Einstieg in dynamische Webtechnologien

Schülerlabor Informatik | RWTH AACHEN UNIVERSITY

Runde 2: Von der Statik zur Dynamik - Team Nav

Es gibt eine neue Nachricht von Alice...

Von: mail@alice-kleiner-shop.de
An: topteam@schuelerlabor-informatik.de
Betreff: Re:Anfrage bezüglich Erstellung eines Online-Shops

Hallo, liebes Entwicklerteam!
Euer Designentwurf hat mir schon sehr gut gefallen und ich denke, wir können somit das Hauptprojekt in Angriff nehmen.

Das Team, das an der Datenbank arbeitet, hat mir nun auch eine Beschreibung der Schnittstellen zukommen lassen, sodass ihr nun eure bisherige Arbeit mit Werten aus der Datenbank füttern könnt.
Die Beschreibung der Schnittstellen befindet sich wieder im Anhang...

Gruß,
Alice

Ah - jetzt kann es also losgehen mit der Dynamik, also mit dem Einfügen von Werten, wenn die Seite angefragt wird. Um dies zu bewältigen, werden wir in diesem Abschnitt:

- ✘ Die PHP-Umgebung **'<?php ...?>'** kennenlernen.
- ✘ Den PHP-Befehl **'echo'** kennenlernen.
- ✘ Den Umgang mit **Variablen** in PHP erlernen.
- ✘ Die sogenannten **Klassen** Datenbank, Produkt und Kategorie nutzen
- ✘ **Werte** über die URL an unser Script **weiterleiten**

Der echo-Befehl

Los geht es direkt mit dem Befehl **'echo'**. Hier ist der Name Programm, denn echo gibt einfach das aus, was man ihm hineinschreibt:

```
echo "Hallo";
```

fügt also genau an der Stelle, an der es steht einfach ein Hallo ein. Doch Moment - war da nicht etwas eben mit Stellen, die markiert werden, und war das erste Ziel nicht:

- ✘ Die PHP-Umgebung **'<?php ...?>'** kennenlernen???

Genau! Denn wenn ihr den **echo**-Befehl einfach so in euer Dokument schreibt, wird dies nicht als Befehl erkannt.

Alle Befehle, die für den PHP-Interpreter (das war das Programm, was den PHP-Code übersetzt) wichtig sind, müssen in diese Klammern gesetzt werden:

```
<?php BEFEHL; ?>
```

Nur so werden sie erkannt.
Natürlich muss nicht jeder Befehl einzeln in diese Klammern gesetzt werden. Es können auch mehrere Befehle zusammengesetzt werden:

```
<?php
    BEFEHL1;
    BEFEHL2;
?>
```

Wobei man in so einem Fall am besten jeden Befehl in eine eigene Zeile packt.
Auch schon zu sehen und äußerst wichtig ist das Semikolon, das nach jedem Befehl unbedingt da stehen muss - sonst gibt's 'nen Fehler...

Also zurück zum **echo**-Befehl:
Dieser müsste also beim Ausprobieren dann in seiner Umgebung so aussehen:

```
<? echo "Hallo"; ?>
```

Probiert es am besten direkt einmal aus:

Fügt in euren Bereich den **echo**-Befehl ein und testet das Ergebnis

- ✘ Denkt immer daran, dass nach dem Ändern der Datei diese auch hochgeladen werden muss...
- ✘ Überprüft vorher jedoch immer noch einmal den Code - insbesondere auf vergessene Semikolons...

Damit habt ihr euren ersten PHP-Befehl geschrieben und ausgeführt. Aber wenn man ehrlich ist, hätte man für diese Aufgabe auch einfach "Hallo" an die entsprechende Stelle schreiben können...
Seine eigentliche Funktion entfaltet echo erst, wenn man mit Variablen arbeitet.

Variablen in PHP

Variablen sind, wie ihr eben gesehen habt, so eine Art Ablagefach für alles Mögliche, sowohl für einfache Werte wie 3 oder "Hallo", als auch für ganze Stapel von Anweisungen (sogenannte Funktionen).
Variablen werden in PHP immer mit einem '$' markiert.

Um einen Wert einer Variable zuzuweisen schreibt man ihn hinter die Variable mit einem Gleichheitszeichen:

```
<?php $Gruss = "Hallo"; ?>
```

Schreibt man nun

```
<?php
    $Gruss = "Hallo";
    echo $Gruss;
?>
```

, so wird an der Stelle wieder 'Hallo' ausgegeben. Doch damit noch nicht genug.
Man kann Variablen und Werte auch kombinieren, indem man sie mit einem Punkt verbindet:

```
<?php
    $Gruss = "Hallo";
    $Welt = "Welt";
    $InfoSphere = "InfoSphere";
    echo $Gruss." ".$Welt;
    echo $Gruss." ".$InfoSphere;
?>
```

führt zu: **'Hallo WeltHallo InfoSphere'**. Dieses Beispiel zeigt auch: echo gibt wirklich nur das aus, was man ihm sagt. Wenn man also ein Leerzeichen haben möchte, so muss man es selbst ergänzen.

