

Bazy danych II

Andrzej Grzybowski

Instytut Fizyki, Uniwersytet Śląski

Wykład 10

Wprowadzenie do architektury oprogramowania
zdalnie dostępnych baz danych

Modele architektury oprogramowania baz danych

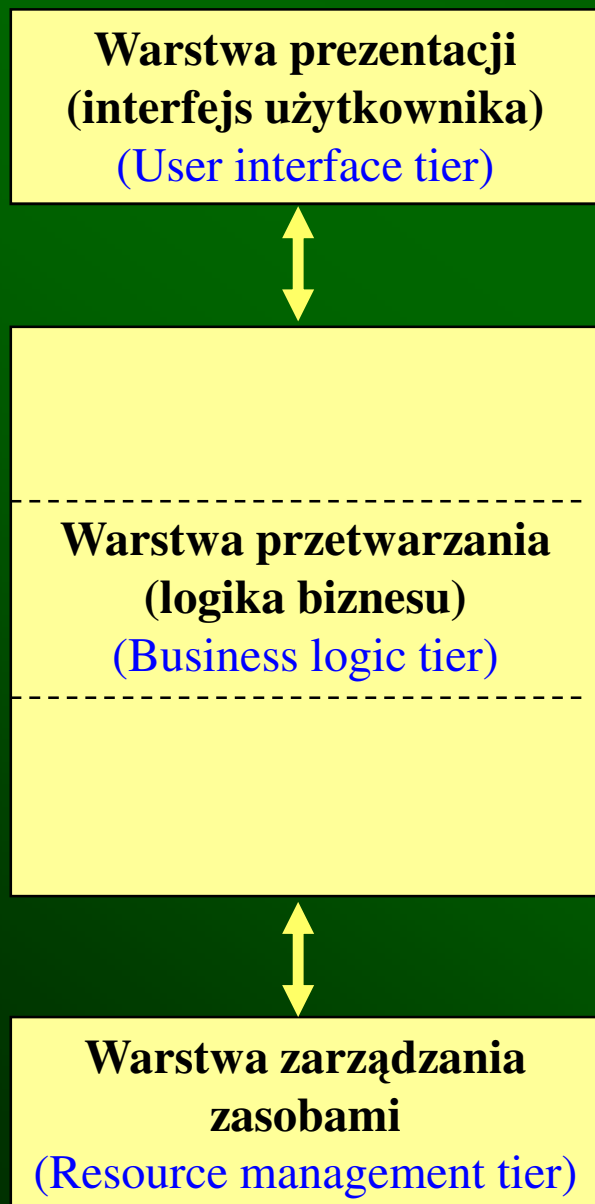
➤ **Architektura klient-serwer**

- model dwuwarstwowy
 - wersja z cienkim klientem
 - wersja z grubym klientem
- model trójwarstwowy i wielowarstwowy

➤ **Architektura obiektów rozproszonych**

Architektura trójwarstwowa i wielowarstwowa

(Three-layer and multi-layer / three-tier and multi-tier architecture)



- Podstawową cechą **architektur warstwowych** jest wyraźne rozdzielenie warstw zarówno na poziomie projektu (logicznym), jak i implementacji (funkcyjnym). Pozwala to na duży stopień niezależności podczas rozbudowy poszczególnych warstw.
- Architektura trójwarstwowa stanowi obecnie wygodną podstawę do analizy modeli typu klient-serwer.
- Architektura trójwarstwowa jest często utożsamiana z architekturą wielowarstwową.
- Bardziej precyzyjna definicja określa **architekturę wielowarstwową** jako architekturę trójwarstwową, w której środkowa warstwa składa się z kilku warstw.
- Oczywiście także warstwa prezentacji może zawierać wielu klientów, a warstwa zarządzania może zajmować się wieloma zasobami, ale to nie wpływa na zmianę ogólnego typu architektury.

Architektura dwuwarstwowa

(Two-layer / two-tier architecture)

jako uproszczona architektura trójwarstwowa

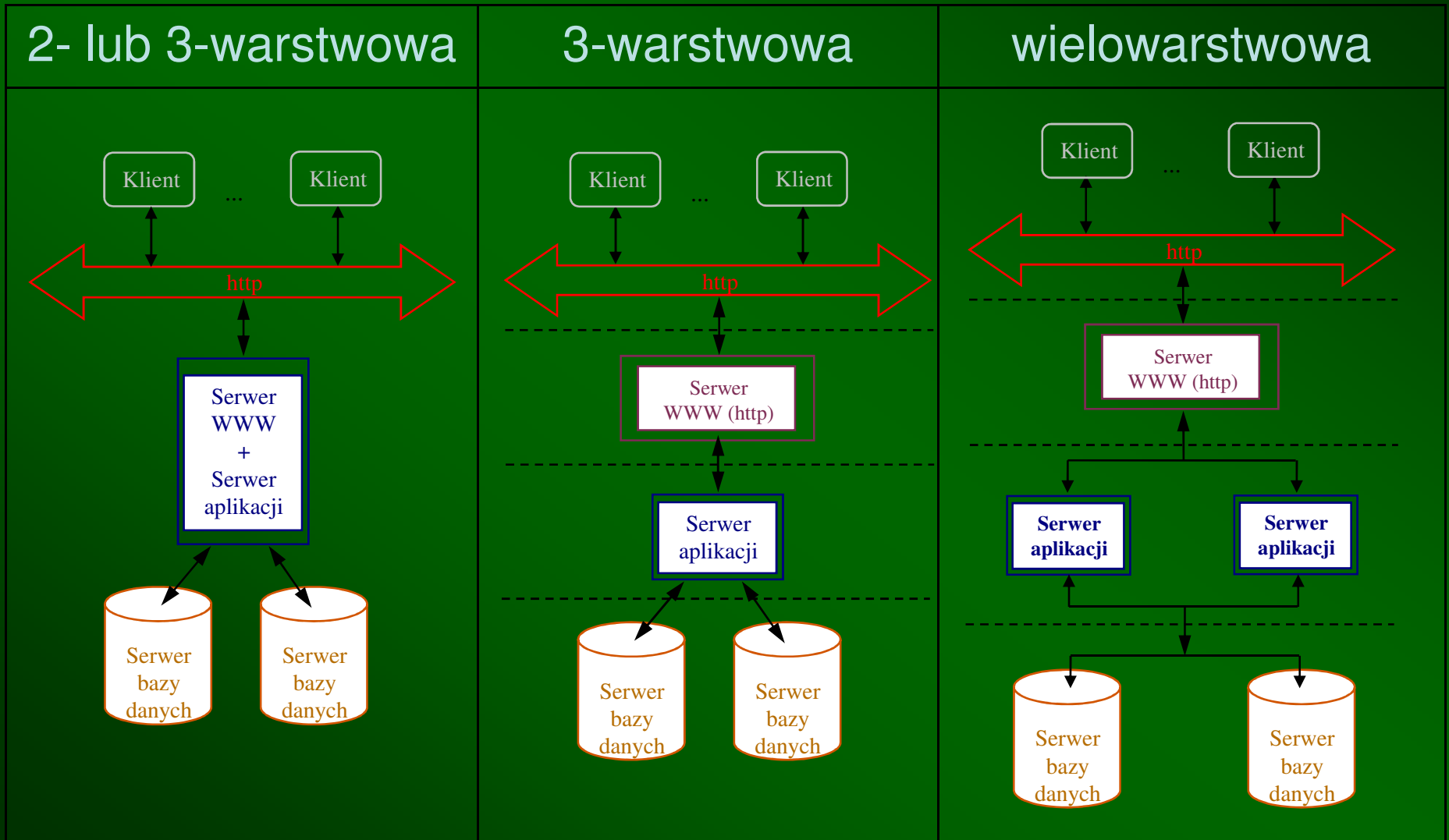


Uwaga: terminy cienkiego i grubego klienta mogą być również stosowane w przypadku architektury trójwarstwowej i wielowarstwowej

