

**HOCHSCHULE  
HANNOVER**  
UNIVERSITY OF  
APPLIED SCIENCES  
AND ARTS

—  
*Fakultät III  
Medien, Information  
und Design*

# **Refugeeks / IntoCode**

## Dynamische Websites

**22.09.2020 / 01:**  
**Einführung**



## Current State

- Einführung
  - Client – Server – Grundlagen
  - Erste Schritte (index.php – Seite aufrufen – erste Ausgabe)



## Next Step I

Vars I

Was ist PHP?

Vars II

## Next Step II

Warum PHP?

HTTP-Request I

## Next Step III

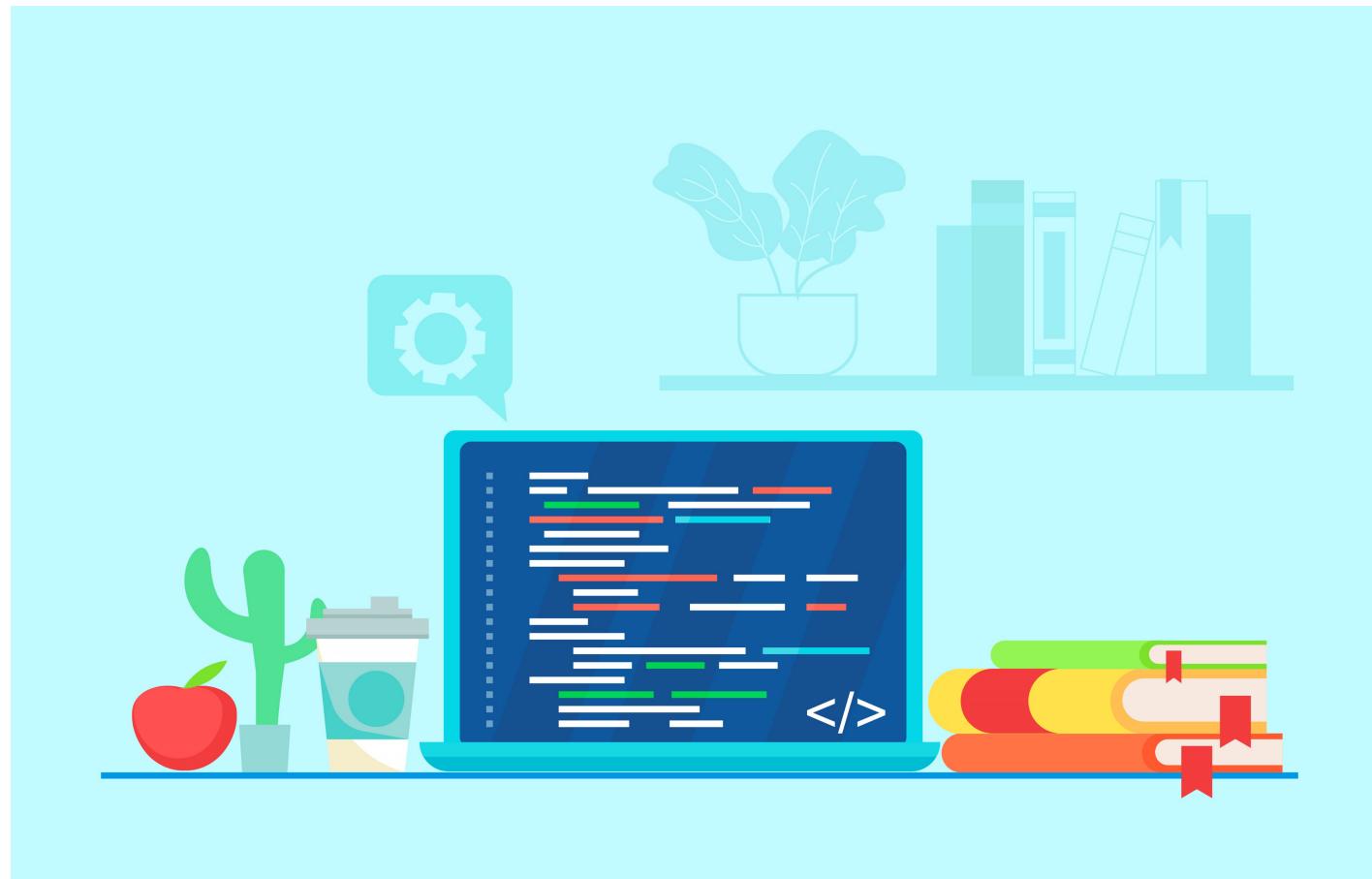
Was kann PHP?



designed by  freepik



# Dynamische Websites – Ausgaben & Variablen



H

# Dynamische Website

## PHP – Variablen - Datentypen

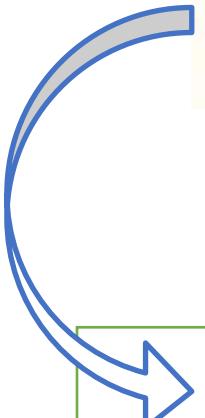
- Integer
- Fließkomma-Zahlen
- String
- Boolean
- Array



# Dynamische Website

## PHP – Variablen - String

```
$output = "Ein kurzer Text";
```

 \$ => Kennzeichnung (für eine Variable)  
 output => Variablenbezeichnung  
 = => Zuweisung  
 "irgendeinText" => String (der jetzt in der Variable ist)  
 ; => Beendet die Anweisung



# Dynamische Website

## PHP – Ausgabe – Variablen

```
$output = "Ein kurzer Text";  
echo $output;
```



```
echo  
$output  
;
```

- => Anweisung zur Ausgabe
- => Variable (die ausgegeben werden soll)
- => Beendet eine Anweisung



# Dynamische Website

## PHP – Client-Anzeige

Ein kurzer Text



# Dynamische Website

## PHP – Lang-Kommentar im Code

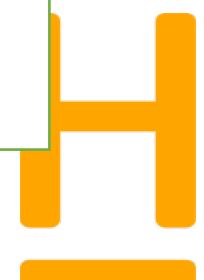
```
/*  
 * Verschiedene Datentypen  
 * String  
 * int and so on  
 */
```

Wird nicht mit an den Client geliefert



```
/*  
 * irgendeinText  
 * nochMehrText  
 */
```

=> Kommentar ab hier  
=> Kommentar Text  
=> Kommentar Text  
=> Kommentar beendet



# Dynamische Website

## PHP – Kurz-Kommentar im Code

Wird nicht mit an den Client geliefert

// String - Beispiele

//  
irgendeinText

=> Kommentar nur für eine Zeile  
=> Kommentar Text



# Dynamische Website

## PHP – Variablen – Konkatenation (Zusammenführung von Zeichenketten)

```
$output = "Ein kurzer Text";
```

```
$anotherOutput = "noch ein kurzer Text";
```

```
echo $output." zusammen mit ".$anotherOutput;
```



