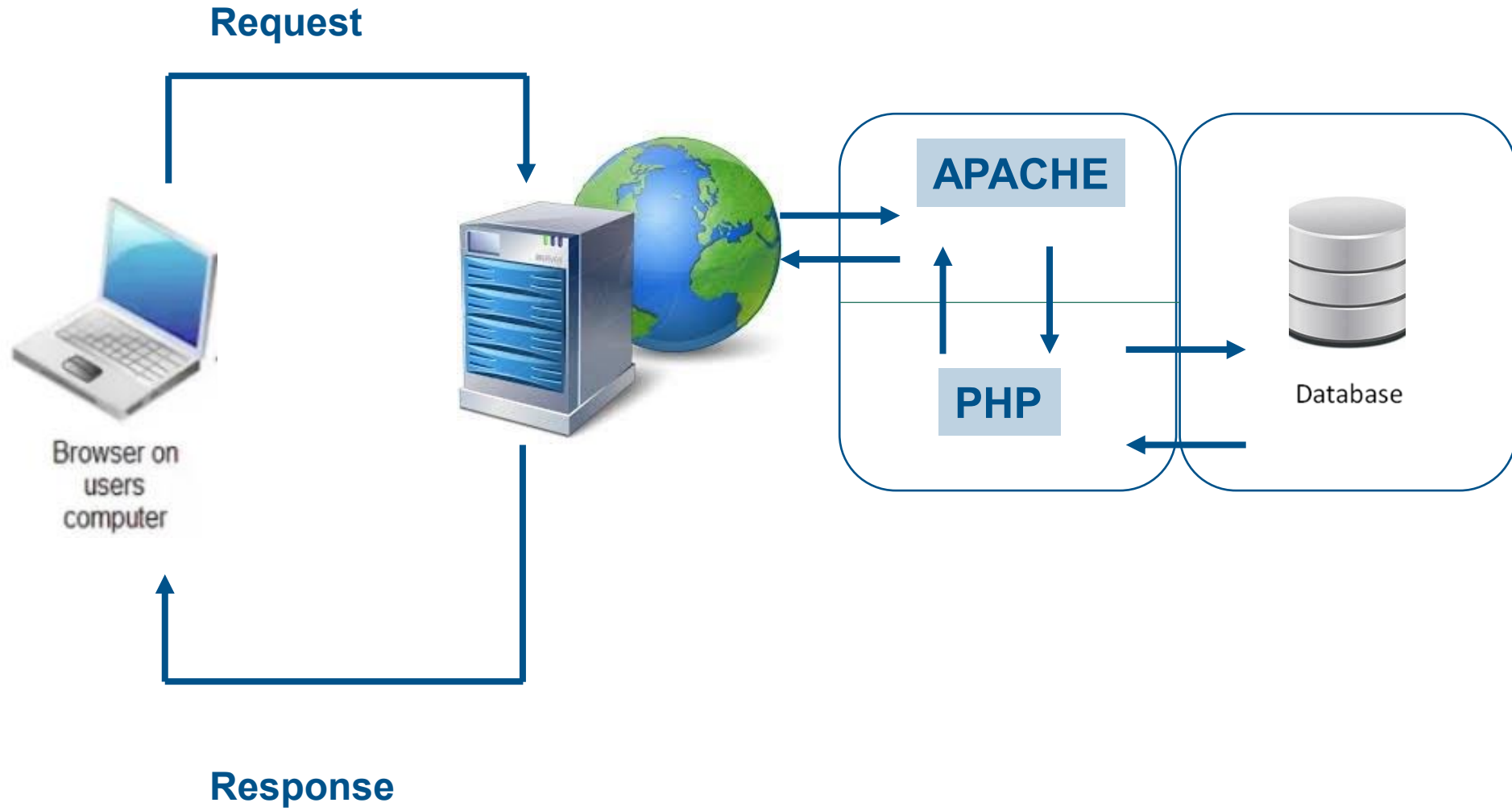
# Szenario





- PHP ist eine serverseitige Open-Source-Anwendung
- Es wird auf einem Webserver (IIS oder Apache) installiert
- PHP-Code kann in HTML integriert, oder HTML per PHP erzeugt werden
- Ermöglicht Webseiten die Anbindung von Datenbanken
- Mittels in PHP eingebetteter SQL-Befehle werden Datenbankabfragen realisiert
- Die Abarbeitung der Scripte erfolgt vor Ausgabe auf den Client-PC
- Nach Ausgabe ist der PHP-Code im Quelltext der Webseiten nicht mehr sichtbar
- Durch zahlreiche Funktionsbibliotheken sehr flexibel einsetzbar
- Gängige Dateiendungen sind: .php, .php3, .php4, .phtm, .phtml

# Ablauf eines Aufrufes einer Webdatenbank mit Ausgabe



# Ablauf eines Aufrufes einer Webdatenbank mit Ausgabe

1. Request (Anfrage) an die Servermaschine.
2. Weiterleitung an den internen Webserver (Apache).
3. Weiterleitung an den PHP-Interpreter.
4. Interpretation der PHP-Befehle und damit Anfrage an die MySQL-Datenbank.
5. Ausführung der SQL-Befehle und Rückgabe des Ergebnisses.
6. Verarbeitung und Layout der Ergebnisse.
7. Weiterleitung der Ergebnisse an den internen Webserver
8. Response (Antwort) im HTML-Format.

