

Refugeeks / IntoCode

Dynamische Websites

23.09.2020 / 03:
Template-Erstellung, HTTP-Method-Post





Dynamische Website

PHP – Variablen - Datentypen

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

Hinweis:
Möchte man während der Entwicklung mehr Infos über eine Variable an einer bestimmten Stelle haben, kann man die Funktion `var_dump();` verwenden



Dynamische Website

Variablen - Boolean

```
// boolean - Beispiele  
echo "<h5>Boolean</h5>";  
$isTrue = true;
```



\$isTrue
;

=> Variable (boolescher Wert(true/false))

=> Beendet eine Anweisung



Dynamische Website

Variablen - Boolean

```
// boolean - Beispiele
echo "<h5>Boolean</h5>";
$isTrue = true;

echo "Die Boolean-Variable gibt folgendes aus bei <strong>true</strong> ". "= <i>".$isTrue."</i>";
$isTrue = false;

echo "<br>";
$spaces6="&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;";
echo "Die Boolean-Variable gibt folgendes aus bei <strong>false</strong> ". "= <i>".$isTrue."</i>".$spaces6."(nichts)";
```



Dynamische Website

Ausgabe - Variablen - Boolean

Boolean

Die Boolean-Variable gibt folgendes aus bei **true** = 1

Die Boolean-Variable gibt folgendes aus bei **false** = (nichts)



Dynamische Website

Variablen – var_dump();

```
// boolean - var_dump() Beispiele
echo "<h5>Boolean</h5>";
$isTrue = true;
echo "Die Boolean-Variable gibt folge
echo "<br>";
var_dump($isTrue);

$isTrue = false;
echo "<br>";

$spaces6="&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;";

echo "Die Boolean-Variable gibt folge
echo "<br>";
var_dump($isTrue);

$output="erster Output als Text";
echo "<br>";
var_dump($output);

$output=2535;
echo "<br>";
var_dump($output);

$output="wieder als Text";
echo "<br>";
var_dump($output);
```

Boolean

Die Boolean-Variable gibt folgendes aus bei **true** = 1

```
bool(true)
```

Die Boolean-Variable gibt folgendes aus bei **false** = (nichts)

- bool(false)

```
string(22) "erster Output als Text"
```

```
int(2535)
```

```
string(15) "wieder als Text"
```

Datentyp
(Zahl)

```
: bool / string / int
```

: Anzahl Zeichen bei String

: Wert bei Integer

“...Text...”


: Der Wert innerhalb der Var



Dynamische Website

Kontrollstruktur-Einführung - if

Bestimmter Code soll nur ausgeführt werden, wenn eine Bedingung zutrifft



```
$isTrue = true;

if($isTrue) {
    echo "die Bedingung ist wahr";
}
```

`$isTrue` => Variable (boolescher Wert(true/false))
`if($variable)` => Prüft ob die Bedingung wahr ist
{ => start if
`echo...` => auszuführender Code, wenn Bedingung = true
} => ende if



Dynamische Website

Kontrollstruktur-Einführung - if

die Bedingung ist wahr




Dynamische Website

Kontrollstruktur-Einführung - if

Bestimmter Code soll nur ausgeführt werden, wenn eine Bedingung zutrifft

```
$isTrue = false;

if($isTrue) {
    echo "die Bedingung ist wahr";
}
else{
    echo "die Bedingung ist nicht wahr";
}
```



else	=> Code der stattdessen ausgeführt werden soll
{	=> start else
echo...	=> auszuführender Code
}	=> ende else



Dynamische Website

Kontrollstruktur-Einführung - if

die Bedingung ist nicht wahr





Dynamische Website

Einfache Template-Erstellung

Anforderung:

Die Website/Anwendung soll so konzipiert werden, dass sie jederzeit ohne großen Aufwand um einzelne Elemente/Features erweitert werden kann.

IST-Stand:

Jede Datei enthält vollständigen HTML und PHP-Code

SOLL:

Semantische Trennung der Elemente
(Header, Nav, Content, Footer)

Lösungsansatz:

