Dynamic site

General – Increment & Decrement

Increment:

- Shorthand notation for increasing an integer or string value in a variable

Beforehand

```
// Initial Wert
$i = 1;

// Erhöhung um 1
$i = $i + 1;
```

Decrement:

- Abbreviated notation for decrementing an integer value in a string value variable

Beforehand

```
// Initial Wert
$i = 5;

// Minderung um 1
$i = $i - 1;
```

Most common use in loops to control the number of loop iterations

Dynamic site

General - Increment

2 variations:

pre-increment:

```
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 increment:

```
// 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>";
```

Dynamic site

General - Decrement

2 variations:

pre-decrement:

```
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 decrement:

```
// 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--;
```

Dynamic site

General – Arithmetic operator: modulus

- Arithmetic operation that returns the remainder of a division

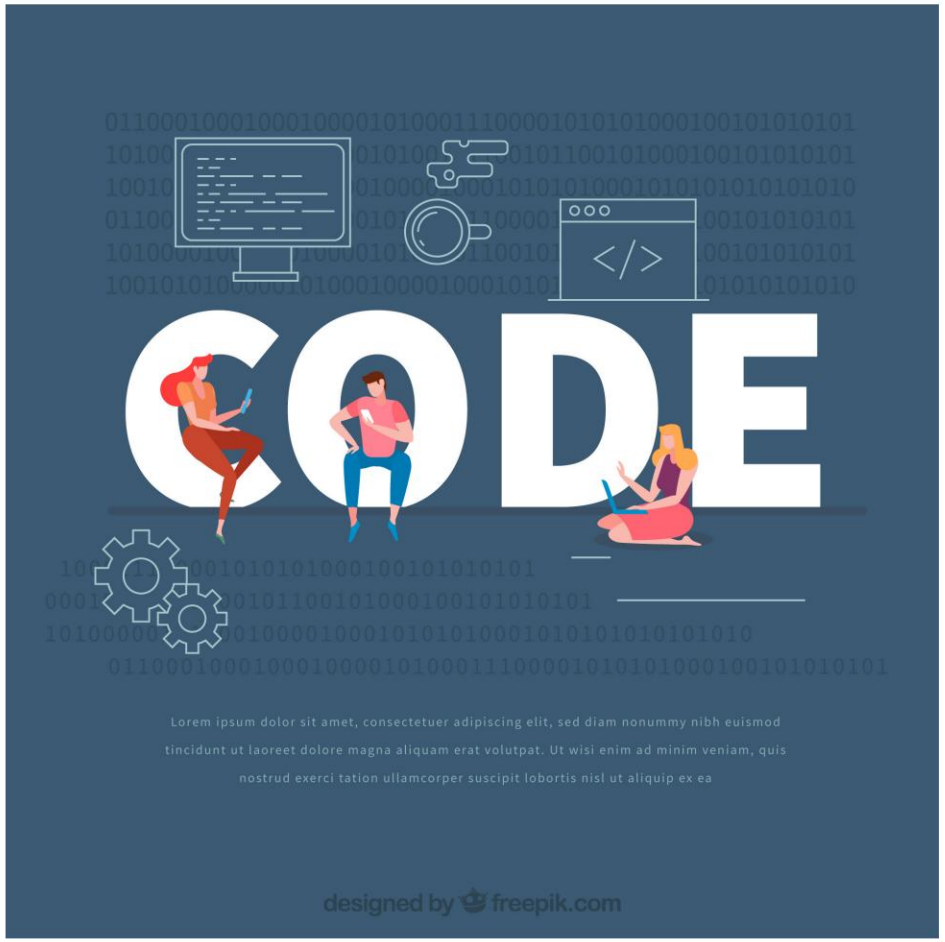
```php
// Modulus-Beispiel
$number1 = 8;

$number2 = 4;

$result = $number1 % $number2;

echo "Modulus ist: ".$result;
```

Modulus ist: 0

```php
// Modulus-Beispiel
$number1 = 9;

$number2 = 4;

$result = $number1 % $number2;

echo "Modulus ist: ".$result;
```