Porównanie modeli cienkiego i grubego klienta

Cienki klient	Gruby klient
<ul style="list-style-type: none">• Klient posiada niewielką moc przetwarzania, ograniczoną do prezentacji danych na ekranie, np. przeglądarka WWW, co implikuje jego niewielkie wymagania sprzętowe.• Wadą jest duże obciążenie serwera i linii komunikacyjnych.	<ul style="list-style-type: none">• Klient posiada bogatsze możliwości przetwarzania, ponieważ oprócz prezentacji danych może realizować przetwarzanie aplikacji (tzw. logikę biznesu).• Potrzebuje większej mocy komputera klienta do przetwarzania zarówno prezentacji, jak i logiki biznesu.• Serwer zajmuje się głównie obsługą transakcji bazy danych, a niekiedy, pełniąc także rolę serwera plików, wysyła aplikacje do klienta.• Projektowanie i zarządzanie w modelu grubego klienta jest bardziej złożone.• Typowym przykładem grubego klienta jest bankomat.

Architektury aplikacji WWW typu klient-serwer

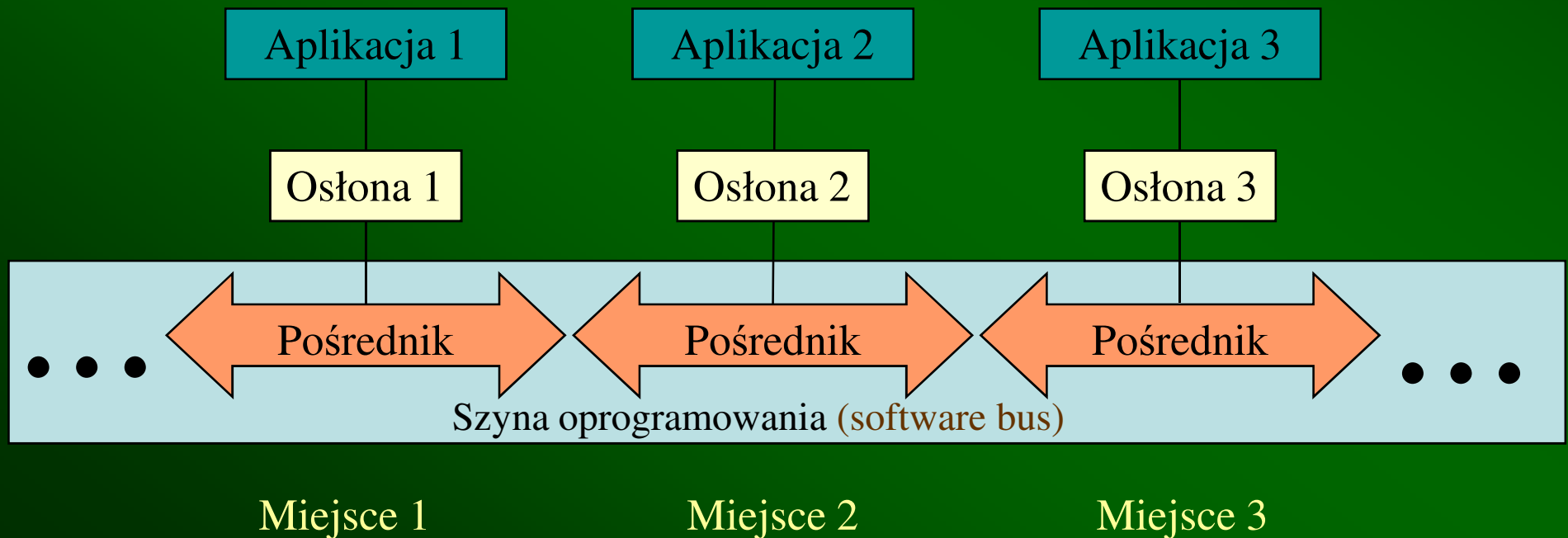


Architektura obiektów rozproszonych (komponentowa)

- Architektura typu klient-serwer nawet w swojej wielowarstwowej wersji dla wielu złożonych zastosowań jest mało elastyczna i zapewnia zbyt małą skalowalność. Jest to w dużej mierze spowodowane wyraźnym rozróżnieniem pomiędzy klientem a serwerem, w tym szczególnie brakiem bezpośredniej komunikacji pomiędzy klientami.
- Dlatego architektura obiektów rozproszonych usuwa podział na klientów i serwery, w takim sensie, że każde miejsce w systemie rozproszonym jest zarówno klientem, jak i serwerem.
- Wymóg dostępności dowolnego obiektu z dowolnego miejsca systemu rozproszonego implikuje sprowadzenie wszystkich danych i usług do jednego standardu.

Standard danych i usług w architekturze obiektów rozproszonych musi zawierać:

- ✓ Model danych i usług, który pozwoli przyjąć wszystkie możliwe dane i usługi, mogące kiedykolwiek pojawić się w systemie rozproszonym i uczynić ich lokalizację transparentną (niewidoczną);
- ✓ Pośrednik (broker) – oprogramowanie dostosowane do wspólnego modelu danych i usług, które umożliwia ich udostępnianie w dowolnych miejscach systemu rozproszonego;
- ✓ Osłonę (adapter, mediator) - oprogramowanie, które przystosowuje konkretne miejsce do modelu przyjętego przez pośrednika.



