

Refugeeks / IntoCode

Dynamische Websites

23.09.2020 / 05:

Array, Schleifen, Datenbank-Anbindung



Dynamische Website

Allgemeines – Increment & Decrement

Inkrement:

- Verkürzte Schreibweise für die Erhöhung eines Integer oder String-Wertes in einer Variable

Vorher

```
// Initial Wert  
$i = 1;  
  
// Erhöhung um 1  
$i = $i + 1;
```

Decrement:

- Verkürzte Schreibweise für die Minderung eines Integer-Wertes String-Wertes in einer Variable

Vorher

```
// Initial Wert  
$i = 5;  
  
// Minderung um 1  
$i = $i - 1;
```

Häufigste Verwendung bei Schleifen um die Anzahl von Schleifen-Durchgängen zu steuern



Dynamische Website

Allgemeines – Increment

2 Variationen:

Prä-Inkrement:

```
echo "<h4> Increment </h4>";  
// Prä-Increment - Beispiel  
$i = 1;  
  
echo "Ausgabe Initial-Wert: ".$i;  
echo "<br>";  
  
echo "Ausgabe mit Prä-Inkrement: " . ++$i;  
echo "<br>";
```

Post-Inkrement:

```
// Post-Increment - Beispiel  
$i = 1;  
  
echo "Ausgabe Initial-Wert: ".$i;  
echo "<br>";  
  
echo "Ausgabe mit Post-Inkrement vorher: " . $i++;  
echo "<br>";  
  
echo "Ausgabe mit Post-Inkrement nachher: " . $i++;  
echo "<br>";
```



Dynamische Website

Allgemeines – Decrement

2 Variationen:

Prä-Dekrement:

```
echo "<h4> Decrement </h4>";  
// Prä-Dekrement - Beispiel  
$i = 5;  
  
echo "Ausgabe Initial-Wert: ".$i;  
echo "<br>";  
  
echo "Ausgabe mit Prä-Inkrement: ".$i;  
echo "<br>";
```

Post-Dekrement:

```
// Post-Dekrement - Beispiel  
$i = 5;  
  
echo "Ausgabe Initial-Wert: ".$i;  
echo "<br>";  
  
echo "Ausgabe mit Post-Inkrement vorher: ".$i;  
echo "<br>";  
  
echo "Ausgabe mit Post-Inkrement nachher: ".$i;
```



Dynamische Website

Allgemeines – Arithmetischer Operator: Modulus

- Rechenoperation, die den Rest einer Division zurückgibt

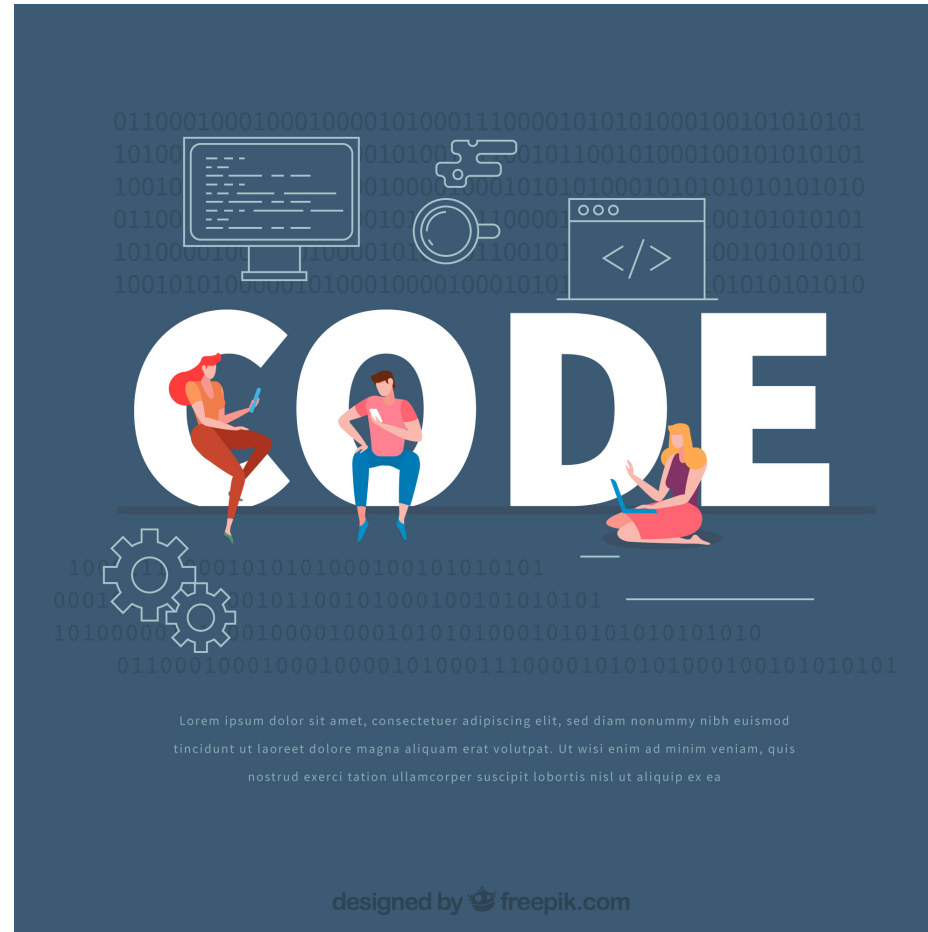
```
// Modulus-Beispiel  
$number1 = 8;  
  
$number2 = 4;  
  
$result = $number1 % $number2;  
  
echo "Modulus ist: ".$result;
```