# Dynamische Website

## PHP – Client-Anzeige

Ein kurzer Text zusammen mit noch ein kurzer Text



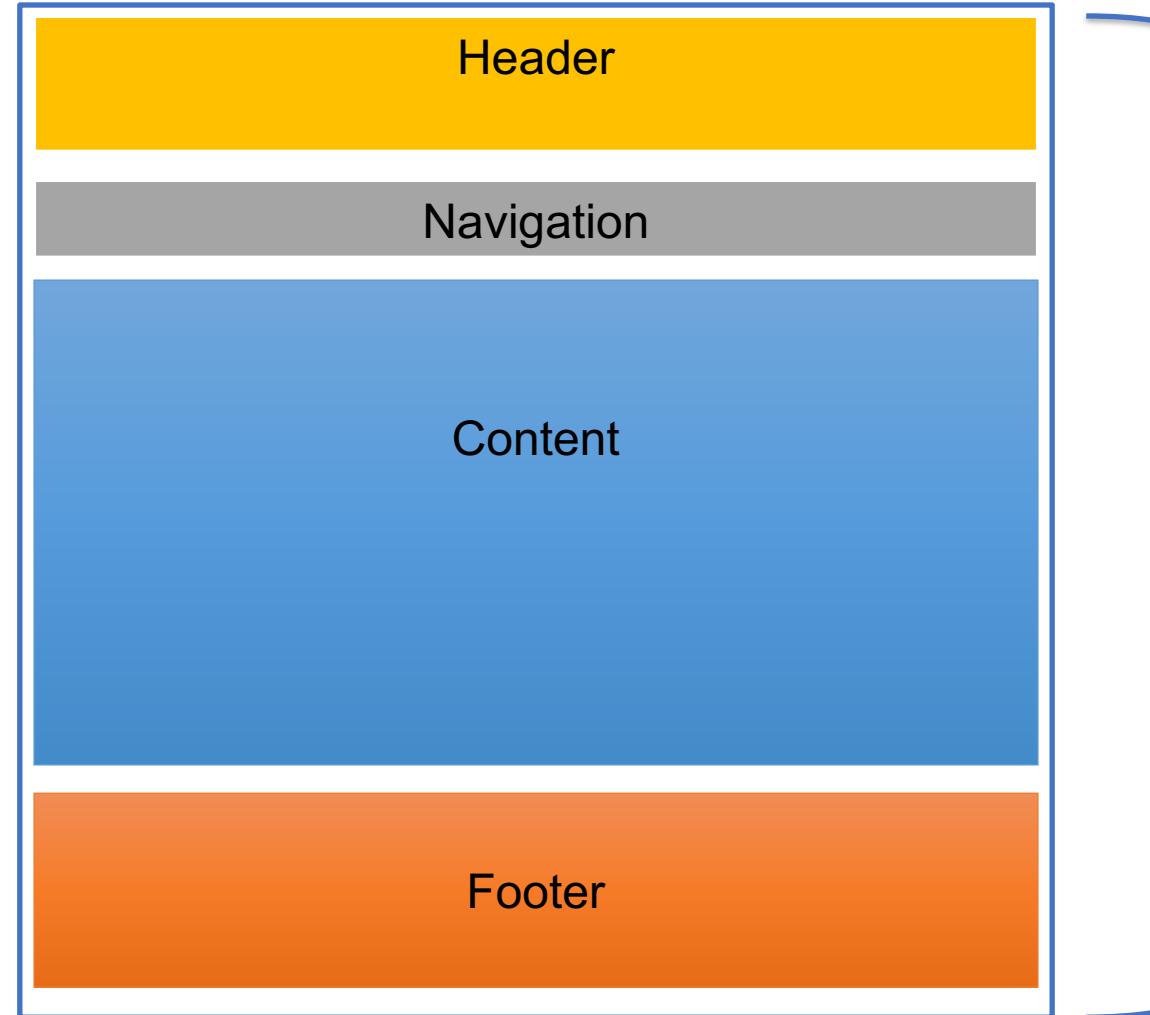
# Aufgabe 1

1. Erweitere in der index.php das HTML Gerüst folgendermaßen
2. Verlinke ein Stylesheet

Verwende die Tags

- <header>
- <nav>
- <section>
- <footer>

- Ordne die Container richtig an
- Setze borders (2px solid black)