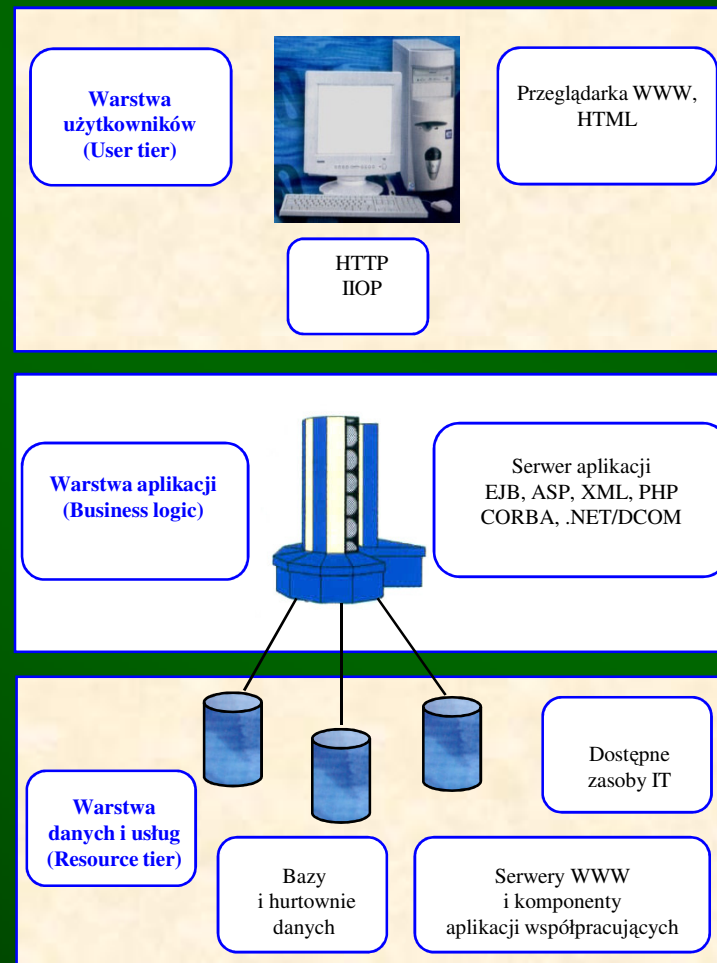
Logika funkcjonowania obiektów rozproszonych:

- ✓ Szyna oprogramowania udostępnia jedną przestrzeń obiektów.
- ✓ Obiekty są dostępne dla dowolnego miejsca poprzez operacje zgrupowane w klasach.
- ✓ Miejsca i sposoby implementacji obiektów są niewidoczne.
- ✓ Aplikacje korzystają z całego zasobu obiektów.
- ✓ Aplikacje są budowane z gotowych lub tworzonych samodzielnie komponentów obiektowych.

Przykłady istniejących technologii o architekturze rozproszonej

- ✓ *OMG CORBA (Object Management Group Common Object Request Broker Architecture)* z protokołem komunikacyjnym *IIOP (Internet Inter-ORB Protocol)*
- ✓ *Sun EJB (Enterprise JavaBeans)* z protokołem komunikacyjnym *Sun RMI (Remote Method Invocation)*
- ✓ *SOAP (Simple Object Access Protocol)* z protokołem komunikacyjnym *HTTP*, intensywnie wykorzystujący *XML*
- ✓ *Microsoft .NET* lub *ServiceComponent* np. z protokołem *DCOM (Distributed Component Object Model)*

Architektura rzeczywistych aplikacji WWW jest najczęściej hybrydowa, tzn. wielowarstwowa i rozproszona (komponentowa)



Wstęp do PHP

historia, zastosowania,
podstawy języka (zmienne, stałe, proste i specjalne typy danych)

Co to jest PHP?

- **PHP (PHP: Hypertext Preprocessor)**
- PHP jest popularnym językiem skryptowym ogólnego zastosowania, dopasowanym do potrzeb aplikacji WWW, z możliwością zagnieżdżania w HTML.
- PHP posiada szeroki zestaw narzędzi umożliwiających szybkie pisanie dynamicznych stron WWW i do tego celu jest głównie wykorzystywany w skryptach wykonywanych po stronie serwera. Jednak nie jest to jego jedyne przeznaczenie.
- Składnia języka bazuje na językach C, Java i Perl, i jest łatwa do nauczenia. PHP jest tworzony na zasadach Open Source.

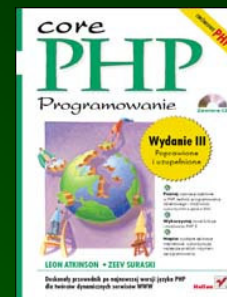
Literatura do PHP

PHP. Programowanie. Wydanie III

Leon Atkinson, Zeev Suraski

Helion 04/2004, stron: 952, zawiera CD-ROM

Tytuł oryginału: [Core PHP Programming, 3 Edition](#)

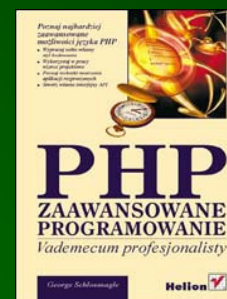


PHP. Zaawansowane programowanie. Vademecum profesjonalisty

George Schlossnagle

Helion 10/2004, stron: 624, przykłady na ftp: 72 kB

Tytuł oryginału: [Advanced PHP Programming](#)



PHP i MySQL. Dynamiczne strony WWW. Szybki start

Larry Ullman

Helion 01/2004, stron: 580

Tytuł oryginału: [PHP and MySQL for Dynamic Web Sites VQPG](#)

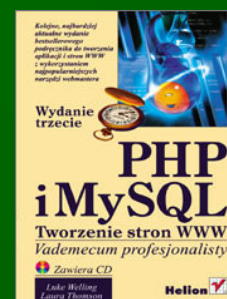


PHP i MySQL. Tworzenie stron WWW. Vademecum profesjonalisty.

Wydanie trzecie, Luke Welling, Laura Thomson

Helion 02/2005, stron: 912, Zawiera CD-ROM

Tytuł oryginału: [PHP and MySQL Web Development, 3rd Edition](#)



http://www.php.net

The screenshot shows the PHP website as it appeared in early 2005. The browser window title is "PHP: Hypertext Preprocessor - Microsoft Internet Explorer". The address bar shows "http://www.php.net". The page features a purple header with the PHP logo and navigation links: "downloads", "documentation", "faq", "getting help", "mailing lists", "reporting bugs", "php.net sites", "links", and "my php.net". A search bar is visible with the text "search for MSOffice/cltreq.asp?UL=1&AC in the function list".

What is PHP?
PHP is a widely-used general-purpose scripting language that is especially suited for Web development and can be embedded into HTML. If you are new to PHP and want to get some idea of how it works, try the [introductory tutorial](#). After that, check out the online [manual](#), and the example archive sites and some of the other resources available in the [links section](#).

Ever wondered how popular PHP is? see the [Netcraft Survey](#).

Thanks To

- [easyDNS](#)
- [Directi](#)
- [pair Networks](#)
- [EVIServers](#)
- [Server Central](#)
- [Hosted Solutions](#)
- [Spry VPS Hosting](#)
- [eZ systems / HiT](#)

Related sites

- [Apache](#)
- [MySQL](#)
- [PostgreSQL](#)

php|symphony

[10-Feb-2005] php|architect magazine has introduced [php|symphony](#), an ongoing series of online talks given by some of today's leaders in the PHP world. Each talk in the series features an in-depth discussion on a specific topic, such as security, enterprise development and so on. Talks are delivered entirely online using an interactive system compatible with most OSs and browsers.

The series will start on February 23rd with a talk by security expert Chris Shiflett titled "Mastering PHP Security".

ApacheCon Europe Call for Papers

[08-Feb-2005] ApacheCon Europe will be held from July 18th to July 22th in Stuttgart, Germany. The conference organizers are looking for speakers willing to have a talk of any of the topics mentioned on the [website](#). If you think that you have material for this conference and want to have a talk there, you can submit a paper [here](#).