Modulus ist: 0

```
// Modulus-Beispiel  
$number1 = 9;  
  
$number2 = 4;  
  
$result = $number1 % $number2;  
  
echo "Modulus ist: ".$result;
```

Modulus ist: 1

Dynamische Websites – Vars IV - Array



Dynamische Website

PHP – Variablen - Datentypen

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



Dynamische Website

Variablen – Definition & Arten von Arrays

- Ermöglicht die Speicherung von mehreren Werten innerhalb einer Variable
- Arrays werden z.B. dann wichtig, wenn die Datenmengen aus Datenbankzugriffen stark zunehmen

Arten von Arrays:

- Indizierte Arrays
- Assoziative Arrays
- Mehrdimensionale Arrays



Dynamische Website

Variablen – Indexierte Arrays (Standard-Form)

- Jeder Wert wird einem Index(Zahl) zugeordnet – automatisch und manuell
- Beispiel für automatisch – Index beginnt immer bei **0**:

```
$penColor = "blau"; // einfache String-Variable mit einer Farbe  
$penColors = array("blau", "rot", "grün"); // Array-Variable mit mehreren Farben
```

- Beispiel für manuell:

```
$penColors[0] = "blau"; // Wertzuweisung an der Stelle 0 des Arrays  
$penColors[1] = "rot";  // Wertzuweisung an der Stelle 1 des Arrays  
$penColors[2] = "grün"; // Wertzuweisung an der Stelle 2 des Arrays
```



Dynamische Website

Variablen – Indexierte Arrays (Standard-Form)



- Ausgabe :

```
echo "Die Stiftfarbe ist: <strong style='color: blue'>".$penColors[0]."</strong><br>";  
echo "Die Stiftfarbe ist: <strong style='color: red'>".$penColors[1]."</strong><br>";  
echo "Die Stiftfarbe ist: <strong style='color: green'>".$penColors[2]."</strong><br>";
```

Indizierte Arrays

Die Stiftfarbe ist: **blau**

Die Stiftfarbe ist: **rot**

Die Stiftfarbe ist: **grün**



Dynamische Website

Variablen – Assoziative Arrays

- Jeder Wert wird einem Schlüssel zugeordnet (Key – Value)
- Zwei Möglichkeiten für Key-Value-Zuweisungen

Beispiel I:

```
// Assoziative Zuweisung mit array-Funktion
$numberOfPens = array("bluePen" => 2, "redPen" => 10, "greenPen" => 0);
```