# Unterschied Programmier- und Skriptsprachen

## Programmiersprache

---

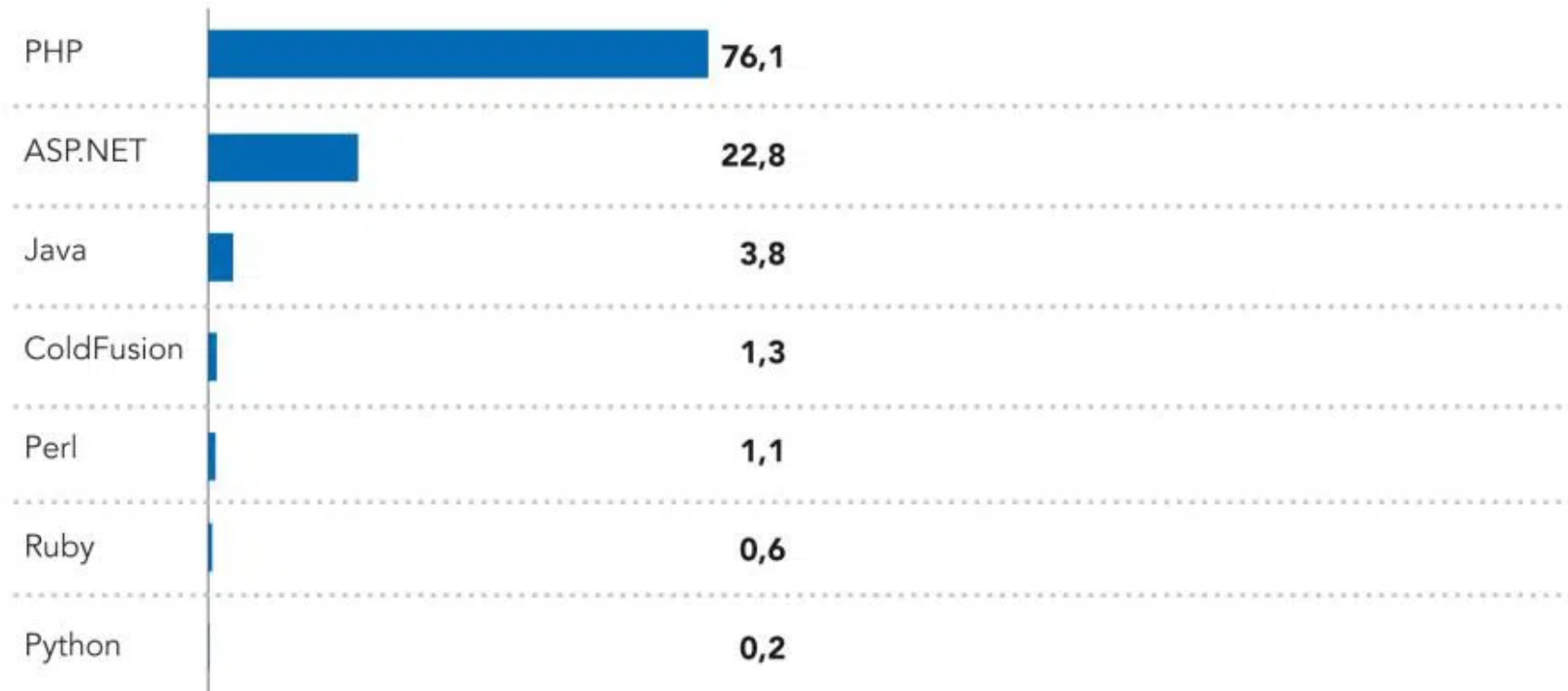
- Quelltext-Datei
- Compiler (Übersetzer)
  - Variablen definiert?
  - Passen die Typen von Funktionsargumenten zu den Funktionsdefinitionen?
  - Syntaxfehler vorhanden?
- Binärcode-Datei
  - Stark optimiert
- Eignet sich für rechenintensive Aufgaben und größere Systeme

## Skriptsprache

---

- Keine Kompilierung (keine Binärdatei)
- In Skriptsprachen geschriebene Programme sind für gewöhnlich weniger effizient in der Ausführung
- Skriptsprachen **entlasten den Programmierer**, indem sie **den Prozessor stärker belasten**
- Besonders gut für kleinere bis mittelgroße Programme geeignet
- Keine Angabe des Typs einer Variablen notwendig
- Keine *main()*-Funktion
- Einsatz von Interpretern