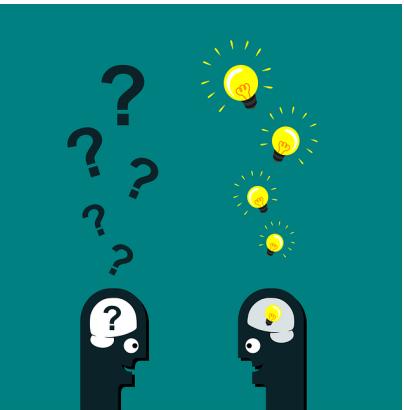
<body>

H

# Dynamische Websites – Was ist PHP?



# Dynamische Websites –Was ist PHP?



PHP – rekursives Akronym für: Hypertext Preprocessor

... weit verbreitet

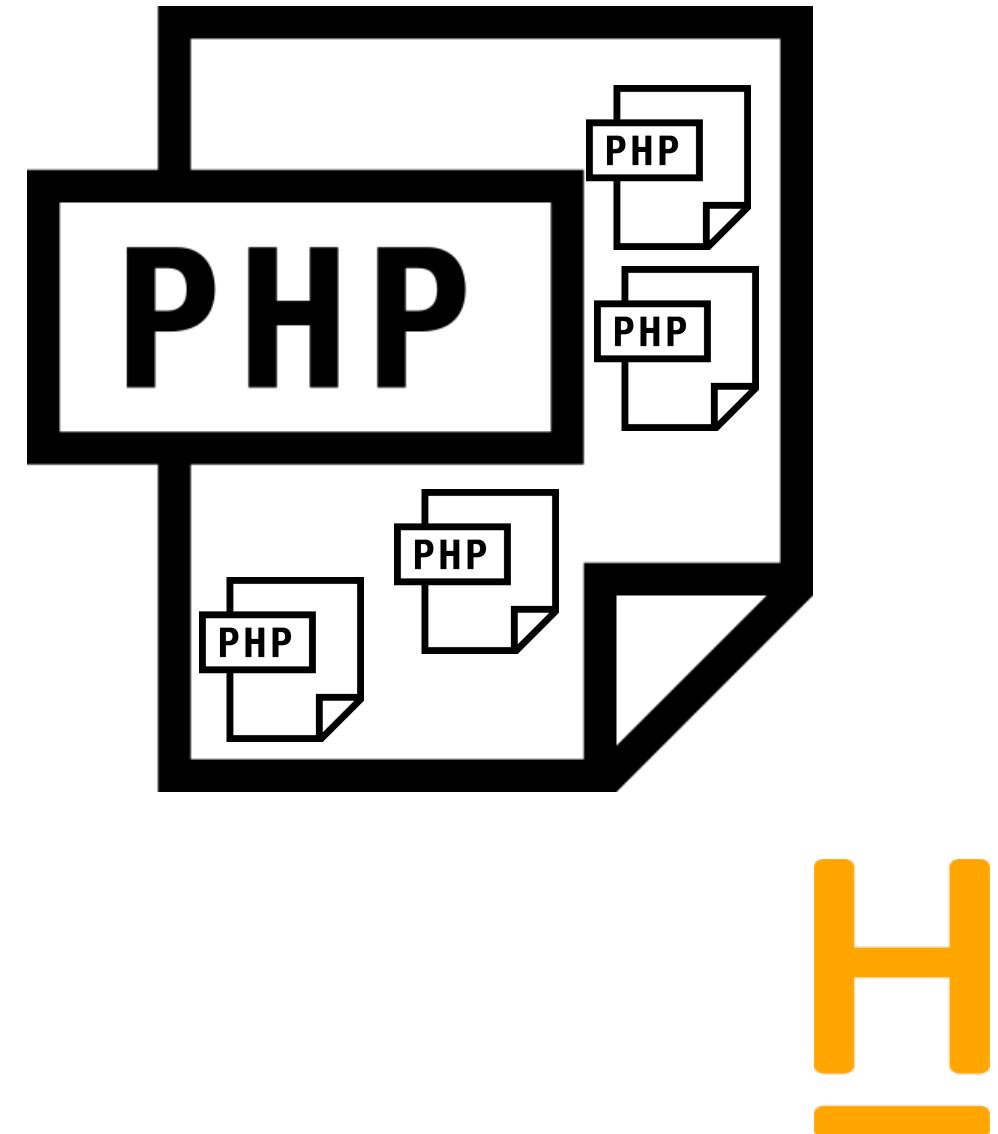
... eine Open-Source-Skriptsprache

... leicht in HTML zu integrieren

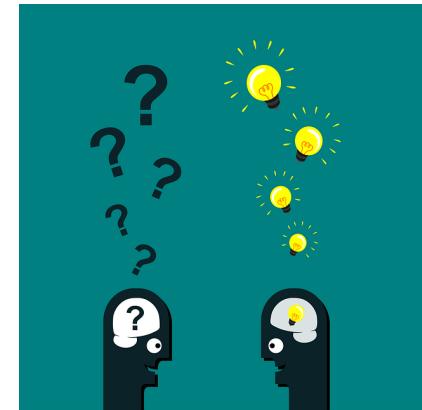


# PHP Seite

```
<html>
<head>
<title>k0dy hostSite</title>
<link href="style.css" rel="stylesheet">
</head>
<body style="background-color:<?php echo $backgroundColor; ?>">
    <header> Hi ! W&amp;uuml;hle deine Aufgabe ! </header>
    <nav>
        <ul>
            <li><a href="?content=output"><strong> Ausgabe </strong></a></li>
            <li><a href="?content=vars"><strong> Variablen </strong></a></li>
            <li><a href="?content=backgroundChange"><strong> Hintergrund </strong></a></li>
            <li><a><strong> Kontrollstrukturen </strong></a>
                <ul>
                    <li><a href="?content=controlStructuresI"><strong> Kontrollstrukturen I </strong></a></li>
                    <li><a href="?content=controlStructuresII"><strong> Kontrollstrukturen II </strong></a></li>
                </ul>
            </li>
        </ul>
    </nav>
    <section id="content" style="background-color:<?php echo $contentColor; ?>">
        <?php
            //Content prüfen und entsprechenden Inhalt setzen
            if (@$content == 'output'){
                include ('content/output.php');
            }
            else if(@$content == 'backgroundChange'){
                include ('content/backgroundChange.php');
            }
            else if(@$content == 'vars'){
                include ('content/vars.php');
            }
            else if(@$content == 'controlStructuresI'){
                include ('content/controlStructuresI.php');
            }
            else {
                echo "kenn ich nicht...";}
        ?>
    </section>
    <footer></footer>
</body>
</html>
```



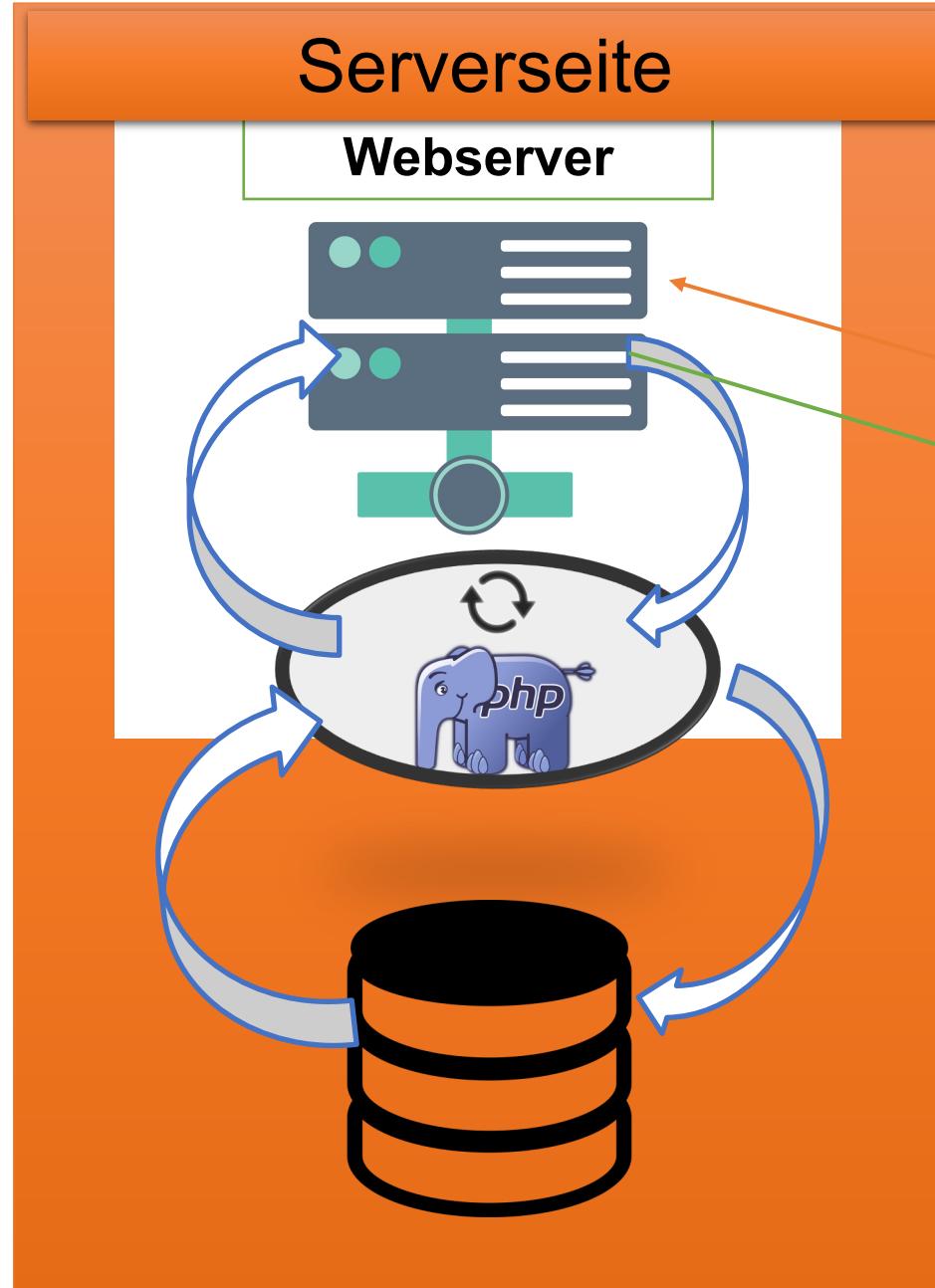
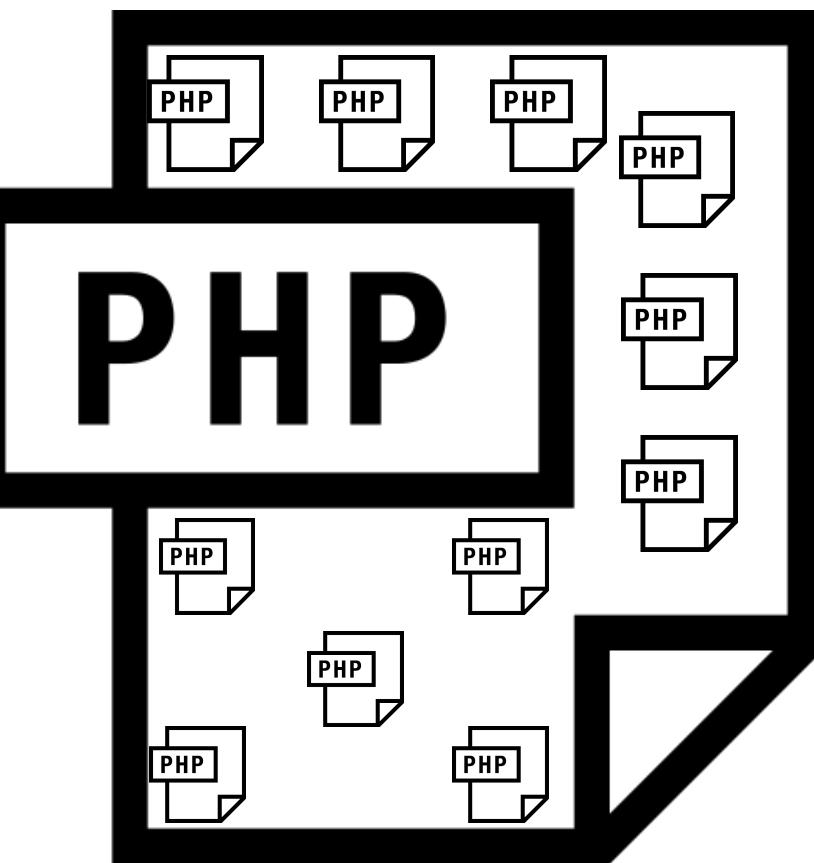
# Dynamische Websites –Was ist PHP?