PHP Security Consortium

[31-Jan-2005] An international group of PHP experts today announced the official launch of the [PHP Security Consortium](#) (PHPSC), a group whose mission is to promote secure programming practices within the PHP community through education and exposition while maintaining high ethical standards.

Members of the PHPSC seek to educate PHP developers about security through a variety of resources, including documentation, tools, and standards. In addition to their educational efforts, the PHPSC engages in exploratory and experimental research in order to develop and promote standards of best practice for PHP application development.

PHP SECURITY CONSORTIUM

This mirror sponsored by:

WebHosting TALK
THE PLACE FOR EVERYTHING ABOUT WEBHOSTING
"Now Featuring PHP Forums"
www.webhostingtalk.com

Upcoming Events [\[add\]](#)

February

User Group Events

- 19. [Kansas City](#)
- 19. [Los Angeles LAMP SIG](#)
- 19. [Miami Linux Users Group](#)
- 20. [A meeting for Taiwan PHP Users](#)
- 22. [New York](#)
- 22. [AzPHP](#)
- 23. [Munich](#)
- 24. [Arabic PHP Group Meeting](#)
- 25. [Malaysia PHP Group First Meeting](#)
- 25. [MMUG-ar | Aplicaciones Web](#)
- 26. [PHP Devroom @ FOSDEM](#)
- 28. [Long Island PHP Users](#)

Taskbar: Dyskusje, Subskrybuj..., Dyskusje są niedostępne na http://www.php.net/

<http://www.orafaq.com/faqphp.htm>

The screenshot shows a Microsoft Internet Explorer browser window. The title bar reads "Oracle FAQ: Oracle/ PHP FAQ - Microsoft Internet Explorer". The address bar contains "http://www.orafaq.com/faqphp.htm". The page content includes a navigation menu with "Index", "Search", "Add FAQ", and "Ask Question". Below the menu are two promotional boxes: one for "Blob calls without bother" and another for "Oracle" comparing SAP vs. Oracle vs. PeopleSoft. The main heading is "Oracle/ PHP FAQ", followed by metadata: "\$Date: 26-Jul-2003 \$", "\$Revision: 1.01 \$", and "\$Author: Frank Naudé \$". A "Topics" section lists several FAQ topics as links. At the bottom, there is a link "Back to Oracle FAQ Index" and a section titled "What is PHP and what's it got to do with Oracle?" with a paragraph of introductory text. The browser's status bar at the bottom shows "Dyskusje są niedostępne na http://www.orafaq.com/".

Oracle FAQ: Oracle/ PHP FAQ - Microsoft Internet Explorer

Plik Edycja Widok Ulubione Narzędzia Pomoc

Wstecz Wyszukaj Ulubione Multimedia

Adres <http://www.orafaq.com/faqphp.htm> Przejdź Łącza

[Index](#) [Search](#) [Add FAQ](#) [Ask Question](#)

Blob calls without bother
Download writes JDBC Calls for SQL Statements That Work With Blobs
www.orindabuild.com

Oracle
SAP vs. Oracle vs. PeopleSoft. Choose Clarity, then choose an ERP.
www.Niku.com

Oracle/ PHP FAQ

\$Date: 26-Jul-2003 \$
\$Revision: 1.01 \$
\$Author: [Frank Naudé](#) \$

Topics

- [What is PHP and what's it got to do with Oracle?](#)
- [What is the difference between the OCI and ORA extension modules?](#)
- [How does one configure PHP to use Oracle?](#)
- [How does one connect to Oracle?](#)
- [Why do we get error "Call to undefined function: ora_logon\(\)/ ocilogon\(\)"?](#)
- [How does one SELECT, INSERT, UPDATE and DELETE data from PHP?](#)
- [How are database transactions handled in PHP?](#)
- [How are database errors handled in PHP?](#)
- [How does one call stored procedures from PHP?](#)
- [Does PHP offer Oracle connection pooling?](#)
- [Where can one find more info about PHP and Oracle?](#)

[Back to Oracle FAQ Index](#)

What is PHP and what's it got to do with Oracle?

PHP is a recursive acronym for "PHP Hypertext Preprocessor". It is an open source, interpretive, HTML centric, server side scripting language. PHP is especially suited for Web

Dyskusje są niedostępne na <http://www.orafaq.com/>

<http://www.oracle.com/technology/products/jdev/htdocs/partners/addins/exchange/php/index.html>

ORACLE
TECHNOLOGY NETWORK

(Sign In / Register for a free Oracle Web account)

ORACLE.COM PARTNERS BUY DOWNLOAD SUPPORT CONTACT US

Developer8i PHP Extension SEARCH SELECT COUNTRY

PRODUCT CENTERS
TECHNOLOGY CENTERS
COMMUNITY

SERVICES

- Downloads
- Documentation
- Discussions
- Articles
- Sample Code
- Training
- Support

RESOURCES FOR

- Developers
- DBAs
- ISVs
- Small & Midsize Businesses
- Getting Started

The Oracle JDeveloper 10g PHP Extension

Overview

The PHP Extension makes it easy to create, edit, and run PHP scripts in Oracle JDeveloper 10g.

The PHP Extension, combined with JDeveloper's comprehensive functionality, help make web and database development productive for the PHP developer.

- PHP, HTML, CSS, XML, Javascript, SQL and PL/SQL support
- Structure view
- Online Help
- Pre-supplied code blocks
- Database connectivity
- PL/SQL debugger
- Database object browser
- Visual database table creation

The PHP Extension can be used in combination with the Application Migration Assistant Extension to migrate non-Oracle based PHP applications to Oracle Database.

Prerequisites

- Oracle JDeveloper 10g 9.0.5.x
- PHP 4.3.4 or later.
- A web server such as Apache 1.3 or later.

Installation

1. Download the PHP Extension, [php-addin.zip](#).
2. Extract the file *php-addin.jar* from the zip file into the `<jdev_root>/jdev/lib/ext` directory

Downloads

- [Download PHP Extension](#)
- [Download Oracle JDeveloper](#)
- [Download PHP](#)
- [Download Apache Web Server](#)
- [Download Application Migration Assistant](#)

Resources

- [PHP Extension Release Notes](#)
- [Oracle and PHP Installation FAQ](#)
- [Oracle and PHP FAQ](#)
- [Oracle PHP Discussion Forum](#)
- [JDeveloper Discussion Forum](#)
- [JDeveloper Home Page](#)
- [The PHP Site](#)

Dyskusje Subskrybuj... Dyskusje są niedostępne na <http://www.oracle.com/>

Historia PHP

- **PHP 1.0** (**P**ersonal **H**ome **P**age **T**ools 1.0) zaprojektował **Rasmus Lerdorf** w 1994 roku. Był to mały zestaw skryptów Perla (**P**actical **E**xtraction and **R**eport **L**anguage) zastosowany przez przepisany zbiór binariów CGI (**C**ommon **G**ateway **I**nterface) napisanych w C, służący do filtrowania podanych mu informacji i zamieniania prostych poleceń na HTML. Pierwszym praktycznym zastosowaniem PHP 1.0 było śledzenie gości przeglądających internetową stronę autora, ale już wtedy możliwe było umieszczanie zapytań SQL na stronach WWW.
- **PHP 2.0** (**PHP/FI**, tzn. **PHP/Form** Interpreter) – pod wpływem ogromnego zainteresowania użytkowników sieci WWW Rasmus Lerdorf konstruuje interpreter formularzy i udostępnia drugą wersję PHP, która:
 - pozwalała umieszczać kod programowania strukturalnego (pętle, warunki, złożone struktury danych) w plikach HTML,
 - potrafiła analizować dane przesłane z formularzy HTML, komunikować się z bazami danych i wykonywać *w locie* skomplikowane obliczenia.

