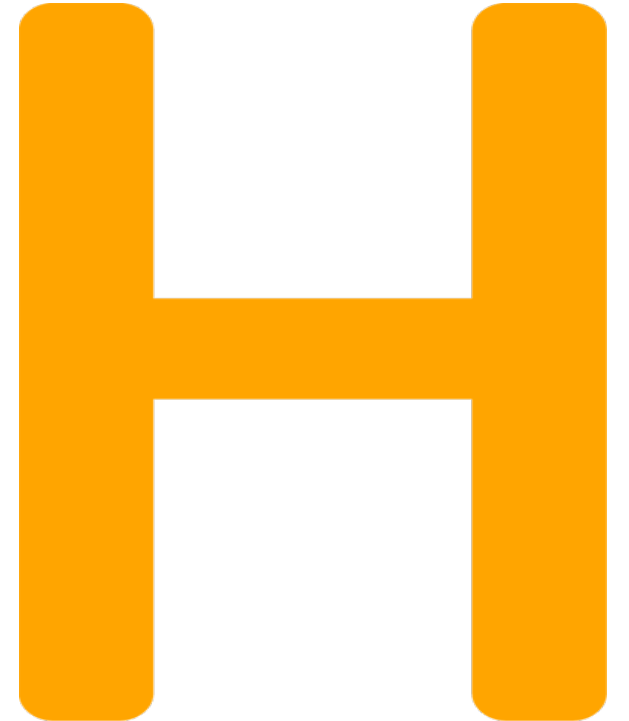


Refugeeks / IntoCode

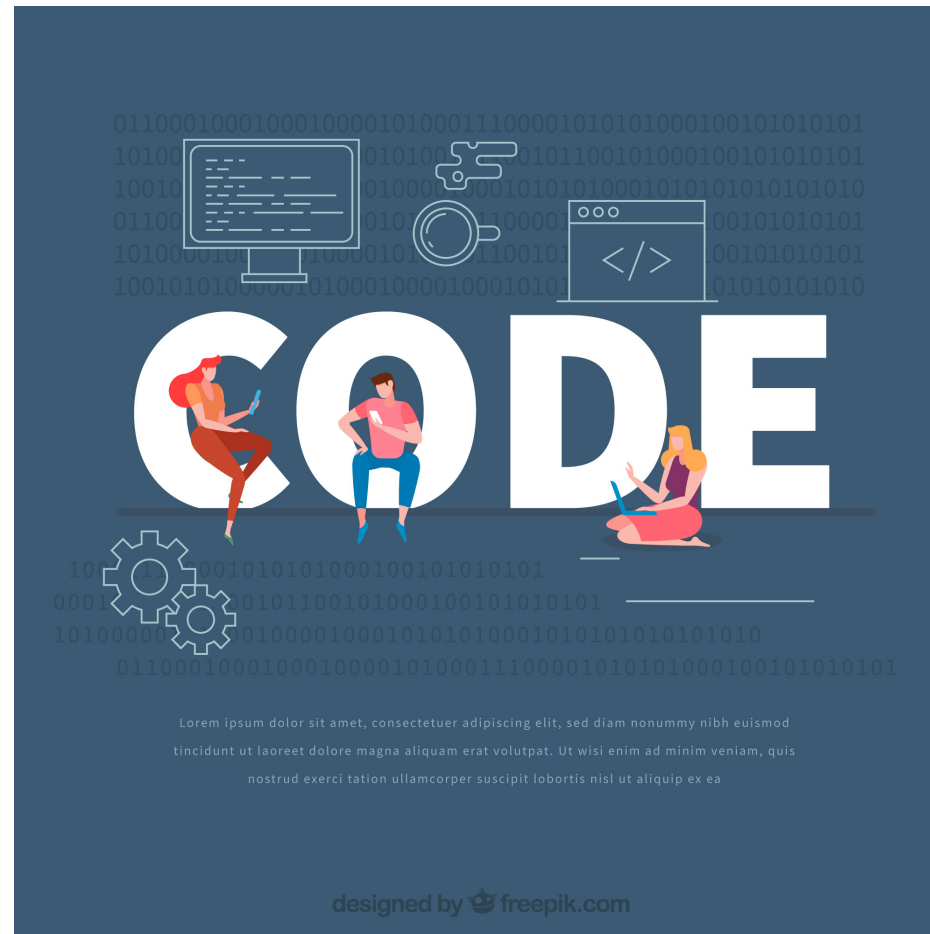
Dynamische Websites

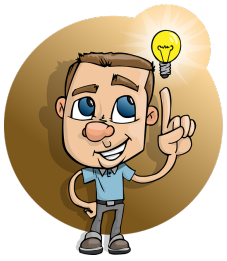
23.09.2020 / 04:

If-else, dynamisches Design, dynamischer Content



Dynamische Websites – Kontrollstruktur-if





Kontrollstruktur-Einführung - if

Bestimmter Code soll nur ausgeführt werden, wenn eine Bedingung zutrifft

Bausteine eines if-Blocks

```
if($bedingung){  
    php-code;
```

(notwendiger Teil)

```
}  
else if($eineAndereBedingung) {  
    php-code;
```

(else if ist optional – kann
beliebig oft angefügt werden)

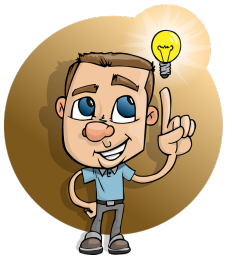
```
}  
else {  
    php-code;  
}
```

(else ist optional – wird immer ausgeführt, wenn
keine Bedingung zutrifft)

Sobald eine Bedingung `true` ist, wird der Code innerhalb der geschweiften Klammern ausgeführt und der gesamte if-Block damit abgeschlossen



Dynamische Website



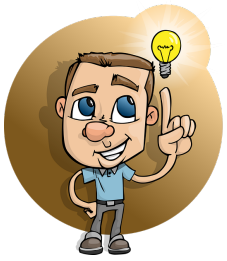
Kontrollstruktur-Einführung – if - Negation

- Zur Negation einer Bedingung wird “ ! “ benutzt

```
$isTrue = true;  
  
if(!$isTrue){  
    .....  
    echo "der Wert ist nicht wahr";  
}
```

Code wird nicht
ausgeführt





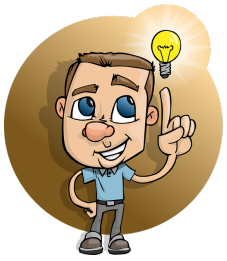