PHP – rekursives Akronym für: Hypertext Preprocessor

- ... weit verbreitet
- ... eine Open-Source Skriptsprache
- ... leicht in HTML zu integrieren
- ... es wird auf einem Server ausgeführt





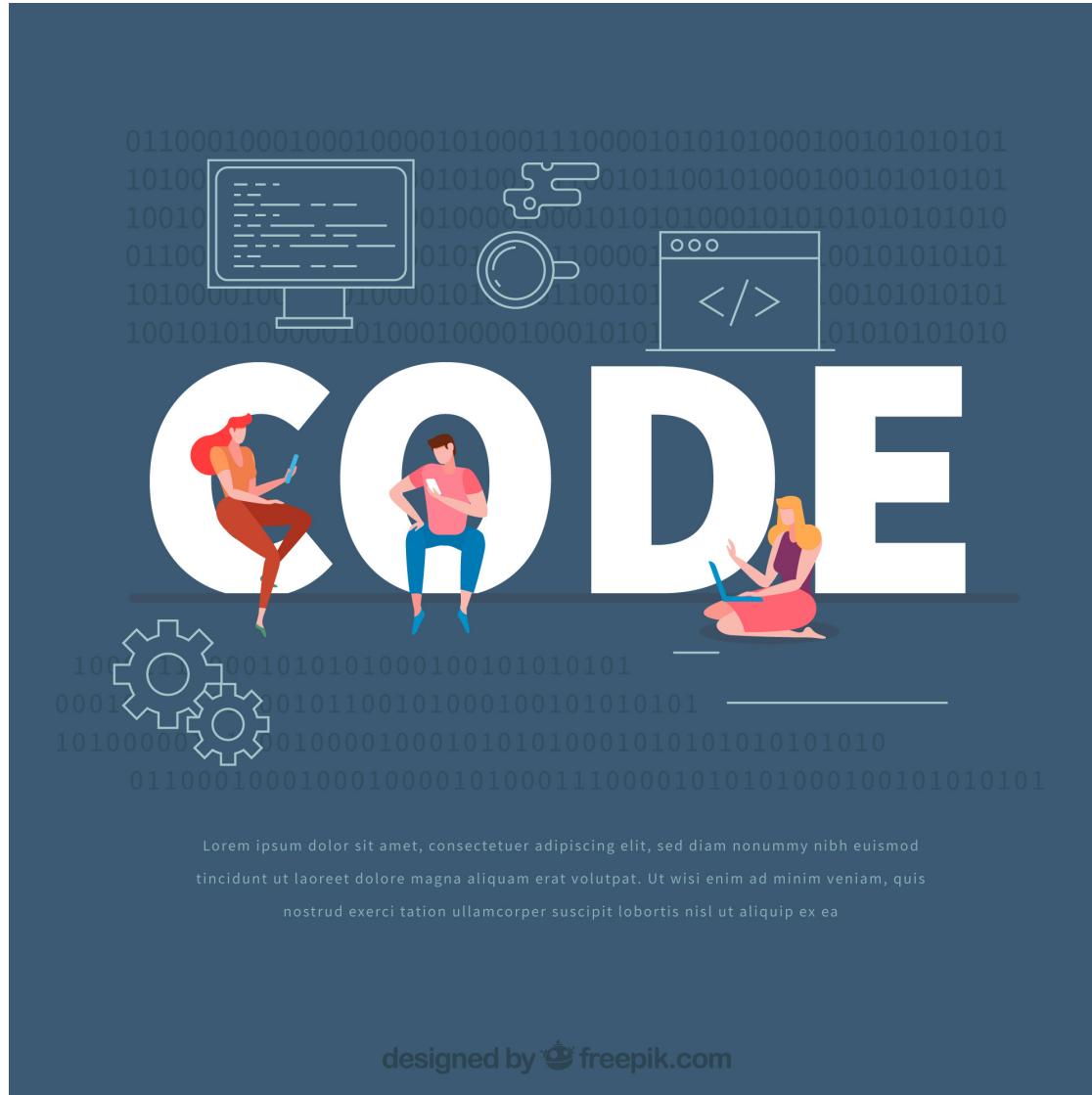
# Dynamische Websites –Was ist PHP?



PHP – rekursives Akronym für: Hypertext Preprocessor

- ... weit verbreitet
- ... eine Open-Source Skriptsprache
- ... im Besonderen für die Webprogrammierung geeignet
- ... leicht in HTML zu integrieren
- ... es wird auf einem Server ausgeführt
- ... im Besonderen für die Webprogrammierung geeignet





H

# Dynamische Website

## PHP – Variablen - Integer

```
/* Integer */  
echo "<br><br>";  
echo "/* Integer */";  
$output = 1;  
echo "<br>";  
echo $output;
```



\$	=> Kennzeichnung (für eine Variable)
output	=> Variablenbezeichnung
=	=> Zuweisung
1	=> Integer
;	=> Beendet die Anweisung



# Dynamische Website

## PHP – Client-Anzeige

```
/* Integer */  
1
```



# Dynamische Website

## PHP – Variablen – Konkatenation (Zusammenführung von Zeichenkette und Zahl)

```
$anotherOutput = "noch ein kurzer Text";
```

```
$output = 1;
```

```
/* String + Integer */  
echo "<br><br>";  
echo "/* String + Integer */";  
echo "<br>";  
echo $output." zusammen mit ".$anotherOutput;
```