Template-Erstellung



Dynamische Website

Einfache Template-Erstellung - Vorteile

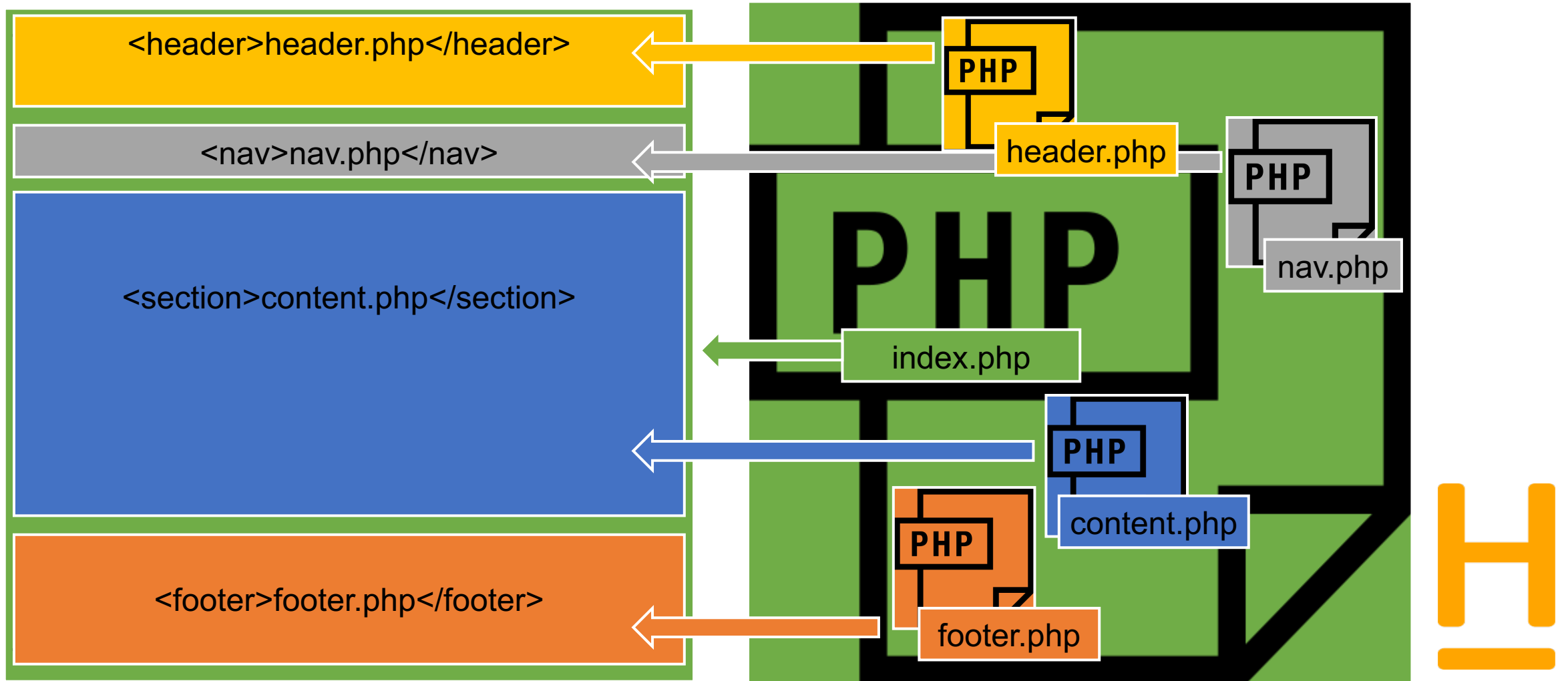
Templates dienen der Trennung von Design und Logik

- Gesamtüberblick bleibt leichter erhalten
- Konzentration auf einzelne Elemente
- Fehlerquellen werden verkleinert
- Kein bis geringer Einfluss auf andere Elemente
- Einzelne Elemente sind schnell und einfach austauschbar
- Ebenso sind Erweiterungen schnell einzubinden
- Etc.



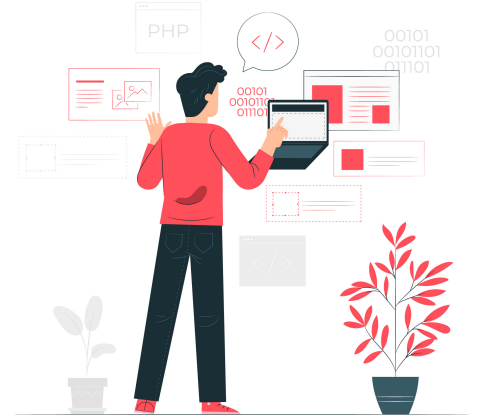
Dynamische Website

Einfache Template-Erstellung – Semantische Aufteilung



Dynamische Website

Einfache Template-Erstellung



Einbindung bzw. Ausführung von externen Dateien mit den Funktionen:

```
include("meineDatei.php"); / include_once("meineDatei.php");  
oder require("meineDatei.php");
```

Unterschied:

- | | |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| Include | : Nachfolgender Code wird ausgeführt, auch bei Problemen mit der Datei (Fehlerhaft/Nicht vorhanden)
: Ausgabe einer Warnung (E_WARNING) |
| Include_once | : Wie include aber die ausgewählte Datei wird nur <i>einmal</i> eingebunden |
| Require | : Nachfolgender Code wird nicht ausgeführt. Die Anwendung wird beendet.
: Wirft einen Compiler Fehler (E_COMPILER_ERROR) |

Einsatz hängt vom Anwendungsfall ab.

Darf oder soll nachfolgender Code ausgeführt werden, wenn es ein Problem gibt?

Dynamische Website

Einfache Template-Erstellung – header.php



designed by freepik.com



header.php

PHP

HTML-Code

```
<div id="headerWrapper">  
  
    <h1>header</h1>  
  
</div>
```



Dynamische Website

Einfache Template-Erstellung – index.php



designed by freepik.com

A large, stylized icon of a document. It has a thick black border. Inside, the text "Index.php" is in a small box at the top, and "PHP" is written in large, bold, black letters in the center.

