



Grundlagen Internet-Technologien

INF3171

Einführung in PHP

Version 1.0

13.06.2024





Aktuelles





Grundprinzip von php

„PHP ist die Abkürzung für "PHP: Hypertext Preprocessor", eine weitverbreitete Open Source Skriptsprache speziell für Webentwicklungen. PHP läßt sich in HTML einbinden. Seine Syntax erinnert an C, Java und PERL und es ist einfach zu erlernen. Das Hauptziel dieser Sprache ist es, Webentwicklern die Möglichkeit zu geben, schnell dynamisch generierte Webseiten zu erzeugen. Aber PHP kann noch mehr.“

(PHP Manual)





Ressourcen

- »Hauptseite«
<http://www.php.net>
- <http://www.zend.com>
die Seite von (u.a.) Zeev Suraski
- daneben zahlreiche weitere Angebote





php.net

php Downloads Documentation Get Involved Help php 8.2 Search

php

A popular general-purpose scripting language that is especially suited to web development. Fast, flexible and pragmatic, PHP powers everything from your blog to the most popular websites in the world.

What's new in 8.2 Download

8.2.7 · Changelog · Upgrading | 8.1.20 · Changelog · Upgrading | 8.0.29 · Changelog · Upgrading

PHP 8.3.0 Alpha 2 available for testing » 22 Jun 2023

The PHP team is pleased to announce the second testing release of PHP 8.3.0, Alpha 2. This continues the PHP 8.3 release cycle, the rough outline of which is specified in the [PHP Wiki](#).

For source downloads of PHP 8.3.0 Alpha 2 please visit [download page](#).

Please carefully test this version and report any issues found in the [bug reporting system](#).

Please DO NOT use this version in production, it is an early test version.

For more information on the new features and other changes, you can read the [NEWS](#) file, or the [UPGRADING](#) file for a complete list of upgrading notes. These files can also be found in the release archive.

The next release will be Alpha 3, planned for 6 July 2023.

The signatures for this release can be found in [the manifest](#) or on [the QA site](#).

Upcoming conferences

- International PHP Conference Munich 2023
- SymfonyCon Brussels 2023

Conferences calling for papers

- Longhorn PHP 2023
- Dutch PHP Conference 2023

User Group Events

Special Thanks

Social media

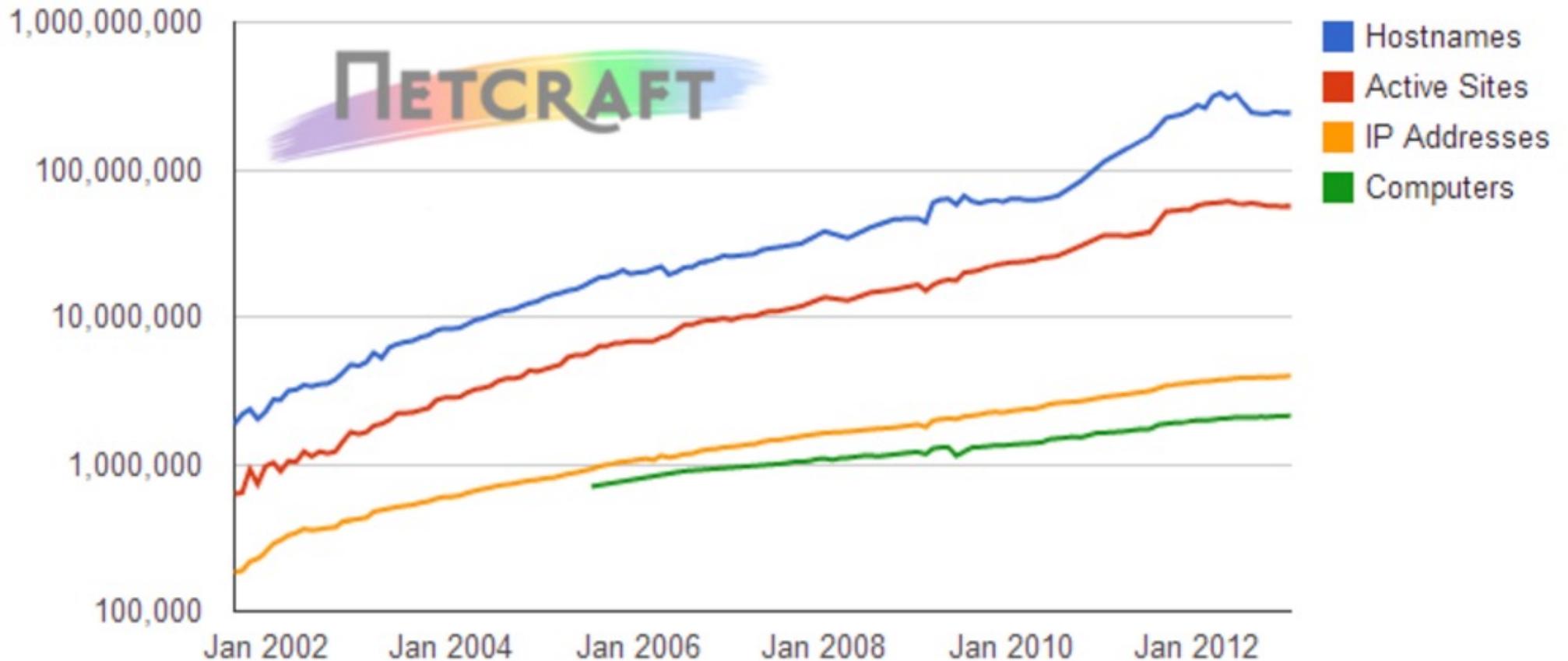
- [@official_php](#)



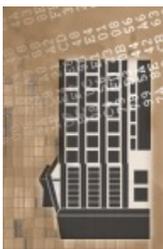


php Sites

PHP Trend (Logarithmic Scale)



Quelle: <http://www.php.net/usage.php>





TIOBE index June 2023

Jun 2023	Jun 2022	Change	Programming Language	Ratings	Change
1	1		 Python	12.46%	+0.26%
2	2		 C	12.37%	+0.46%
3	4	▲	 C++	11.36%	+1.73%
4	3	▼	 Java	11.28%	+0.81%
5	5		 C#	6.71%	+0.59%
6	6		 Visual Basic	3.34%	-2.08%
7	7		 JavaScript	2.82%	+0.73%
8	13	▲	 PHP	1.74%	+0.49%



Versionen von php

- php wurde 1994 von Rasmus Lerdorf »begründet«
- 1995: PHP/FI Version 2
- ab 1997: Entwicklung von PHP3 in größerem Team (Lerdorf, Zmievski, Suraski, Gutmans et. al.)
- seit PHP4 auch andere Webserver als Apache
- aktuell:php 8.2.7, 8.1.20, 8.0.29
 - wir behandeln php Versionen 7 und 8
- Version 5/7/8 sind gut abwärtskompatibel
 - relevante Ausnahme: neue Schlüsselworte





php8

Released!

PHP 8.0 is a major update of the PHP language. It contains many new features and optimizations including named arguments, union types, attributes, constructor property promotion, match expression, nullsafe operator, JIT, and improvements in the type system, error handling, and consistency.

[Upgrade to PHP 8 now!](#)





Documentation

The PHP Manual is available online in a selection of languages. Please pick a language from the list below.

More information about php.net URL shortcuts by visiting our [URL howto page](#).

Note, that many languages are just under translation, and the untranslated parts are still in English. Also some translated parts might be outdated. The translation teams are open to contributions.

Formats	Destinations
View Online	English , Brazilian Portuguese , Chinese (Simplified) , French , German , Japanese , Romanian , Russian , Spanish , Turkish
Downloads	For downloadable formats, please visit our documentation downloads page.

Documentation for PHP 4 has been removed from the manual, but there is archived version still available. For more information, please read [Documentation for PHP 4](#).

More documentation

- If you are interested in how the documentation is edited and translated, you should read the [Documentation HOWTO](#).
- [PHP-GTK related documentation](#) is hosted on the PHP-GTK website.
- [Documentation of PEAR and the various packages](#) can be found on a separate server.
- You can still read a copy of the original [PHP/FI 2.0 Manual](#) on our site, which we only host for historical purposes. The same applies to the [PHP 3 Manual](#), and the [PHP 4 Manual](#).





php.net/manual/en/migration71.new-features.php

php Downloads Documentation Get Involved Help Search

PHP Manual > Appendices > Migrating from PHP 7.0.x to PHP 7.1.x « Migrating from PHP 7.0.x to PHP 7.1.x New functions »

Change language: English Edit Report a Bug

New features

Nullable types

Type declarations for parameters and return values can now be marked as nullable by prefixing the type name with a question mark. This signifies that as well as the specified type, **NULL** can be passed as an argument, or returned as a value, respectively.

```
<?php

function testReturn(): ?string
{
    return 'elePHPant';
}

var_dump(testReturn());

function testReturn(): ?string
{
    return null;
}
```

» New features
New functions
New global constants
Backward incompatible changes
Deprecated features in PHP 7.1.x
Changed functions
Other changes
Windows Support



Gibt es Komplikationen mit altem Code?

Da PHP 8 eine neue Hauptversion ist, muss man damit rechnen, dass alter Code nicht mehr kompatibel sein kann. Die meisten Änderungen, die zu Komplikationen führen könnten, waren allerdings schon in Version [7.2](#), [7.3](#) und [7.4](#) veraltet.

