

Was ist PHP?

- PHP: Hypertext Preprocessor
- Serverseitige Skriptsprache
- Syntax ähnlich C, Perl, Java, ...
- Eingebettet in HTML
- Ermöglicht dynamische Webseiten
- Anbindung an Datenbanken (z.B. MySQL)
- hat globale, statische und dynamische Variable
- hat Konstanten
- Variablen
- Kontrollstrukturen (if, case)
- Schleifen (for, while)
- Datentypen (Integer, Double, Bool, Strings)
- Arrays (einfache, multidimensionale, indizierte, assoziierte)
- Funktionen
- Bild-Erstellung
- OOP, ab Version 5.0
- Anbindung an eine Datenbank
- Dateioperationen
- Dateiendungen (php, phtml)

PHP-Abschnitte

Anfang PHP: `<?php`

Ende PHP: `?>`

Datentypen

Einfache Datentypen (skalar):

- Boolean (Werte: TRUE und FALSE)
- Integer (Ganze Zahlen)
- Double (Fließkommazahlen)
- String (Zeichenketten)
- Abfragen
 - `is_numeric`
 - `intval`
 - `doubleval`
 - `floatval`
 - `is_int($variable)` Testaufruf
 - `is_float($variable)` Testaufruf

Numeric-Test in PHP

```
is_numeric('123.4')    true

is_numeric('123')      true

is_numeric('a123.4')   false

is_numeric('12XDF')    false


is_int(12XDF)          false

is_int(123)            true

is_int(123.33);        false

$k=12.3;

is_int($k);            false

$k=12.3;

is_float($k);          true
```

Zusammengesetzte Datentypen:

- Array
 - Klassen / Objekte
-
- Variablen brauchen in PHP vor der Benutzung nicht deklariert werden.
 - Sie werden bei der Benutzung automatisch deklariert.
 - Jede Variable muss bei der Benutzung mit einem **Dollar** beginnen.
 - Die Variablen werden nach Groß- und Kleinschreibung unterschieden

Strings, Zeichenketten

– Stringoperatoren zum Verbinden von Strings

- `$a="Hallo " . "Welt!";`
- `$a="Hallo ";`
- `$a .= "Welt";`

Beispiele für Zuweisungen:

```
$vorname;           // Deklaration
$vorname="33";       // Deklaration und Zuweisung
$i = 10;             // Deklaration und Zuweisung
echo $i;             // Ausgabe
$a = 5;              // legt einen Integer an, gettype() liefert integer
$a = 5.2;            // legt eine Fließkommazahl an, gettype() liefert double
$a = "Heute ist Montag"; // gettype() liefert einen string
```

Ausgabe mit syso:

```
echo "hier ist Text";
echo ("hier ist Text");
echo ('hier ist Text');
```

Hochkommata:

Einfache und doppelte Hochkommata

- Variablen werden in einfachen Hochkommas nicht ersetzt
- Variablen werden in doppelte Hochkommas ersetzt
- XHTML zwingt zu Hochkommas im HTML-Code
- XHTML zwingt zu Hochkommas im HTML-Code
- `<input type="text" name="nachname" value="Anton" size="12" />`
- `echo "<input type=\"text\" name=\"nachname\" value=\"Anton\" size=\"12\" />";`
- `// hier ist die Begrenzung das ' Zeichen`
- `echo '<input type="text" name="nachname" value="Anton" size="12" />';`
- **Lösung:**
- `$item="text";`
- `echo '<input type="$item">'; // keine Ersetzung`
- `echo '< input type = ' ' . $item . ' ">'; // Aufteilen der Strings`
- `echo '< input type = " ' , $item , ' ">'; // Aufteilen der Strings mit Kommata`
- `echo "< input type =\"$ item\">"; // Sonderzeichen \`

Beispiel einer Liste:

```
<?php
```

```
echo "<ul>";
```

```
echo " <li> Grundlagen der Informatik </li>";
```

```
echo " <li> Mathematik </li>";
```

```
echo " <li> Physik </li>";
```

```
echo " <li> Englisch </li>";
```

```
echo "</ul>";
```

```
?>
```

Funktionen

- Funktionen - Beispiel
 - <html>
 - <body>
 - <?php
 - \$a=2;
 - \$b=3;
 - \$c=addieren(\$a,\$b);
 - echo ("Summe von \$a und " . \$b . " ist " . \$c);
 - **function addieren (\$summand1, \$summand2)**
 - **{**
 - **return \$summand1 + \$summand2;**
 - **}**
 - ?>
 - </body>
 - </html>

Formulare

Formulare auslesen

- Funktionen
 - isset oder empty
 - is_numeric

- POST

Im Formular: <form action="? " method="get">

Auslesen: \$vorname=\$_POST['vorname'];

- GET

Im Formular: <form action="? " method="get">

Auslesen: \$_GET['Variablenname']

- Übertragung der Daten im Klartext in der URL

http://localhost/VariablenFormular.php?passwort=Horst&login=...