## PHP, Python, Ruby, Java: Web-Programmiersprachen im Vergleich

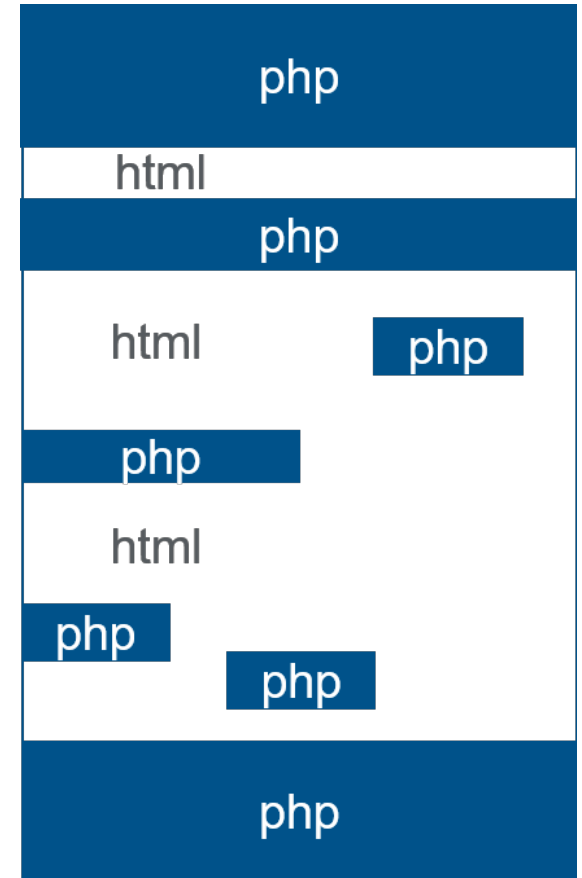


Der Programming Community Index von Tiobe Software listet die Popularität von Programmiersprachen anhand der im jeweiligen Kontext gemessenen Aktivitäten in Suchmaschinen wie Google, Bing und Yahoo.

Foto: Tiobe Software

# PHP-Grundlagen

- **P**ersonal **H**ome **P**age Tools oder **H**ypertext **P**reprocessor
- PHP ist eine serverseitige, in HTML eingebettete Scriptsprache
- Sie ist beeinflusst von C, Java und Perl
- Mit PHP und Datenbankunterstützung lassen sich Webseiten oder Webanwendungen dynamisch gestalten
- Erweitert um Objektorientierte Programmierung
- Um viele nützliche Bibliotheken erweiterbar
- Erweiterbar um diverse Widget-Sammlungen
- PHP-Programme werden ohne Umwege interpretiert



# Erste Schritte

## Grundaufbau/Kommentare/Funktionen/Ausgabe

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8" />
  <title>Mein erstes PHP-Script</title>
</head>
```

```
<body>
  <h1>Herzlich Willkommen</h1>
  <p>Dies ist meine erste PHP-Datei. Eine Mischung aus HTML und php. <br>
  Ausgabe in Farbe sind die Ausführungen der php-Blöcke<br>
```

```
<h2 style="color:blue;">
```

```
<?php
```

```
# php-Block
```

```
echo "Mittels echo können Daten ausgegeben werden";
```

```
?> </h2></p>
```

```
<p> Ein einfaches Beispiel aktuelles Datum auszugeben. <br>
```

```
<h2 style="color:green;">
```

```
<?php
```

```
// weiterer php-Block
```

```
echo "Datum: ". date("d.m.Y H:i:s");
```

```
/* Das Datum kann in unterschiedlichen Formaten ausgegeben werden. siehe Dokumentation
```

```
y = Jahr zweistellig 97, 08, Y = Jahr vierstellig 1997, 2008,
```

```
H = Stunden im 24-Stunden Format, zweistellig 03,20; G = Stunden im 24-Stunden-Format, einstellig 3, 20
```

```
*/
```

```
?>
```

```
</h2></p>
```

```
</body>
```

```
</html>
```

### Herzlich Willkommen

Dies ist meine erste PHP-Datei.

Einfache Textausgabe:

### Mittels echo können Daten ausgegeben werden

Ein einfaches Beispiel aktuelles Datum auszugeben:

**Datum: 16.03.2021 11:21:32**

php-Block

Kommentare

Funktionen

# Erste Schritte

- Variablen
- Rechenoperationen
- Anhängen (concat)
- Inkrementieren

## Variablen

Das 1.Ergebnis lautet: 38.5

Das 2.Ergebnis lautet: 12.5

Das 3.Ergebnis lautet: 13.5

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8" />
    <title>Mein erstes PHP-Script</title>
</head>
<body>
    <h1>Variablen</h1>
    <?php
        $ausgabertext="Ergebnis "; /* ein String*/
        $zahl1 = 7; /* eine ganze Zahl */
        $zahl2 = 5.5; /* eine Fließkommazahl */
        $ergebnis = $zahl1 * $zahl2; /* Multiplikation */
        $text= "Das 1.". $ausgabertext. "lautet: ". $ergebnis;
        echo $text; echo "<br /><br />";
        $ergebnis = $zahl1 + $zahl2; /* Addition */
        $text= "Das 2.". $ausgabertext. "lautet: ". $ergebnis;
        echo $text; echo "<br /><br />";
        /* Inkrementieren */
        $ergebnis++;
        $text= "Das 3.". $ausgabertext. "lautet: ". $ergebnis;
        echo $text;
    ?>
</body>
</html>
```



