



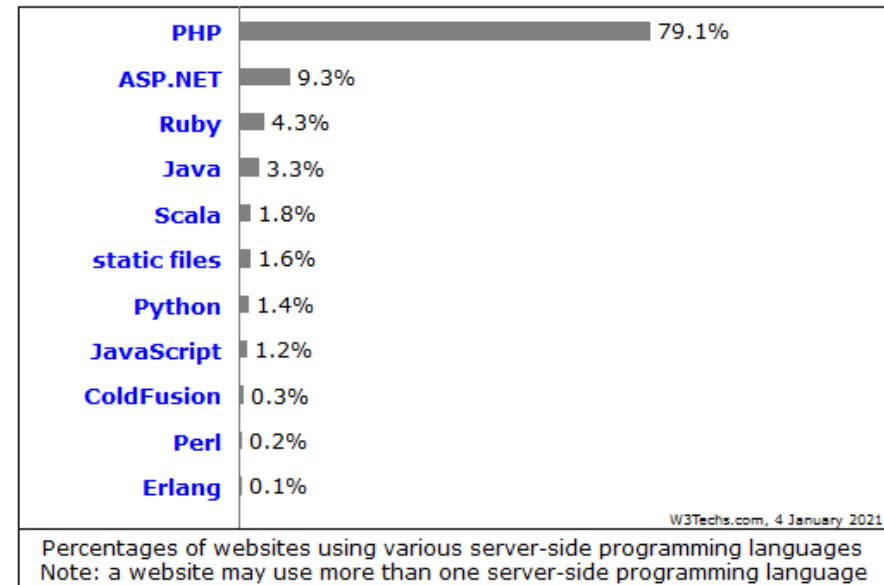
PHP 01



EINFÜHRUNG

PHP – WAS IST DAS

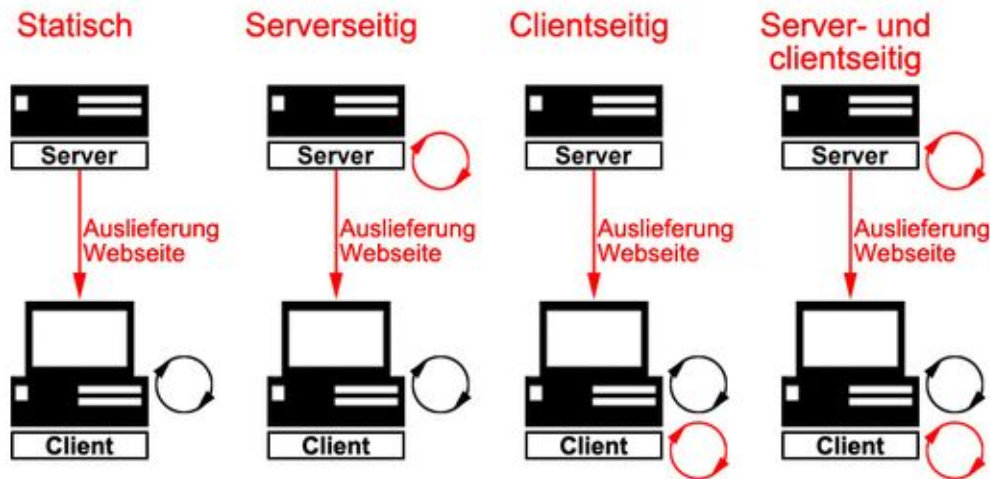
- World Wide Web entstand 1989 mit einfachen HTML-Seiten
- Erste Version 1995 vorgestellt
 - ▶ Rasmus Lerdorf hatte hierfür einige Skripte in Perl erstellt
 - ▶ Ausgangspunkt: Zugriffe auf seinen Online-Lebenslauf protokollieren
 - ▶ Daraus entstand umfangreiche Skriptsammlung die er für Gestaltung seiner persönlichen Homepage nutzte
 - ▶ Nannte es: Personal Home Page Tools – PHP
 - ▶ Außer dem Namen ist nicht mehr viel übrig
- Nächste Version und folgende in C entwickelt
- 1997 änderte sich Entwicklerteam mit Hauptverantwortliche: Andi Gutmans und Zeev Suraski
 - ▶ PHP neue Bedeutung: Hypertext Preprocessor
- Großer Durchbruch mit PHP 4
- Mit PHP 5 wurde PHP eine objektorientierte Programmiersprache
- PHP auf fast allen Webservern vorinstalliert
- PHP für Erstellung dynamischer Webseiten
- https://w3techs.com/technologies/overview/programming_language