# Dynamische Website

## PHP – Client-Anzeige

```
/* String + Integer */  
1 zusammen mit noch ein kurzer Text
```



# Dynamische Website

## PHP – Variablen – Integer - Rechnen

```
/*Rechnung*/  
echo "<br><br>";  
echo "/*Rechnung*/";  
echo "<br>";  
$four = 4;  
$six = 6;  
$plus = "+";  
$equal = "=";  
  
$result = $four + $six;  
echo $four.$plus.$six.$equal.$result;
```

### Rechen-Operatoren:

+	=	Addition
-	=	Subtraktion
*	=	Multiplikation
/	=	Division

\$result  
=   
+

=> Ergebnisvariable  
=> Zuweisung  
=> Additionsoperator

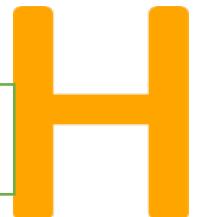


# Dynamische Website

## PHP – Variablen – Integer - Rechnen

```
/*Rechnung*/  
echo "<br><br>";  
echo "/*Rechnung*/";  
echo "<br>";  
$four = 4;  
$six = 6;  
$plus = "+";  
$equal = "=";  
  
$result = $four + $six;  
echo $four.$plus.$six.$equal.$result;
```

Ausgabe der Rechnung inklusive Ergebnis



# Dynamische Website

## PHP – Client-Anzeige

```
/*Rechnung*/  
4+6=10
```



# Dynamische Websites – Warum PHP?



# Dynamische Websites – Warum PHP?



# Dynamische Websites – Warum PHP?

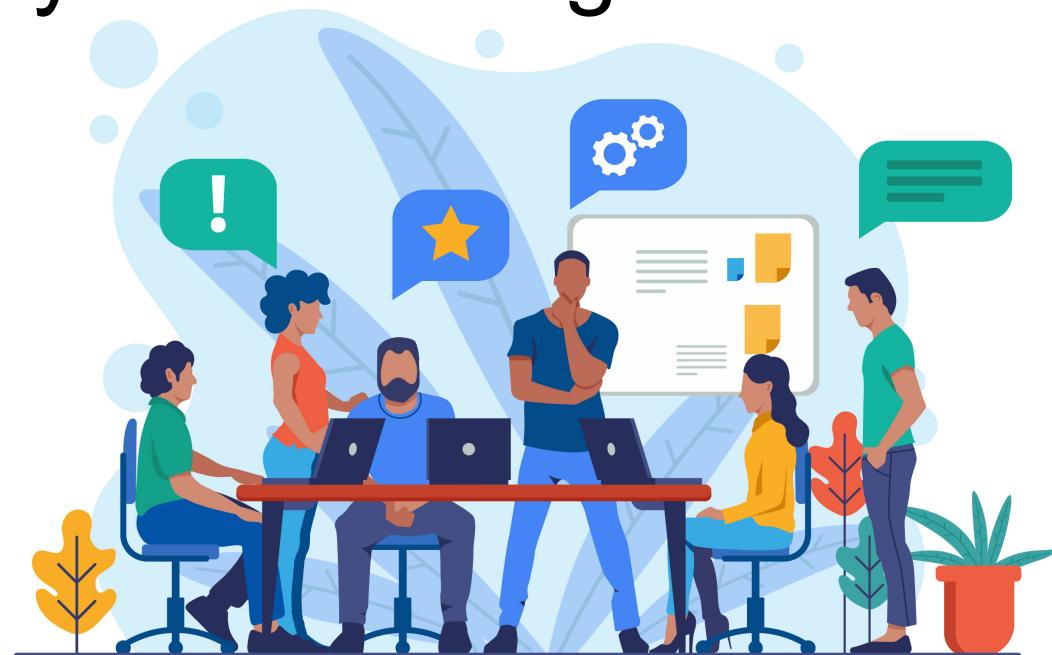


- Läuft allen Systemen (Windows, macOS etc.)
- Easy Coding
  - Bei Neu-Erstellung oder Aktualisierungen
- Leicht und schnell zur ersten dynamische Website



# Dynamische Websites – Warum PHP?

## easy start & integration



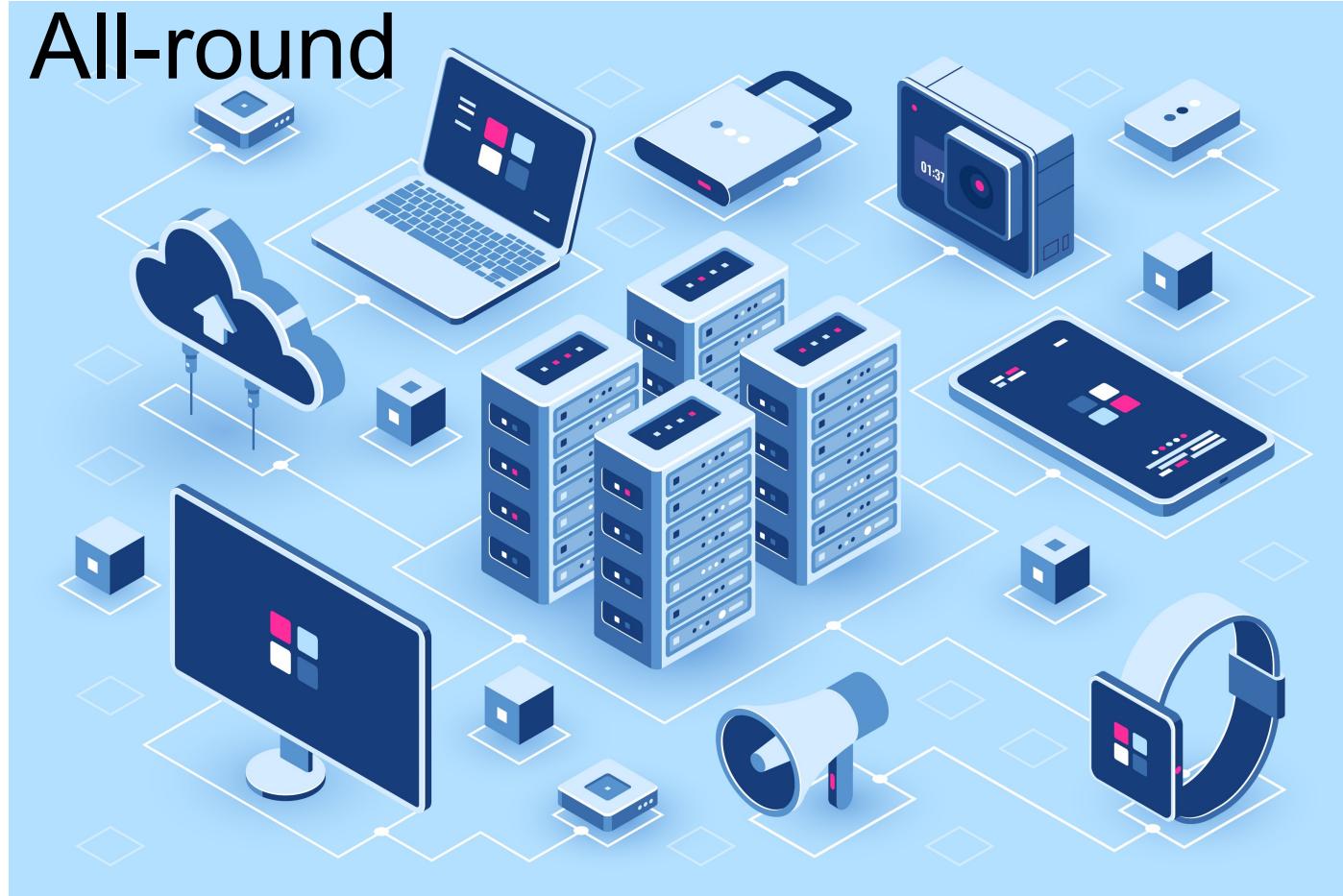
designed by freePik

- Keine Genies fürs Team notwendig
- Leichter Einstieg



# Dynamische Websites – Warum PHP?

All-round



- PHP ist eins mit dem Web
- Plattform unabhängig
- Vielseitigkeit
  - unterstützt viele verschiedene Datenbanken
- Skalierung leicht zu handhaben