Zu den aktuellen Änderungen gehören:

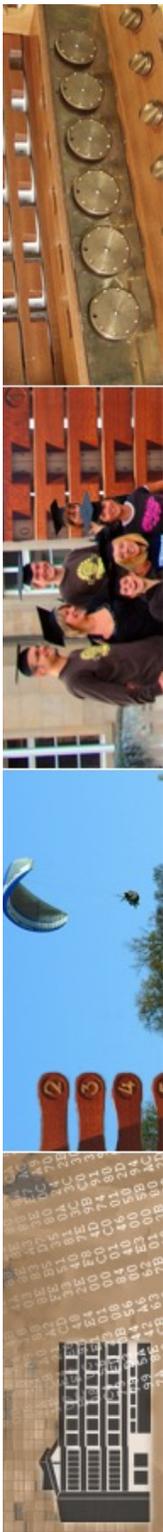
- The *real* type
- Magic quotes legacy
- *array_key_exists()* with objects
- *FILTER_SANITIZE_MAGIC_QUOTES* filter
- Reflection *export()* methods
- *mb_strrpos()* with encoding as 3rd argument
- *implode()* parameter order mix
- Unbinding *\$this* from non-static closures
- *hebrevc()* function
- *convert_cyr_string()* function
- *money_format()* function
- *ezmlm_hash()* function
- *restore_include_path()* function
- *allow_url_include* ini directive

Wer seinen Code bisher immer up to date gehalten hat, wird kaum Probleme haben, auch wenn das neue Update ein Major-Release ohne abwärtskompatible Änderungen ist. Eine komplette Auflistung aller Änderungen kann man auf der offiziellen [PHP-Projektseite](#) einsehen.



Installation von php

- php macht nur in Verbindung mit einem Webserver Sinn
- php kann (inzwischen) mit vielen Webservern betrieben werden, „klassisch“ ist php mit Apache (und MySQL)
- php arbeitet mit vielen DBMS, „default“ war MySQL für PHP 4 und SQLite oder MySQL ab PHP 5
- **LAMP: Linux - Apache - MySQL - php**
 - es gibt auch WAMP... ;-)
- einfachere Möglichkeit der Installation: php als cgi
- php als Modul des Webserverns hingegen bedeutet, dass der php-Interpreter stets zur Verfügung steht
 - man unterscheidet noch dynamische und statische Module - php ist typisch ein dynamisches Modul





XAMPP Apache + MariaDB + PHP + Perl

Was ist XAMPP?

XAMPP ist die beliebteste PHP-Entwicklungsumgebung

XAMPP ist eine vollständig kostenlose, leicht zu installierende Apache-Distribution, die MariaDB, PHP und Perl enthält. Das XAMPP Open-Source-Paket wurde für eine extrem einfache Installation und Nutzung eingerichtet.



Herunterladen

Klicken Sie hier für weitere Versi

 XAMPP für **Windows**
8.2.4 (PHP 8.2.4)

 XAMPP für **Linux**
8.2.4 (PHP 8.2.4)

 XAMPP für **OS X**
8.2.4 (PHP 8.2.4)

New XAMPP release
8.2.4, 8.1.17 and
8.0.28

Hi Apache Friends!

We just released a new version of XAMPP for Windows, Linux and OS X for PHP versions 8.2.4, 8.1.17 and 8.0.28. You can download these new installers at <http://www.apachefriends.org...>

[Weiterlesen »](#)



php Konfiguration

- es gibt eine zentrale Konfigurationsdatei **php.ini**
 - diese legt u.a. die Art der php-Tags fest
 - lernen wir noch genauer kennen

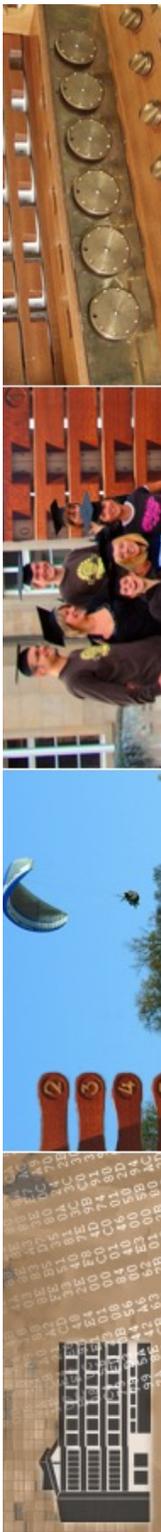
```

[PHP]

;;;;;;;;;;;;;;;;;;;;;;;;;
; About php.ini        ;
;;;;;;;;;;;;;;;;;;;;;;;;;
; PHP's initialization file, generally called php.ini, is responsible for
; configuring many of the aspects of PHP's behavior.

; PHP attempts to find and load this configuration from a number of locations.
; The following is a summary of its search order:
; 1. SAPI module specific location.
; 2. The PHPRC environment variable. (As of PHP 5.2.0)
; 3. A number of predefined registry keys on Windows (As of PHP 5.2.0)
; 4. Current working directory (except CLI)
; 5. The web server's directory (for SAPI modules), or directory of PHP
; (otherwise in Windows)
; 6. The directory from the --with-config-file-path compile time option, or the
; Windows directory (C:\windows or C:\winnt)
; See the PHP docs for more specific information.
; http://php.net/configuration.file

; The syntax of the file is extremely simple.  Whitespace and Lines
--More-- (1%)
    
```

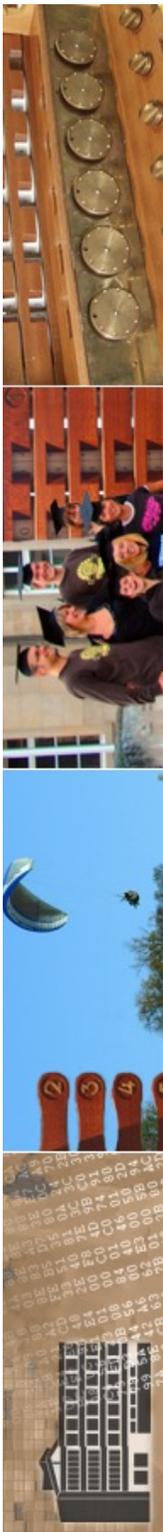




Testen der Installation

- Datei `info.php` abgelegt im Ordner `htdocs` mit folgendem Inhalt:

```
<?php    phpinfo () ;    ?>
```





phpinfo0 phpinfo0
127.0.0.1:82/phpinfo.php

PHP Version 5.3.2



System	Windows NT MOUTON 6.1 build 7600 ((nu
Build Date	Mar 3 2010 19:38:00
Compiler	MSVC6 (Visual C++ 6.0)
Architecture	x86
Configure Command	cscrip /nologo configure.js "--enable-snap disable-isapi" "--without-mssql" "--without-p sdk\oracle\instantclient10\sdk,shared" " "--with-oci8-11g=D:\php-sdk\oracle\insta

Server API
Virtual Directory Support
Configuration File (php.ini) Path
Loaded Configuration File
Scan this dir for additional .ini files
Additional .ini files

phpinfo0 phpinfo0
134.2.2.38/~zrvwa01/phpinfo.php

PHP Version 5.2.6-1+lenny10



phpinfo0
127.0.0.1:82/phpinfo.php

PHP Version 5.4.4

System	Windows NT PE
Build Date	Jun 13 2012 21
Compiler	MSVC9 (Visual
Architecture	x86
Configure Command	cscrip /nologo c without-mssql" " sdk\oracle\insta sdk\oracle\insta sdk\oracle\insta with-mcrypt=sta
Server API	Apache 2.0 Han
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\Windows

phpinfo0
127.0.0.1/webst1/php/phpinfo.php

PHP Version 5.5.12



System	Windows NT CARRUADESMOUTON 6.2 build 9200 (Windows 8 Business Edition) AMD64
Build Date	Apr 30 2014 11:15:47
Compiler	MSVC11 (Visual C++ 2012)
Architecture	x64
Configure Command	cscrip /nologo configure.js "--enable-snapshot-build" "--disable-isapi" "--enable-debug-pack" "--without-mssql" "--without-pdo-mssql" "--without-pi3web" "--with-pdo-oci=C:\php-sdk\oracle\x64\instantclient10\sdk,shared" "--with-oci8=C:\php-sdk\oracle\x64\instantclient10\sdk,shared" "--with-oci8-11g=C:\php-sdk\oracle\x64\instantclient11\sdk,shared" "--enable-object-out-dir=../obj/" "--enable-com-dotnet=shared" "--with-mcrypt=static" "--disable-static-analyze" "--with-pgo"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\Windows
Loaded Configuration	(none)

PHP Version 5.6.40-0+deb8u4



System	Linux infodienste 4.9.0-6-amd64 #1 SMP Debian 4.9.82-1+deb9u3 (2018-03-02) x86_64
Build Date	Jun 3 2019 09:23:41
Server API	CGI/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/cgi
Loaded Configuration File	/etc/php5/cgi/php.ini
Scan this dir for additional .ini files	/etc/php5/cgi/conf.d
Additional .ini files parsed	/etc/php5/cgi/conf.d/05-opcache.ini, /etc/php5/cgi/conf.d/10-pdo.ini, /etc/php5/cgi/conf.d/20-json.ini, /etc/php5/cgi/conf.d/20-mysql.ini, /etc/php5/cgi/conf.d/20-mysqli.ini, /etc/php5/cgi/conf.d/20-pdo_mysql.ini, /etc/php5/cgi/conf.d/20-readline.ini
PHP API	20131106
PHP Extension	20131226
Zend Extension	220131226
Zend Extension Build	API220131226,NTS
PHP Extension Build	API20131226,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	enabled
Registered PHP Streams	https, ftps, compress.zlib, compress.bzip2, php, file, glob, data, http, ftp, phar, zip
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, sslv3, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib.*, bzip2.*, convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk