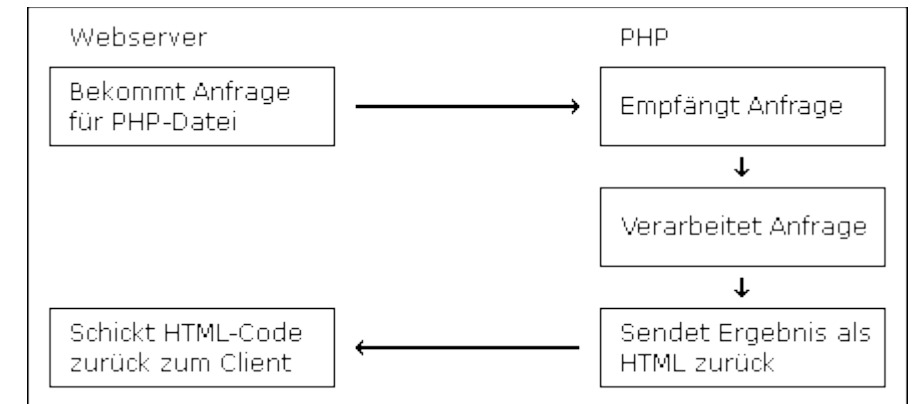
STATISCHE UND DYNAMISCHE WEBSEITEN

➤ Reine HTML-Seiten = statische Seiten

Verarbeitung



<https://images.app.goo.gl/CQ9H5aGpnHyv3mZ6A>



<https://images.app.goo.gl/wKs7T4xFTMxHnE9d9>

ANWENDUNGSMÖGLICHKEITEN FÜR PHP

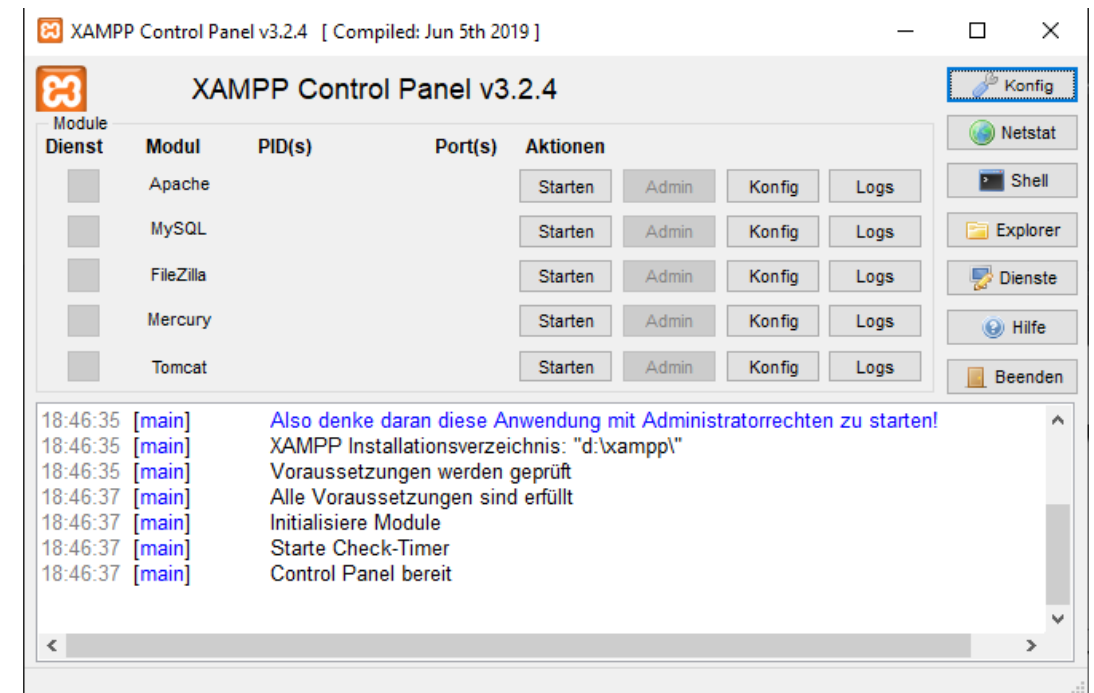
- Content Management Systeme
 - ▶ Wordpress
 - ▶ Serendipity
 - ▶ Joomla
 - ▶ Typo3
 - ▶ Drupal
- Shopsysteme
 - ▶ Magento
- Foren-Software
 - ▶ phpBB
- Customer Relationship Management Systeme
 - ▶ SugarCRM
- Und viele viele mehr