Beispiel:

```
if ( isset($_GET['vorname']) ){           Ist gesetzt?

    $vorname = $_GET['vorname'];

}

if ( isset($_GET['nachname']) ){           Ist gesetzt?

    $nachname = $_GET['nachname'];

}

if ( isset($_GET['zahl']) && is_numeric($_GET ['zahl']) )

{

    $zahl=$_GET['zahl'];

}

else {

    $zahl="Keine Zahl";

}
```

```
echo "$vorname $nachname hat Lieblingszahl $zahl?;
```

```
if ( isset($_GET['vorname']) && isset($_GET['nachname'])  
    && isset($_GET['zahl']) && is_numeric($_GET['zahl']) ){           sind alle gesetzt?  
    // Verarbeitung  
}  
else {  
}
```

Prüfung und Konvertierung der Parameter

```
if ( is_numeric($sn) {  
    $n = intval ( $sn );  
    if ( $n > 0 ) {  
        // Aktion  
    }  
    else {  
        echo "Fehlerhafte Zahl: " . $sn;  
    }  
}
```

HTML-Formulare:

Textzeile (bsp5.xhtml):

```
<input type="text" name="nachname" value="Meier" size="12"/>
```

ComboBox (bsp6.xhtml):

```
<select name="obst" >
  <option value="pfs"> Pfirsich </option >
  <option selected="selected" value="apf"> Apfel </option >
  <option value="org"> Orange </option >
  <option value="ban"> Banane, gelb </option >
</select>
```

In den Options-Tag wurde das Attribut „value“ eingefügt. Dies ist sinnvoll, um kürzere Schlüsselwörter zu erhalten

Übergabe in PHP: obst=apf

```
Abfrage in PHP:     if ( isset($_GET['obst']) ) {                     // Ist gesetzt?
                     $obst = $_GET['obst'];
                     }
                     else {
                         echo "Fehlerhafte &#220;bertragung des Obst";
                         $bOk=false;
                     }
```

Radiobutton (bsp7.xhtml):

```
<input type="radio" name="obst" value="pfs" checked="checked"/> Pfirsich <br />
<input type="radio" name="obst" value="apf" /> Apfel <br />
<input type="radio" name="obst" value="org" /> Orange <br />
<input type="radio" name="obst" value="ban"/> Banane grün <br />
```


In den Radiobuttons gibt es wieder das Attribut „value“. Dieses ist sinnvoll, um kürzere Schlüsselwörter zu erhalten.

Übergabe in PHP: obst=pfs

```
Abfrage in PHP:     if ( isset($_GET['obst']) ) {                     // Ist gesetzt?
                       $obst = $_GET['obst'];
                   }
                   else {
                       echo "Fehlerhafte &#220;bertragung des Obst";
                       $bOk=false;
                   }
                   }
```

CheckBox (bsp8.xhtml):

<input type="checkbox" name="obst" value="gesund" /> Obst ist gesund

Übergabe in PHP: obst=gesund wenn CheckBox gesetzt

Übergabe in PHP: nichts, wenn CheckBox gesetzt

```
Abfrage in PHP:     if ( isset($_GET['obst ']) ) {                     // Ist gesetzt?
                       $obst = true;
                   }
                   else {
                       $obst = false;
                   }
                   }
```

echo in PHP

- Variablen werden in einfache Hochkommas nicht ersetzt
- Variablen werden in doppelte Hochkommas ersetzt

```
$name="meier";
```

```
echo ("Der Name lautet " . $name);
```

```
echo "Der Name lautet " . $name;
```

```
echo("Der Name lautet \"$name\");             // Sonderzeichen für XHTML
```

- PHP-Sonderzeichen müssen beachtet werden:

```
$name="meier";
```

```
echo("Der Name lautet \"$name\");     // Ausgabe: "meier"
```