PHP Version 7.3.27-1~deb10u1

System	Linux infodienste 4.19.0-9-amd64 #1 SMP Debian 4.19.118-2+deb10u1 (2020-06-07) x86_64
Build Date	Feb 13 2021 16:31:40
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.3/fpm
Loaded Configuration File	/etc/php/7.3/fpm/php.ini
Scan this dir for additional .ini files	/etc/php/7.3/fpm/conf.d
Additional .ini files parsed	/etc/php/7.3/fpm/conf.d/10-mysqld.ini, /etc/php/7.3/fpm/conf.d/10-opcache.ini, /etc/php/7.3/fpm/conf.d/10-pdo.ini, /etc/php/7.3/fpm/conf.d/15-xml.ini, /etc/php/7.3/fpm/conf.d/20-calendar.ini, /etc/php/7.3/fpm/conf.d/20-ctype.ini, /etc/php/7.3/fpm/conf.d/20-dom.ini, /etc/php/7.3/fpm/conf.d/20-exif.ini, /etc/php/7.3/fpm/conf.d/20-fileinfo.ini, /etc/php/7.3/fpm/conf.d/20-ftp.ini, /etc/php/7.3/fpm/conf.d/20-gd.ini, /etc/php/7.3/fpm/conf.d/20-gettext.ini, /etc/php/7.3/fpm/conf.d/20-iconv.ini, /etc/php/7.3/fpm/conf.d/20-json.ini, /etc/php/7.3/fpm/conf.d/20-mysqli.ini, /etc/php/7.3/fpm/conf.d/20-pdo_mysql.ini, /etc/php/7.3/fpm/conf.d/20-phar.ini, /etc/php/7.3/fpm/conf.d/20-posix.ini, /etc/php/7.3/fpm/conf.d/20-readline.ini, /etc/php/7.3/fpm/conf.d/20-shmop.ini, /etc/php/7.3/fpm/conf.d/20-simplexml.ini, /etc/php/7.3/fpm/conf.d/20-sockets.ini, /etc/php/7.3/fpm/conf.d/20-sysvmsg.ini, /etc/php/7.3/fpm/conf.d/20-sysvsem.ini, /etc/php/7.3/fpm/conf.d/20-sysvshm.ini, /etc/php/7.3/fpm/conf.d/20-tokenizer.ini, /etc/php/7.3/fpm/conf.d/20-wddx.ini, /etc/php/7.3/fpm/conf.d/20-xmlreader.ini, /etc/php/7.3/fpm/conf.d/20-xmlwriter.ini, /etc/php/7.3/fpm/conf.d/20-xsl.ini, /etc/php/7.3/fpm/conf.d/20-zip.ini
PHP API	20180731
PHP Extension	20180731
Zend Extension	320180731
Zend Extension Build	API320180731,NTS
PHP Extension Build	API20180731,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled



PHP Version 7.4.33

System	Linux infodienste 5.10.0-12-amd64 #1 SMP Debian 5.10.103-1 (2022-03-07) x86_64
Build Date	Jun 9 2023 16:51:37
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.4/fpm
Loaded Configuration File	/etc/php/7.4/fpm/php.ini
Scan this dir for additional .ini files	/etc/php/7.4/fpm/conf.d
Additional .ini files parsed	/etc/php/7.4/fpm/conf.d/10-mysqld.ini, /etc/php/7.4/fpm/conf.d/10-opcache.ini, /etc/php/7.4/fpm/conf.d/10-pdo.ini, /etc/php/7.4/fpm/conf.d/15-xml.ini, /etc/php/7.4/fpm/conf.d/20-calendar.ini, /etc/php/7.4/fpm/conf.d/20-ctype.ini, /etc/php/7.4/fpm/conf.d/20-dom.ini, /etc/php/7.4/fpm/conf.d/20-exif.ini, /etc/php/7.4/fpm/conf.d/20-ffi.ini, /etc/php/7.4/fpm/conf.d/20-fileinfo.ini, /etc/php/7.4/fpm/conf.d/20-ftp.ini, /etc/php/7.4/fpm/conf.d/20-gd.ini, /etc/php/7.4/fpm/conf.d/20-gettext.ini, /etc/php/7.4/fpm/conf.d/20-iconv.ini, /etc/php/7.4/fpm/conf.d/20-json.ini, /etc/php/7.4/fpm/conf.d/20-mysqli.ini, /etc/php/7.4/fpm/conf.d/20-pdo_mysql.ini, /etc/php/7.4/fpm/conf.d/20-phar.ini, /etc/php/7.4/fpm/conf.d/20-posix.ini, /etc/php/7.4/fpm/conf.d/20-readline.ini, /etc/php/7.4/fpm/conf.d/20-shmop.ini, /etc/php/7.4/fpm/conf.d/20-simplexml.ini, /etc/php/7.4/fpm/conf.d/20-sockets.ini, /etc/php/7.4/fpm/conf.d/20-sysvmsg.ini, /etc/php/7.4/fpm/conf.d/20-sysvsem.ini, /etc/php/7.4/fpm/conf.d/20-sysvshm.ini, /etc/php/7.4/fpm/conf.d/20-tokenizer.ini, /etc/php/7.4/fpm/conf.d/20-xmlreader.ini, /etc/php/7.4/fpm/conf.d/20-xmlwriter.ini, /etc/php/7.4/fpm/conf.d/20-xsl.ini, /etc/php/7.4/fpm/conf.d/20-zip.ini
PHP API	20190902
PHP Extension	20190902
Zend Extension	320190902
Zend Extension Build	API320190902,NTS
PHP Extension Build	API20190902,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled





erstes Script

- php-Script wird **direkt in die HTML-Datei eingefügt** (im Gegensatz zu Python/CGI oder etwa Java Servlets)
- Tags zur Begrenzung des Scripts:
 - `<?php ... ?>` oder auch `<?PHP ... ?>`
 - `<?= ... ?>` für direkte Ausgabe
 - `<script language="php"> ... </script>` (nicht mehr in Version 7)
 - `<% ... %>` (ASP-Nomenklatur - beachte php.ini, nicht mehr in Version 7)
- typisches Suffix der Datei: **.php** (Abhängig von Server-Konfiguration)

```
zrvwa01@infodienste => more helloworld.php
<?php
/*
 * Grundlagen Internet-Technologien
 * Webapplikationen mit PHP
 *
 * HelloWorld in PHP
 */
?>
<!DOCTYPE html>
<HTML>
    <HEAD>
        <TITLE>Grundlagen Internet-Technologien: HelloWorld mit PHP</TITLE>
        <link rel="stylesheet" type="text/css" href="../css/webkompendium.css"
    >
        <link rel="shortcut icon" href="../css/favicon.ico">
    </HEAD>
    <BODY>
        <H2><HR><CENTER>

                <!-- Script Anfang -->

        <?php
                                // Ausgabe im php-Script
                                echo "Grundlagen Internet-Technologien<BR>I li
ke PHP";
                                ?>

                <!-- Script Ende -->

        <HR><IMG src="php.gif"><HR></CENTER></H2>
    </BODY>
</HTML>
zrvwa01@infodienste => █
```



← → ↻ 🔖 ⚠ Nicht sicher | 134.2.6.167/~zrvwa01/php/helloworld.php 🔍 📄 📁 VPN Aktualisieren ☰

Grundlagen Internet-Technologien I like PHP



```
zrvwa01@infodienste => more helloworld2.php
<?php
/*
 * Grundlagen Internet-Technologien
 * Webapplikationen mit PHP
 *
 * unterbrochenes HelloWorld-Script
 */
?>
<!DOCTYPE html>
<HTML>
  <HEAD>
    <TITLE>Grundlagen Internet-Technologien: HelloWorld mit PHP, Version 2</TITLE>
    <link rel="stylesheet" type="text/css" href="../css/webkompendium.css">
    <link rel="shortcut icon" href="../css/favicon.ico">
  </HEAD>
  <BODY>
    <H2><CENTER>

    <?php
      // Variablenbelegung
      $phrase = "Grundlagen Internet-Technologien<BR>zweiteiliges PHP-Script";
```

< > ↻ 🗖 | ⚠ 134.2.6.167/~zrvwa01/php/helloworld2.p 🔍 📷 ✓ ▶ ❤ | ⬇ ☰

Grundlagen Internet- Technologien zweiteiliges PHP-Script

```
</HTML>
```

```
zrvwa01@infodienste =
```



Editoren für php

- einfacher Editor genügt: Atom
- auch starke Web-Editoren wie DreamWeaver sind php-fähig
- für die IDE-Eclipse gibt es ein Plugins:
 - phpeclipse
 - PDT: PHP Development Tools
 - ZEND Studio