Modulus ist: 1

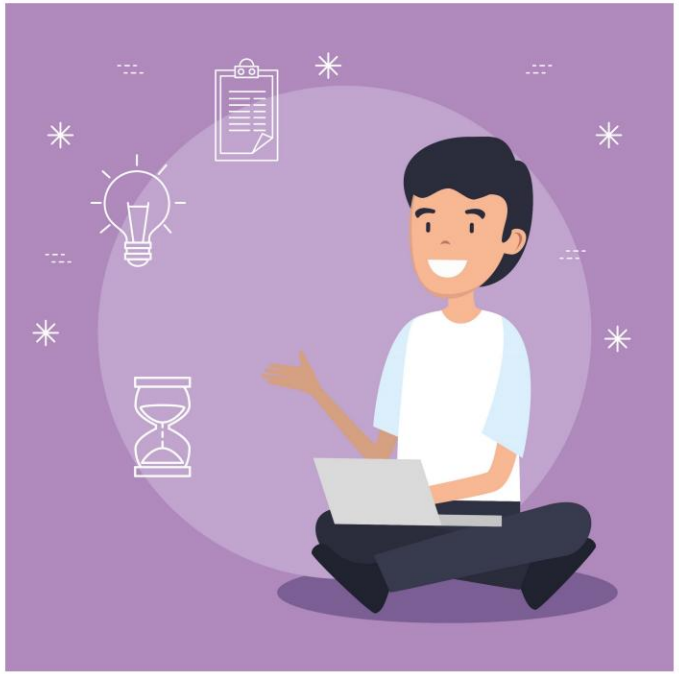
Dynamic websites – Vars IV - Array



Dynamic site

PHP – Variables - Data Types

- Integers
- Floating point numbers
- string
- Booleans
- Arrays



Dynamic site

Variables - definition & types of arrays

- Allows multiple values to be stored within one variable
- Arrays become important, for example, when the amount of data from Database access is increasing

Types of Arrays:

- Indexed arrays
- Associative arrays
- Multidimensional arrays

Dynamic site

Variables - Indexed Arrays (Standard Form)

- Each value is assigned an index (number) - automatically and manually
- Example for automatic - index always starts at 0:

```
$penColor = "blau"; // einfache String-Variable mit einer Farbe

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

- Example for manual:

```
$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
```

Dynamic site

Variables - Indexed Arrays (Standard Form)

- Output :

```
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**

Dynamic site

Variables - Associative Arrays

• Each value is assigned to a key (Key – Value) • Two options for key-value assignments

Example I:

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

Example II:

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

Dynamic site

Variables - Associative Arrays

- Output

```
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>";
```

Assosiative Arrays

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

Dynamic site

Variables - Multidimensional arrays

- Array variables can also be assigned to an array variable

- Example two-dimensional array:

```php
// 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]);
```

Dynamic site

Variables - Multidimensional arrays

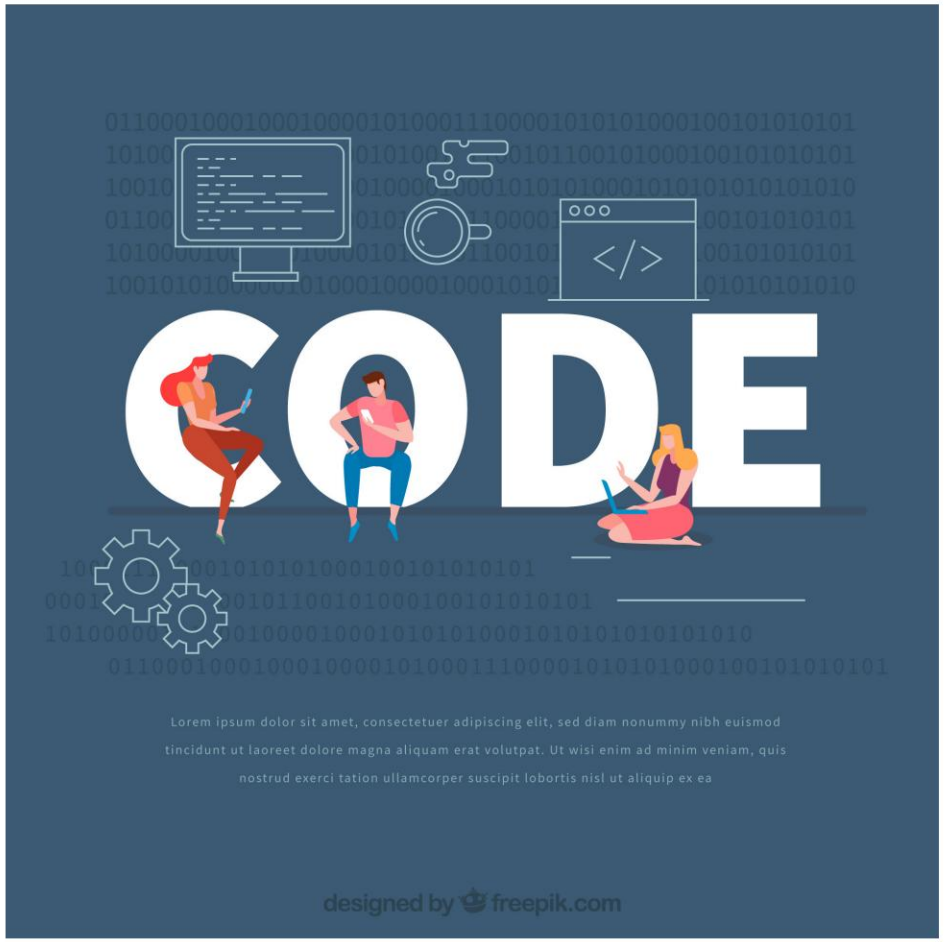
- Output

```
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

Dynamic websites –
Control Structure II - Loops



Dynamic site

Loops - General

Why grind?

• Instead of constantly repeating certain code itself, it can be repeated with loops become.

• This code is executed as long as a condition is met

  • for a self-defined number of repetitions

  • for an indefinite number of repetitions

There are different types of loops: • For loop •
While loop • Do-
While loop • Foreach
loop

Dynamic site

Loops – Without a loop

code without a loop