# Erste Schritte

- Integer: Eine Integer-Variable enthält nur Ganzzahlen, d. h. Zahlen ohne Komma.
- String: Dies ist eine Variable, die einen Text/Satz/Wort enthält.
- Float: Eine Fließkommazahl, d. h. eine Zahl mit Komma. Man verwendet aber die englische Schreibweise und somit einen Punkt statt dem deutschen Komma.
- Double: In PHP das gleiche wie float.
- bool: Dies sind die Werte true und false

# Operatoren

Operatoren	Bedeutung
+, -, *, /, %	Rechenoperatoren
%	bedeutet Modulo
++	bedeutet Addition um 1
--	bedeutet Subtraktion um 1
+=	bedeutet Addition mit sich selbst
*=	bedeutet Multiplikation mit sich selbst
.	bedeutet Addition einer Zeichenkette (hinten anhängen)
.=	bedeutet Voranstellen einer Zeichenkette
and, &&, or,   , not, !	Logische Operatoren
=, ==, ===, !=, !==, <=, >=, <, >	Vergleichsoperatoren

# Operatoren ( ==, ===)

```
<?php
    $zahl1=1;           /* ein Integer */
    $text="1";          /* ein String  */

    var_dump($zahl1);
    var_dump($text);
    echo "<p></p>";
    /* Die Funktion var_dump() gibt strukturierte Informationen über einen oder mehrere Ausdrücke aus,
    darunter auch den entsprechenden Typ und Wert. Arrays und Objekte werden rekursiv durchlaufen und die
    jeweiligen Werte eingerückt dargestellt, um die Struktur zu verdeutlichen. */

    if ($zahl1==$text){
        echo "Werte sind gleich, aber Typ nicht";
    }
    echo "<p></p>";

    if ($zahl1 === $text){ /* Neben dem reinen Wert wird auch der Typ verglichen */
        echo "zahl ist gleich text";
    } else {
        echo " === nicht gleicher Typ";
    }
    echo "<p></p>";

    $zahl2=1;
    var_dump($zahl1);
    var_dump($zahl2);
    echo "<p></p>";
    if ($zahl1=== $zahl2){
        echo "\nzahl1 ist gleich zahl2 und gleicher Typ";
    }
?>
```

## Typ-sicher?

int(1) string(1) "1"

Werte sind gleich, aber Typ nicht

=== nicht gleicher Typ

int(1) int(1)

zahl1 ist gleich zahl2 und gleicher Typ

# Schleifen

## Kopfgesteuerte Schleife

### while-Schleife

```
$i = 0;
while ($i <= 3)
{
    echo "Ausgabe: $a [
i ] "; i++;
}
```

## Fußgesteuerte Schleife

### do-while-Schleife

```
$i = 0;
do
{
    echo Ausgabe: $a [ i
]; i++;
} while ($i <= 3)
```

## Zähl-Schleife

### for-Schleife

```
for ($i = 0; i <= 3; i++)
{
    echo "Ausgabe: $i
";
}
```

## Schleife für jedes Element in einer Liste

### Foreach-Schleife

```
foreach ($a as $i)
{
    echo Ausgabe: $a [ i
];
}
```

# Bedingungen

## Syntax

```
if (Ausdruck)
{
    Anweisung;
}
```

```
if (Ausdruck)
{
    Anweisung;
}
else
{
    Anweisung;
}
```

## Syntax

```
(Ausdruck) ?
    Anweisung1 :
    Anweisung2 ;
```

```
if (Ausdruck1)
{
    Anweisung1;
}
elseif (Ausdruck2)
{
    Anweisung2;
}
else
{
    Anweisung3;
}
```

## Syntax

```
switch (Ausdruck)
{
    case Konstante1:
        Anweisung;
        break;
    case Konstante2:
        Anweisung;
        break;
}
```

# Wichtigste Funktionen

Hilfreich:

<https://www.php-einfach.de/php-tutorial/die-wichtigsten-php-funktionen/>

<https://www.php-kurs.com/funktionen-von-php.htm>

- **Arrayfunktionen**  
(Sortierung, Anzahl Elemente, etc.)
- **Dateifunktionen**  
(Dateigröße, lesbar, beschreibbar, löschen etc.)
- **Datum / Uhrzeitfunktionen**  
(Formatierung, Aktuelles Datum/Zeit etc.)
- **Mathematische Funktionen**  
(Absolutwert, Maximum, Zufallszahl, Runden, etc.)
- **MySQL-Funktionen**  
(DB-Verbindung erzeugen, SQL-Befehl ausführen, durch Ergebnis iterieren, etc.)
- **Stringfunktionen**  
(Replace, Länge des Strings, Zeichensuche, etc.)
- **Sonstige Funktionen**  
(Datei einbinden, mail versenden, Session starten, etc.)