Beispiel II:

```
// Direkte assoziative Zuweisung
$numberOfPens['bluePen'] = 2;
$numberOfPens['redPen'] = 10;
$numberOfPens['greenPen'] = 0;
```



Dynamische Website

Variablen – Assoziative Arrays



- Ausgabe

```
echo "Anzahl der blauen Stifte: <strong style='color: blue'>".$numberOfPens['bluePen']."</strong><br>";  
echo "Anzahl der roten Stifte: <strong style='color: red'>".$numberOfPens['redPen']."</strong><br>";  
echo "Anzahl der grünen Stifte: <strong style='color: green'>".$numberOfPens['greenPen']."</strong><br>";
```

Assoziative Arrays

Anzahl der blauen Stifte: **2**
Anzahl der roten Stifte: **10**
Anzahl der grünen Stifte: **0**



Dynamische Website

Variablen – Mehrdimensionale Arrays

- Einer Array-Variable können auch Array-Variablen zugewiesen werden
- Beispiel zwei-dimensionales Array:

```
// Zwei-dimensionales Array (pen => [0] = Farbe, [1] = Anzahl)
$pen[0] = array("blue", 2);
$pen[1] = array("red", 10);
$pen[2] = array("green", 0);

$pens = array($pen[0], $pen[1], $pen[2]);
```



Dynamische Website

Variablen – Mehrdimensionale Arrays



- Ausgabe

```
echo "Stiftfarbe: <strong style='color: blue'>".$pens[0][0]."</strong> Anzahl: ".$pens[0][1]."<br>";  
echo "Stiftfarbe: <strong style='color: red'>".$pens[1][0]."</strong> Anzahl: ".$pens[1][1]."<br>";  
echo "Stiftfarbe: <strong style='color: green'>".$pens[2][0]."</strong> Anzahl: ".$pens[2][1]."<br>";
```

Mehrdimensionale Arrays

Stiftfarbe: **blue** Anzahl: 2
Stiftfarbe: **red** Anzahl: 10
Stiftfarbe: **green** Anzahl: 0



Dynamische Websites – Kontrollstruktur II - Schleifen



Dynamische Website

Schleifen - Allgemeines

Wozu Schleifen?

- Anstatt bestimmten Code ständig selbst zu wiederholen, kann dieser mit Schleifen wiederholt werden.
- Dieser Code wird ausgeführt solange eine Bedingung erfüllt ist
 - für eine selbst definierte Anzahl von Wiederholungen
 - für eine unbestimmte Anzahl von Wiederholungen

Es gibt verschiedene Schleifen-Typen:

- For-Schleife
- While-Schleife
- Do-While-Schleife
- Foreach-Schleife



Dynamische Website

Schleifen – Ohne Schleife

Code ohne Schleife

```
echo "<h4> Ausgabe ohne Schleife </h4>";  
echo "Ohne Schleife Ausgabe Nr.: 0";  
echo "<br>";  
echo "Ohne Schleife Ausgabe Nr.: 1";  
echo "<br>";  
echo "Ohne Schleife Ausgabe Nr.: 2";  
echo "<br>";  
echo "Ohne Schleife Ausgabe Nr.: 3";  
echo "<br>";  
echo "Ohne Schleife Ausgabe Nr.: 4";  
echo "<br>";  
echo "Ohne Schleife Ausgabe Nr.: 5";  
echo "<br>";
```

Ausgabe

Ausgabe ohne Schleife

Ohne Schleife Ausgabe Nr.: 0
Ohne Schleife Ausgabe Nr.: 1
Ohne Schleife Ausgabe Nr.: 2
Ohne Schleife Ausgabe Nr.: 3
Ohne Schleife Ausgabe Nr.: 4
Ohne Schleife Ausgabe Nr.: 5

Dynamische Website

for-Schleife

- Häufige Verwendung bei festgelegter Anzahl von Wiederholungen

Beispiel:

```
echo "<h4> Ausgabe mit For-Schleife </h4>";  
  
for ($i = 0; $i < 6;$i++){  
    echo "For-Schleife Ausgabe Nr.: ".$i;  
    echo "<br>";  
}
```

Ausgabe:

Ausgabe mit For-Schleife

For-Schleife Ausgabe Nr.: 0
For-Schleife Ausgabe Nr.: 1
For-Schleife Ausgabe Nr.: 2
For-Schleife Ausgabe Nr.: 3
For-Schleife Ausgabe Nr.: 4
For-Schleife Ausgabe Nr.: 5



Dynamische Website

while-Schleife

- Schleife wird solange ausgeführt, wie eine Bedingung erfüllt ist – ähnlich wie beim `if($bedingung)`

Beispiel:

```
echo "<h4> Ausgabe mit While-Schleife </h4>";

$loops = 0;

while($loops < 6) {
    echo "While-Schleife Ausgabe Nr.: ".$loops;
    echo "<br>";
    $loops++;
}
```

Ausgabe:

Ausgabe mit While-Schleife

While-Schleife Ausgabe Nr.: 0
While-Schleife Ausgabe Nr.: 1
While-Schleife Ausgabe Nr.: 2
While-Schleife Ausgabe Nr.: 3
While-Schleife Ausgabe Nr.: 4
While-Schleife Ausgabe Nr.: 5



Dynamische Website

do-while-Schleife

- Code wird immer mind. 1x ausgeführt.
- Ein abschließendes **while(\$bedingung)** prüft, wie lange die Schleife daraufhin fortgeführt wird

Beispiel:

```
echo "<h4> Ausgabe mit Do-While-Schleife </h4>";  
  
$loops = 0;  
  
do{  
    echo "Do-While-Schleife Ausgabe Nr.: ".$loops;  
    echo "<br>";  
    $loops++;  
}while($loops < 6);
```

Ausgabe:

Ausgabe mit Do-While-Schleife

Do-While-Schleife Ausgabe Nr.: 0
Do-While-Schleife Ausgabe Nr.: 1
Do-While-Schleife Ausgabe Nr.: 2
Do-While-Schleife Ausgabe Nr.: 3
Do-While-Schleife Ausgabe Nr.: 4
Do-While-Schleife Ausgabe Nr.: 5

Dynamische Website

foreach-Schleife

Diese Schleife funktioniert nur mit Arrays
Jeder Wert innerhalb eines Arrays wird einmal durchlaufen

Beispiel:

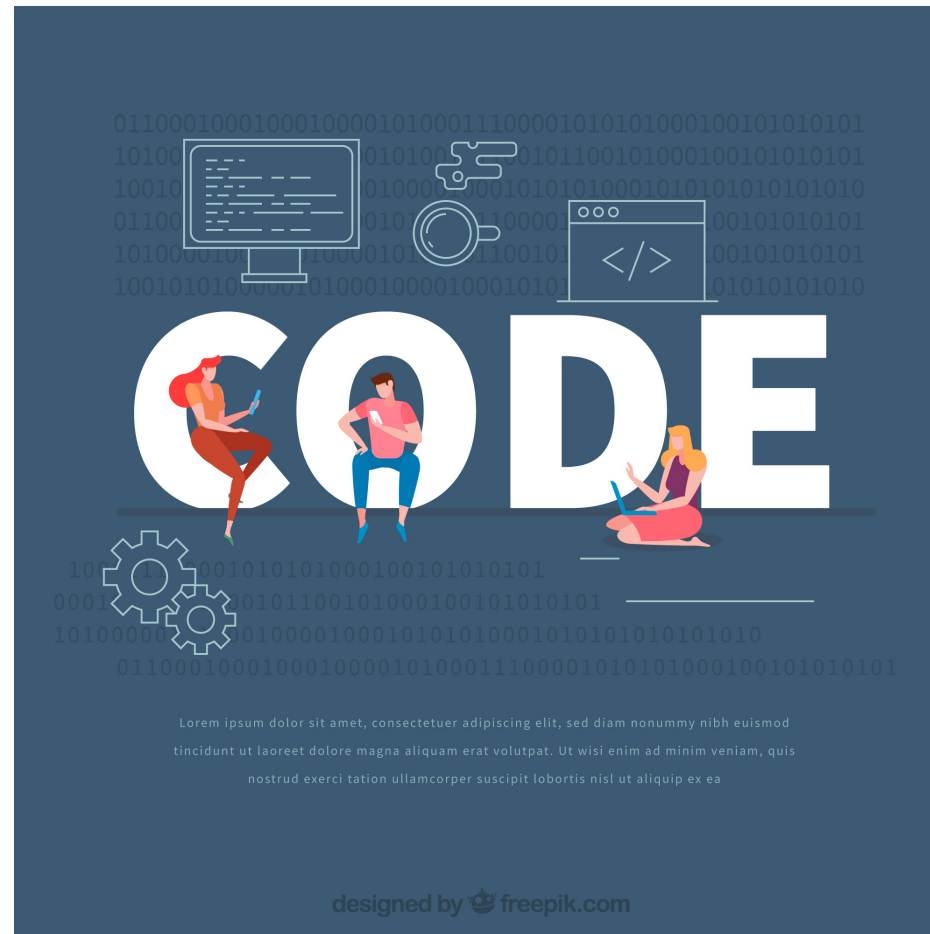
```
echo "<h4> Ausgabe mit Foreach-Schleife </h4>";  
  
$heroes = array("Luke", "Leia", "Han", "Rey");  
  
foreach ($heroes as $hero){  
    echo "Foreach-Schleife Ausgabe Helden-Name: ".$hero;  
    echo "<br>";  
}
```