Kontrollstruktur- if

Möglichkeiten zur Überprüfung einer Bedingung

- boolean-Variable
- < (kleiner als)
- > (größer als)
- <= (kleiner als oder gleich)
- >= (größer als oder gleich)
- == (gleich [wichtig: 2x "=" für einen Vergleich])
- != (ungleich)

Beispiele

- \$var = TRUE; = true
- 1 < 2 = true
- 1 > 2 = false
- 1 <= 2 = true
- 1 >= 2 = false
- 1 == 2 = false
- 1 != 2 = true
- "String" == "Vergleichen" = false
- "String" == "String" = true
- "String" != "Vergleichen" = true



Kontrollstruktur- if - Code-Beispiele

```
if(1 < 2){
    echo "1 ist kleiner als 2";
}
if(1 > 2){
    echo "1 ist größer als 2";
}
if(1 == 2){
    echo "1 ist gleich als 2";
}
```

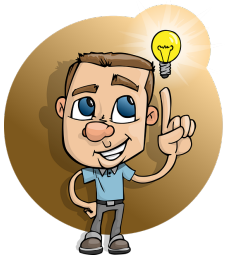
```
if(1 <= 1){
    echo "1 ist kleiner oder gleich 1";
    echo "<br>";
}else if(1 <= 2){
    echo "1 ist kleiner oder gleich 2";
    echo "<br>";
}
```

```
if(1 < 2){
    echo "1 ist kleiner als 2";
}else if(1 > 2){
    echo "1 ist größer als 2";
}else if(1 == 2){
    echo "1 ist gleich als 2";
}else{
    echo "irgendwas passt da nicht";
}
```

```
if("Wort" == "keinWort"){
    echo "Wort ist gleich keinWort";
    echo "<br>";
}
```

```
if("Wort" != "keinWort"){
    echo "Wort ist ungleich keinWort";
    echo "<br>";
}
```