# Eigene Funktionen schreiben und aufrufen

## Syntax

```
function ausgabe()  
{  
    echo "Es wird etwas  
ausgegeben";  
}
```

```
function rechne($a,$b)  
{  
    $c = $a + $b  
    echo "Ergebnis: $c";  
}
```

```
function rechne($a,$b)  
{  
    $c = $a + $b  
    return $c;  
}
```

## Aufruf

```
ausgabe();
```

```
rechne(2,4)
```

```
$a=100;  
$b=150;  
$d = rechne($a,$b)  
echo "Ergebnis: $d";
```

```
echo "Ergebnis: ",  
rechne(2,4);
```

# Formularauswertung

**\$\_POST/\$\_GET**

```
1 <?php
2 echo "eingegebener Vorname: ". $_POST['vorname'];
3 echo " ";
4 echo "eingegebener Nachname: ". $_POST['nachname'];
5 ?>
6
7
8 <form action="sendenPOSThtmlphp.php" method="post">
9   <table>
10     <tr>
11       <td> Vorname: </td>
12       <td> <input type="text" name="vorname" /> </td>
13     </tr>
14     <tr>
15       <td> Nachname: </td>
16       <td> <input type="text" name="nachname" /> </td>
17     </tr>
18     <tr>
19       <td colspan="2"> <input type="submit" value="Senden" /> </td>
20     </tr>
21   </table>
22 </form>
23
```

localhost/test/sendenPOSThtmlphp.php

Lernplattform der GFN... Bundeseinheitliche Pr... EUROPATHEK | Europa... Faktoren | Lernen sicht... EBooks – OneDrive Binary Clock - Online ...

**Notice:** Undefined index: vorname in C:\xampp\htdocs\test\sendenPOSThtmlphp.php on line 2  
eingegebener Vorname:  
**Notice:** Undefined index: nachname in C:\xampp\htdocs\test\sendenPOSThtmlphp.php on line 4  
eingegebener Nachname:

Vorname:

Nachname:

localhost/test/sendenPOSThtmlphp.php

Lernplattform der GFN... Bundeseinheitliche Pr... EUROPATHEK | Europa... Faktoren

eingegebener Vorname: Klara eingegebener Nachname: Fall

Vorname:

Nachname:

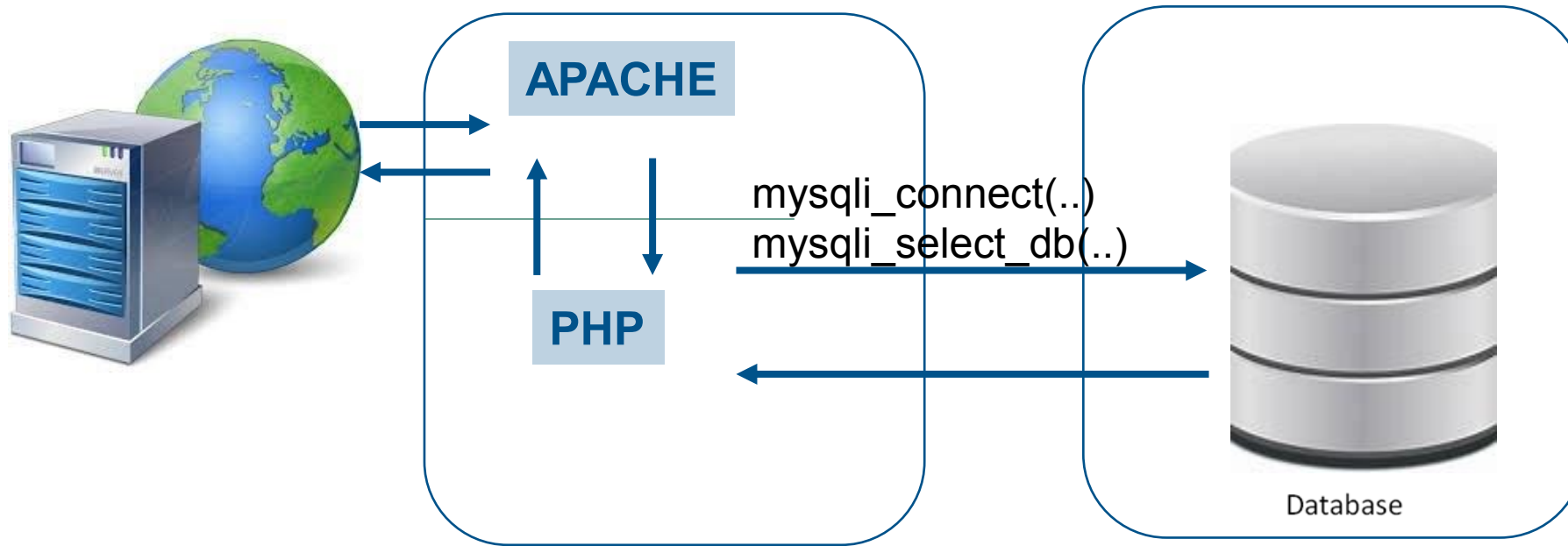
view-source:http://localhost/test/sendenPOSThtmlphp.php