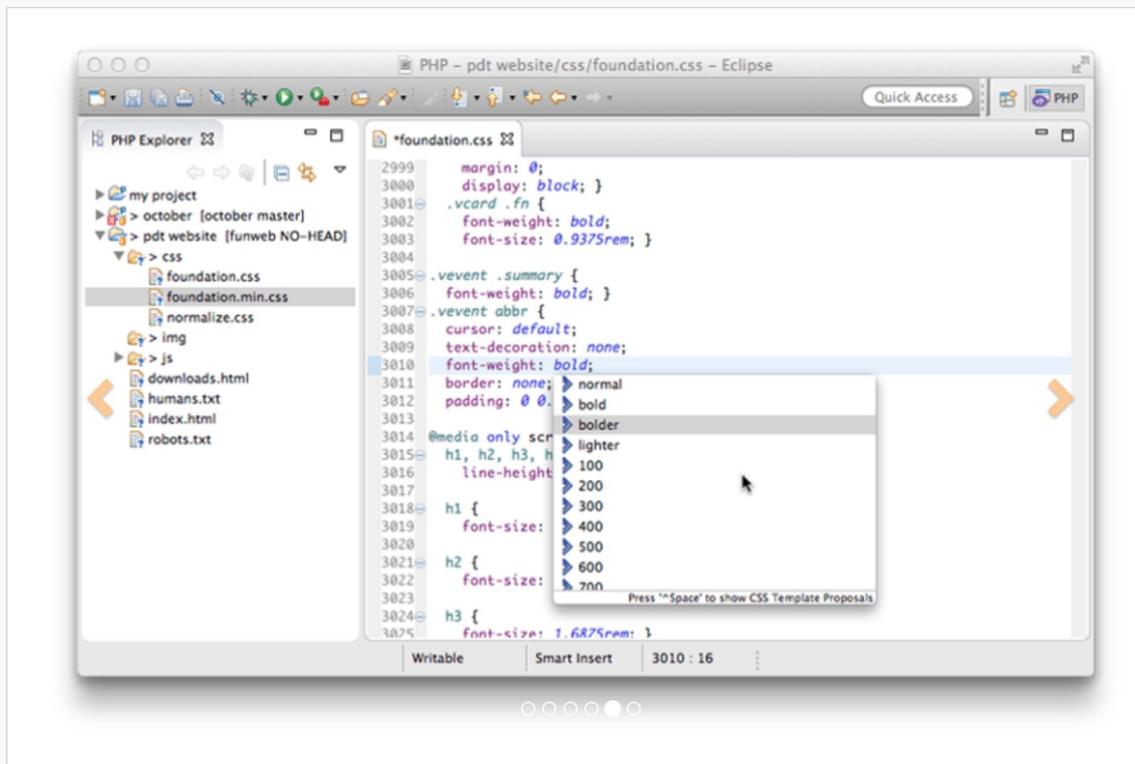




Eclipse PHP Development Tools

About

The PHP IDE project delivers a PHP Integrated Development Environment framework for the Eclipse platform. This project encompasses the development components necessary to develop PHP-based Web Applications and facilitates extensibility. It leverages the existing Web Tools Project in providing developers with PHP capabilities.



▶ About

▶ Features

▶ Download

▶ Extensions

▶ Contributing

Project

▶ News & Noteworthy

▶ Marketplace

▶ Project Repository

▶ Project Details

Support

▶ Manual

▶ PDT Forum

▶ Wiki

▶ Mailing List

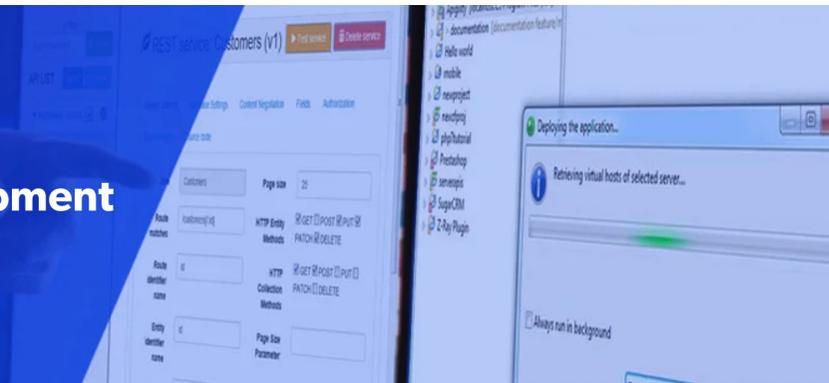
▶ Submit A Bug

Fork me on GitHub



Zend Studio PHP IDE

Your Proven PHP Integrated Development Environment



PHP Development Tools

Zend Guard

Zend Studio

BUY NOW

Develop PHP Apps with Zend Studio

Zend Studio is an integrated development environment (IDE) for PHP 5.5 and newer. With this PHP IDE, you gain tools for:



Intelligent Code Editing

Save time with intuitive code assist.



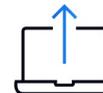
Debugging and Profiling

Use independently or with Zend Server.



Interoperability

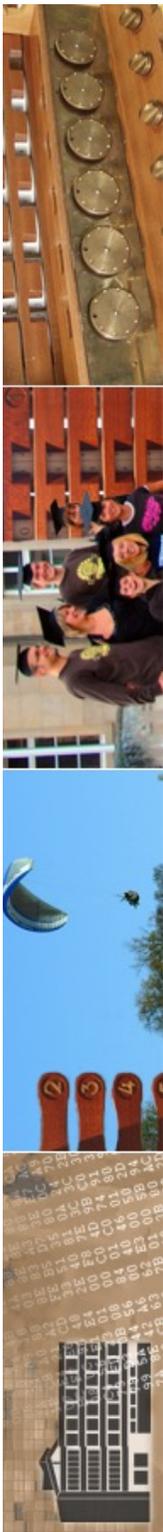
Leverage SVN, CVS, Git, and GitHub.



Simplifying Deployment

Deploy apps on private and public clouds.





PHP - GIT PHP/helloworld.php - Eclipse

File Edit Refactor Source Navigate Search Project Run Window Help

Quick Access

Java Perl JavaScript PHP Ruby

dbi1.rb dbi.rb autohtml.rb fakultaet2.rb fakultaet3.rb helloworld.php »32

```

1 <?php
2 /*
3  * Grundlagen Internet-Technologien
4  * Webapplikationen mit PHP
5  *
6  * HelloWorld in PHP
7  */
8 ?>
9 <HTML>
10 <HEAD>
11     <TITLE>Grundlagen Internet-Technologien: HelloWorld mit PHP</TITLE>
12     <link rel="stylesheet" type="text/css" href="/css/webkompendium.css">
13     <link rel="shortcut icon" href="/css/favicon.ico">
14 </HEAD>
15 <BODY>
16     <H2><HR><CENTER>
17
18         <!-- Script Anfang -->
19
20         <SCRIPT language="php">
21             // Ausgabe im php-Script
22             echo "Grundlagen Internet-Technologien<BR>I like PHP";
23         </SCRIPT>
24
25         <!-- Script Ende -->
26
27         <HR><IMG src="/images/php.gif"><HR></CENTER></H2>
28 </BODY>
29 </HTML>
    
```

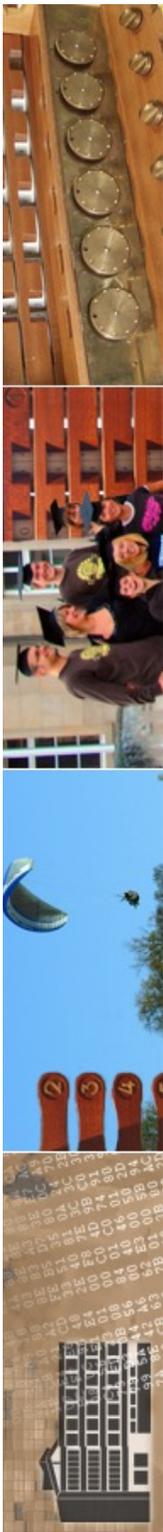
Writable Smart Insert 1:1





Kommentare

- im php-Script können folgende Kommentare verwendet werden:
 - // Kommentar
 - /*
 Kommentar
 ueber mehrere Zeilen
 */
 - # noch ein Kommentar: geht auch!





das Dienstprogramm phpdoc

- es gibt eine Adaption von javadoc für php: **phpdoc**
- Arbeitsweise und Syntax ähnlich zu javadoc
- nützlich: <http://www.phpdoc.de>
- insbesondere für php in OO-Kontext sinnvoll
 - phpdoc-Kommentar wie in Java:

- `/**`

`phpDoc-comment`

- `*/`

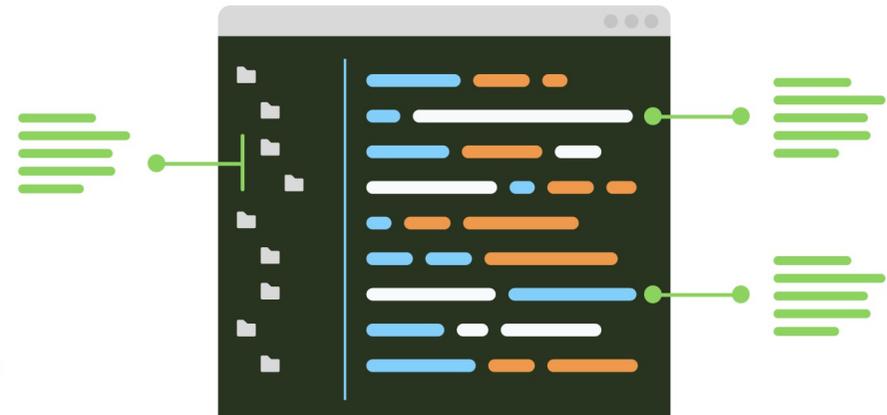




Because **code** and **documentation** are meant to be together.

phpDocumentor is the de-facto documentation application for PHP projects. Your project can benefit too from more than 20 years of experience and setting the standard for documenting PHP Applications.

 DOCUMENTATION



Features

What can phpDocumentor 3 do to help you with your documentation



✔ **Super easy to install and use**
Thanks to the power of Docker or PHAR

✔ **Creates Beautiful Documentation**

✔ **More control using DocBlocks**
Uses the information from your DocBlocks to provide even more insight.