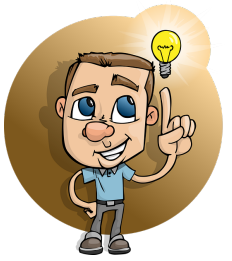




Kontrollstruktur- if - Verknüpfung von Bedingungen mit booleschen Operatoren

Operator	Bedeutung	Beispiel
<code>\$isHigher and \$isLower</code>	Und	TRUE wenn sowohl <code>\$isHigher</code> als auch <code>\$isLower</code> TRUE ist.
<code>\$isHigher or \$isLower</code>	Oder	TRUE wenn <code>\$isHigher</code> oder <code>\$isLower</code> TRUE ist.
<code>\$isHigher xor \$isLower</code>	Entweder Oder	TRUE wenn entweder <code>\$isHigher</code> oder <code>\$isLower</code> TRUE ist, aber nicht beide.
<code>! \$isHigher</code>	Nicht	TRUE wenn <code>\$isHigher</code> nicht TRUE ist.
<code>\$isHigher && \$isLower</code>	Und	TRUE wenn sowohl <code>\$isHigher</code> als auch <code>\$isLower</code> TRUE ist.
<code>\$isHigher \$isLower</code>	Oder	TRUE wenn <code>\$isHigher</code> oder <code>\$isLower</code> TRUE ist.





Kontrollstruktur- if - Code-Beispiele

```
$number1 = 1;
$number2 = 2;
$word1 = "Wort";
$word3 = "Wort";

$conditionTrue = true;
$conditionFalse = false;
$isLower = $number1 < $number2; // true
$isSameWord = $word1 == $word3; // true

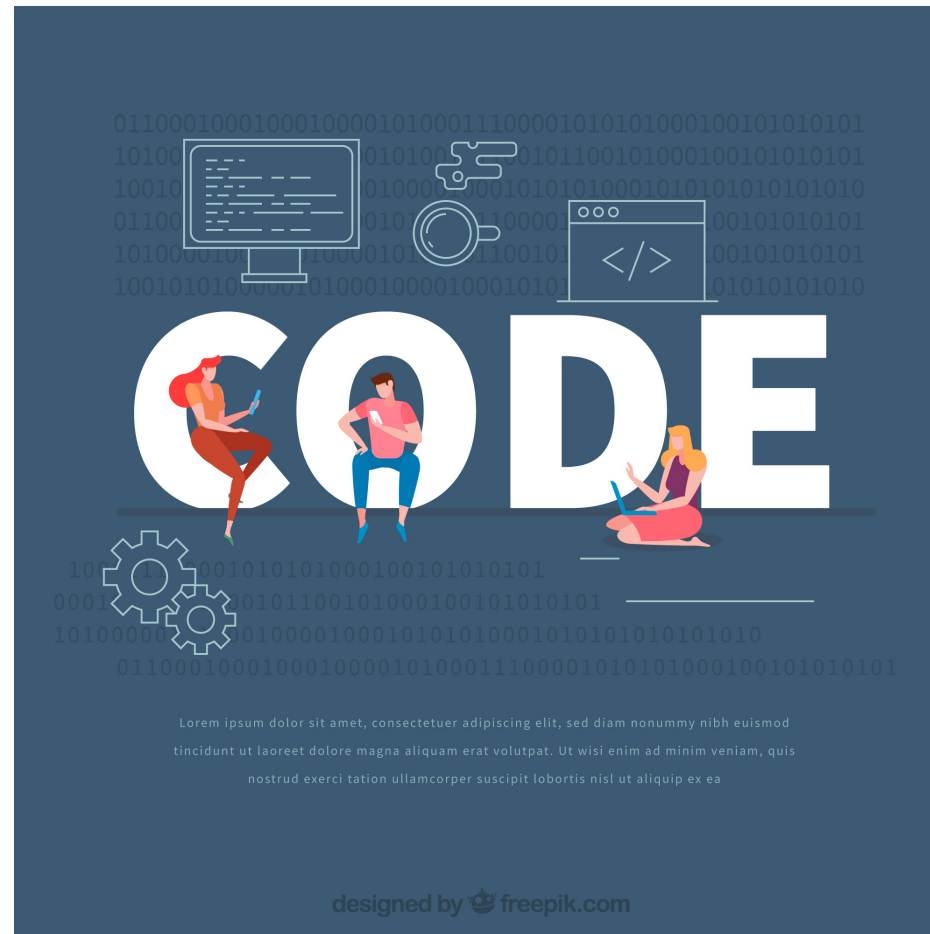
if($conditionTrue and !$conditionFalse){
    echo "die Bedingungen sind beide wahr";
    echo "<br>";
}

if($conditionFalse xor $isLower){
    echo "entweder ist eine Bedingung wahr oder die andere";
    echo "<br>";
}

if($conditionTrue && !$isSameWord){
    echo "Bitte Ausgabe machen, wenn Beides wahr ist";
}
```