Wichtig ist, dass ihr bei den Variablen auf Groß- und Kleinschreibung achtet...
$Hallo ist für PHP eine ganz andere Variable als **$hallo** ...

Doch jetzt seid ihr erst einmal wieder dran:

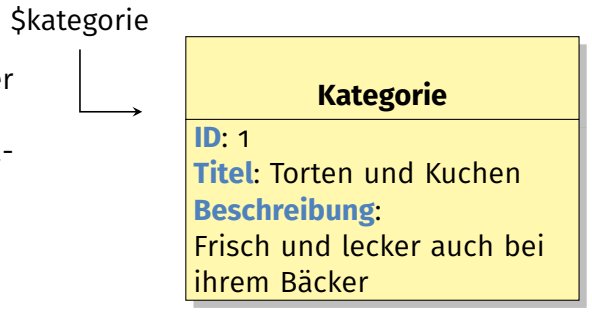
1. Erstellt mehrere Variablen und weist ihnen Werte zu.
2. Baut aus den Variablen nun Sätze, die ihr dann mit dem echo-Befehl ausgebt.
3. **Kleiner Tipp**: Man kann die Variablen auch in einer anderen '<?php ...?>' Umgebung erstellen, als man sie nutzt, z.B.:

```
<?php $Variable=1; ?>
Der Wert von Variable ist: <?php echo $Variable; ?>
```

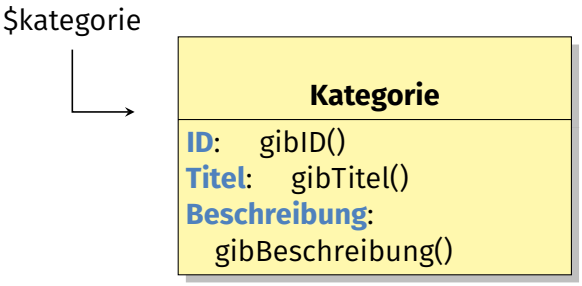
Produkte und Kategorien in Variablen

Neben Zahlen und Zeichenketten kann man auch noch mehr in einer Variablen speichern. Dieses Konzept nennt man Klasse. Was das genau ist, lernt ihr in der Oberstufe.
Man kann sich das aber in etwa so vorstellen:

Statt der Zahl oder der Zeichenkette wird in der Variablen ein Pfeil gespeichert, der auf einen Zettel zeigt. Auf dem Zettel stehen dann die Daten der Kategorie oder des Produktes.



Doch wie kommt man nun an die Werte auf dem Zettel?
Hierzu gibt es Funktionen die, zu dem Zettel gehören:



Über den Pfeil -> kann man von der Variablen auf die Funktion zugreifen.
Um zum Beispiel an den Titel zu kommen schreibt man:

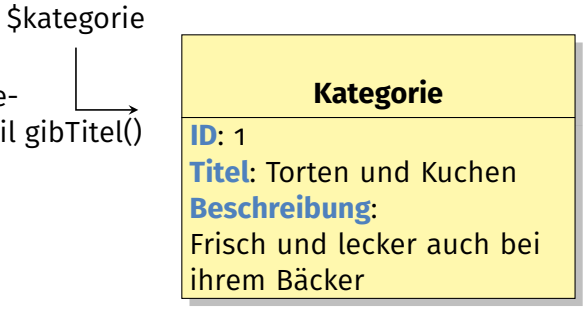
```
$kategorie->gibTitel();
```

Welche Funktionen es für die unterschiedlichen Klassen gibt, seht ihr auf dem **Beiblatt 'Klassen'**.
Um das ganze zu verdeutlichen hier einmal ein Beispiel für die Ausgabe des Titels einer Kategorie:
Nehmen wir an, wir hätten eine Variable **$kategorie**, die bereits eine Kategorie enthält (wie man dahin kommt, sehen wir später).
Nun soll der Titel ausgegeben werden:

```
<?php echo $kategorie->gibTitel(); ?>
```

Nun passiert folgendes:

$kategorie verweist auf den Zettel mit der Kategorie. Die Funktion gibTitel() folgt also dem Pfeil gibTitel() und schaut nach, was bei **Titel** steht.



PHP macht also folgende Schritte:

```
<?php echo $kategorie->gibTitel(); ?>
        ⇓ wird zu
<?php echo "Torten und Kuchen"; ?>
        ⇓ wird zu
Torten und Kuchen
```

Und wie kommt das Ganze in die Variable?

Bis jetzt sind wir ja einfach davon ausgegangen, dass unsere Kategorie bereits in der Variablen ist. Doch wie bekommt man es nun dort hinein?
Hierfür gibt es eine für euch vorgefertigte Variable namens $datenbank.