# Dynamische Websites – Warum PHP?

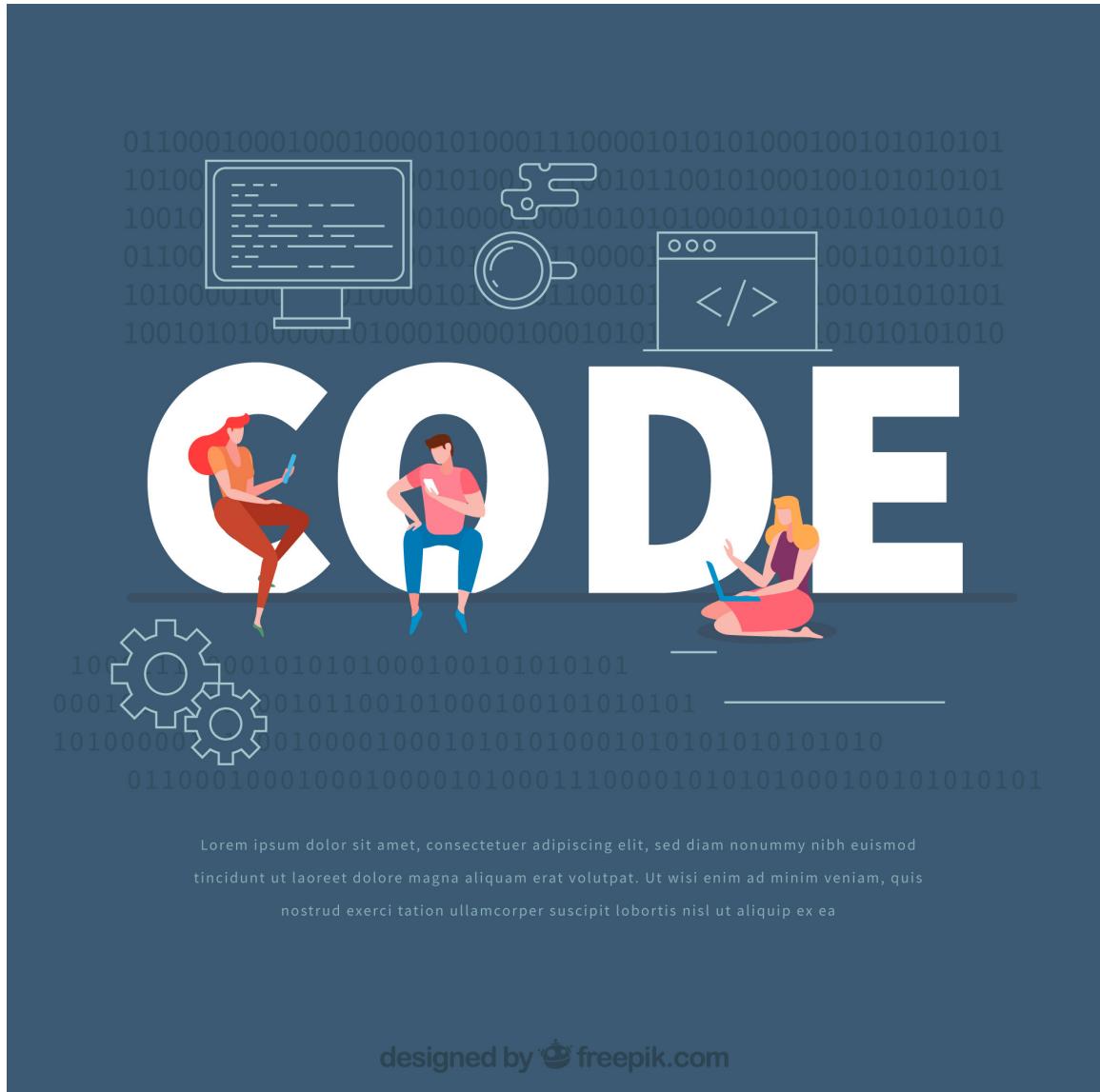
good documentation



designed by  **fotolia**

- Gute Dokumentation
  - Einstieg ([w3schools](#), [php.net](#))
- Für unzählige Probleme gibt es bereits unzählige Lösungen
- [Blogs](#) / [StackOverflow](#) / [Reddit](#) etc.





H

# Dynamische Website

## PHP – HTML-Form

Anforderung:

Erstelle ein Formular zur Eingabe von Zahlen, die miteinander addiert werden und auf der gleichen Seite ein Ergebnis ausgeben

`/* Eingabe Felder zum Addieren */`

`addieren`

`/* Eingabe Felder zum Addieren */`

`addieren`

Das Ergebnis lautet:  
4



# Dynamische Webseiten



Benutze einfach aus der  
HTTP-Request-Method  
die GET-Parameter



H

# Dynamische Website

## PHP – HTML-Form-Design

```
/* Eingabe Felder zum Addieren */
<br><br>
<form action="" method="GET" id="form">
    <input type="text" name="number1" value="" />
    <br><br>
    <input type="text" name="number2" value="" />
    <br><br>
    <input type="submit" value="addieren" style="background:lightgrey;width:100px;height:30px" />
    <br><br>
</form>
```

/\* Eingabe Felder zum Addieren \*/


# Dynamische Website

## PHP – HTML-Form - Weiterleitung

Ohne Weiterleitung (bleibt einfach leer)

```
/* Eingabe Felder zum Addieren */
<br><br>
<form action="" method="GET" id="form">
    <input type="text" name="number1" value="" />
    <br><br>
    <input type="text" name="number2" value="" />
    <br><br>
    <input type="submit" value="addieren" style="background:lightgrey;width:100px;height:30px" />
    <br><br>
</form>
```



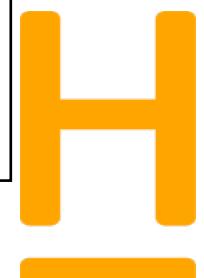
# Dynamische Website

## PHP – HTML-Form – Hyper Text Transfer Protocol

HTTP-Request-Method (Anfragemethode) festlegen

Hier GET

```
/* Eingabe Felder zum Addieren */
<br><br>
<form action="" method="GET" id="form">
    <input type="text" name="number1" value="" />
    <br><br>
    <input type="text" name="number2" value="" />
    <br><br>
    <input type="submit" value="addieren" style="background:lightgrey;width:100px;height:30px" />
    <br><br>
</form>
```