```
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>";
```

output

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

Dynamic site

for loop

- Frequent use with fixed number of repetitions

Example:

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

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

Output:

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

Dynamic site

while loop

- Loop is executed as long as a condition is met - similar to if **($condition)**

Example:

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

$loops = 0;

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

Output:

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

Dynamic site

do-while loop

- Code is always executed at least once.
- A final **while($condition)** checks how long the loop will then continue

Example:

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

$loops = 0;

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

Output:

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

Dynamic site

foreach loop

This loop only works with arrays
Each value within an array is traversed once

Example:

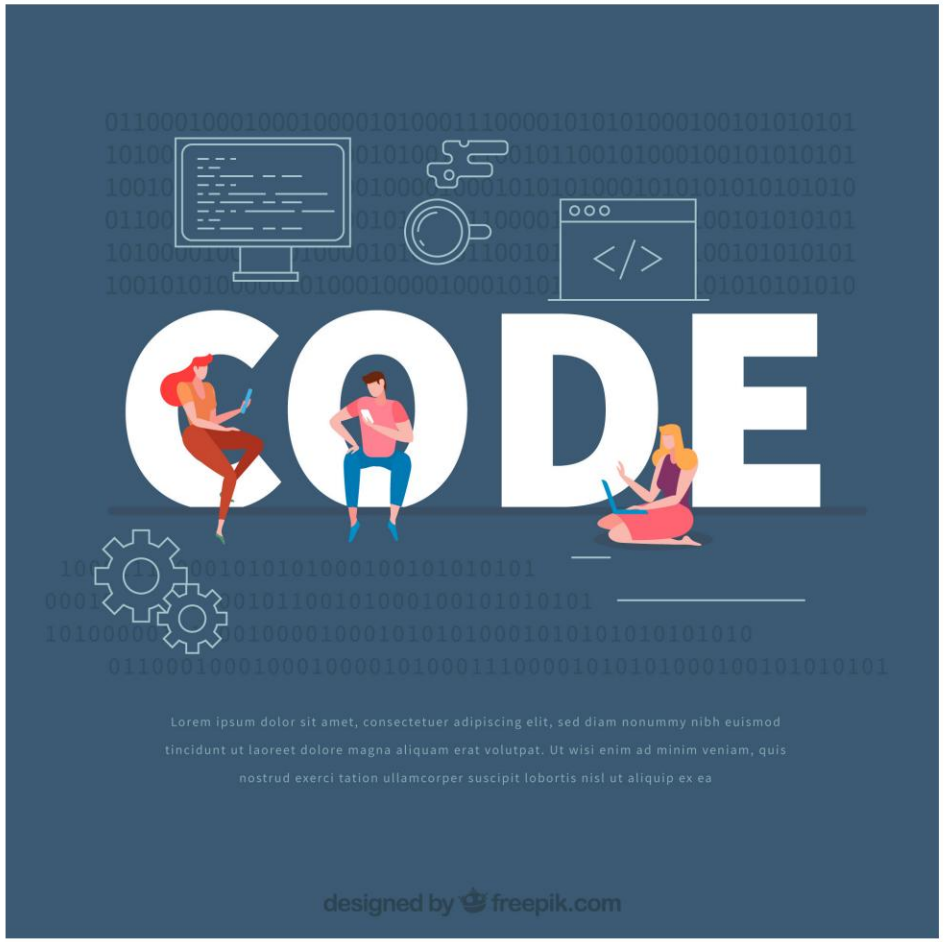
```php
echo "<h4> Ausgabe mit Foreach-Schleife </h4>";

