

FAQ grupy pl.comp.lang.php

pl.comp.lang.php FAQ group
Ostatnie zmiany: 2004-06-03 17:47:38
Data generacji: 4 czerwca 2004

4 czerwca 2004

Spis treści

1	Informacje o FAQ	4
1.1	Co to jest FAQ? Czemu ma służyć?	4
1.2	Gdzie można znaleźć najnowszą wersję tego FAQ?	4
1.3	Kto tworzy FAQ?	4
1.4	Czy FAQ jest rozwijane?	5
1.5	Czy mogę umieścić FAQ na swojej stronie?	5
2	Grupa dyskusyjna pl.comp.lang.php	6
2.1	Czego dotyczy grupa?	6
2.2	Jakie zasady obowiązują na pl.comp.lang.php?	6
2.3	Gdzie można znaleźć archiwum grupy?	6
2.4	Jakie napisać zapytanie SQL, żeby...	7
2.5	Raz po raz widzę w odpowiedzi skróty RTFM lub NTG. Co one oznaczają?	7
2.6	Dlaczego nikt nie odpowiada na moje pytanie?	7
2.7	Grupa pl.comp.lang.php nie spełnia moich oczekiwań. Gdzie jeszcze można podyskutować o PHP?	7
3	Podstawy PHP	9
3.1	Co to jest PHP? Co oznacza skrót PHP?	9
3.2	Do czego potrzebne jest PHP?	9
3.3	Jakie są największe różnice pomiędzy PHP3 a PHP4?	9
3.4	Skąd wziąć PHP?	10
3.5	Jak skompilować PHP?	10
3.6	Jakie opcje można podać przy kompilacji PHP?	10
3.7	Czy PHP ma plik konfiguracyjny? Co można w nim ustawić?	10
3.8	Z jakim serwerem WWW może współpracować PHP?	11
3.9	Z jakimi bazami danych współpracuje PHP?	11
3.10	Skąd można czerpać informacje o PHP w Internecie?	11
3.11	Czy są jakieś książki o PHP?	11
3.12	Gdzie można znaleźć skrypt, który...	12
3.13	Jakie serwisy Internetowe możecie polecić osobom znającym PHP?	12
3.14	W jakim edytorze pisać skrypty PHP?	12
3.15	Jak napisać skrypt, który...	13
3.16	Korzystam ze skryptu X, który nie chce działać. Co zrobić?	14
3.17	Czy znacie bezpłatne serwery, które obsługują PHP?	14
3.18	Gdzie mogę się pochwalić moim skryptem PHP?	15
3.19	Czy z poziomu PHP można tworzyć grafikę?	15

3.20	Czy w PHP można tworzyć tylko strony internetowe?	15
3.21	Czy z poziomu JavaScript można wywołać funkcję PHP?	15
3.22	Co to są wyrażenia regularne i do czego służą?	15
3.23	Czym różni się include() i require()?	16
3.24	Jakie nowości zostały wprowadzone w PHP 4.1.0?	17
3.25	Gdzie znajduje się plik konfiguracyjny PHP?	17
3.26	Jaka jest różnica w zastosowaniu isset() i !empty() ?	18
4	Porady praktyczne	19
4.1	Co oznacza błąd...	19
4.1.1	Cannot add header information - headers already sent by...	19
4.1.2	open(/tmp\sess_[...], O_RDWR) failed: m(2)	19
4.1.3	Unexpected character in input...	19
4.1.4	Cannot redeclare [class] foo()...	19
4.1.5	Call to undefined function:	20
4.1.6	Failed opening 'foo.php' for inclusion (include_path='.')	20
4.1.7	Maximum execution time of 30 seconds exceeded in...	20
4.1.8	First argument to array_*() should/needs to be an array	20
4.1.9	Supplied argument is not a valid MySQL result resource	20
4.2	PHP nie zgłosił żadnego błędu, ale skrypt nie działa poprawnie bądź PHP zawiesza się przy jego wykonywaniu. Co zrobić?	20
4.3	Jak korzystając z PHP ograniczyć dostęp do strony, bądź pozbawić kogoś całkowicie dostępu do witryny?	21
4.4	Mam kilka skryptów do wyboru. Który polecacie?	21
4.5	Jak przekierować przeglądarkę na inną stronę?	21
4.6	Jak zasugerować przeglądarce nazwę pliku do zapisania?	22
4.7	Jak skonwertować polskie znaki pomiędzy stronami kodowymi WIN-1250 i ISO-8859-2?	22
4.8	Jak wysłać SMS z poziomu PHP?	22
4.9	Czy można ukryć kod PHP?	23
4.10	Jak uzyskać dostęp do danych przesłanych formularzem?	23
4.11	Jak uruchomić skrypt w regularnych odstępach czasu?	24
4.12	Słyszałem o sesjach. Co to jest? Jak ich używać? Gdzie znaleźć więcej informacji na ten temat?	25
4.13	Jak wysłać list z załącznikiem?	25
4.14	Jak sprawdzić poprawność adresu e-mail?	25
4.15	Czy w PHP można sprawdzić poprawność numeru...?	26
4.15.1	NIP	26
4.15.2	REGON	27
4.15.3	PESEL	28
4.16	Jak zarejestrować funkcję callback, która jest metodą w klasie?	28
4.17	Co to jest referencja? Do czego jej użyć?	28
4.18	Jak korzystać z plików konfiguracyjnych w stylu php.ini?	29
4.19	Jak zoptymalizować kod napisany w PHP?	29
4.20	Jak odczytać w PHP rozdzielczość ekranu klienta?	31
4.21	Gdzie można znaleźć bazę danych z datami imienin?	32
4.22	W jaki sposób umieścić dane wewnątrz istniejącego już pliku tekstowego?	32
4.23	Jak zapobiec powtórnemu submitowi formularza?	33
4.24	Czemu nie działa mi autoryzacja HTTP?	33