Dynamisches Design mit PHP - Styles



Dynamische Website

Dynamisches Design mit PHP - Styles

Anforderung:

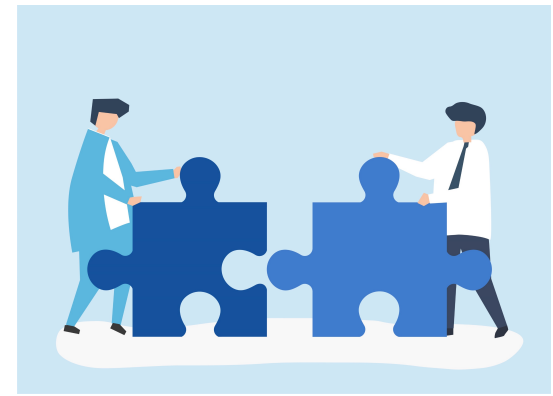
Der Benutzer soll durch Betätigung eines Buttons den Hintergrund verändern können

IST-Stand:

Style wird statisch definiert. (Direkt im HTML-Tag, im Header oder CSS-Datei)

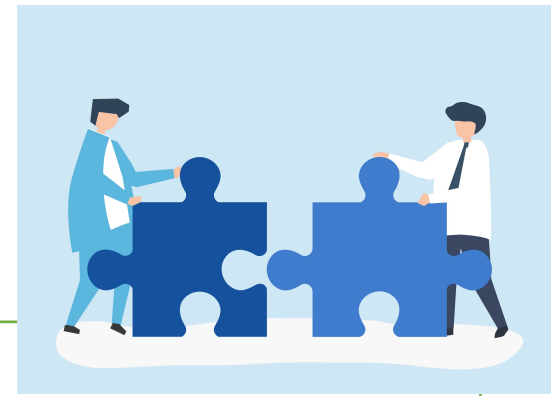
Lösungsansatz:

- Buttons(oder andere Elemente) bereitstellen, die eine Interaktion mit dem Benutzer ermöglichen
- In PHP Variablen definieren, die eine Steuerung der Aktionen ermöglichen
- Ausgabe der Variablen, wo Style-Attribute definiert werden



Dynamische Website

Dynamisches Design mit PHP - Styles



- Die dynamische Style-Steuerung erfolgt über HTTP mit Hilfe der GET-Parameter
- GET-Parameter sind nicht an Formulare gebunden.
- **<a>-Tag** zur Verlinkung von Dateien/Webseiten, kann GET-Parameter übertragen

Beispiel:

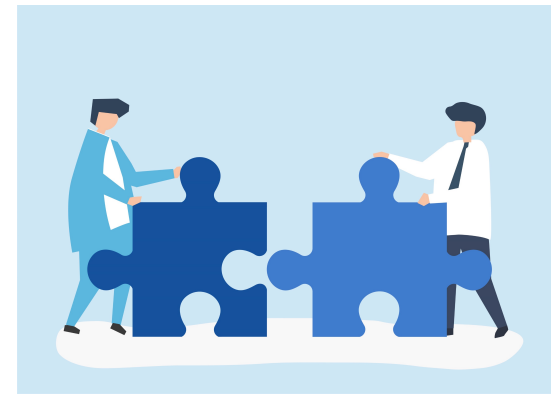
```
<a style="color:white;" href="?backgroundColor=green"><strong> Change me to Green </strong></a>
```

GET-Parameter

dynamischer Wert



Dynamische Website



Dynamisches Design mit PHP - Styles

```
$backgroundColor = "white";  
  
if ( isset ( $_GET['backgroundColor'] )) {  
    $backgroundColor = $_GET['backgroundColor'];  
}
```

Beim ersten Aufruf einer Seite wird ein Standard-Wert benötigt (z.B. für die Startseite)

```
<a style="color:white;" href="?backgroundColor=green"><strong> Change me to Green </strong></a>
```

Nach Betätigung des Links wird der dynamische Wert der Variable zugewiesen



Dynamische Website

Dynamisches Design mit PHP - Styles

Ausgabe der Variable dort, wo der entsprechende Style verändert werden soll.

Beispiel*:

```
<body style="background-color:<?php echo $backgroundColor; ?>">
```

*URL beachten



Dynamisches Design mit PHP – Content-Steuerung



Dynamische Website

Dynamisches Design mit PHP – Content-Steuerung

Anforderung:

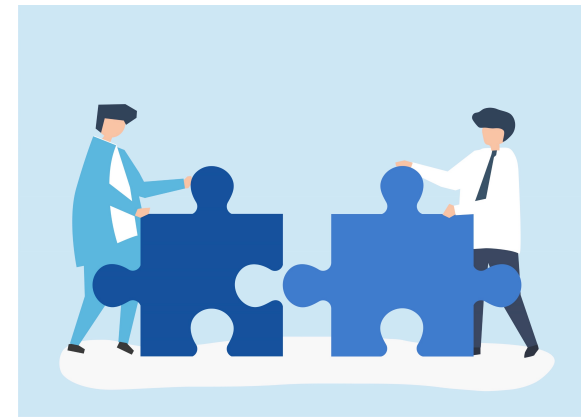
Erst nach einer Benutzerinteraktion sollen die Inhalte zukünftig dynamisch in die Website eingebunden und angezeigt werden

IST-Stand:

- Ein Template ist vorhanden,
- Navigation besitzt Buttons ohne Funktion

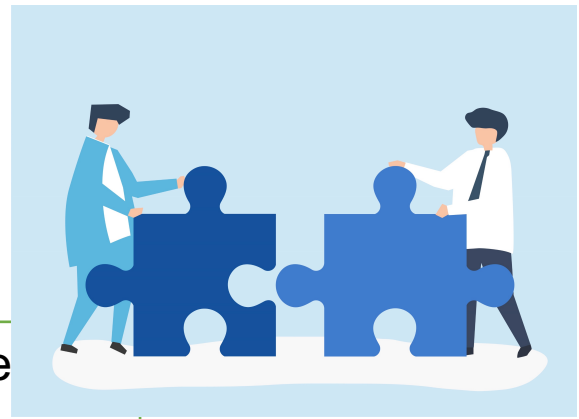
Lösungsansatz:

- Navigation anpassen (Buttons die auf Aktionen reagieren)
- Variablen in PHP definieren, die eine Steuerung der Inhalte ermöglichen
- Kontrollstrukturen implementieren, die die Variablen auswerten
- Dateien mit Inhalten bereitstellen



Dynamische Website

Dynamisches Design mit PHP – Content-Steuerung



- Die dynamische Content-Steuerung erfolgt über HTTP mit Hilfe der GET-Parameter
- GET-Parameter sind nicht an Formulare gebunden.
- **<a>-Tag** zur Verlinkung von Dateien/Webseiten, kann GET-Parameter übertragen