# Dynamische Website

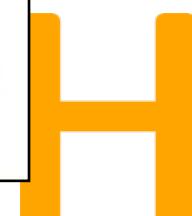
## PHP – HTML-Form – HTTP-Request-Method GET

GET-Parameter bestimmen



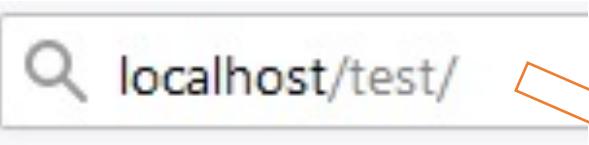
Und wozu ist das  
jetzt gut?

```
/* Eingabe Felder zum Addieren */
<br><br>
<form action="" method="get">
    <input type="text" name="number1" value="" />
    <br><br>
    <input type="text" name="number2" value="" />
    <br><br>
    <input type="submit" value="addieren" style="background:lightgrey;width:100px;height:30px" />
    <br><br>
</form>
```



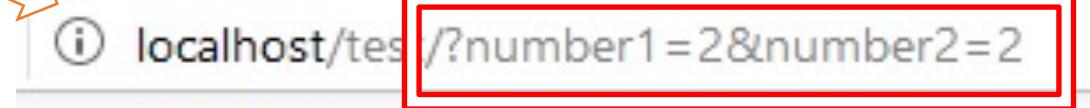
# Dynamische Website

## PHP – HTML-Form – HTTP-Request-Method GET



/\* Eingabe Felder zum Addieren \*/

addieren



/\* Eingabe Felder zum Addieren \*/

addieren

Drücke auf „addieren“ und schaue im Browser in deine Adressleiste



GET-Parameter



# Dynamische Website

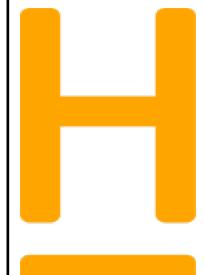
## PHP – HTML-Form – HTTP-Request-Method GET

/\* Eingabe Felder zum Addieren \*/

ⓘ localhost/test/?number1=2&number2=2

```
/* Eingabe Felder zum Addieren */
<br><br>
<form action="" method="GET" id="form">
  <input type="text" name="number1" value="" />
  <br><br>
  <input type="text" name="number2" value="" />
  <br><br>
  <input type="submit" value="addieren" style="background:lightgrey;width:100px;height:30px" />
  <br><br>
</form>
```



# Dynamische Website

## PHP – HTML-Form – HTTP-Request-Method GET

Parameter werden an die URL geheftet mit ?

localhost/test/?number1=2&number2=2

Parameter werden verbunden mit &

Was macht man jetzt damit?



# Dynamische Website

## PHP – HTTP-Request-Method GET

① `localhost/test/?number1=2&number2=2`

```
$number1 = $_GET['number1'];
$number2 = $_GET['number2'];
```

PHP kann die GET-Parameter mit `$_GET['variable']` auslesen und weiterverwenden



```
/* Eingabe Felder zum Addieren */
<br><br>
<form action="" method="GET" id="form">
  <input type="text" name="number1" value="" />
<br><br>
  <input type="text" name="number2" value="" />
<br><br>
  <input type="submit" value="addieren" style="background:lightgrey;width:100px;height:30px" />
<br><br>
</form>
```



# Dynamische Website

## PHP – HTTP-Request-Method GET

Auf der Website wird eine Meldung angezeigt,  
wenn die Seite aufgerufen wird



```
$number1 = $_GET['number1'];  
$number2 = $_GET['number2'];
```

/\* Eingabe Felder zum Addieren \*/

addieren

Notice: Undefined index: number1 in C:\xampp\htdocs\test\content\vars.php on line 67

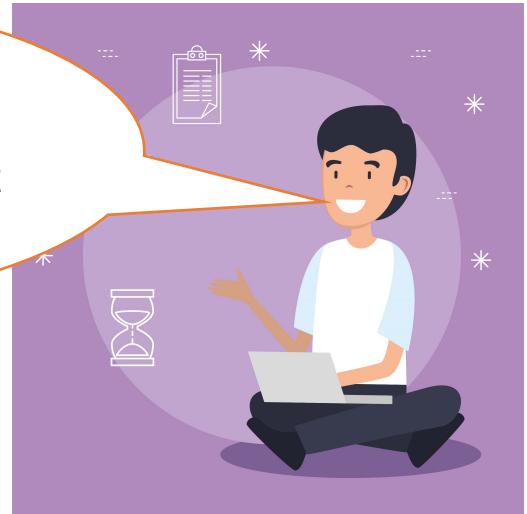
Notice: Undefined index: number2 in C:\xampp\htdocs\test\content\vars.php on line 68



# Dynamische Website

## PHP – HTTP-Request-Method GET

Request-Parameter müssen vor der Zuweisung zu einer Variable geprüft werden (damit es nicht zu Fehlern kommt)



```
isset($_GET['number1'])
```

**isset** => prüft ob eine Variable existiert und ob sie nicht NULL ist  
=> gibt einen boolean zurück  
=> [mehr dazu](#)



# Dynamische Website

# PHP – HTTP-Request-Method GET

```
if(isset($_GET['number1']) && isset($_GET['number2'])) {  
    //...  
}
```

\* Genaueres zu **if's** kommt in späterer Unterrichtseinheit



# Dynamische Website

## PHP – PHP + Rechnung

```
<?php

if(isset ($_GET['number1']) && isset ($_GET['number2'])) {
    $number1 = $_GET['number1'];
    $number2 = $_GET['number2'];

    $result = $number1 + $number2;

    echo "Das Ergebnis lautet:";
    echo "<br>".$result;
}

?>
```



# Dynamische Website

## PHP – HTML-Form + PHP

```
/* Eingabe Felder zum Addieren */
<br><br>
<form action="" method="GET" id="form">
  <input type="text" name="number1" value="" />
  <br><br>
  <input type="text" name="number2" value="" />
  <br><br>

  <input type="submit" value="addieren" style="background:lightgrey;width:100px;height:30px" />
  <br><br>
</form>
<?php

if(isset($_GET['number1']) && isset($_GET['number2'])){
  $number1 = $_GET['number1'];
  $number2 = $_GET['number2'];

  $result = $number1 + $number2;

  echo "Das Ergebnis lautet:";
  echo "<br>".$result;
}

?>
```



designed by freepik.com



# Dynamische Websites – Was kann PHP?



# Dynamische Websites – Was kann PHP?

## Kommandozeilenprogrammierung

```
File Edit View Search Terminal Help
$_SERVER["SCRIPT_FILENAME"] => /var/www/html/infophp.php
$_SERVER["PATH_TRANSLATED"] => /var/www/html/infophp.php
$_SERVER["DOCUMENT_ROOT"] =>
$_SERVER["REQUEST_TIME_FLOAT"] => 1436341155.45
$_SERVER["REQUEST_TIME"] => 1436341155
$_SERVER["argv"] => Array
(
    [0] => /var/www/html/infophp.php
)
$_SERVER["argc"] => 1

PHP License
This program is free software; you can redistribute it and/or modify
it under the terms of the PHP License as published by the PHP Group
and included in the distribution in the file: LICENSE

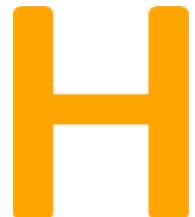
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

If you did not receive a copy of the PHP license, or have any
questions about PHP licensing, please contact license@php.net.
avi@deb:~$
```

