

PHP Introduction (2)

What is a web application and
how can PHP help us



Mario Peshev

Technical Trainer

<http://peshev.net/>

Software University

<http://softuni.bg>

Table of Contents

1. Basic functions
2. Built-in PHP Function Libraries
3. Some predefined variables
4. Strings escaping
5. PHP – advantages and disadvantages



PHP Object and Resource Types

- PHP supports "object" variable type
 - Will be explained further in the OOP lecture
- "Resource" variable type
 - The resource type means the variable is holding reference to resource or data, external to your script
 - Example – opened file, database connection, etc

PHP Basic Expressions

- PHP expressions are similar to C
 - "=" - assigning value to variable
 - +, -, /, *, % - arithmetic operations
 - ==, <=, >=, !=, <, > - comparison
 - +=, -=, /=, *=, %+=, ++, --, etc – prefix/postfix operators
 - (and) – for expressions combining
 - &, |, >>, <<, ^, ~ - bitwise operators

PHP Basic Expressions

- String operators
 - "." (period) – string concatenating
- ==, != comparison
 - different from ===, !==
 - "10"==10 will produce true, while "10"===10 will produce false
 - Strict comparison – `$a === $b` :
 - TRUE if `$a` is equal to `$b`, and they are of the same type.
 - Note: Assignment of value to variable returns as result the value being assigned
 - We can have `$a = $b = $c = 7;`

PHP Constants

- In PHP constants are defined with the `define` function

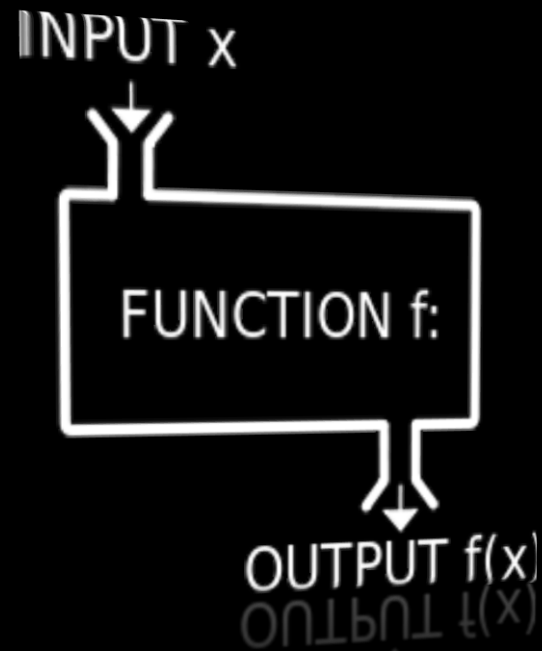
```
<?php
define("CONSTANT", "Hello world.");
echo CONSTANT; // outputs "Hello world."
define("GREETING", "Hello you.", true); // not recommended
echo GREETING; // outputs "Hello you."
echo Greeting; // outputs "Hello you."
?>
```

- Cannot change value
- Doesn't start with \$
- Can hold any scalar value

PHP Constants

Live Demo

PHP Basic Functions



PHP Function

Start with
"function"

Function name

Input arguments

```
<?php
function foo($arg_1, $arg_2, /* ..., */ $arg_n) {
    echo "Example function.\n";
    return $retval;
}
foo($arg_1, $arg_2, /* ..., */ $arg_n);
?>
```

Return value

Function body

Call function with
arguments

PHP Functions

Live Demo

phpinfo();

Live Demo

Built-in PHP Functions

Built-in PHP Functions

- PHP is very rich in terms of Built-in functions. Here is the list of various important function categories. There are various other function categories which are not covered here.
 - PHP Array Functions
 - PHP Calendar Functions
 - PHP Class/Object Functions
 - PHP Character Functions
 - PHP Date & Time Functions
 - PHP Directory Functions

Built-in PHP Functions (2)

- PHP Error Handling Functions
- PHP File System Functions
- PHP MySQL Functions
- PHP Network Functions
- PHP ODBC Functions
- PHP String Functions
- PHP SimpleXML Functions
- PHP XML Parsing Functions