HTML+PHP-Code

```
<header>
  <?php
    include_once("header.php");
  ?>
</header>
```





Dynamische Website

HTTP-Request-Method POST

Anforderung:

Es wird ein Formular zur Eintragung von sensiblen Daten benötigt

Besonderheiten:

Persönliche Daten dürfen nicht in der URL auftauchen, da sie sonst vom Browser gespeichert werden.

Lösungsansatz:

HTTP-Request-Method - POST

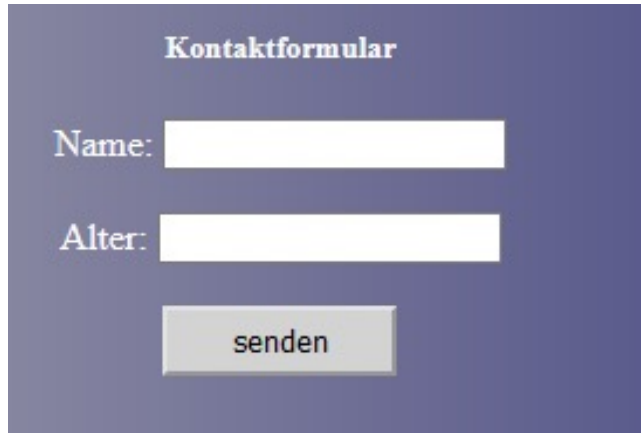


Dynamische Website

Bisher: HTTP-Request-Method GET

Diese Methode kennen wir

```
<h5> Kontaktformular </h5>
<form action="" method="GET" id="form">
  <label for="name">Name: </label>
  <input type="text" id="name" name="name" value=""/>
  <br><br>
  <label for="age">Alter: </label>
  <input type="text" id="age" name="age" value=""/>
  <br><br>
  <input type="submit" value="senden" style="background:lightgrey;width:100px;height:30px"/>
  <br><br>
</form>
```



Kontaktformular

Name:

Alter:

localhost/test/?name=Rainer&age=25



Dynamische Website

HTTP-Request-Method POST

Kontaktformular

Name:

Alter:

Senden

Kontaktformular

Name:

Alter:

localhost/test/

```
<h5> Kontaktformular </h5>
<form action="" method="POST" id="form">
  <label for="name">Name: </label>
  <input type="text" id="name" name="name" value="" />
  <br><br>
  <label for="age">Alter: </label>
  <input type="text" id="age" name="age" value="" />
  <br><br>
  <input type="submit" value="senden" style="background:lightgrey;width:100px;height:30px"/>
  <br><br>
</form>
```

Die Parameter werden nicht über die URL übermittelt



Dynamische Website

HTTP-Request-Method Unterschied GET & POST



GET:

- Übermittlung der Daten mittels URL
- Übergabe der Daten ist begrenzt (URL max. 1024 Zeichen)
- Übertragende Daten sind direkt sichtbar
- Ergebnisseite mit Daten kann gespeichert werden(bookmark)

POST:

- Übermittlung der Daten über [HTTP-Header](#)
- Keine Begrenzung der Daten
 - (Verwendung bei File-Uploads, Bildern oder großen/langen Texten)
- Ergebnisse eine POST-Formulars sind nicht speicherbar



Dynamische Website

HTTP-Request-Method POST

- Daten sind nicht (vollständig) unsichtbar oder verschlüsselt.
- HTTP-Header auslesbar mittels Entwicklerkonsole im Browser
(z.B. Firefox | Extras \Rightarrow Web-Entwickler \Rightarrow Werkzeuge einblenden \Rightarrow Das Formular ausführen)



Inspector Konsole Debugger Netzwerkanalyse Stilbearbeitung Laufzeitanalyse Speicher Web-Speicher Barrierefreiheit

Adressen durchsuchen

Status	Methode	Host	Datei	Ursprung	Typ	Übertragen	Größe
200	POST	localhost	/test/	document	html	1,40 KB	1,16 KB
200	GET	localhost	style.css	stylesheet	css	2,59 KB	2,29 KB
404	GET	localhost	favicon.ico	img	html	Aus Cache	1,15 KB

Kopfzeilen Cookies **Parameter** Antwort Zeit

Anfrageparameter durchsuchen

▼ Formulardaten

name: "Rainer"
age: "25"

▼ Anfrageinhalt

1 name=Rainer&age=25



Dynamische Website

HTTP-Request-Method Unterschied POST

Formular-Daten mit POST auslesen

```
<?php
    if(isset ($_POST['name']) && isset ($_POST['age'])) {
        $name = $_POST['name'];
        $age = $_POST['age'];

        echo "Der Name lautet: ". $name;
        echo "<br>";
        echo "und er ist ". $age. " Jahre alt";
    }
?>
```

Kontaktformular

Name:

Alter:

Der Name lautet: Rainer
und er ist 25 Jahre alt



Finish

Dynamische Websites

PHP II