Auf dem **Beiblatt 'Klassen'** findet ihr in dem Kästchen zur **Datenbank** wieder eine ganze Liste von gib...()-Funktionen - insbesondere auch die Funktion gibKategorieMitID($ID).
Diese Funktion schaut in der Datenbank nach der Kategorie und gibt diese an die davorstehende Variable weiter

Wollen wir nun also die Kategorie mit der Nummer 1 aus der Datenbank holen und in unserer Variable $kategorie speichern, so geht dies, indem man wieder die Funktion mit gibKategorieMitID($ID) mit dem -> an die Variable $datenbank hängt:

```
<? $kategorie = $datenbank->gibKategorieMitID(1); ?>
```

Doch zurück zur Frage, wie die Variable nun gefüllt wird:
Die Werte von Kategorie 1 sind nun von der Datenbank in die Variable $kategorie geladen worden und können durch z.B. die Funktion gibTitel() abgefragt werden:

```
<? $kategorie = $datenbank->gibKategorieMitID(1); ?>
<h1><?php echo $kategorie->gibTitel(); ?></h1>
```

Dann kommen wir von der Theorie mal zur Praxis:

1. Erstellt am Anfang der Datei nav.php eine Variable und füllt sie mit der Kategorie mit der Nummer 1 aus der Datenbank.
2. Gebt den Titel des Produktes an der entsprechenden Stelle in eurer bisher statischen Website aus.
3. Schaut euch das Ergebnis im Browser an.
4. Ladet nun die Kategorie mit der Nummer 2 aus der Datenbank, indem ihr die 1 aus $datenbank->gibKategorieMitID(1) durch eine zwei ersetzt.
5. Schaut euch das Ergebnis erneut im Browser an.
6. Ändert nun die restlichen Stellen in eurer Website, wo bis jetzt noch die Testdaten standen.
7. Testet nun das Gesamtergebnis auf eurer Seite mit verschiedenen Produkten aus der Datenbank.

Werte an den Server übermitteln

Eure Seite wird zwar nun bereits mit Werten aus der Datenbank gefüllt, aber eine wirkliche Erleichterung bietet das natürlich noch nicht, wenn jedesmal die Website geändert und hoch geladen werden muss.Um dieses Problem zu lösen gibt es die Möglichkeit Werten per URL zu übergeben.

Runde 2: Von der Statik zur Dynamik - Team Nav

Vielleicht habt ihr das ja schon einmal gesehen z.B. bei der Suche im Internet:

https://www.google.de/search?q=Schuelerlabor+Informatik&gws_rd=cr,ssl&ei=CIwzVafaLcPLaMGfg

Alles hier Markierte sind Daten, die in diesem Fall an Google übergeben werden.
Was man an diesem Beispiel schon schön sehen kann, ist der Aufbau der URL, wenn hier Daten mitgegeben werden sollen:

- ✘ **?** markiert den Start der Daten - alles was hiernach folgt, wird als Variablen und Werte angesehen.
- ✘ **variable=wert** sind dann die einzelnen Kombinationen aus Variablennamen und Werten, die übergeben werden.
- ✘ **&** nutzt man, um mehrere Variablen/Werte zu trennen

Besprecht euch mit eurem Header Team und eurem Main Team und legt zwei Namen für die Variablen fest mit denen die Nummer der Kategorie und des Produktes übergeben werden sollen.

Nachdem ihr im Team Variablennamen festgelegt habt, kommen wir nun zu deren Verwendung:
Bis jetzt habt ihr beim Testen eurer Seite ja die Seite

http://shopXY.schuelerlabor.informatik.rwth-aachen.de/

aufgerufen, wobei XY eurer Teamnummer entspricht.
Wenn ihr nun stattdessen

http://shopXY.schuelerlabor.informatik.rwth-aachen.de/?<Kategorienummer>=1

, wobei **<Kategorienummer>** durch euren gewählten Variablennamen ersetzt wird, aufruft, so übergebt ihr damit euren ersten Wert an euere Website...

Doch wie arbeitet man nun damit?
Um an den übergebenen Wert zu kommen, gibt es in PHP eine ganz besondere Variable: **$_GET**. Diese Variable ist eigentlich wieder eine ganze Reihe von Variablen. Für uns ist an der Stelle nur wichtig, dass man in eckigen Klammern angeben kann, welchen übergebenen Wert man nutzen möchte:

```
$_GET['<Kategorienummer>']
```

1. Ersetzt die Stelle, die ihr bis jetzt von Hand bearbeitet habt, durch die neue Variable $_GET['<Kategorienummer>'].
2. Testet eure Website nun mit der neuen Adresse und tauscht an der Stelle die Nummern aus

Wow, damit habt ihr den wesentlichen Teil der Dynamik einer Website verstanden!!!
Wieder könnt ihr verbliebene Zeit für weitere Verschönerungen an eurer Seite nutzen.

Das nächste ist dann wohl ein bisschen Feintuning...