Historia PHP

- **PHP 3.0 (PHP/FI 2.0)** – dwóch studentów, Andi Gutmans i Zeev Suraski, realizując projekt sklepu internetowego przy pomocy PHP znalazło wiele błędów, często wynikających ze złamania zasad programowania. W 1997 roku napisali oni parser PHP niemal od początku, czyniąc język PHP spójnym i skutecznym, a pełna oficjalna trzecia wersja PHP została udostępniona w 1998. Nowi twórcy zmienili też nazwę języka na jej współczesną rekursywną wersję **PHP: Hypertext Preprocessor**. Jednak pod wpływem swojego wykładowcy Gutmans i Suraski skontaktowali się z Rasmusem Lerdorfem, który zaakceptował ich implementację i w ten sposób powstała grupa programistów PHP (PHP Core Team, zwana obecnie PHP Group).

Ważne pojęcia:

Preprocesor to program, który pobiera tekst na wejściu i dokonuje jego leksykalnej konwersji.

Parser to program komputerowy lub jego komponent, który analizuje gramatyczną strukturę tego co pobrał na wejściu w odniesieniu do danej gramatyki formalnej.

Historia PHP

- **PHP 4.0** (wspomagany przez **Zend Engine 1.0**) – Gutmans i Suraski, znając najlepiej niedostatki kodu PHP 3.0, natychmiast rozpoczęli pracę nad nowym jądrem PHP, które nazwali Zend Engine i już na początku 1999 roku poinformowali o jego utworzeniu. Wstępne testy pokazały, że czas wykonywania skryptów w nowej wersji spadł 100-krotnie względem PHP 3, dlatego szybko zaniechano rozwijania poprzedniego projektu, a PHP 4.0 swoją oficjalną premierę miało w 2000 roku. Oprócz poprawek, ulepszeń i optymalizacji dodano nowe istotne elementy:
 - wewnętrzna obsługa sesji,
 - buforowanie zawartości strony,
 - warstwa abstrakcji serwera sieciowego.
- **PHP 5.0** (wspomagany przez **Zend Engine 2**) – Obiektowość wprowadzona została już do wersji PHP 3, ale nie wykraczała ona wiele ponad poziom składni obiektów, których wewnętrzna implementacja niewiele różniła się od tablic. W 2004 roku ukończono piątą wersję PHP, opartą oprócz języków C i Perl na składni języka Java, która umożliwia wygodne programowanie obiektowe, udostępniając rozbudowany system obiektów.

Zalety PHP

- Darmowo dostępne kody źródłowe.
- Modyfikowalność – posiada możliwość samodzielnego dodawania nowych elementów, co już spowodowało powstanie wielu użytecznych rozszerzeń.
- Swoboda wyboru między programowaniem strukturalnym a obiektywnym.
- Łatwy dostęp do kodu programu i wyników jego działania.
- Niezależność od platformy systemowej – współpracuje z różnymi systemami operacyjnymi (UNIX, LINUX, Windows, Macintosh OS X)
- Współpraca z różnymi serwerami WWW – PHP został zaprojektowany przede wszystkim do pracy z serwerem *Apache*, ale skrypty PHP mogą być bez zmian przenoszone na różne platformy serwerowe, np. Microsoft IIS (Internet Information Server), PWS (Personal Web Server), Netscape FastTrack (iPlanet), O'Reilly Website Pro, Caudium, Xitami, OmniHTTPd. Można to osiągnąć zarówno poprzez dołączony do serwera WWW moduł PHP, jak i program CGI, np. IIS 4.0 lub nowszy na Windows NT/2000/XP umożliwia wykonywanie skryptów PHP dzięki Microsoft ISAPI (interfejs modułów Internet Server Application Interface) lub wykonywalny program CGI (php.exe).

Zalety PHP: obsługa baz danych

- **Niezwykle łatwa obsługa baz danych ze strony WWW**
 - Obecnie PHP daje możliwość współpracy z następującymi systemami baz danych:

Adabas D	Ingres	Ovrimos
dBase	InterBase	PostgreSQL
Empress	FrontBase	Solid
FilePro (tylko do odczytu)	mSQL	Sybase
Hyperwave	Direct MS-SQL	Velocis
IBM DB2	MySQL	Unix dbm
Informix	Oracle (OCI7 i OCI8)	

- PHP umożliwia połączenie niemal z dowolną bazą danych poprzez ODBC (Open Database Connection) albo abstrakcyjne rozszerzenie DBX.

Miejsce PHP wśród innych narzędzi programistycznych

PHP jest znaczącą alternatywą dla innych narzędzi do tworzenia dynamicznych stron WWW:

- **CGI/Perl (Common Gateway Interface / Practical Extraction and Report Language)** – główną wadą tego popularnego układu jest wykonywanie nowego procesu serwera WWW na każde żądanie klienta (przeglądarki WWW), co prowadzi do przeciążeń często używanych serwerów.
- **Sun Microsystems JSP/Java (JavaServer Pages / Java)** – skrypty JSP są oparte na kodzie Javy z możliwością wbudowywania go do HTML lub XML. Są one kompilowane (najczęściej na serwerze aplikacji) przez kompilator JSP, najpierw parsujący JSP, a następnie transformujący je do wykonywalnych serwletów Javy (Java servlets), które mogą występować w dwóch postaciach:
 - prekompilowanej (w celu uzyskania lepszej wydajności na komputerze klienta) – serwlet jest generowany w postaci tzw. kodu bajtowego (*bytecode*) i może być wykonany np. przez JVM (Java Virtual Machine), która oprócz interpretera Javy zawiera środowisko, umożliwiające wykonanie serwleta (*runtime*); serwlet w postaci *bytecode* może być następnie skompilowany do kodu maszynowego przy pomocy kompilatora JIT (just-in-time compiler)
 - skompilowanej (w celu lepszego sprawdzenia błędów) – serwlet jest generowany w kodzie Javy, a następnie kompilowany przez kompilator Javy do postaci wykonywalnej (do kodu maszynowego).

Kompilator JSP, stanowiący najczęściej jedno z narzędzi serwera aplikacji, jest uruchamiany przy pierwszym dostępie do kodu JSP i funkcjonuje jako jeden proces, który może obsłużyć kolejne fragmenty tego kodu, kolejne strony JSP. Współczesne przeglądarki potrafią wyświetlać serwlety Javy.