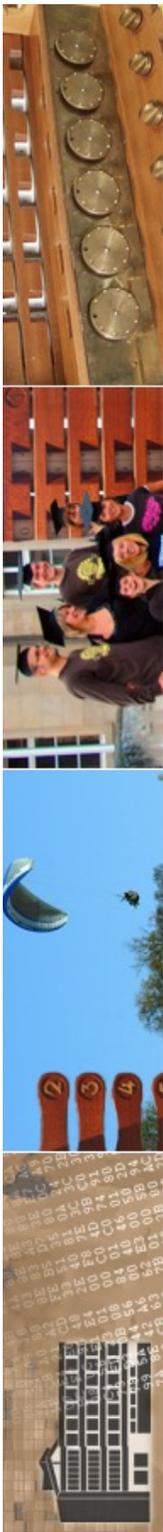
Ausgabe in php

- es gibt die Methoden echo und print
- bezüglich ' und " unterschiedlich (wie z.B. bei PERL)
 - `print ('php costs $0');`
 - `print ("php costs $0");`
- es gibt in php wie in Python und Python auch HERE-Documents
 - `echo <<<ENDE`
 - `...`
 - `ENDE`
 - `;`



Bezeichner

- Bezeichner für Variablen in php:
 - erstes Zeichen Buchstabe oder _
 - Groß- und Kleinschreibung wird unterschieden
- es gibt in php ~ 50 reservierte “typische” Schlüsselwörter
 - bei PHP 5 einige neue Schlüsselwörter





skalare Variablen

- ...hier finden wir vieles wieder, was wir schon in JavaScript und Python kennen gelernt haben...
- die Variablen sind *zunächst nicht typisiert*

```

- $wert = 4711;           // Integer
  $pi = 3.1415;          // real
  $e = 1.602e-19;        // real
  $text = "php";         // String
  $text2 = "I like $text"; // Ersetzung
  $text3 = 'php costs $0'; // keine Ersetz.
  $a = $b = $c = $text;  // Dreifachzuweis
  
```



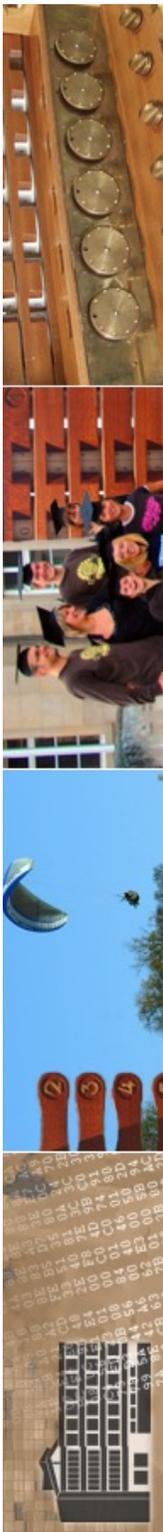
interne Datentypen

- folgende Datentypen sind vorhanden- und es existiert jeweils ein CAST dafür:
 - `int` und `integer`
 - `double` und `real`
 - `string`
 - `array`
 - `object`
- daneben gibt es einen expliziten CAST-Operator für die Typen `double`, `string` und `integer`
- einige Änderungen in PHP 7



mehr zu den Datentypen

- int und integer
 - hat - meistens - die Größe von 32 bit
 - mit führender 0 : Octaldastellung
 - mit führenden 0x : Hexadezimal
- double und real
 - meistens 64 bit im IEEE-Format
 - über php.ini auch alternativ festlegbar
- string
 - es gibt die üblichen escapes: \n, \", \\, ...
 - Komposition mit .





Konstanten in php

- Konstanten in php (entsprechend final-Variablen in Java)
 - haben *kein* führendes \$ im Bezeichner
 - typische Bezeichner aus Großbuchstaben
 - keine Ersetzung in String mit "
 - Deklaration durch Methode `define(bezeichner, wert)`

```
define( 'LANG' , "php" );
echo (LANG) ;           ###   Ausgabe:   php
```



Vordefinierte Konstanten

- **PHP_VERSION**
- **PHP_OS**
- **__FILE__**
- **__LINE__**
- **TRUE**
- **FALSE**
- **NULL**
- ...und ein paar weitere, die insbesondere mit den Superglobals von Bedeutung werden



```
zrvwa01@infodienste => more konstanten.php
```

```
<?php
/*
 * Grundlagen Internet-Technologien
 * Webapplikationen mit PHP
 *
 * Konstanten in PHP
 */
?>
<!DOCTYPE html>
<HTML>
    <HEAD>
        <TITLE>Grundlagen Internet-Technologien: Konstanten PHP</TITLE>
        <link rel="stylesheet" type="text/css" href="../css/webkompendium.css">
        <link rel="shortcut icon" href="../css/favicon.ico">
    </HEAD>
    <BODY>
        <HR><H2><CENTER>

        <?php

            define('TITEL',"Grundlagen Internet-Technologien");

            print(TITEL);
            print("<BR>Konstanten in PHP");
            print("</CENTER></H2><HR><H3>");

            echo "Betriebssystem: ".PHP_OS."<BR>";
            echo "PHP-Version: ".PHP_VERSION."<BR>";
            echo "Datei: ".__FILE__."<BR>";
            echo "dies ist Zeile ".__LINE__." im php-Script";

        ?>

        </H3><HR>
    </BODY>
</HTML>
```

```
zrvwa01@infodienste => █
```





134.2.6.167/~zrvwa01/php/konstanten.php



Grundlagen Internet-Technologien

Konstanten in PHP

Betriebssystem: Linux

PHP-Version: 7.3.27-1~deb10u1

Datei: /home/zrvwa01/public_html/php/konstanten.php

dies ist Zeile 30 im php-Script



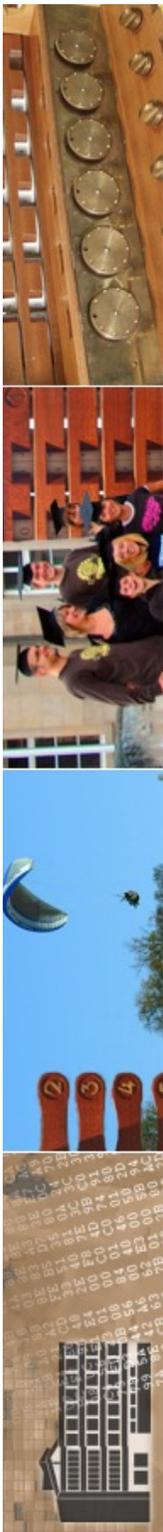


vordefinierte Variablen

- php hat einige vordefinierte Variablen (“Superglobals”), insbesondere zum Sessionhandling und zur Variablenübermittlung aus Formularen, z.B.

- `$_COOKIE[]`
- `$_ENV[]`
- `$_SESSION[]`
- `$_SERVER[]`

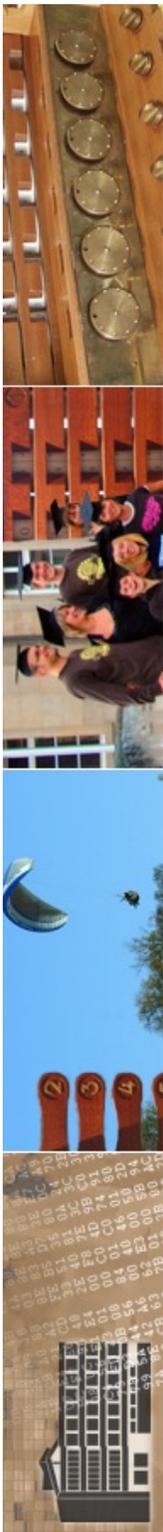
- `$_GET[]`
- `$_POST[]`
- `$_REQUEST[]`





Superglobals

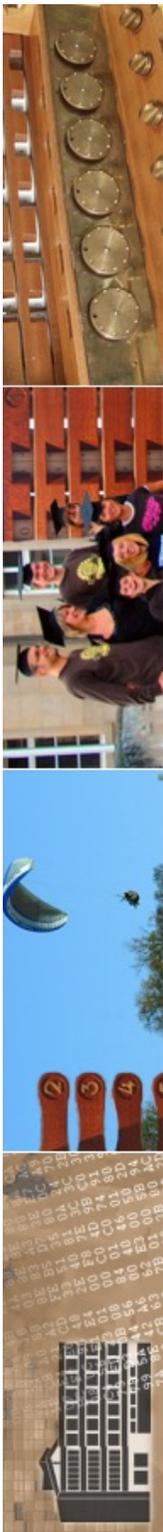
- drei Superglobals für die Übergabe von Formularwerten (sind jeweils assoziative Listen):
 - \$_REQUEST
 - \$_GET
 - \$_POST





Gültigkeit der Variablen

- Variablen in php sind zunächst lokal
- mittels des Schlüsselwortes **global** kann eine globale Variable deklariert werden
 - `global $studiengang;`
`$studiengang = 'Medizininformatik';`





```
zrvwa01@infodienste => more globalevariable.php
<?php
/*
 * Grundlagen Internet-Technologien
 * Webapplikationen mit PHP
 *
 * globale und lokale Variablen in PHP
 */
?>
<!DOCTYPE html>
<HTML>
  <HEAD>
    <TITLE>Grundlagen Internet-Technologien: globale Variablen in PHP</TITLE>
    <link rel="stylesheet" type="text/css" href="../css/webkompodium.css">
    <link rel="shortcut icon" href="../css/favicon.ico">
  </HEAD>
  <BODY>
    <H2><HR><CENTER>

    <?php

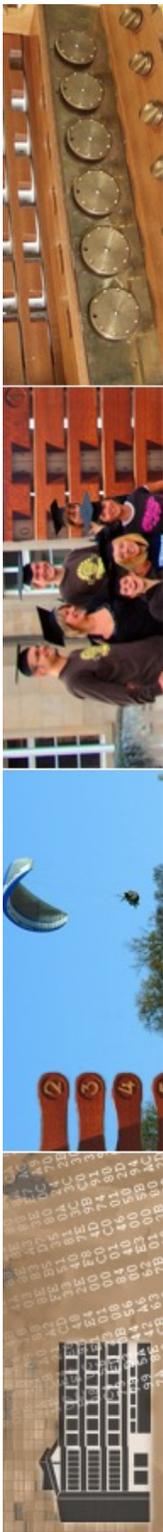
        function globalisierung() {
            global $variable;
            $variable = "globale Variable";
        } // globalisierung

        $variable = "lokale Variable";
        echo ("Wert der Variablen vor Funktionsaufruf: $variable");
```

