

# PHP

---

## Grundlagen:

- Variablen immer mit \$-Zeichen: \$VAR;
- Stringconcat mit . nicht mit +;
- Jedes Statement mit ; enden !!!!!!!!;
- echo "\$VAR" → gibt Inhalt aus;
- echo '\$VAR' → gibt Variablennamen aus;
- include „bla.php“;

## Variablen Scopes:

Lokal, global, static, superglobal;

**Lokal → normal**

**Global:**

```
$x = 5;
function pow(){
    global $x;
    $x = $x*$x;
}
```

**Static:**

```
function pow(){
    static $x;
    $x = $x*$x;
}
```

**SUPERGLOBAL:**

- \$\_SERVER['PHP\_SELF'] → Name der PHP-Datei
- \$\_SERVER['REMOTE\_ADDR'] → Server IP
- \$\_POST[]
- \$\_GET[]
- \$\_FILES['<name>']['name']
- \$\_FILES['<name>']['size']
- \$\_FILES['<name>']['tmp\_name']
- \$\_FILES['<name>']['type']
- \$\_FILES['<name>']['error']
- \$\_COOKIE
- \$\_SESSION

## Formulare:

```
<form method="GET" action="php5Auswertung.php">
Name: <input type="text" name="nachname">
Vorname: <input type="text" name="vorname">
<input type="reset" value="del">
<input type="submit" value="send">
</form>
```

### php5Auswertung.php:

```
$name = $_GET['nachname'];
$vorname = $_GET['vorname'];
echo ...
```

## Formulare2:

```
<form method="POST" action="php6Auswertung.php" enctype="multipart/form-data">
Name: <input type="text" name="nachname">
Vorname: <input type="text" name="vorname">
Bild: <input type="file" name="bild">
<input type="reset" value="del">
<input type="submit" value="send">
</form>
```

### php6Auswertung.php:

```
if ( isset( $_FILES['bild'] ) ){
    $nach = $_POST['name'];
    $vor = $_POST['vor'];
    echo ...;

    $source=$_FILES['fupload']['tmp_name'];
    $target="D://xampp/apache/Bilder/".$_FILES['fupload']['name'];

    if(source <> ""){
        move_uploaded_file($source, $target) or die ("FEHLER");
    }
}
```

## Arrays:

```
$temperaturen = array (29.1, 22.3, 14.3, 27.6, 31.2, 29.1, 33.1);
$temperaturen = [29.1, 22.3, 14.3, 27.6, 31.2, 29.1, 33.1];
$temparaturen[] = 555; → Element anhängen;
anzahl = sizeof($temperaturen);
```

```
sort($temperaturen, SORT_NUMERIC);
rsort($temperaturen, SORT_NUMERIC); //
```

*SORT\_NUMERIC || SORT\_STRING*

## Felder (Hashmaps):

```
$lufttemp=array ("Rom" => "28 Grad",
"London" => "14 Grad",
"Paris" => "30 Grad",
"Berlin" => "20 Grad",
);
$lufttemp= ["Rom" => "28 Grad",
"London" => "14 Grad",
"Paris" => "30 Grad",
"Berlin" => "20 Grad",
]
$lufttemp["Ansbach"] = "24 Grad";

foreach($lufttemp as $stadt=>$temp){echo "<br>In $stadt hat es $temp";}

asort($lufttemp, SORT_STRING);
ksort($lufttemp, SORT_STRING);
```

## Dateien:

- *file\_exists(„wetter.txt“);*
- *fopen(„bla.txt“, „r“); // fclose; // „r+“ read/write („a“) append*
- *feof(\$datei) → End of FILE?*
- *fgets(\$datei) → Lies Zeile?*
- *explode(' ', \$Zeile) → tokenizer*
- *fseek(\$datei, 5);*
- *fwrite(\$date, \$text);*
- *fputs(\$datei, \$Zeile);*
- *flocks(\$datei, LOCK\_EX) // LOCK\_UN*
- *stat(\$datei) → \$info[x] 8 -> zugriff, 9-> mod, 10 -> last change*