- **Microsoft ASP/VBScript/JScript (Active Server Pages / Visual Basic Scripting Edition / JavaScript)** – w skryptach głównie używa się VBScript lub JScript, ale działanie tej technologii jest podobne do CGI, tzn. po żądaniu przeglądarki o daną stronę WWW serwer generuje odpowiedni kod HTML i wysyła go do przeglądarki.
- **Microsoft ASP.NET/VBScript.NET/JScript.NET** - bazuje na technologii ActiveX i wymaga kompilacji, działa podobnie do JSP, ale w przeciwieństwie do apletów Javy sterowanie ActiveX daje pełny dostęp do systemu operacyjnego Windows, co daje większe możliwości, ale jednocześnie stwarza zagrożenia dla danych i programów zarówno na komputerze klienta, jak i na serwerze.
- **Allaire/Macromedia ColdFusion** z językiem znaczników **CFML (Cold Fusion Markup Language)**, który jest zintegrowany z bazami danych (jego pierwszą nazwą z 1995 roku to DataBase Markup Language) i podobnie jak JSP lub ASP.NET wymaga kompilacji.

Prekompilacja w PHP 5

- W PHP 5 rolę maszyny wirtualnej spełnia Zend Engine 2. Silnik Zend dokonuje **parsowania** i **prekompilacji** skryptu do kodu pośredniego (*intermediate code*), będącego trzyadresowym kodem bajtowym, który zawiera dwa operandy (dwa wejścia), jedno wyjścia i mechanizm obsługi operandów. Operandy są stałymi (reprezentującymi wartości statyczne) lub adresami tymczasowych zmiennych, które z funkcjonalnego punktu widzenia są rejestrami maszyny wirtualnej Zend.
- Silnik Zend potrafi wykonać tak przygotowane, prekompilowane skrypty PHP.

Sposoby używania PHP

- **Skrypty wykonywane po stronie serwera** – najbardziej popularne zastosowanie, do którego realizacji potrzebne są serwer WWW z parserem PHP (plik wykonywalny CGI lub moduł) i przeglądarka WWW.
- Skrypty uruchamiane z linii poleceń – do uruchomienia wymagają jedynie parsera PHP (np. php.exe, znajdujący się w podkatalogu CLI głównego katalogu instalacji PHP) i mogą służyć jako skrypty systemu operacyjnego lub analizatory tekstu.
- Aplikacje z interfejsem użytkownika wykonywane po stronie klienta – PHP nie jest wygodnym narzędziem do tego celu, ale istnieje niewłaśczone do głównej dystrybucji PHP rozszerzenie: pakiet PHP-GTK, będący obiektowo zorientowanym interfejsem do GTK+ (zestawu bibliotek graficznych GLib, GDK, GTK napisanych w C), służącym do tworzenia uruchamianych po stronie klienta, samodzielnych, wieloplatformowych aplikacji z graficznym interfejsem użytkownika (GUI), które jednak nie mogą być wyświetlane przez przeglądarkę WWW.

Skrypty PHP

- Skrypty PHP wywoływane w przeglądarce WWW (ale oczywiście wykonywane na serwerze WWW) powinny mieć rozszerzenie: **php**, chociaż niekiedy dozwolone są także rozszerzenia: phtm, phtml, php3, php4.
- Skrypty PHP mogą zawierać:
 - tylko *czysty* kod PHP,
 - tylko *czysty* kod HTML zawarty między znacznikami PHP,
 - wbudowane do kodu HTML fragmenty kodu PHP.
- Skrypty PHP mogą służyć jako generatory całych dokumentów HTML, wówczas każdy bajt przesłany do przeglądarki WWW jest wygenerowany przez parser PHP. Zaletą takiego podejścia jest pełniejsza i łatwiejsza kontrola nad wyglądem strony WWW, co znacznie ułatwia pracę, szczególnie w przypadku dużego projektu.

Składnia PHP: znaczniki otwierające i zamykające kod

- Zalecane ze względu na dostosowanie do XML

`<?php ... ?>`

- Popularne, ale źle działające z innymi narzędziami (np. z interpreterem XML lub edytorem Front Page). Dostępne, jeżeli w pliku konfiguracyjnym PHP.INI wybrano opcję `short_open_tag = On` lub dokonano kompilacji z opcją `--enable-short-tags`

`<? ... ?>`

- Takie jak w przypadku ASP. Dostępne, jeżeli w pliku konfiguracyjnym PHP.INI wybrano opcję `asp_tags = Off`

`<% ... %>`

- Podobne jak w przypadku JavaScript

`<script type="text/php" language="php"> ... </script>`

Składnia PHP: oddzielanie instrukcji i komentarze

- Każda instrukcja musi być zakończona średnikiem

instrukcja;

- Mamy dwa główne sposoby wprowadzania komentarzy do kodu PHP

- komentarze jednoliniowe, działające do końca wiersza lub do końca bloku PHP, jeśli wystąpi on najpierw

```
<?php
```

```
// komentarz jednoliniowy w stylu C++
```

```
# komentarz jednoliniowy w stylu shella
```

```
instrukcja; // tutaj też można wstawić komentarz jednoliniowy
```

```
instrukcja; # ten komentarz obowiązuje do końca bloku PHP ?>
```

– komentarz wieloliniowy

```
<?php
/*
    komentarz
    wieloliniowy
*/
?>
```

Nie powinno się zagnieżdżać komentarzy wieloliniowych

```
<?php
/*
    komentarz
        /* Ten zagnieżdżony komentarz
           spowoduje błąd! */
    wieloliniowy
*/
?>
```

Składnia PHP: wybrane sposoby rozwiązania problemu polskich czcionek

Można skorzystać z dwóch niezależnych i niedostępnych jednocześnie metod:

- Zastosowanie rozszerzenia *iconv* – wymaga ustawienia w pliku PHP.INI opcji `output_handler = ob_iconv_handler`, a w przypadku PHP 4 trzeba wprowadzić to rozszerzenie w pliku PHP.INI dodatkowo np. komendą `extension=php_iconv.dll`. Wówczas można w PHP.INI zdefiniować parametry (przykład zawiera definicję polskiej strony kodowej Latin 2):

```
iconv.input_encoding = "ISO-8859-2"
```

```
iconv.output_encoding = "ISO-8859-2"
```

```
iconv.internal_encoding = "ISO-8859-2"
```

W przypadku kompilacji należy użyć opcji `--with-iconv[=DIR]`, gdzie `DIR` jest ścieżką dostępu do katalogu rozszerzenia *iconv*.