134.2.6.167/~zrvwa01/php/globalevariable.php

Wert der Variablen vor Funktionsaufruf: lokale Variable
Wert der Variablen nach Funktionsaufruf: globale Variable

</HT
zrvw





Listen/Arrays in php

- Deklaration eines Arrays

```
– $a[0] = "Hello";  
  $a[1] = "World";
```

→ Array mit 2 Elementen

```
$a[ ] = "php";
```

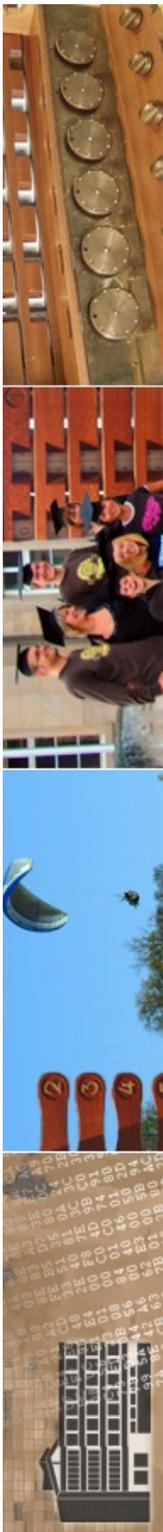
nun Array mit 3 Elementen, `$a[2] == "php";`

– → Arrays in PHP sind immer auch Listen!



Erzeugung mittels „array(...)“

- mittels der üblichen PHP-Methode `array (...)` kann direkt ein Array erzeugt werden:
 - `$feld = array("eins", "zwei", "drei");`
- die PHP-Methode `range (start, ende)` füllt Array mit int-Bereich
 - `$feld = range (1, 100);`





assoziative Arrays

- Arrays mit beliebigen keys (wie in Python: Hash)

```
– $vip["chef"]           = "Christian";
  $vip["alterchef"]     = "Horst";
  $vip["partner"]       = "Angela";
```

```
echo $vip["partner"]; // Ausgabe "Angela"
```

- ...oder auch...

```
$vip = array("chef"       => "Christian",
            "alterchef"  => "Horst",
            "partner"    => "Angela"
            );
```



```
zrvwa01@infodienste => more liste.php
```

```
<?php
/*
 * Grundlagen Internet-Technologien
 * Webapplikationen mit PHP
 *
 * Listen in PHP
 */
?>
<!DOCTYPE html>
<HTML>

    <HEAD>
        <TITLE>Grundlagen Internet-Technologien: Listen in PHP</TITLE>
        <link rel="stylesheet" type="text/css" href="../css/webkompendium.css">
        <link rel="shortcut icon" href="../css/favicon.ico">
    </HEAD>
    <BODY>
        <CENTER><HR><H2>
            Grundlagen Internet-Technologien<BR>Listen in PHP
        </H2><HR></CENTER>

        <?php

            $buecher = array('liebling' => 'MediaFotografie',
                'auchGut' => 'Kompendium der Web-Programmierung');

            ### Erweiterung
            $buecher['neu'] = 'Kompendium der Mediengestaltung';

            $favorit = $buecher['liebling'];

            print("mein Lieblingsbuch ist $favorit");

            print("<BR><BR>Alle Titel:<BR>");
            foreach ($buecher as $titel) {print "$titel<BR>"; }

        ?>

        <HR>
    </BODY>
</HTML>
```

```
zrvwa01@infodienste => _
```





< > ↻ ☰ | ⚠ 134.2.6.167/~zrvwa01/php/liste.php



Grundlagen Internet-Technologien

Listen in PHP

mein Lieblingsbuch ist MediaFotografie

Alle Titel:

MediaFotografie

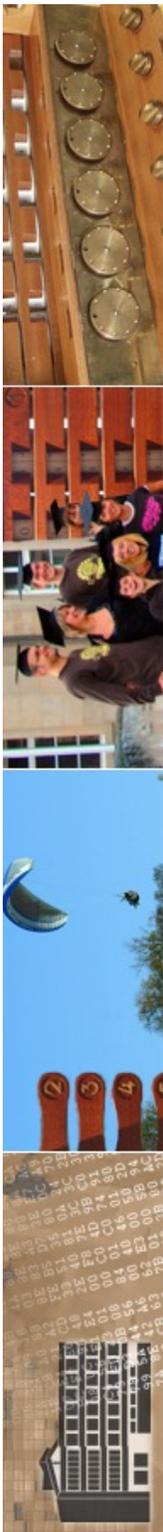
Kompendium der Web-Programmierung

Kompendium der Mediengestaltung



Weiteres nützliches

- ...für Arrays stehen zahlreiche nützliche Funktionen zur Verfügung, etwa
 - `unset ($array)` löscht das Array `$array`
 - `count ($array)` zählt die Elemente
 - `asort ($array)` sortiert ein Array, aufsteigend
 - `rsort ($array)` sortiert ein Array absteigend
 - `ksort ($array)` sortiert ein Array nach Key
 - mit `each ($array)` kann ein Array durchlaufen werden (dies werden wir bei Schleifen nutzen)





Operatoren in PHP

- +, -, * wie »erwartet«
- / ist je nach Datentyp div (bei int) oder geteilt (bei real)
- % ist mod bei int
 - `$i = 4711 % 2;` // `$i` hat den Wert 1
- »String-Addition« durch .
- Zuweisungsoperatoren: »wie üblich«
 - `$x *= 3;`
 - `$text .= "\n";`
 - `$x++`



Logik in PHP

- Prinzip (wie in JavaScript und Python):
 - 0 ist falsch, alle anderen Zahlenwerte sind wahr
 - "" (der leere String) ist falsch, alle anderen sind wahr
- Vergleichsoperatoren
 - `$a == $b` und `$a != $b`
 - `$a < $b` und `$a > $b`
 - `$a <= $b` und `$a >= $b`
- logische Operatoren
 - `$a && $b` und (auch als »and«)
 - `$a || $b` oder (auch als »or« - es gibt auch xor)
 - `! $a` nicht



Kontrollstrukturen in PHP

- die Grundtypen
 - Verzweigung
 - Switch
 - Schleifen

- werden wir im folgenden wie üblich auch in php wiederfinden





Verzweigung - if

- ```
if (bedingung) { wahr_anweisung; }
else { falsch_anweisung; }
```

  - else kann entfallen
  - geschweiftes Klammerpaar nicht zwingend
- Beispiel:
  - ```
if ($vip["chef"] == "Angela")
    {echo "Merkel ist Kanzler";}
else
    {echo "abgel&ouml;st";}
```



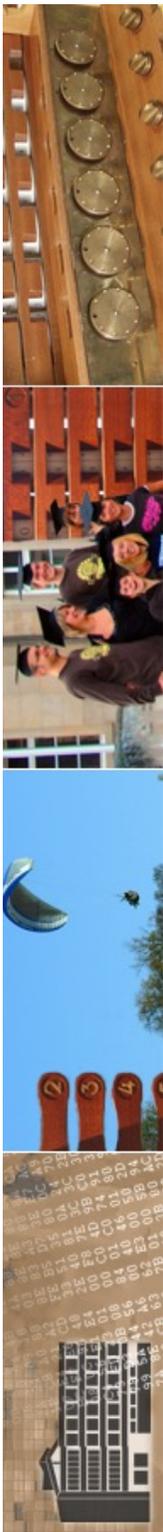
Verzweigung - elseif

- mit **elseif** kann ein else-Zweig eine weitere Bedingung abprüfen

```

- if (bedingung)    wahr_anweisung;
  elseif (bedingung2)
    falsch_wahr_anweisung;
  else
    falsch_falsch_anweisung;

```



komplexere Verzeifung: switch

- Verzweigung mit »Schalter«
 - `switch ($schalter) {`
 - `case konstante : anweisung; break;`
 - `...`
 - `default : sonst_anweisung;`
 - `}`



- Beispiel:
 - `switch ($i) {`
 - `case 0 : echo "i ist 0"; break;`
 - `case 1 : echo "i ist 1"; break;`
 - `default : echo "i ist sonstwas";`
 - `}`



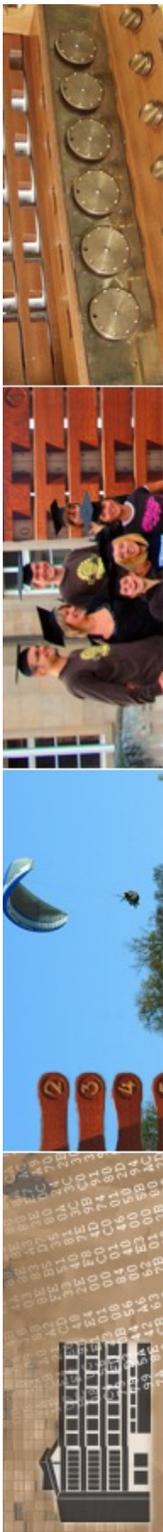
Schleifen

- es gibt die vier Schleifentypen

- `while`
- `do-while`
- `for`
- `foreach` (ab php 4)



- wieder weitgehend analog zu Java





```
<?php
```

```
// while
$summe = 0; $i = 1;
while ($i <= 100) {
    $summe += $i;
    $i++;
}
echo "Summation mittels while ergibt: $summe <BR>";

// do-while
$summe = 0; $i = 1;
do {
    $summe += $i;
    $i++;
} while($i <= 100);
echo "Summation mittels do-while ergibt: $summe <BR>";

// for
$summe = 0;
for ($i = 1; $i <= 100; $i++)
    $summe += $i;
echo "Summation mittels for ergibt: $summe <BR>";

// foreach
$liste = range(1,100); $summe = 0;
foreach ($liste as $i)
    $summe += $i;
echo "Summation mittels foreach ergibt: $summe <BR>";
```

```
?>
```