Ausgabe:

Ausgabe mit Foreach-Schleife

Foreach-Schleife Ausgabe Helden-Name: Luke
Foreach-Schleife Ausgabe Helden-Name: Leia
Foreach-Schleife Ausgabe Helden-Name: Han
Foreach-Schleife Ausgabe Helden-Name: Rey

Datenbank-Anbindung



Dynamische Website

Datenbank-Anbindung

Anforderung:

An ein bestehendes System soll eine Datenbank angebunden werden um die Persistenz von Daten zu ermöglichen

IST-Stand:

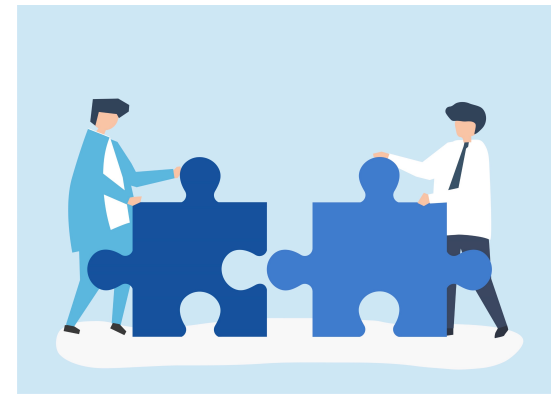
MySQL-Datenbank vorhanden aber keine Anbindung