## Printformat:

- *printf(““, arg...); (<prev> </prev> !!!!!)*
- *% -20s → string linksbündig 20 lang*
- *% 20s → analog zu oben nur rechtsbündig*
- *% 10.0f → 10 Stellen 0 Nachkomma*
- *% 9.2f → 9 Stellen 2 Nachkomma*
- *date(„d.m.y h:i:s“)*

## MYSQL Datenbanken:

- `$mysql = new MySQLi('localhost', 'root', '', '<DB>');`
- `$result = $mysql->query($sql)`
- `$mysql->errno`
- `$mysql->sqlstate`
- `$mysql->error`
- `$mysql->close()`
- `$row = $result->fetch_array(MYSQLI_NUM); // MYSQLI_BOTH, MYSQLI_ASSOC`

## XML Sax:

- `Xml_parser_create();`
- `Xml_set_element_handler($parser, '<func>', '<func>');`
- `Xml_set_character_data_handler($parser, '<func>');`
- `Xml_parse($parser, file_get_contents('<path to XML>'), true);`
- **function start(\$parser, \$el, \$att) → \$att['TYP'] zugriff auf typ**
- **function ende(\$parser, \$el)**
- **function cdata(\$parser, \$data) // printet Daten aus**

## XML Simple XML:

- `$xml = simplexml_load_file('<path to XML>')`
- `$xml->lkw[0]->hersteller[0]`
- `$xml->lkw[0]->fahrer[0]`
- `$lkw['typ'] → Zugriff auf typ`
- *Ausgeben mit foreach(\$xml as \$lkw) schleifen*
- *Ausgeben mit foreach(\$lkw->fahrer as \$fahrer) schleife*
- ---
- *Modding xml*
- ---
- `addChild("<name>", "<inhalt>")`
- `addAttribute("<name>", "<inhalt>")`
- `$xml->asXML("<path>"); → Speichern`
- `Echo $xml->asXML();`

## XML mit XMLReader:

- `$xml = new XMLReader();`
- `$xml->open("<path to XML>");`
- Ausgabe mit `while($xml->read())` Schleife → Parsed nächsten Knotenpunkt mit jeder Iterat
- --
- `$xml->name !!!`
- `$xml->nodeType` // Knotentyp XMLReader:: ELEMENT und XMLReader:: TEXT !!!!  
`XMLReader ::END_ELEMENT`
- `$xml->hasAttributes;`
- `$xml->attributeCount;`
- `$xml->hasValue;`
- `$xml->value; !!!!`
- `$xml->getAttribute("<name>") !!!!`

## XML mit DOM Object:

- `error_reporting(E_ALL ^ E_NOTICE); !!!!!???`
- `$dom = new DOMDocument();`
- `$dom->preserveWhiteSpace=false;`
- `$dom->load("./xml/fuhrpark.xml");`
- -- SETUP
- --
- `$lkws = $dom. getElementsByTagName("<TAG>");`
- `Foreach($lkws as $lkw)`
- `$lkw->firstChild->nodeName; !!! hersteller->childnode == Textnode mit inhalt`
- `$lkw->firstChild->firstChild->data !!!`
- `$lkw->nextSibling; !!!`
- `$lkw->attributes["typ"]->value`
- -
- `$dom->schemaValidate('abc.xsd');`

## Cookies:

- `$_COOKIE` → arr
- `Setcookie("<name>", <val>, time()+3600*24*7)` EINE Woche gültig

## Sessions:

- `Session_start()` → neue Session starten oder bestehende fortsetzen
- `Session_destroy()` || `Session_unset()`
- `Session_name();`
- `Session_id();`
- `$_SESSION` → Daten container → arr
- --
- Login Bsp
- `Exit()`

## MALEN BULLSHIT:

- `Header(„Content-type: image/jpeg“); !!!!`
- `$img = imagecreatetruecolor(1000, 1000);`
- `$schwarz = imagecolorallocate($img, 0, 0, 0)`
- ----
- `Imagefilledrectangle($img, x, y, width, height, <farbe>)`
- `imagefilledellipse ($img, x, y, width, height, <farbe>)`
- `imagefilledarc ($img, x, y, width, height, arcBegin, arcEnd, <farbe>, 0)`
- `imagesetpixel ($img, x, y, <farbe>)`
- --
- `Imageline($img, x, y, x2, y2, <farbe>);`
- --
- `Imagejpeg($img, null, 100);`