VORBEREITUNGSMAßNAHMEN FÜR DAS PROGRAMMIEREN MIT PHP

WEBSERVER-SOFTWARE FÜR DIE AUSFÜHRUNG EINES PHP-PROGRAMMS

- PHP Programme sind Skripte
- Code wird nicht kompiliert => Interpreter notwendig
- Liste von Texteditoren: https://de.wikipedia.org/wiki/Liste_von_Texteditoren
- XAMPP
 - ▶ Dateipfad (meistens): C:\xampp\htdocs\meinEigenerOrdnerFuerDieUebungen
 - ▶ Browser URL: localhost/meinEigenerOrdnerFuerDieUebungen/meinePHPDatei.php





DAS ERSTE PROGRAMM MIT PHP GESTALTEN

DAS ERSTE PROGRAMM SCHREIBEN

- PHP wird geöffnet mit `<?php` und geschlossen mit `?>`
- Dazwischen ist der Programmcode
- `print` erzeugt eine Ausgabe am Bildschirm
- `echo` erzeugt ebenfalls eine Ausgabe am Bildschirm
 - ▶ Kurzschreibweise: `<?=$foo?>`
- Unterschiede
 - ▶ `echo` hat keinen Rückgabewert, während `print` einen Rückgabewert von 1 hat, sodass es in expressions verwendet werden kann. `Echo` kann mehrere Parameter annehmen (obwohl eine solche Verwendung selten ist), während `print` ein Argument verwendet werden kann. `Echo` ist geringfügig schneller als `print`.

```
1 <!doctype html>
2 <html lang="de">
3 <head>
4     <meta charset="UTF-8">
5     <title>Mein erstes PHP Programm</title>
6 </head>
7 <body>
8     <p>mein erstes Programm</p>
9     <p>255</p>
10    <p>500</p>
11    <p>1850</p>
12 </body>
13 </html>
```

```
<!doctype html>
<html lang="de">
<head>
    <meta charset="UTF-8">
    <title>Mein erstes PHP Programm</title>
</head>
<body>
    <p><?php
        // Zeichenkette
        echo "mein erstes Programm";
    ?></p>
    <p><?php
        // Zahl
        print 255;
    ?></p>
    <p><?php
        // Zahl als String
        echo "500";
    ?></p>
    <p><?php
        // Rechnen
        echo 50*37;
    ?></p>
</body>
</html>
```

ÜBUNG

1. Erstelle ein Programm, das die Besucher deiner Seite begrüßt
2. Zeige drei verschiedene Möglichkeiten auf, um die Zahl 8 als Ausgabe eines PHP-Programms darzustellen und erkläre die Unterschiede unter jeder Ausgabe im HTML File



PHP UND HTML MITEINANDER VERBINDEN

HTML-CODE MIT PHP AUSLESEN

```
<!doctype html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Übung</title>
</head>
<body>
<?php
echo "<h1>Willkommen</h1>";
?>
</body>
</html>
```

```
<!doctype html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Übung</title>
</head>
<body>
<?php
echo "<h1>Willkommen</h1>";
echo "<p>1. Absatz</p>";
echo "<p>2. Absatz</p>";
?>
</body>
</html>
```

```
1 <!doctype html>
2 <html>
3 <head>
4   <meta charset="UTF-8">
5   <title>Übung</title>
6 </head>
7 <body>
8 <h1>Willkommen</h1></body>
9 </html>
```

```
1 <!doctype html>
2 <html>
3 <head>
4   <meta charset="UTF-8">
5   <title>Übung</title>
6 </head>
7 <body>
8 <h1>Willkommen</h1><p>1. Absatz</p><p>2. Absatz</p></body>
9 </html>
```

```
<!doctype html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Übung</title>
</head>
<body>
<?php
echo "<h1>Willkommen</h1>\n";
echo "<p>1. Absatz</p>\n";
echo "<p>2. Absatz</p>\n";
?>
</body>
</html>
```

```
1 <!doctype html>
2 <html>
3 <head>
4   <meta charset="UTF-8">
5   <title>Übung</title>
6 </head>
7 <body>
8 <h1>Willkommen</h1>
9 <p>1. Absatz</p>
10 <p>2. Absatz</p>
11 </body>
12 </html>
```