< > ↻ ☰ | ⚠ 134.2.6.167/~zrvwa01/php/schleifen.php



Grundlagen Internet-Technologien

Schleifen in PHP

Summation mittels while ergibt: 5050

Summation mittels do-while ergibt: 5050

Summation mittels for ergibt: 5050

Summation mittels foreach ergibt: 5050



Verlassen von Schleifen

- in php werden Schleifenstrukturen “wie üblich” (Java) verlassen:
 - mit **break**; wird die Schleifenausführung insgesamt beendet und mit der ersten Anweisung am Schleifenende fortgefahren
 - mit **continue**; wird mit dem nächsten Schleifendurchlauf fortgefahren, also mit der ersten Anweisung des neuen Schleifendurchlaufs

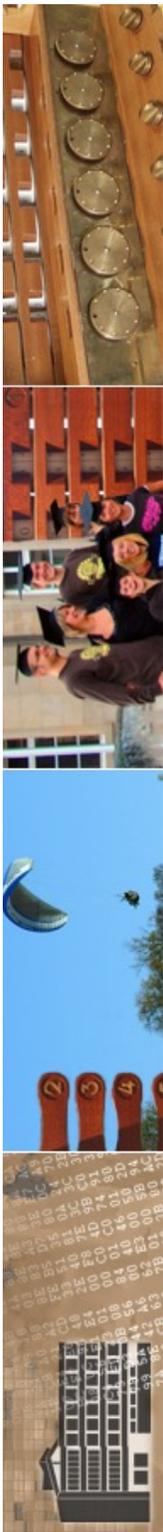




Verlassen von php

- php hat zwei „Abbruch-Anweisungen“:
 - `exit`
 - `die`

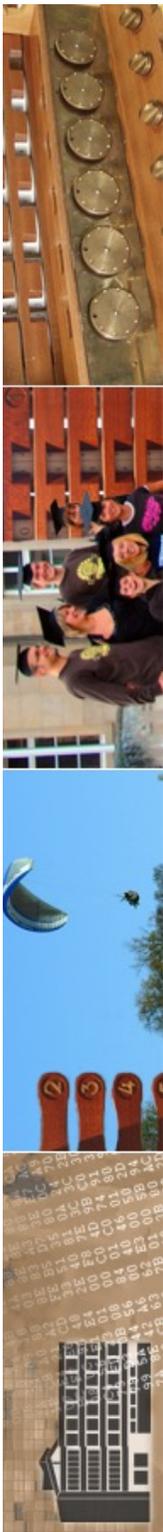
- es kann als Parameter eine Zeichenkette übergeben werden
 - `die` ist nur Alias für `exit`





Arbeiten mit Dateien

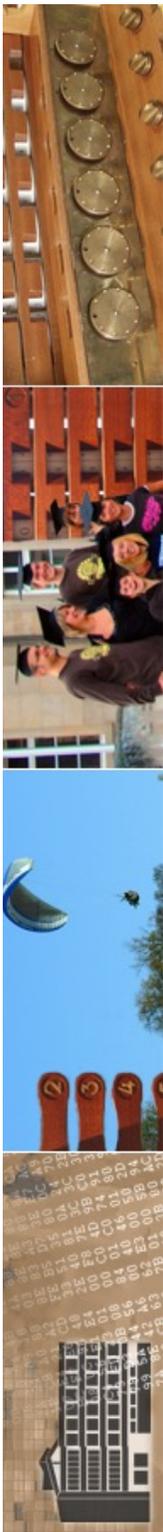
- mit der php-Methode `require()` kann eine Datei eingefügt werden
 - auch mit `include()` wird eine Datei eingefügt
- wo ist der Unterschied???





Öffnen einer Datei

- der Dateizugriff erfolgt wiederum über Filehandles
 - `$fh = fopen ($datei, $mode) ;`
- mögliche Modes:
 - r : lesen
 - r+ : lesen und schreiben
 - w: schreiben
 - w+: lesen und schreiben
 - a : anfügen





Lesen und Schreiben

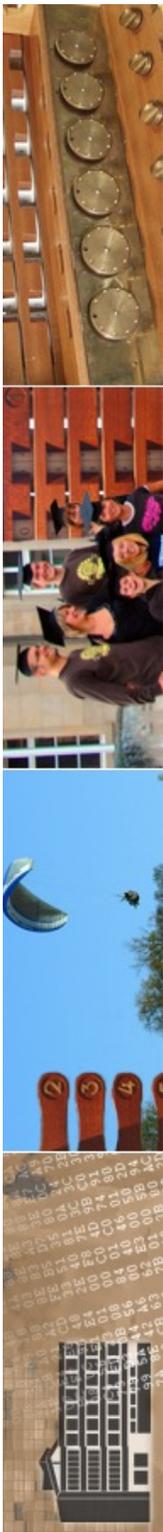
- Lesen einer Zeile:

- `$string = fgets($fh, $length);`

liest eine Zeile oder maximal `$length` viele Zeichen

- `$i = fputs($fh, $string); # schreibt Zeile`

- `$i = fclose($fh); # schließt Datei`





Funktionen in php

- Funktionen werden durch folgende Syntax deklariert:
- ```
function beispiel($arg_1, $arg_2, ...) {
 ...
}
```
- z.B. (was geschieht hier???):
  - ```
function beispiefunktion($q, $n) {
    $res = 1;
    for ($i = 1; $i <= $n; $i++)
        $res *= $q;
    return $res;
}
```



zrvwa01@infodienste => more funktion_1.php

```
<?php
/*
 * Grundlagen Internet-Technologien
 * benutzerdefinierte Funktionen in php
 */

        // Funktion zum Potenzieren
function hoch($q, $n)  {
    $res = 1;

    if ($n < 0) {
        $n *= -1;
        $q = 1.0 / $q;
    } // if

    for ($i = 1; $i <= $n; $i++)
        $res *= $q;
    return $res;
} // function
```

?>

```
<!DOCTYPE html>
<HTML>
```

```
<HEAD>
```

```
<TITLE>Methoden in php</TITLE>
```

```
<link rel="stylesheet" type="text/css" href="../css/webkompendium.css">
```

```
</HEAD>
```

```
<BODY>
```

```
<CENTER><HR><H2>Grundlagen Internet-Technologien</H2>
```

```
<H3>selbstdefinierte Methoden</H3><HR>
```

```
<H3><BR>2 hoch 16 ist
```

```
<?php
```

```
    echo hoch(2,16);
```

```
?>
```

```
</H3><HR></CENTER>
```

```
</BODY>
```

```
</HTML>
```



< > ↻ ☰ | ⚠ 134.2.6.167/~zrvwa01/php/funktion_1.php



Grundlagen Internet-Technologien

selbstdefinierte Methoden

2 hoch 16 ist 65536



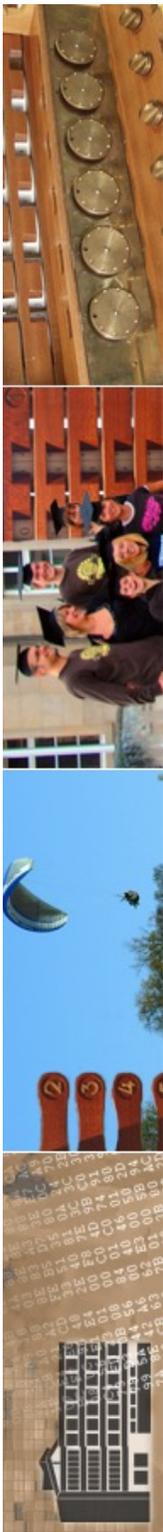
Auslagerung in separate Datei

- typisches Suffix: `.inc`
- Importieren mit
 - `include`
 - `include_once`
 - `require`
 - `require_once`



Parameterübergabe

- php unterstützt keine »variablen Parameterlisten«, d.h. die Anzahl (nicht jedoch der Typ) der Argumente muss bei der Funktionsdeklaration bereits feststehen
- standardmäßig werden beim Funktionsaufruf Werte übergeben (Call-by-Value)
 - Call-by-Reference ist möglich durch Voranstellen des Referenzierungsoperators &





Defaultwerte von Methoden

- bei der Funktionsdeklaration können Default-Werte gesetzt werden, so dass für »fehlende« Parameter diese verwendet werden

```
- function who_is_who($wer, $job =
  "Bundeskanzler(in)") {
    echo("$wer ist $job \n");
  }
```

```
who_is_who("Angela");
who_is_who("Micky", "Maus");
```



Referenzen in php

- Referenzen in php erlauben wie üblich das Ansprechen einer Variablen mit verschiedenen Namen

Syntax:

- `$a = &$b;`

- bedeutet, daß \$a und \$b zum *gleichen* Speicherplatz zeigen
- & ist der Referenzierungsoperator



Objektorientierung in PHP

- php ab Version 4 bietet eine *grundlegende Objektorientierung*
 - Definition von Klassen mit Attributen und Methoden
 - Erzeugung von Objekten
 - Konstruktoren
 - Vererbung
 - es gibt in PHP 4 keine Kapselung a la private/protected, erst ab PHP 5