$heroes = array("Luke", "Leia", "Han", "Rey");

foreach ($heroes as $hero){
    echo "Foreach-Schleife Ausgabe Helden-Name: ".$hero;
    echo "<br>";
}
```

Output:

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

Database connection



Dynamic site

Database connection

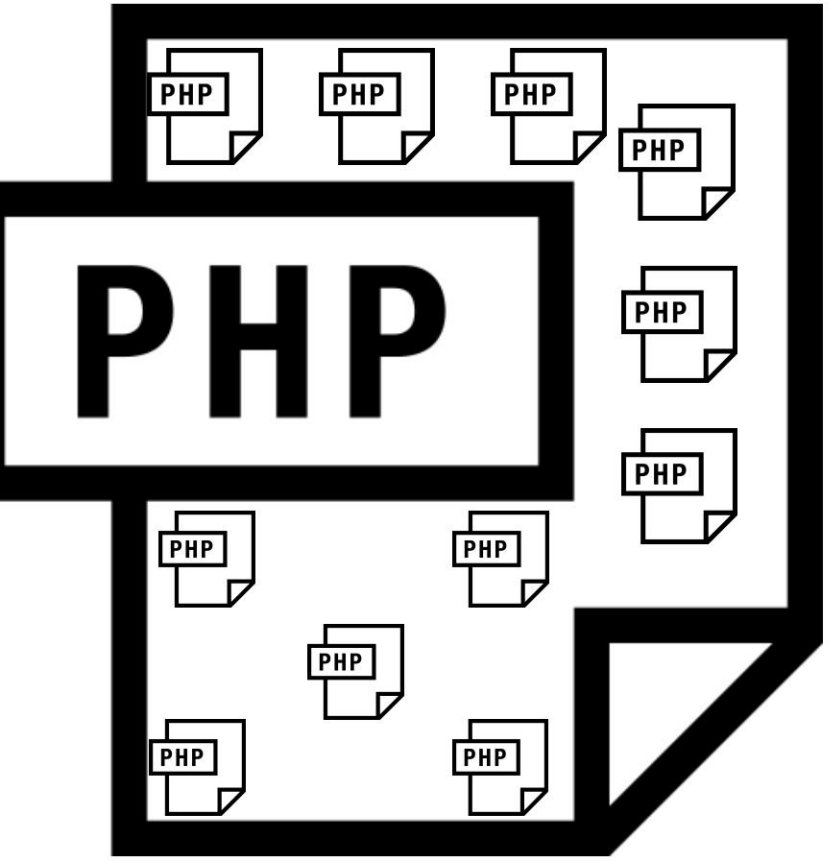
Requirement:
A database is to be connected to an existing system in order to enable persistence of data

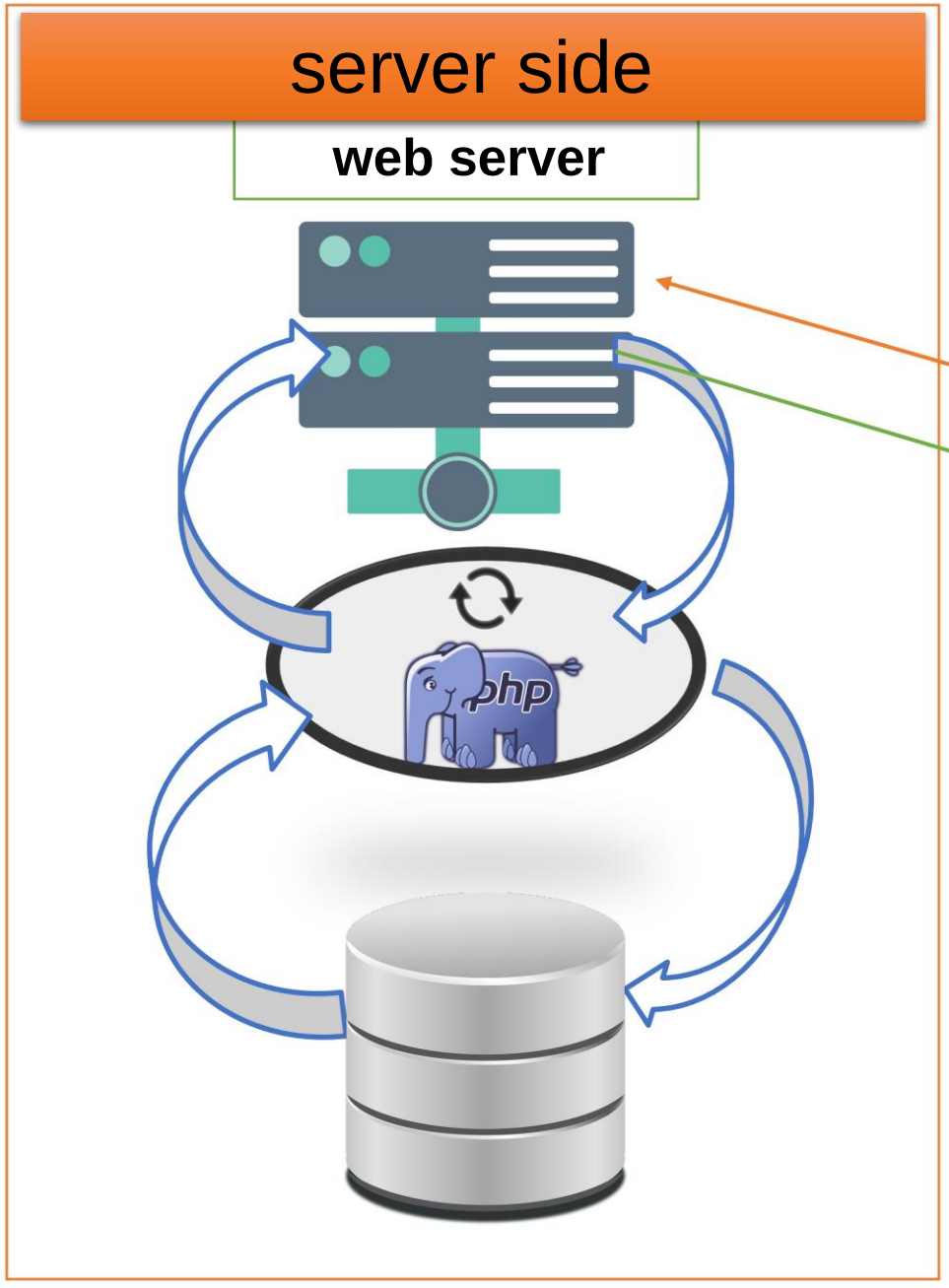
ACTUAL status:
MySQL database available but no connection

Solution approach/prerequisites:
Clarify connection details

- Host name (server on which the DB is located)
- User name
- Password
- Database name

Selection of a PHP function for DB connection





Dynamic site

Database connection - functions

- MySQLi
  - Only for connecting MySQL databases • Object-oriented or procedural use possible

- PDO (PHP Data Objects)
  - Connection of many different databases • Only object-oriented use possible • Advantageous if the database is exchanged

Dynamic site

Database connection – MySQLi (object-oriented)

• Assignment of the connection
details • Transfer of the connection details to establish the
connection • Saving the connection in a variable

```php
$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";
}
```

Dynamic site

Database connection – MySQLi (procedural)

• Assignment of the connection details • Transfer of the connection details to establish the connection • Saving the connection in a variable

```php
$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";
}
```

Dynamic site

Database connection - PDO

• Assignment of the connection
details • Transfer of the connection details to establish the
connection • Saving the connection in a variable

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

// Verbindung herstellen
try {
    $conn = new PDO("mysql:host=$host;dbname=$database", $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();
    }
```

Dynamic site

Database connection – outsource connection details

Provide PHP file with connection details:

File name: config.inc.php

Contents:

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

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

finish

Dynamic websites

PHP IV