Lernplattform der GFN... Bundeseinheitliche Pr... EUROPATHEK | Europa... Faktoren | Lernen sicht...

```
1 eingegebener Vorname: Klara eingegebener Nachname: Fall
2
3 <form action="sendenPOSThtmlphp.php" method="post">
4   <table>
5     <tr>
6       <td> Vorname: </td>
7       <td> <input type="text" name="vorname" /> </td>
8     </tr>
9     <tr>
10      <td> Nachname: </td>
11      <td> <input type="text" name="nachname" /> </td>
12    </tr>
13    <tr>
14      <td colspan="2"> <input type="submit" value="Senden" /> </td>
15    </tr>
16  </table>
17 </form>
```



# PHP und Datenbanken



**Connect DB-Server→**

**\$dbserver:**

**localhost (127.0.0.1)**

**\$dbuser:**

**root**

**\$dbpassword:**

**Connect DB**

**\$dbname:**

**adressbuch**

# PHP und MySQL

- APIs
  - mysql („Deprecated“ seit PHP 5.5.0)
  - mysqli (i = „improved“)
  - PDO (PHP Data Objects)
- mysql, um mit älteren MySQL Versionen zu arbeiten (< 4.1.3)
- Empfehlung, mysqli oder PDO zu verwenden

# mysqli

- Bietet zwei Schnittstellen an
  - Prozedural–Objektorientiert
- Unterstützt „prepared“ Statements
- Unterstützt mehrere Statements
- Unterstützt Transaktionen
- Verbesserte Debugging-Möglichkeiten

# mysqli - prozedural

## Verbindung aufbauen

```
$c = mysqli_connect("localhost", "user", "password", "mydb");
```

## Datenbank wechseln

```
mysqli_select_db("mydb");
```

## PHP-Befehl für MySQL-Query

```
$results = mysqli_query($c, $query);
```

## Ergebnisse verarbeiten

```
mysqli_fetch_array($result);  
mysqli_fetch_array($result, MYSQLI_NUM);  
mysqli_fetch_array($result, MYSQLI_ASSOC);
```

## Verbindung beenden

```
mysqli_close($c);
```

# OO in PHP – Klasse – Einbinden einer Klasse

classlernfeld.php

```
<?php
class classLernfeld{
    var $name="Lernfeld 10";
    var $beschreibung="10a Benutzerschnittstellen gestalten und entwickeln";
    var $anzahlUE=135;
    var $anzahlTage=15;
    var $klausur="Freitag: 10:30 - 12:00 Uhr";

    function setKlausur($klausur){
        $this->klausur=$klausur;
    }

    function getKlausur(){
        return $this->klausur;
    }
}
```

?>

Attribute

Methoden

# OO in PHP – Klasse – Einbinden einer Klasse

lernfeld.php

```
<?php

require_once("classLernfeld.php");
$lf= new classLernfeld();

echo "<pre>Name:          ".$lf->name."</br></pre>";
echo "<pre>Beschreibung: ".$lf->beschreibung."</br></pre>";
echo "<pre>Anzahl UEs:     ".$lf->anzahlUE."</br></pre>";
echo "<pre>Anzahl Tage:    ".$lf->anzahlTage."</br></pre>";

$lf->setKlausur("Donnerstag: 11-12 Uhr");
echo "<pre style=\"color:blue; font-weight:bold\" >Klausur:      ".$lf->getKlausur(). "</br></pre>";

?>
```

Ausgabe

```
Name:          Lernfeld 10

Beschreibung:  10a Benutzerschnittstellen gestalten und entwickeln

Anzahl UEs:    135

Anzahl Tage:   15

Klausur:      Donnerstag: 11-12 Uhr
```

# mysqli - objektorientiert

## Verbindung aufbauen

```
$c = new mysqli("localhost", "user", "password", "mydb");
```

## PHP-Befehl für MySQL-Query

```
$results = $c->query($query);
```

## Ergebnisse verarbeiten

```
$results->fetch_assoc();  
$results->fetch_row();  
  
$results->fetch_all(MYSQLI_BOTH);  
$results->fetch_all(MYSQLI_ASSOC);  
$results->fetch_all(MYSQLI_NUM);
```

## Verbindung beenden

```
$c->close();
```

# Datenbankverbindung herstellen – als Funktion

## Utility für DB-Zugriff

```
function new_db_connect ($database, $user, $pass) {  
    $host = "127.0.0.1";  
    $connID = new mysqli($host, $user, $pass, $database);  
    return $connID;  
}
```

## Verwendung des Utilities

```
include ("util.inc.php");  
  
$conn = new_db_connect ("mydb", "root", "");
```



# Formulardaten auslesen – DB-Zugriff

## auslesen

```
// E-Mail-Adresse aus dem Formular auslesen  
$mail = $_POST["mail"]
```

## DB-Zugriff mit Formularwert

```
// E-Mail-Adresse aus der Datenbank heraussuchen  
$sqlstmt = "SELECT email FROM kunden WHERE email=\"{$mail}\"";  
$query = $conn->query ($sqlstmt);  
// Existiert ein Eintrag?  
if ($query->num_rows > 0) {  
    ....  
}
```