Quelle: <https://www.tecmint.com/run-php-codes-from-linux-commandline/>

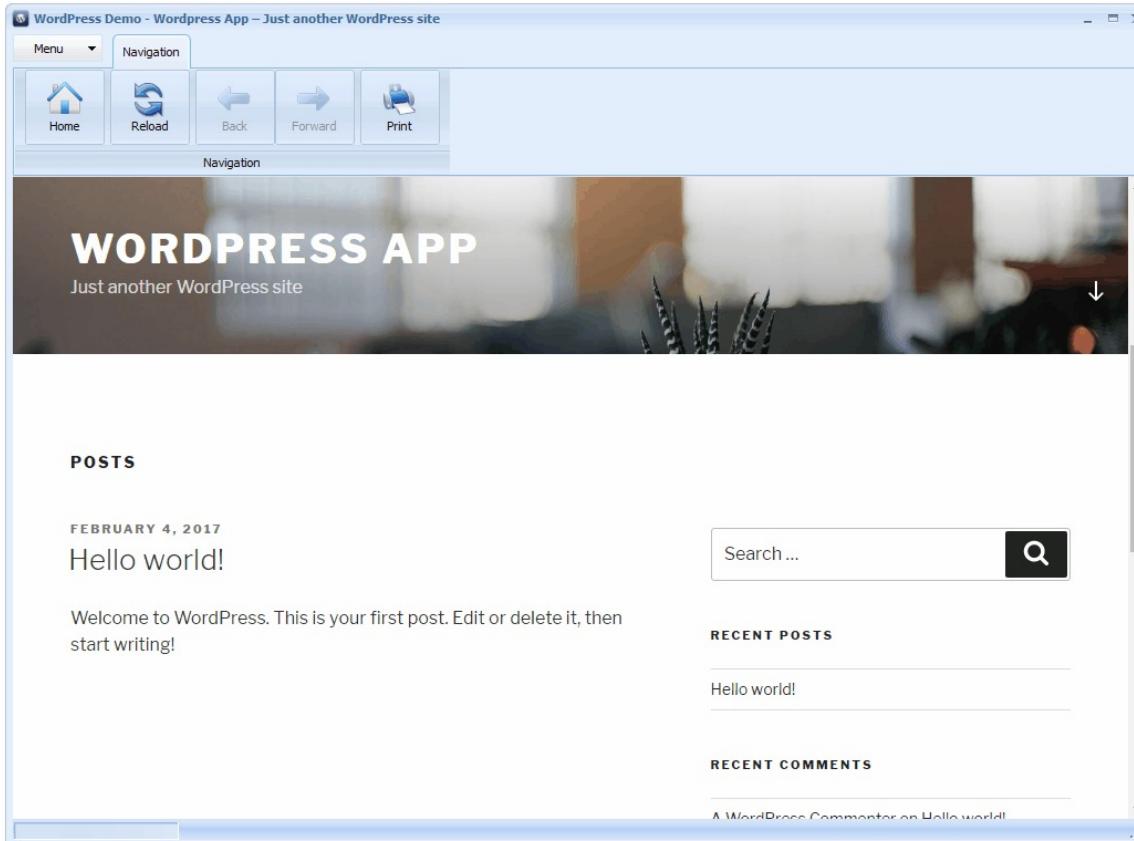
- Kein Server / Browser
- nur PHP-Parser notwendig
- Beispielhafte Verwendung
  - Cron-Jobs\*
  - Einfache Textprozessierung ([Tokenisierung](#), [Part-of-speech-Tagging](#))

\*Bestimmte Aufgaben die zu definierten Terminen automatisiert ausgeführt werden



# Dynamische Websites – Was kann PHP?

## Desktop – Applikationen (Rich-Client)



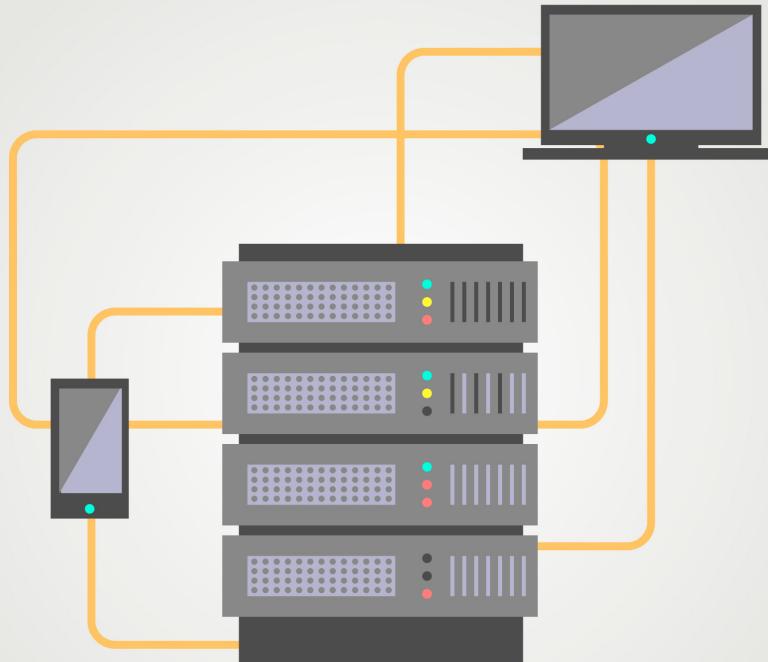
- Plattform übergreifend
- Eher für Fortgeschrittene
- (nicht die beste Alternative für Desktop-Applikationen)

[\(http://gtk.php.net/\)](http://gtk.php.net/)

Quelle: <https://www.exeoutput.com/samples/wordpress-desktop-app>

# Dynamische Websites – Was kann PHP?

## Serverseitige Programmierung



designed by  freepik

- Stellt den Nutzungsschwerpunkt von PHP dar
- Es wird benötigt:
  1. PHP-Parser (für gewöhnlich integriert)
  2. einen Webserver
  3. Webbrower

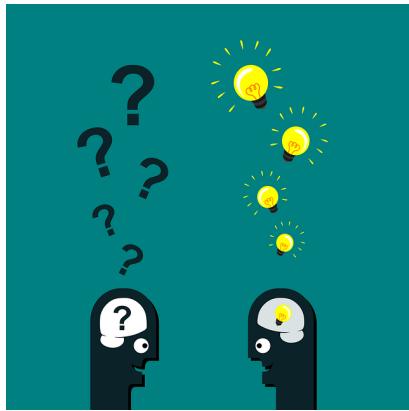


# Dynamische Website

## Was kann PHP im Detail?

- Prozedurale Programmierung (z.B. imperativer Ansatz)
- Objektorientierte Programmierung
- Kommunikation mit diversen Service
- Anbindung vieler verschiedener Datenbanken möglich
  - [databases](#)
- Öffnung von Netzwerk-Sockets (->Clients)
- Dynamische Generierung von Inhalten (z.B. HTML, Pdf-Dateien, Bilder etc.)
- Etc.

Die wiederum diverse Protokolle verwenden  
(LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM)



# Finish

# Dynamische Websites

# PHP I