➤ \n hat keine Auswirkung auf die Darstellung im Browser. Bewirkt aber einen Zeilenumbruch im Quellcode

PHP-SKRIPT IN EINE HTML SEITE INTEGRIEREN

```
<!doctype html>
<html>
<head>
    <?php
    echo "<meta charset=\"UTF-8\">\n";
    echo '<meta name="description" content="Beschreibung der Seite">';
    echo "\n";
    echo "<meta name='author' content='Beschreibung der Seite'>\n";
    ?>
    <title>Übung</title>
</head>
<body>
<h1>Willkommen</h1>
<?php
echo "<p>1. Absatz</p>\n";
echo "<p>2. Absatz</p>\n";
?>
</body>
</html>
```



VARIABLEN

TEXT MIT EINER VARIABLEN ERFASSEN

```
<?php
/* Variablen werden mit einem Dollarzeichen gekennzeichnet
   Variablennamen dürfen keine Sonderzeichen enthalten */
$textbaustein = "Meine erste Variable";
echo $textbaustein;
?>
```

ZAHLEN ALS VARIABLE ABSPEICHERN

```
<?php
/* Variablen können neben Strings auch numerische oder boolsche Werte besitzen */
$ganzeZahl = 3;
$kommaZahl = 5.39281;
$boolean = true;
echo "Ganze Zahl: " . $ganzeZahl . "<br>Kommazahl: " . $kommaZahl . "<br>Boolean: " . $boolean;
?>
```

Ganze Zahl: 3
Kommazahl: 5.39281
Boolean: 1

8 Ganze Zahl: 3
Kommazahl: 5.39281
Boolean: 1

- Zeichenketten und Variablen können in einem ausgegeben werden.
- In JS trennt man sie mit einem Plus (+)
- In PHP trennt man sie mit einem Punkt (.)

OPERATIONEN MIT VARIABLEN

- Additionen, Multiplikationen, Divisionen funktionieren im gleichen Schema wie in JS

```
<?php
$bestand = 5;
echo "<p>Verfügbare Artikel: " . $bestand . "</p>\n";
$bestand = $bestand - 1;
/* Kurzschreibweise: $bestand--; */
echo "<p>Verfügbare Artikel: " . $bestand . "</p>\n";
?>
```

Verfügbare Artikel: 5

Verfügbare Artikel: 4

```
8 <p>Verfügbare Artikel: 5</p>
9 <p>Verfügbare Artikel: 4</p>
```

NUMERISCHES ARRAY

```
<?php
/* Array deklarieren */
$produkt = array();
$produkt[0] = 1;
$produkt[1] = "Roggensemmel";
$produkt[2] = 0.79;
$produkt[3] = "Bioweckerl vom feinsten";
$produkt[4] = 50;
foreach($produkt as $info) {
    echo "<p>" . $info . "</p>\n";
}
?>
<hr>
<?php
/* oder: kürzere Schreibweise */
$produktAlternative = array(1, "Roggensemmel", 0.79, "Bioweckerl vom feinsten", 50);
foreach($produktAlternative as $info) {
    echo "<p>" . $info . "</p>\n";
}
?>
```

```
1
Roggensemmel
0.79
Bioweckerl vom feinsten
50


---


1
Roggensemmel
0.79
Bioweckerl vom feinsten
50
```

```
echo "<p>" . $produkt[4] . "</p>\n";
```