Predefined Variables

Predefined Variables

- PHP provides a lot predefined variables and constants
- `__FILE__`, `__LINE__`, `__FUNCTION__`,
`__METHOD__`, `__CLASS__` - contain debug info
- `PHP_VERSION`, `PHP_OS`, `PHP_EOL`,
`DIRECTORY_SEPARATOR`, `PHP_INT_SIZE` and others are
provided for easy creating cross-platform applications

Predefined Variables

- `$_SERVER` – array, holding information from the web server – headers, paths and script locations
 - `DOCUMENT_ROOT` – the root directory of the site in the web server configuration
 - `SERVER_ADDRESS`, `SERVER_NAME`, `SERVER_SOFTWARE`, `SERVER_PROTOCOL`
 - `REMOTE_ADDR`, `REMOTE_HOST`, `REMOTE_PORT`
 - `PHP_AUTH_USER`, `PHP_AUTH_PW`, `PHP_AUTH_DIGEST`
 - And others

Predefined Variables

- `$_GET`, `$_POST`, `$_COOKIE` - arrays hold the parameters from the URL, from the post data and from the cookies accordingly
- `$_FILES` - array holds information for successfully uploaded files over multipart post request
- `$_SESSION` – array holds the variables, stored in the session

Variable variables

- PHP supports \$\$ syntax- variable variables

```
<?
$var1= 'var2';
$var2= 'Hello World';
echo $$var1; // outputs Hello World
?>
```

- The variable `$var1` is evaluated as 'var2' and so `$$var1` is evaluated as `$var2`

Predefined Variables

Live Demo

Strings escaping



Strings escaping

- Special chars in strings are escaped with backslashes (C style)
 - The escape sequences for double quoted string:
 - `\n` – new line (10 in ASCII)
 - `\r` – carriage return (13 in ASCII)
 - `\t` – horizontal tab
 - `\v` – vertical tab
 - `\\` - backslash
 - `\$` - dollar sign
 - `\"` – double quote

String escaping

- Single-quoted strings escape the same way

```
$str1 = 'Start with single quotes "and \' escape single quotes'';
```

- Difference is that instead of \" you need \' to escape the closing quotes
- No other escaping sequences will be expanded
- In both single and double quoted strings, backslash before any other character will be printed too!

Variables in strings

- Double quoted strings offer something more:

```
$var1= "content";  
$str1 = "We can put variable to get $saying";  
// this will output:  
// We can put variable to get content
```

- Variables in double-quoted strings are evaluated

Heredoc syntax

- Define strings with heredoc syntax ('<<<')

```
$bar = <<<EOT
bar
    EOT;
}
echo $bar;// output EOT bar EOT
```

- After the <<< we put "ending delimiter" – string goes all the way to this delimiter
 - The delimiter must be followed by new line
 - The ending delimiter must be alone on the last line, starting from first column
- Same escaping behavior as double-quoted string
- In single and double quoted strings you can embed new lines too

String escaping

Live Demo

Advantages and Disadvantages

Advantages and disadvantages

■ Advantages

- Easy to learn, open source, multiplatform and database support, extensions, community and commercial driven.
- Considered to be one of the fastest languages

■ Disadvantages

- Too loose syntax – risk tolerant, poor error handling, poor OOP (before version 6 a lot things are missing!)

Summary

- Built-in PHP Function Libraries
- Global variables
- Strings escaping
- Data sanitization
- PHP – advantages and disadvantages



PHP & MySQL

Questions?

A network diagram consisting of several circular nodes, each containing a large question mark. These nodes are interconnected by thin lines, forming a web-like structure. The background is dark with some abstract, flowing lines in shades of brown and orange.

License

- This course (slides, examples, demos, videos, homework, etc.) is licensed under the "Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International" license



- Attribution: this work may contain portions from
 - "PHP Manual" by The PHP Group under CC-BY license
 - "PHP and MySQL Web Development" course by Telerik Academy under CC-BY-NC-SA license

Free Trainings @ Software University



- Software University Foundation – softuni.org
- Software University – High-Quality Education, Profession and Job for Software Developers
 - softuni.bg
- Software University @ Facebook
 - facebook.com/SoftwareUniversity
- Software University @ YouTube
 - youtube.com/SoftwareUniversity
- Software University Forums – forum.softuni.bg