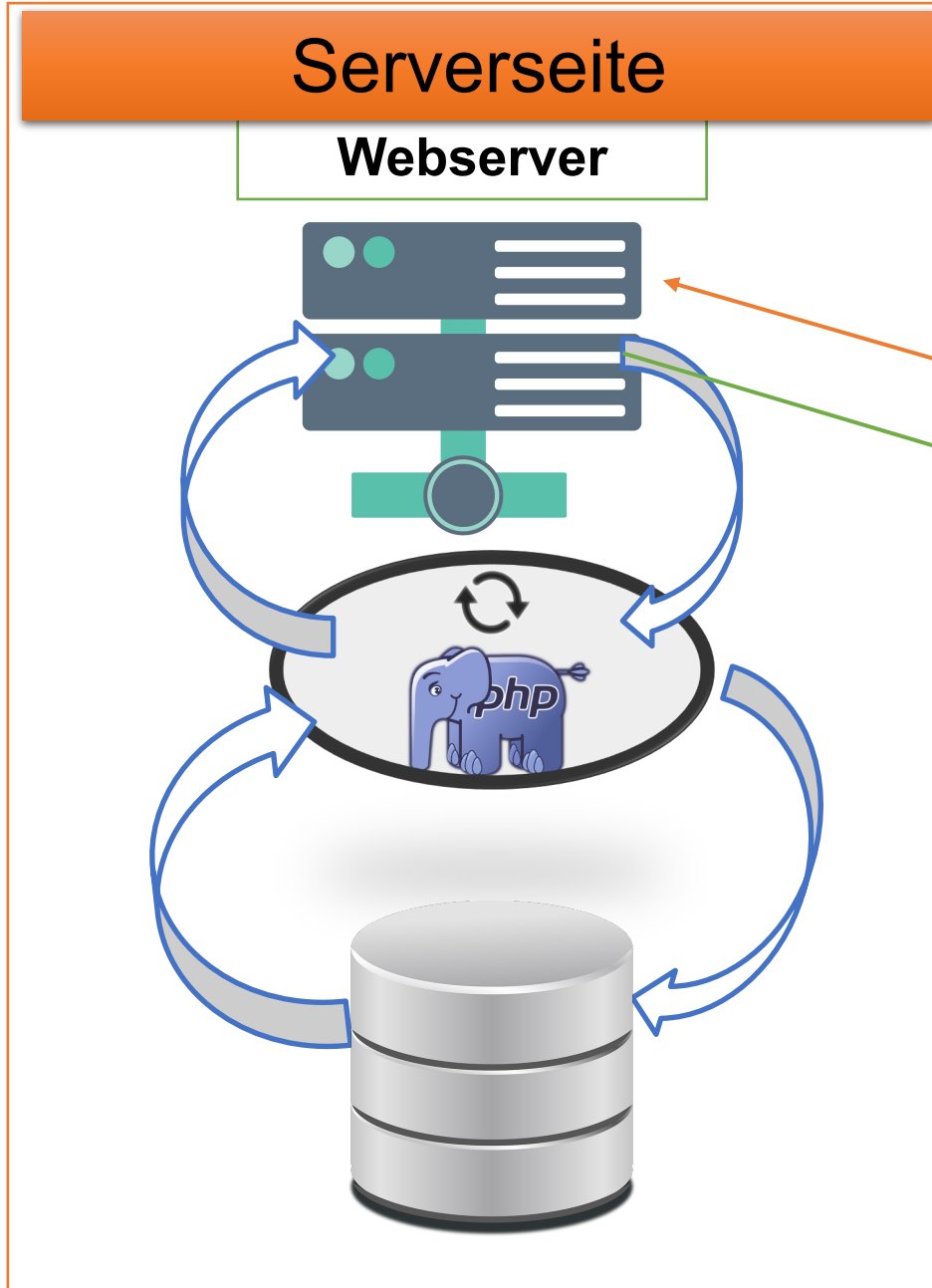
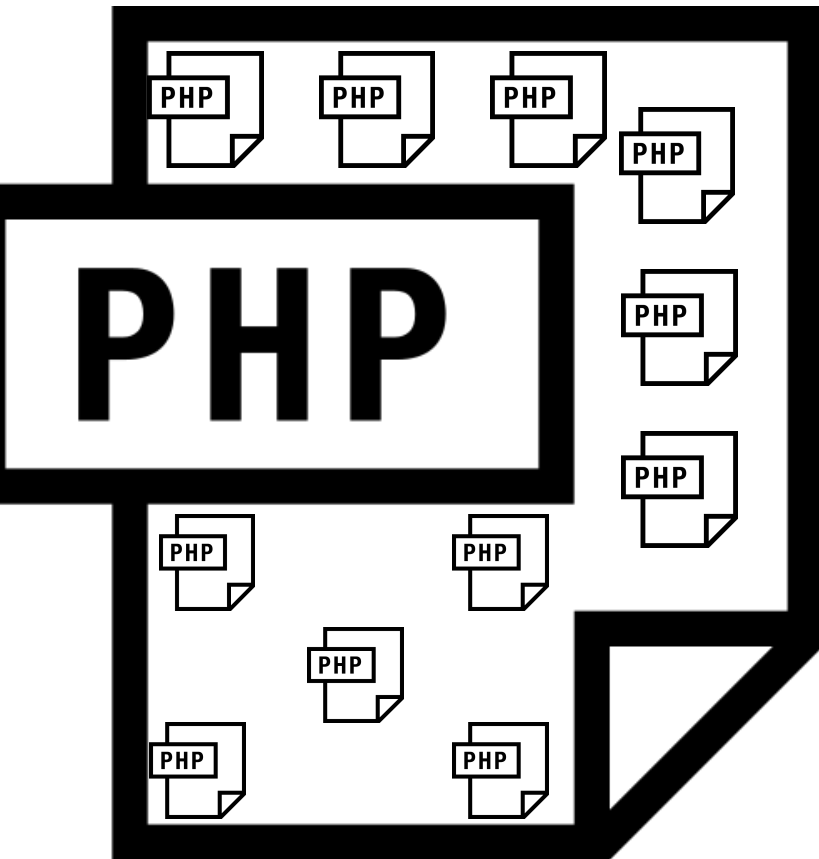
Lösungsansatz/Voraussetzungen:

Verbindungsdetails klären

- Hostname (Server auf dem sich die DB befindet)
- Username
- Passwort
- Datenbankname

Auswahl einer PHP-Funktion zur DB-Anbindung





Dynamische Website

Datenbank-Anbindung – Funktionen

- MySQLi
 - Nur für Anbindung von MySQL-Datenbanken
 - Objekt-orientierte oder prozedurale Nutzung möglich
- PDO (PHP Data Objects)
 - Anbindung viele verschiedener Datenbanken
 - Nur objekt-orientierte Nutzung möglich
 - Vorteilhaft, wenn die Datenbank ausgetauscht wird



Dynamische Website

Datenbank-Anbindung – MySQLi (Objekt-orientiert)

- Zuweisung der Verbindungsdetails
- Übergabe der Verbindungsdetails zur Verbindungsherstellung
- Speichern der Verbindung in einer Variable

```
$host = "webhosting-db";  
$username = "deinUsername";  
$password = "deinPasswort";  
  
// Create connection  
$conn = new mysqli($host, $username, $password);  
  
// Check connection  
if ($conn->connect_error) {  
    die("Verbindung fehlgeschlagen: " . $conn->connect_error);  
}  
else{  
    echo "Erfolgreich verbunden";  
}
```



Dynamische Website

Datenbank-Anbindung – MySQLi (Prozedural)

- Zuweisung der Verbindungsdetails
- Übergabe der Verbindungsdetails zur Verbindungsherstellung
- Speichern der Verbindung in einer Variable

```
$host = "webhosting-db";
$username = "deinUsername";
$password = "deinPasswort";

// Create connection
$conn = mysqli_connect($host, $username, $password);

// Check connection
if (!$conn) {
    die("Verbindung fehlgeschlagen: " . mysqli_connect_error());
}
else{
    echo "Erfolgreich verbunden";
}
```



Dynamische Website

Datenbank-Anbindung – PDO

- Zuweisung der Verbindungsdetails
- Übergabe der Verbindungsdetails zur Verbindungsherstellung
- Speichern der Verbindung in einer Variable

```
$host = "webhosting-db";
$username = "deinUsername";
$password = "deinPasswort";
$databasename = "deinDatenbankName";

// Verbindung herstellen
try {
    $conn = new PDO("mysql:host=$host;dbname=$databasename", $username, $password);
    // Error-Mode für die Exception festlegen
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    echo "Erfolgreich verbunden";
}
catch(PDOException $e)
{
    echo "Verbindung fehlgeschlagen: " . $e->getMessage();
}
```



Dynamische Website

Datenbank-Anbindung – Verbindungsdetails auslagern

PHP-Datei mit Verbindungsdetails bereitstellen:

Dateiname: config.inc.php

Inhalt:

```
// Daten befinden sich auf dem Webserver in der Datei README
$sql['host'] = 'webhosting-db';
$sql['uid'] = 'deinUsername';

$sql['pwd'] = 'deinPasswort';
$sql['db'] = 'deineDatenbank';
```



Finish

Dynamische Websites

PHP IV