OO-Syntax

- Definition der Klasse über Schlüsselwort `class`
- Konstruktor: Methode `__construct (...)` { ... }
- Destruktor: Methode `__destruct (...)` { ... }
- Datenkapselung
 - `public`
 - `protected`
 - `private`
- Methode `__toString()`
- Instanziierung: `new`





```
zrvwa01@infodienste => more Buch.php
```

```
<?php
```

```
/*
```

```
 * Grundlagen Internet-Technologien
```

```
 * Webapplikationen mit PHP
```

```
 *
```

```
 * Objektorientierung in PHP
```

```
 * Klasse Buch (Klasse mit Attributen, Konstruktor, Methode)
```

```
 */
```

```
class Buch {
```

```
    // Attribute (gekapselt)
```

```
    protected $titel,$nname,$vname,$verlag,$buch,$jahr;
```

```
    // Syntax PHP4:
```

```
    //function Buch($titel,$nname,$vname,$verlag="Springer",$jahr=2021) {
```

```
    // Konstruktor
```

```
    function __construct($titel,$nname,$vname,$verlag="Springer",$jahr=2021) {
```

```
        $this->titel = $titel;
```

```
        $this->nname = $nname;
```

```
        $this->vname = $vname;
```

```
        $this->verlag = $verlag;
```

```
        $this->jahr = $jahr;
```

```
    } // Konstruktor
```

```
    // Methode
```

```
    public function __toString() {
```

```
        $buch = $this->vname." ".$this->nname.": ".$this->titel.
```

```
        " (".$this->verlag.", ".$this->jahr.)";
```

```
        return $buch;
```

```
    } // __toString
```

```
} // class
```

```
?>
```

```
zrvwa01@infodienste =>
```



```
zrvwa01@infodienste =>
zrvwa01@infodienste => more buchhandlung1.php
<?php
/*
 * Grundlagen Internet-Technologien
 * Webapplikationen mit PHP
 *
 * Objekte: Instanzen von Klassen
 */

include_once("Buch.php");

?>
<!DOCTYPE html>
<HTML>
    <HEAD>
        <TITLE>WebKompodium: Instanzen in PHP</TITLE>
        <link rel="stylesheet" type="text/css"
            href="../css/webkompodium.css">
        <link rel="shortcut icon"
            href="../css/favicon.ico">
    </HEAD>
    <BODY>
        <CENTER><HR><H2>Grundlagen Internet-Technologien</H2>
        <H3>Instanzen in PHP</H3><HR><H3>

        <?php
            $meinBuch = new Buch ("Kompodium der Web-Programmierung",
                                "Walter", "Thomas");

            echo $meinBuch;
        ?>

        </H3><HR></CENTER>

    </BODY>
</HTML>
zrvwa01@infodienste =>
```



< > ↻ ☰ | ⚠ 134.2.6.167/~zrvwa01/php/buchhandlung1.php



Grundlagen Internet-Technologien

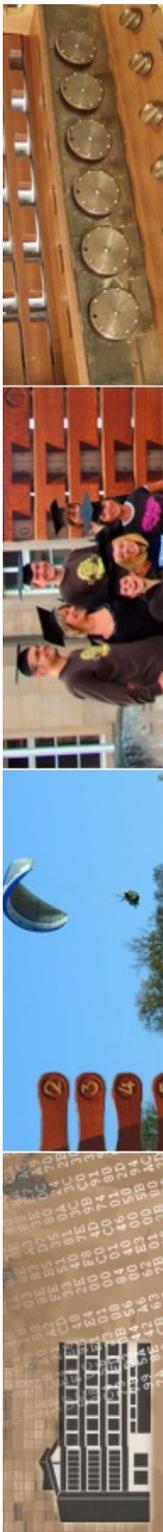
Instanzen in PHP

**Thomas Walter: Kompendium der Web-
Programmierung (Springer, 2021)**



Vererbung in PHP

- PHP bietet die klassische Vererbung
 - `class B extends A`
- Zugriff auf Elternklasse über `parent`
(entspricht `super` von Java)





```
zrvwa01@infodienste => more BuchIsbn.php  
<?php
```

```
/*  
 * Grundlagen Internet-Technologien  
 * Webapplikationen mit PHP  
 *  
 * Objektorientierung in PHP  
 * Vererbung: Ableiten der Klasse "Buch" zur Klasse "BuchIsbn"  
 */  
  
include_once("Buch.php");  
  
class BuchIsbn extends Buch {  
  
    // neues Attribut  
    protected $isbn;  
  
    // Konstruktor  
    function __construct($isbn,$titel,$nname,$vname,  
                        $verlag="Springer",$jahr=2021) {  
        parent::__construct($titel,$nname,$vname,$verlag,$jahr);  
        $this->isbn = $isbn;  
    } // Konstruktor  
  
    // Methoden  
    function __toString() {  
        $buch = parent::__toString()." - ISBN ".$this->isbn;  
        return $buch;  
    } // to_s  
  
} // class  
  
?>  
zrvwa01@infodienste =>
```





```
zrvwa01@infodienste => more buchhandlung2.php
```

```
<?php
```

```
/*
```

```
 * Grundlagen Internet-Technologien
```

```
 * Webapplikationen mit PHP
```

```
 *
```

```
 * Objekte: Instanzen von Klassen
```

```
 */
```

```
require_once("BuchIsbn.php");
```

```
?>
```

```
<!DOCTYPE html>
```

```
<HTML>
```

```
  <HEAD>
```

```
    <TITLE>Grundlagen Internet-Technologien: Instanzen in PHP</TITLE>
```

```
    <link rel="stylesheet" type="text/css"
```

```
        href="../css/webkompendium.css">
```

```
    <link rel="shortcut icon"
```

```
        href="../css/favicon.ico">
```

```
  </HEAD>
```

```
  <BODY>
```

```
    <CENTER><HR><H2>Grundlagen Internet-Technologien</H2>
```

```
    <H3>Instanzen in PHP</H3><HR><H3>
```

```
    <?php
```

```
        $meinBuch = new BuchIsbn ("3-540-33134-4",
```

```
        "Kompendium der Web-Programmierung",
```

```
        "Walter", "Thomas");
```

```
        echo $meinBuch;
```

```
    ?>
```

```
  </H3><HR></CENTER>
```

```
</BODY>
```

```
</HTML>
```

```
zrvwa01@infodienste =>
```



< > ↻ 🗖 | ⚠ 134.2.6.167/~zrvwa01/php/buchhandlung2.php



Grundlagen Internet-Technologien

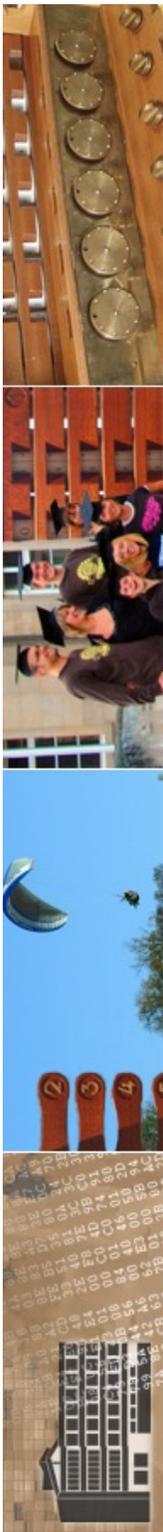
Instanzen in PHP

**Thomas Walter: Kompendium der Web-Programmierung
(Springer, 2021) - ISBN 3-540-33134-4**



weiteres zu OO in PHP

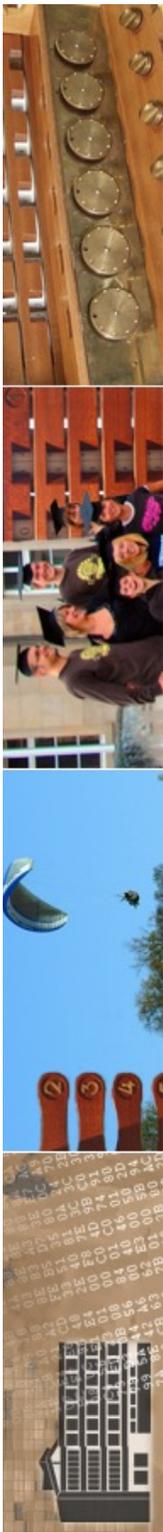
- viele Mechanismen, die wir aus Java kennen:
 - **abstract**
 - **final**
 - schwache Mehrfachvererbung über Interface
 - Ausnahmebehandlung über **try catch finally**
 - Klasseigenschaften und -methoden über **static**





ZEND

- 1999: Andi Gutmans und Zeev Suraski gründen ZEND
 - Kernentwickler von php
 - mehrere Produkte rund um professionelles php
 - ZEND Studio: Entwicklungsumgebung
 - ZEND Engine
 - ZEND Framework
 - ZEND Guard





Enterprise PHP Solutions, Groundbreaking PHP Support.

Support Your Mission-Critical Applications With Secure PHP Runtimes From Zend

TRY ZENDPHP

PHP Expertise & Support on Your Schedule

Working with PHP? Zend can help support, migrate, or modernize your PHP application.



Upgrade Flexibility

Zend supports older versions of PHP with security patches and updates, so you can upgrade on your own schedule.



Long-Term Support

Zend offers 3+ years of security and defect fixes after community support ends, meaning peace of mind for non-LTS PHP apps.





SOLUTIONS

SERVICES

TRAINING

RESOURCES

SUPPORT

TRY FREE

Zend Server

PHP Application Server

Zend Server Full Stack

Enterprise Support

ZendPHP Enterprise

PHP Runtime and Support

Laminas Enterprise Support

Formerly Zend Framework

Groundbreaking PHP Support.

Support Your Mission-Critical Applications With Secure PHP Runtimes From Zend

TRY ZENDPHP





...und nun...

- haben wir das Prinzip und die Grundsyntax von php kennen gelernt und erste php-Applikationen gesehen
- als nächstes mit php:
 - **Datenbankzugriff** mit PHP
 - Design Patterns mit PHP (MVC)
 - Erweiterung von PHP mit Bibliotheken (PEAR)
 - Cookies und Sessions