➤ Array Ausgabe einzelnen Wertes

ASSOZIATIVES ARRAY

```
<?php
/* Array deklarieren */
$produkt = array();
$produkt['Artikelnummer'] = 1;
$produkt['Produktname'] = "Roggensemmel";
$produkt['Preis'] = 0.79;
$produkt['Beschreibung'] = "Bioweckerl vom feinsten";
$produkt['Anzahl'] = 50;
foreach($produkt as $i => $info) {
    echo "<p>" . $i . ": " . $info . "</p>\n";
}
?>
<hr>
<?php
/* Alternative Schreibweise */
$produktAlternative = array(
    'Artikelnummer' => 1,
    'Produktname' => "Roggensemmel",
    'Preis' => 0.79,
    'Beschreibung' => "Bioweckerl vom feinsten",
    'Anzahl' => 50
);
foreach($produktAlternative as $i => $info) {
    echo "<p>" . $i . ": " . $info . "</p>\n";
}
?>
```

```
print "<p>" . $produktAlternative["Preis"] . "</p>\n";
```

➤ Array Ausgabe einzelnen Wertes

Artikelnummer: 1

Produktname: Roggensemmel

Preis: 0.79

Beschreibung: Bioweckerl vom feinsten

Anzahl: 50

Artikelnummer: 1

Produktname: Roggensemmel

Preis: 0.79

Beschreibung: Bioweckerl vom feinsten

Anzahl: 50

MEHRDIMENSIONALES ARRAY

```
<?php
/* Array deklarieren */
$sortiment = array();
$sortiment[0]['Artikelnummer'] = 1;
$sortiment[0]['Produktname'] = "Roggensemmel";
$sortiment[0]['Preis'] = 0.79;
$sortiment[1]['Artikelnummer'] = 2;
$sortiment[1]['Produktname'] = "Kornspitz";
$sortiment[1]['Preis'] = 0.59;
$sortiment[2]['Artikelnummer'] = 3;
$sortiment[2]['Produktname'] = "Semmel";
$sortiment[2]['Preis'] = 0.99;
/* Debugging mit var_dump um zu sehen was in dem Array drin ist */
var_dump($sortiment);
echo "<hr>";

/* Ausgabe einzelner Position */
echo "<p>" . $sortiment[2]['Produktname'] . "</p>\n";
echo "<hr>";

/* Ausgabe des gesamten Arrays */
foreach($sortiment as $array) {
    foreach($array as $i => $entry) {
        echo "<p>" . $i . ": " . $entry . "</p>\n";
    }
}
?>
```

```
array(3) { [0]=> array(3) { ["Artikelnummer"]=> int(1) ["Produktname"]=>
string(12) "Roggensemmel" ["Preis"]=> float(0.79) } [1]=> array(3) {
["Artikelnummer"]=> int(2) ["Produktname"]=> string(9) "Kornspitz"
["Preis"]=> float(0.59) } [2]=> array(3) { ["Artikelnummer"]=> int(3)
["Produktname"]=> string(6) "Semmel" ["Preis"]=> float(0.99) } }
```

Semmel

Artikelnummer: 1

Produktname: Roggensemmel

Preis: 0.79

Artikelnummer: 2

Produktname: Kornspitz

Preis: 0.59

Artikelnummer: 3

Produktname: Semmel

Preis: 0.99

ÜBUNG

- Erstelle ein Programm, das die Besucher deiner Seite begrüßt
- Zeige drei verschiedene Möglichkeiten auf, um die Zahl 8 als Ausgabe eines PHP-Programms darzustellen und erkläre die Unterschiede unter jeder Ausgabe im HTML File
- Schreibe ein Programm, das zwei numerische Variablen verwendet. Dieses soll die beiden Werte zunächst einzeln ausgeben. Anschließend soll es beide Variablen multiplizieren und das Ergebnis ausgeben.
- Schreibe ein Programm, das folgenden Liedtext ausgibt:
„Die Donau ist ins Wasser g'falln,
der Rheinstrom ist verbrannt,
In Frankfurt ist ein Spaß passiert,
der Geisbock hats erzählt“
Dabei soll jede Zeile in einer Variablen erfasst werden. Verwende für die Ausgabe nur einen einzigen print-Befehl
- Erstelle ein Array, das von drei Kunden jeweils die Kundennummer, den Vor- und den Nachnamen enthält. Gebe daraufhin die entsprechenden Werte auf dem Bildschirm aus. Dabei sollen die zusammengehörigen Werte für jeden Kunden jeweils in einer Zeile stehen.



ENDE

QUELLE: PHP & MYSQL FÜR EINSTEIGER
ISBN: 978-3-96645-009-6
KAPITEL 1 - 5