Beispiel:

```
<a href="?content=output"><strong> Ausgabe </strong></a>
```



GET-Parameter

dynamischer Wert

URL:

localhost/beispiele/?content=output



Dynamische Website

Dynamisches Design mit PHP – Content-Steuerung



```
//Standard-Content
$content = 'start';
if ( isset ( $_GET['content'] ) ) {
    $content = $_GET['content'];
}
?>
```

Beim ersten Aufruf einer Seite wird ein Standard-Wert benötigt (z.B. für die Startseite)

```
<a href="?content=output"><strong> Ausgabe </strong></a>
```

Nach Betätigung des Links wird der dynamische Wert der Variable zugewiesen



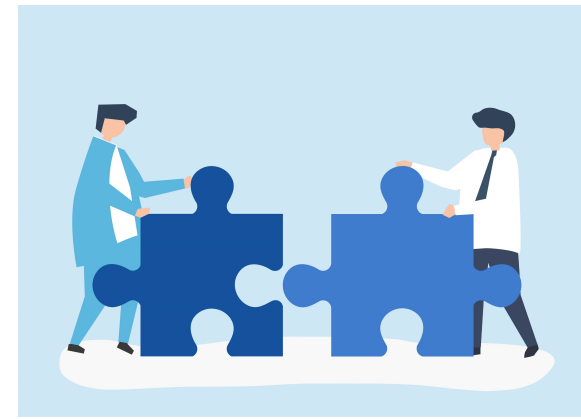
Dynamische Website

Dynamisches Design mit PHP – Content-Steuerung

- Auswertung der Content-Variable mit Hilfe der Kontrollstruktur
- Einbindung der geforderten Inhalte mittels **include_once()**;

```
<div id="contentWrapper" class="classWrapper">

    <h1>Content</h1>
    <?php
        //Content prüfen und entsprechenden Inhalt setzen
        if ($content == 'start'){
            include_once ('content/start.php');
        }
        else if ($content == 'output'){
            include_once ('content/output.php');
        }
        else {
            echo "kenn ich nicht...";
        }
    </?php>
</div>
```



Dynamische Website

Dynamisches Design mit PHP – Content-Steuerung



```
<?php
//Standard-Content
$content = 'start';
if ( isset ( $_GET['content'] ) ) {
    $content = $_GET['content'];
}
?>
```

```
<div id="contentWrapper" class="classWrapper">
```

```
<h1>Content</h1>
```

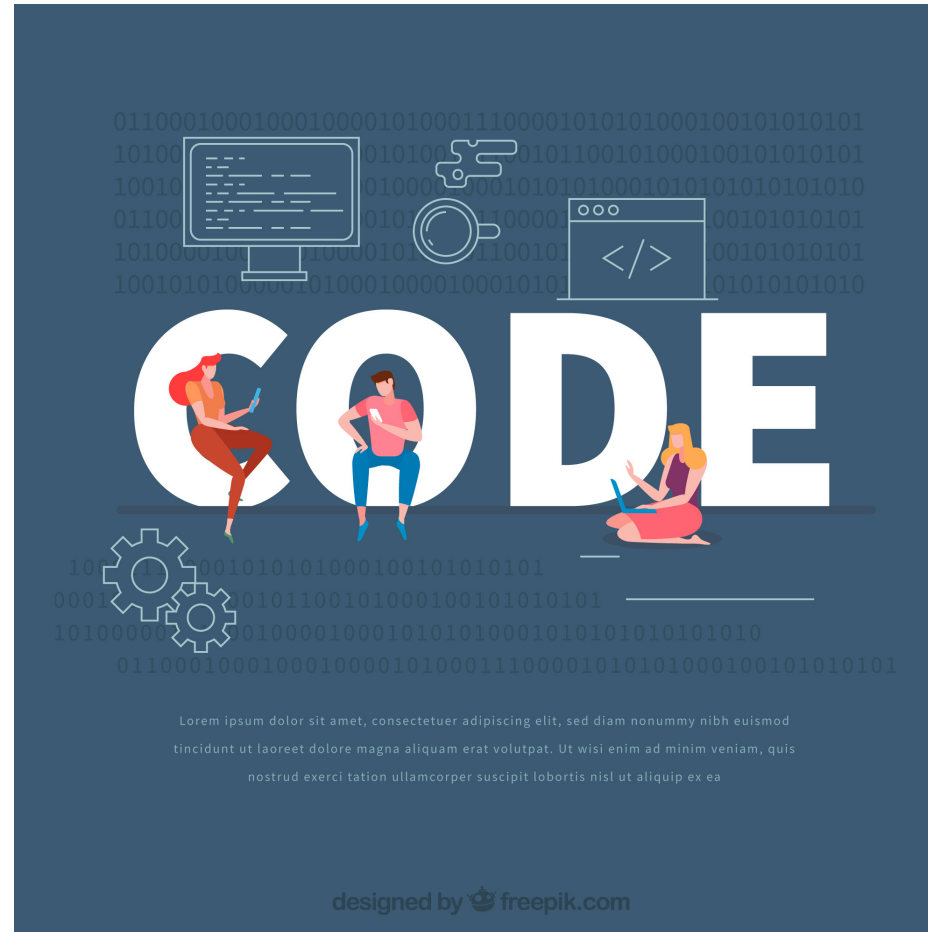
```
<?php
//Content prüfen und entsprechenden Inhalt setzen
if ($content == 'start'){
    include_once ('content/start.php');
}
else if ($content == 'output'){
    include_once ('content/output.php');
}
else {
    echo "kenn ich nicht...";
}
?>
```

```
</div>
```