# Formulardaten auslesen – DB-Zugriff

## auslesen

```
// E-Mail-Adresse aus dem Formular auslesen  
$mail = $_POST["mail"]
```

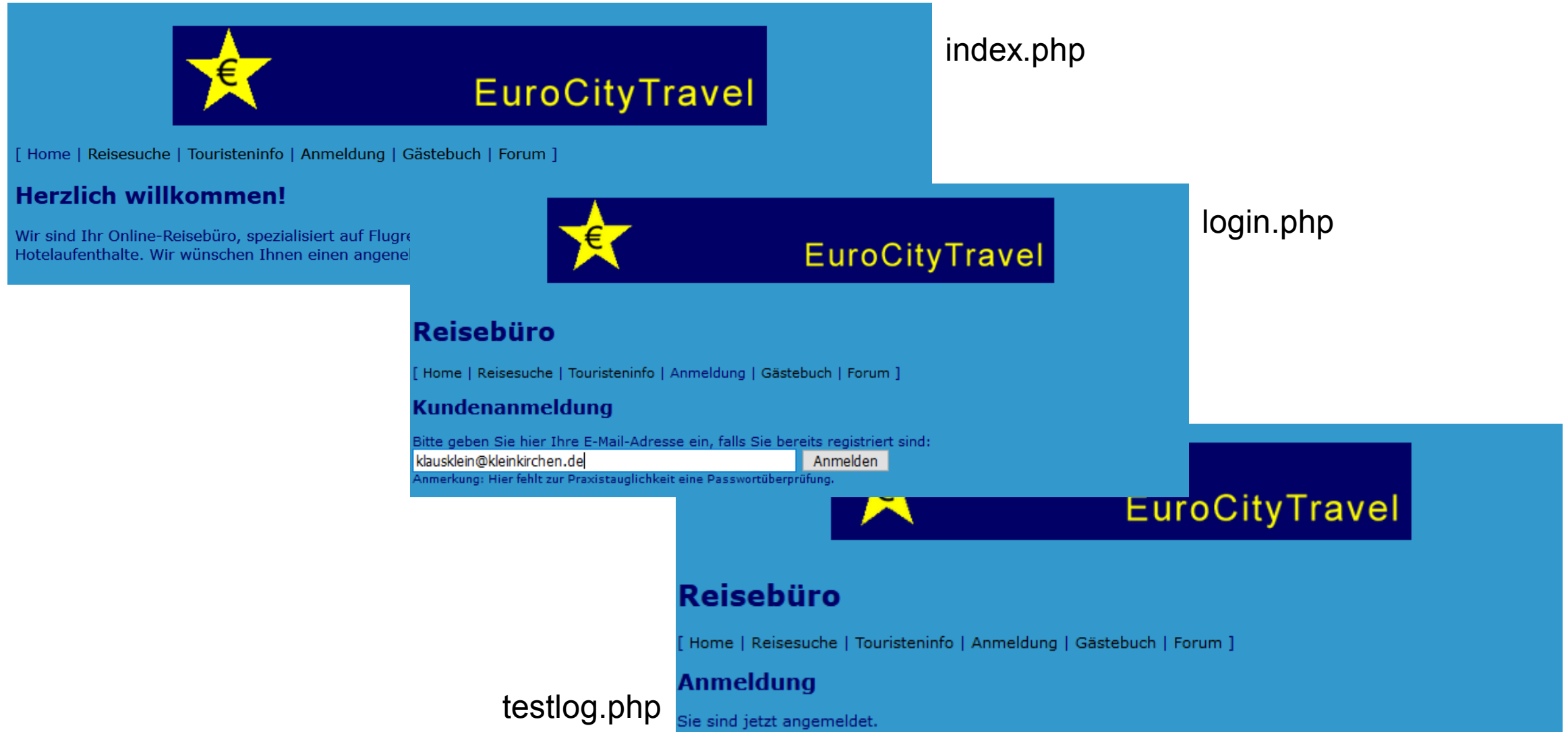
## DB-Zugriff mit Formularwert

```
// E-Mail-Adresse aus der Datenbank heraussuchen  
$sqlstmt = "SELECT email FROM kunden WHERE email=\"{$mail}\"";  
$query = $conn->query ($sqlstmt);  
  
if ($conn->affected_rows == 1)
```

# Aufgabe

- Mit PHP Daten aus den Formularen verarbeiten
- login.php - Anmeldung Datenbankbindung implementieren
- Funktion für den Datenbankzugriff implementieren (util.inc.php)
- Weitere HTML-Seite ausgeben nach Betätigung des Anmelden-Buttons (testlog.php)
- Siehe ff. Seiten und Aufgabenbeschreibung in Word

# Aufgabe



# Aufgabe

- auskunft.php (Reisesuche, Reisauswahl)

**EuroCityTravel**

## Reisebüro

[ [Home](#) | [Reisesuche](#) | [Touristeninfo](#) | [Anmeldung](#) | [Gästebuch](#) | [Forum](#) ]

### Reiseauswahl

Abreiseort:    
Die meisten Flugangebote gehen von/nach Köln/Bonn, einige auch von/nach Frankfurt oder Düsseldorf.