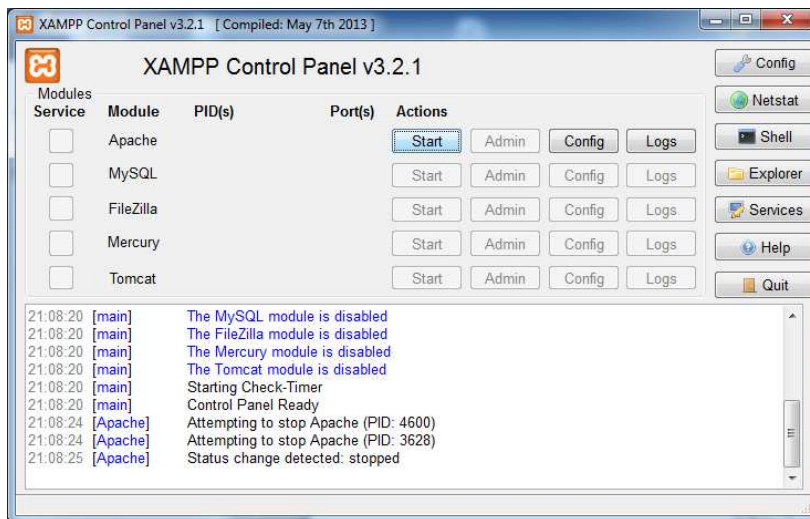
Abfrage der Parameter

Verwendung des eingebauten assoziativen Arrays: Get

```
if ( isset($_GET['zahl']) ){           // Ist gesetzt?
    $sZahl = $_GET['zahl'];
}
else {
    echo "Fehlerhafte &#220;bertragung der Zahl";
    $bOk=false;
}
```

XAMPP

- 1) Download der Datei von meiner Homepage
- 2) Entpacken nach c:\xampp
- 3) Wechseln ins Verzeichnis "c:\xampp"-Verzeichnis
- 4) Starten des Programms „xampp-control.exe“
- 5) Schalter „Start“ betätigen



- 6) Wechseln ins Verzeichnis „C:\xampp\htdocs\xampp\mw\“
In diesem Verzeichnis stehen alle Beispiele
- 7) Aufruf eines Web-Browsers
- 8) Eintragen des ersten Beispiels
`localhost/xampp/mw/bsp01.php`



```
<html>

<head>

  <title> Beispiel 01 </title>

</head>

<body>

  <h2> Meine erste PHP-Datei </h2>

  <?php

    echo "hier ist der erste Text" ;

    echo "<br />";

  ?>

  <p> Abschnitt nach dem PHP-Teil </p>

</body>

</html>
```

```

<?xml version="1.0" encoding="UTF-8" ?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"

    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>

    <title> Bsp_Formular </title>

    <!-- http://localhost/xampp/mw/Bsp_Formular.xhtml -->

</head>

<body>

    <h2>Bsp_Formular</h2>

    <form method="get" action="Bsp_Formular_php.php" >

        Artikel: <input type="text" name="artikel" value="Bugatti Veyron" size="20"/><br /><br />

        Anzahl: <input type="text" name="anz" value="2" size="12"/><br /><br />

        Bezahlung: <select name="bez" >

            <option selected="selected" > bar </option >

            <option > EC-Karte </option >

            <option > PayPal </option >

        </select>

        <p>

            <input type="submit" value="senden" />

            <input type="reset" value="loeschen" />

        </p>

    </form>

</body>

</html>

```

```

<html>

<head>

<title> Bsp_Formular_php.php </title>


</head>

<body>

<h2>Bsp_Formular_php.php</h2>

<?php

$Ok=true;

if ( isset($_GET['artikel']) ){ // Ist gesetzt?

    $artikel = $_GET['artikel'];

}

else {

    echo "Fehlerhafte &#220;bertragung der Variablen artikel";

    $Ok=false;

}

if ( isset($_GET['anz']) ){ // Ist gesetzt?

    $sAnz = $_GET['anz'];

}

else {

    echo "Fehlerhafte &#220;bertragung der Variablen anz";

    $Ok=false;

}


if ( isset($_GET['bez']) ){ // Ist gesetzt?

    $bez = $_GET['bez'];

}

else {

    echo "Fehlerhafte &#220;bertragung der bez";

```

```

$Ok=false;

}

if ($Ok){
    if ( !is_numeric($sAnz) ) {
        $Ok=false;

        echo "<p>Fehlerhafte Eingaben, bitte eine gültige Anzahl eingeben </p>";

        echo "Anzahl: " . $sAnz;
    }
    else {
        $anz=intval($sAnz);

        echo "Artikel: " . $artikel . "<br />\n";

        echo "Anzahl: " . $anz . "<br />\n";

        echo "Bezahlung: " . $bez . "<br />\n";
    }
}

?>

</body>

</html>

```