- Zastosowanie rozszerzenia *mbstring* – wymaga ustawienia w pliku PHP.INI opcji `output_handler = mb_output_handler` i dodania tego rozszerzenia, np. `extension=php_mbstring.dll`, dalej postępowanie jest podobne jak dla rozszerzenia *iconv*. W przypadku kompilacji należy użyć opcji `--enable-mbstring`

Składnia PHP: zmienne i typy danych

- Nazwy zmiennych zaczynają się od symbolu dolara \$
\$nazwa_zmiennej
- *nazwa_zmiennej* może zawierać litery, cyfry i znaki podkreślenia, ale po symbolu dolara jako pierwsza nie może wystąpić cyfra. Istotna jest wielkość liter.
- W PHP nie deklaruje się typu danych dla zmiennej specjalną instrukcją. Typ danych zmiennej jest określany przez przypisanie zmiennej wartości o odpowiednim typie (operator przypisania oznaczamy symbolem =).
- Można wyróżnić trzy grupy typów danych
 - Proste (skalarne) typy danych:
całkowity, zmiennoprzecinkowy, łańcuchowy, logiczny
 - Złożone typy danych:
tablicowy, obiektowy
 - Specjalne typy danych:
identyfikator zasobów, wartość pusta (NULL)

Składnia PHP: liczby całkowite

- Typ całkowity określany jako *integer* oparty został na typie *long* z języka C.
- Jego zakres zależy od architektury procesora i na procesorze 32-bitowym wartości całkowite należą do zbioru liczb całkowitych o kresie dolnym – 2 147 483 648 i kresie górnym 2 147 483 647. PHP nie obsługuje liczb całkowitych bez znaku.
- Gdy zostanie przekroczony zakres typu całkowitego czy to przy podstawianiu wartości, czy też w wyniku obliczeń, to wartość zostanie potraktowana jako zmiennoprzecinkowa.
- Zmienne całkowite mogą być definiowane w trzech notacjach:

`$a = 1234 // dodatnia liczba dziesiętna`

`$b = -123 // ujemna liczba dziesiętna`

`$c = 034 // liczba ósemkowa równa liczbie dziesiętnej 28`

`$d = 0x2A // liczba szesnastkowa równa liczbie dziesiętnej 42`

Składnia PHP: liczby zmiennoprzecinkowe

- Typ zmiennoprzecinkowy określany często jako *float* w rzeczywistości oparty został na typie *double* z języka C.
- Jego zakres także zależy od architektury procesora i na procesorze 32-bitowym najczęściej osiąga wartość bezwzględną równą w przybliżeniu $1.8e308$ przy precyzji 14 cyfr dziesiętnych po przecinku, co jest zgodne z 64-bitowy formatem IEEE (*The Institute of Electrical and Electronic Engineers*)
- Do obliczeń o dowolnej dokładności (dowolnej precyzji) można użyć biblioteki *BCMath* lub grupy funkcji *gmp*. To drugie rozwiązanie nie jest dostępne pod Windows.
- Przykłady definicji zmiennych zmiennoprzecinkowych:

$\$x = 1.245$

$\$y = -1.234$

$\$z = 1.23e6$ // e lub E oznacza podstawę potęgi równą 10

$\$p = 4E-10$

Składnia PHP: łańcuchy znaków

- Typ łańcuchowy określany jako *string* jest łańcuchem znaków.
- W PHP znak jest kodowany w jednym bajcie, zatem możliwych jest 256 różnych znaków w danym zestawie znaków. Dlatego PHP nie posiada żadnej wbudowanej obsługi Unikodu, ale istnieją rozszerzenia (funkcje) umożliwiające konwersję łańcuchów PHP do łańcuchów w Unikodzie.
- Jednak nie ma ustalonej maksymalnej długości łańcucha znaków, więc nie musimy obawiać się, że skonstruujemy za długi łańcuch. Trzeba jednak mieć świadomość, że w przypadku łańcucha o długości przeraczającej kres górny typu całkowitego istotną staje się precyzja zmiennoprzecinkowej reprezentacji liczby całkowitej, gdy chcemy mieć dostęp do znaków o odległych pozycjach w łańcuchu.

Składnia PHP: różne reprezentacje łańcuchów znaków

- Zmienne i wartości typu łańcuchowego można określać na trzy sposoby:
 - poprzez pojedyncze cudzysłowy:
‘Tekst w pojedynczym cudzysłowie’
 - poprzez podwójne cudzysłowy:
”Tekst w podwójnym cudzysłowie”
 - poprzez składnię dokumentów *HERE*
- Do każdego znaku zmiennej łańcuchowej, niezależnie od jej reprezentacji, można dostać się poprzez jego pozycję w łańcuchu, indeksowaną przy pomocy całkowity liczb nieujemnych, korzystając z operatora `{}`

```
$tekst = 'To jest łańcuch znaków.'
```

```
$tekst{0} = 'T'
```

```
$tekst{1} = 'o'
```

```
$tekst{6} = 't'
```

Składnia PHP: różne reprezentacje łańcuchów znaków

- W pojedynczym cudzysłowie zmienne nie są zamieniane na swoje wartości, a dodatkowe kody wprowadzane przy pomocy symbolu ukośnika odwrotnego \ nie są wykonywane, natomiast chcąc wprowadzić apostrof musimy poprzedzić go *backslashem*:

```
$tekst = 'Student powiedział: I\'ll learn it soon.'
```

- Zastosowanie podwójnego cudzysłowu daje dodatkowe możliwości:

- wykorzystywanie w łańcuchach wartości przypisanych zmiennym

```
$t = 'John'
```

```
$tekst = "This is $t's house." /* działa, bo apostrof nie jest  
poprawnym znakiem nazwy */
```

```
$tekst = "$tsons have their house." /* nie działa, bo $tsons  
jest poprawną nazwą zmiennej */
```

```
$tekst = "${t}sons have their house." // to działa poprawnie
```

Składnia PHP: różne reprezentacje łańcuchów znaków

- wykorzystywanie w łańcuchach określonych przez podwójny cudzysłów dodatkowych kodów wprowadzanych przy pomocy symbolu ukośnika wstecznego \

Kod	Opis
\"	Podwójny cudzysłów w tekście
\\	Odwrotny ukośnik (<i>backslash</i>) w tekście
\\$	Znak dolara w tekście
\n	Nowy wiersz (LF, kod ASCII: 0x0A (10))
\r	Powrót karetki (CR, kod ASCII: 0x0D (13))
\t	Tabulacja pozioma (HT, kod ASCII: 0x09 (9))
\nmk	Znak o kodzie <i>nmk</i> w notacji ósemkowej
\xnm	Znak o kodzie <i>nm</i> w notacji szesnastkowej

Składnia PHP: składnia dokumentów *HERE*

- Do długich łańcuchów znaków wygodnie jest stosować zaczerpniętą z Perla możliwość tworzenia dokumentów *HERE* (tzw. dokumentów w miejscu). Wykorzystujemy do tego celu operator `<<<` w przedstawiony poniżej sposób.

```
<<<Identyfikator_dokumentu_HERE
```

```
wielowierszowy tekst
```

```
Identyfikator_dokumentu_HERE;
```

- Identyfikator zamykający *musi* zaczynać się w pierwszej kolumnie nowej linii, a linia zawierająca ten identyfikator *nie może* zawierać żadnych innych znaków z *wyjątkiem* nazwy identyfikatora i następującego po niej średnika.
- Składnia dokumentów *HERE* jest bardzo wygodna: nie wymaga cytowania cudzysłowów, wskazywania dzielenia wierszy oraz zezwala na wykorzystanie wartości zapisanych w zmiennych i dodatkowych kodów wprowadzanych przy pomocy *backslasha*.
- Niestety zakaz stosowania na końcu wiersza znaku powrotu karetki `\r` i zezwolenie jedynie na znak `\n` skutkuje błędami, jeżeli skorzystaliśmy z edytora windowsowego, który opatruje końce linii znakami `\r\n`.