Zielort:  

Abreisedatum: Datum:      

Rückreisedatum: Datum:      

Mit Hotelangebot? ☒ ja ☐ nein



37

# Aufgabe

- ergebnis.php  
(gefundene Angebote  
Flüge, Hotels etc.)

EuroCityTravel

## Reisebüro

[ Home | Reisesuche | Touristeninfo | Anmeldung | Gästebuch | Forum ]

### Gefundene Angebote

#### Hinflüge

☒ **AF1419** Frankfurt Airport (FRA) - Paris / Roissy (CDG): 2021-03-19, 10:40:00 - 100 €

#### Rückflüge

☐ ohne Rückflug  
☐ **AF1418** Paris / Roissy (CDG) - Frankfurt Airport (FRA): 2021-03-21, 07:35:00 - 100 €

Flugbuchung für  Person(en)

#### Hotelangebote

Buchen	Hotelname	Badtyp	Verpflegung	Einzelzimmer / Nacht	Doppelzimmer / Nacht
<input type="radio"/>	kein Hotel buchen				
<input type="radio"/>	Hotel de la Gare	Dusche	Frühstück	85 €	150 €
<input checked="" type="radio"/>	Hotel au Jardin	Bad	Frühstück	110 €	200 €

Buchen:  Einzelzimmer und  Doppelzimmer

Reisedaten ändern

# Aufgabe

- buchung.php  
(Buchungsbestätigung)

**EuroCityTravel**

## Reisebüro

[ [Home](#) | [Reisesuche](#) | [Touristeninfo](#) | [Anmeldung](#) | [Gästebuch](#) | [Forum](#) ]

## Buchungsbestätigung

Bitte bestätigen Sie die folgende Buchung:

### Hinflug

2021-03-19  
**Air France** Flug Nr. **AF1419**  
von **Frankfurt Airport** (FRA) nach **Paris / Roissy** (CDG)  
2 Personen  
Einzelpreis: **100,00 EUR**; Gesamtpreis: **200,00 EUR**

### Gesamtbetrag

**200,00 EUR** incl. 16% Mehrwertsteuer

[Buchung bestätigen](#) [Buchung abbrechen](#)

# Aufgabe

- buchabschl.php  
(Buchungsabschluss mit Bestätigung)





# Aufbau von URLs

`https://www.meineseite.de:80/seite?query#wert`

| | | | |

| | | | +--- Hash, Sprungziel

| | | |

| | | #--- Query String

| | | (PHP oder CGI-Programme)

| | |

| | +--- Unix-Pfadname

| |

| |

| | +--- TCP-Portnummer, kann fast immer  
weggelassen werden

| |

| | +--- Host / Domainname

|

+--- Protokoll (entweder http:// oder https://)

# Aufgabe

- 13 gegebene html-, php-Seiten auf objektorientierten Datenbankzugriff umstellen

# Aufgabe

## Übungen:

- Optional:  
Datenbank-Gewinnspiel in Reisebüro-Website integrieren  
(siehe Aufgabenbeschreibung in Word)
- Datenbank-Gewinnspiel um zeitlich begrenzte Spiele erweitern
- ERM erstellen
- RM ableiten
- Tabellen anlegen bzw. erweitern
- Website entsprechend anpassen
- Webseite für die Verwaltung erstellen

# Typische Fehler

- Component-Tags nicht geschlossen
- Wertzuweisung an eine CSS-Eigenschaft anstelle des Doppelpunktes ein = zu schreiben
- ; vergessen als Trennzeichen der CSS-Eigenschaften
- Geschweifte Klammer nicht geschlossen
- Groß- und Kleinschreibung bei Selektoren
- Kommentare nicht geschlossen

# Webseite „live“ schalten

- Anmelden beim Provider
- Dateien verschieben mit FTP
- Benutzungsrechte ändern
- Terminalsitzung?
  - Sichere Terminalsitzung mit SSH (Secure Shell)
  - Sichere Terminalsitzung mit Putty

# 10a Kompetenzcheck



Welche Aussagen sind wahr?

- a) Ein php-Block wird immer durch `<php> </php>` umschlossen
- b) Variablen werden in PHP mit `$` deklariert
- c) PHP steht für *Hypertext Preprocessor*
- d) Bei Verwendung der POST-Methode sind die Variablen in der URL zu sehen
- e) Funktionen werden folgendermaßen deklariert: *create myFunction()*
- f) Kommentare werden nur mit `/** */` versehen
- g) Include-files müssen die Endung `.inc` haben
- h) PHP wird serverseitig ausgeführt
- i) PHP kann mit Referenzen umgehen
- j) PHP unterstützt die objektorientierte Programmierung