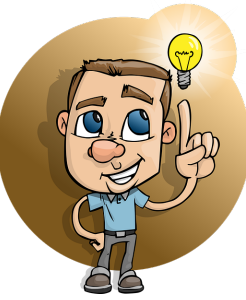
```
<div id="navWrapper" class="classWrapper">
    <h1>Navigation</h1>
</div>

<div >
    <ul>
        <a href="?content=start" class="button"> Start </a>
        <a href="?content=output" class="button"> Ausgabe </a>
    </ul>
</div>
```

Dynamisches Design mit PHP – GET-Parameter weiterleiten/wiederverwenden



Dynamische Website



Dynamisches Design mit PHP – GET-Parameter weiterleiten/ wiederverwenden

Wird ein Formular mit GET-Parametern ausgeführt, gehen z.B. die bestehenden GET-Parameter für die Inkludierung der Inhalte verloren

Die bestehenden GET-Parameter sollen in der URL erhalten bleiben

Lösung:

hidden input field

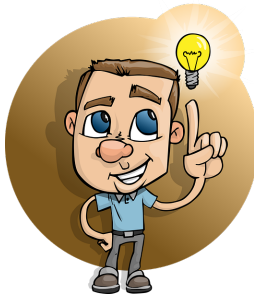
Nach Ausführung des Formulars, wird in den GET-Parameter der Value geschrieben

```
<h4> Eingabe Felder zum Ausrechnen </h4>
<form action='' method='get' id='form'>
  <input type="hidden" name="content" value="controlStructuresI" />
```

URL:

`?content=controlStructuresI&number3=4&operator=%2B&number4=4`

Dynamische Website



Dynamisches Design mit PHP – GET-Parameter weiterleiten/ wiederverwenden

Wird ein Link (<a>) ausgeführt, gehen z.B. die bestehenden GET-Parameter für die Inkludierung der Inhalte verloren

Die bestehenden GET-Parameter sollen in der URL erhalten bleiben

Lösung 1:

Statisch: GET-Parameter [hartkodiert](#) (selbstgeschrieben) zu den links hinzufügen

```
<a style="color:white;" href="?content=backgroundChange&backgroundColor=green">
```

Lösung 2:

Dynamisch: GET-Parameter mit PHP auslesen und ausgeben

```
<a style="color:white;"  
  href="?content=<?php if(isset($_GET['content'])) echo $_GET['content'];?>&backgroundColor=green">
```

Finish

Dynamische Websites

PHP III