Składnia PHP: przykład skryptu z dokumentem HERE

- Przykładowy kod skryptu z dokumentem HERE

```
<?php
```

```
  $x = 5;
```

```
  $str = <<<DOC
```

"Przykład łańcucha znaków", 'zajmującego kilka linii', zapisanego w składni heredoc, w którym można używać cudzysłowy niepoprzedzone backslashem.

Można tu też wstawiać kody specjalne np. \x40 i zmienne, np. x = \$x.

```
DOC;
```

```
print("$str");
```

```
?>
```

- Rezultat wykonania skryptu (przy założeniu w pełni poprawnej zawartości kodu i związanej z tym jego poprawnej interpretacji)

"Przykład łańcucha znaków", 'zajmującego kilka linii', zapisanego w składni heredoc, w którym można używać cudzysłowy niepoprzedzone backslashem.

Można tu też wstawiać kody specjalne np. @ i zmienne, np. x = 5.

Składnia PHP: wartości logiczne

- Typ logiczny określany jako *boolean* posiada dwie predefiniowane wartości, stałe TRUE oraz FALSE.
- Zmienne logiczne określamy w następujący sposób:

```
$test = True;  
$sprawdz = false;
```
- Zmienne innych typów mogą być konwertowane do typu logicznego w sposób jawny poprzez operatory konwersji (bool) lub (boolean) albo automatycznie, gdy operator, funkcja lub struktura kontrolna wymagają argumentu typu *boolean* zgodnie z poniższą tablicą konwersji:

Typ danych	Wartość	Wartość logiczna
całkowity lub zmiennoprzecinkowy	0	FALSE
	każda inna wartość	TRUE
łańcuchowy	łańcuch pusty ""	FALSE
	"0"	FALSE
	każdy inny łańcuch	TRUE
tablicowy	tablica bez elementów	FALSE
	tablic z przynajmniej jednym elementem	TRUE
obiektowy	obiekt, którego egzemplarz utworzono	TRUE
wartość pusta (NULL)	NULL	FALSE

Składnia PHP: typy specjalne (identyfikator zasobów)

- Identyfikator zasobów określany jako *resource* jest odwołaniem (uchwytem) do zewnętrznych struktur danych, takich jak otwarte pliki lub połączenia z bazami danych. Przy próbie użycia zasobu jako łańcucha zwrócona zostanie sensowna wartość, np. **Resource id #1**.
- Identyfikatory zasobów są tworzone i wykorzystywane przez specjalne funkcje. Takie zasoby mają swoje określone nazwy (np. *ftp* – strumień ftp, *mysql link* – łącze do bazy danych MySQL, *oci8 connection* – łącze do bazy danych Oracle), *odbc link* – łącze do bazy danych ODBC)
- Od PHP 4 silnik Zend automatycznie wykrywa, czy nie ma już odnośników do zasobu (jak w Javie) i korzystając z destruktora zasobów zwalnia zasoby systemowe używane przez ten zasób. Wyjątkiem są połączenia stałe z bazami danych, które nie są niszczone przez destruktora zasobów.

Składnia PHP: typy specjalne (identyfikator zasobów)

- Stałe połączenia (persistent connections) to połączenia, które nie są zamykane automatycznie po wykonaniu skryptu. PHP sprawdza, czy istnieje już odpowiednie, wolne połączenie, a jeśli nie to je tworzy. Zyskujemy wówczas na wydajności procesu łączenia z bazą, ale łatwo przekroczyć dozwoloną liczbę połączeń na serwerze bazy danych. Przykładami takich połączeń są: *mysql link persistent* – stałe łącze do bazy danych MySQL, *odbc link persistent* – stałe łącze do bazy danych ODBC.
- Do obsługi identyfikatorów zasobów pomocne są dwie funkcje: *boolean is_source(dowolny_typ zmienna)*, która określa, czy *zmienna* jest zasobem; *string get_resource_type(resource handle)*, która podaje nazwę zasobu o uchwycie *handle*, a jeśli *handle* nie jest uchwycem zasobu to generuje błąd.
- Można samodzielnie zaprogramować obsługę zasobu jako ogólnego wskaźnika z język C odnoszącego się do dowolnych danych, ale to wykracza poza ramy wstępu do PHP.

Składnia PHP: typy specjalne (wartość NULL)

- Wartość pusta (NULL) oznacza brak wartości.
- Służy do inicjalizacji i *czyszczenia* zmiennych
- Przypisanie wartości NULL zmiennej powoduje, że w większości zwalniana jest pamięć przez nią zajmowana i tylko niewielka ilość pamięci jest użyta, by zachować zmienną.
- Należy jednak pamiętać, że PHP nie potrafi sam ostrzec o użyciu niezdefiniowanej zmiennej. Do tego celu służy funkcja `boolean is_null(dowolny_typ zmienna)`, która określa, czy *zmienna* ma wartość NULL

Składnia PHP: deklarowanie stałych

- Stałe w języku PHP są deklarowane funkcją *define()*. Po zadeklarowaniu wartości stałych nie mogą ulegać zmianie.
- Nazwy stałych nie są poprzedzone symbolem dolara, a poza tym podlegają tym samym zasadom, co nazwy zmiennych. Przyjęto, używać dużych liter w ich nazwach i zaleca się takie postępowanie w celu wyróżnienia stałych w kodzie skryptu
- Funkcja *define()* posiada następujący prototyp:

```
boolean define(string nazwa_stalej, wartość_stalej,  
                boolean nie_uwzględniaj_wielkości_liter)
```

nazwa_stalej jest odpowiednim łańcuchem znakowym,
wartość_stalej jest typu prostego (całkowitego, rzeczywistego,
łańcuchowego lub logicznego),
trzeci argument jest opcjonalny, domyślnie nazwy stałych
uwzględniają wielkość
liter

Składnia PHP: deklarowanie stałych

- Przykład deklaracji stałej PI o wartości 3.14:
`define("PI", 3.14159265358979);`
- Wówczas otrzymamy następujące rezultaty:
`echo PI; // na wyjściu otrzymamy 3.14159265358979`
`echo Pi; // na wyjściu otrzymamy Pi`
- Gdy chcemy sprawdzić, czy dana stała istnieje, to możemy użyć funkcji *defined()* o prototypie:

`boolean defined(string nazwa_stałej),`

którą możemy wykorzystać np. w następujący sposób:

```
if (defined("PI"))
{
    $r = 4.5;
    $obw = 2.0*PI*$r;
    print ("Obwód koła o promieniu $r cm wynosi $obw cm.");
}
```