4.25	Czemu nie działają mi zmienne globalne po upgradzie?	33
4.26	Jak ukryć parametry w adresie strony?	34
4.27	Jak sprawdzić, które komputery w sieci są włączone?	34
4.28	Czemu nie działa mi funkcja mail na localhoscie?	34
4.29	Jak przyspieszyć pracę samego PHP?	34
4.30	Jak skonwertować pseudo-unicode odczytany z programu Excel do formatu iso-8859-2?	35
5	Bezpieczeństwo	36
5.1	Jak bezpiecznie skonfigurować PHP?	36
5.2	Jak zabezpieczyć pliki PHP przed dostępem osób niepowołanych? . .	36
5.3	Jak zabezpieczyć biblioteki przed odczytem przez WWW?	36
5.4	Dlaczego należy używać tablic \$HTTP_* zamiast zmiennych globalnych?	36
6	PEAR (PHP Extension and Application Repository)	38
6.1	Co to jest PEAR?	38
6.2	Jak zainstalować PEAR?	38
6.3	Gdzie można znaleźć najnowszą wersję PEAR?	39
6.4	Gdzie można znaleźć dokumentację do PEAR?	39
6.5	Dlaczego używać PEAR?	39
6.6	Jak sprawdzić poprawność danych?	39
6.7	Jak porcjować wyniki zapytania?	39
6.8	Jak wysłać maila z załącznikiem lub w formacie HTML?	39
6.9	Jak rozwiązany jest dostęp do baz danych w PEAR?	40
6.10	Jak PEAR obsługuje upload plików?	40
6.11	Szablony w PEAR	40
6.11.1	Co to są szablony? Po co ich używać?	40
6.11.2	Jak używać szablonów?	40
6.12	Jak udawać przeglądarkę?	41
6.13	Nie mam / Nie chcę używać sendmaila do wysyłania poczty. Co zrobić? . .	41
6.14	Napisałem dwa różne skrypty/funkcje/algorytmy wykonujące te same zadania. Jak sprawdzić, co jest szybsze?	42
6.15	Jak wczytać do tablicy nazwy plików zawartych w danym katalogu lub znaleźć określony plik?	42

Rozdział 1

Informacje o FAQ

1.1 Co to jest FAQ? Czemu ma służyć?

FAQ (angielski skrót od Frequently Asked Questions) to zbiór najczęściej zadawanych pytań wraz z odpowiedziami na nie. Pierwsze FAQ powstały najprawdopodobniej w roku 1980 w Stanach Zjednoczonych.

Obecnie czytasz FAQ polskiej grupy usenetowej `pl.comp.lang.php`. Znajdziesz w nim, zgodnie z tematem grupy, odpowiedzi na najczęściej padające pytania z zakresu języka PHP. Odpowiedzi są najczęściej krótkimi wskazówkami umożliwiającymi samemu rozwiązanie danego problemu. Gdy jest to konieczne zamieszczane są adresy źródeł, w których czytelnik może znaleźć dokładniejsze informacje.

FAQ nie zastąpi zapoznania się z językiem - zalecamy lekturę dokumentacji (lub książek) o PHP. FAQ jest tylko uzupełnieniem informacji, których tam nie znajdziesz.

1.2 Gdzie można znaleźć najnowszą wersję tego FAQ?

Najnowsza wersja FAQ grupy `pl.comp.lang.php` jest dostępna pod adresem <http://php.faq.pl/>. Możesz również pobrać FAQ w formacie spakowanego dokumentu hipertekstowego (HTML), Portable Document Format (PDF), XML oraz tekstowego.

Raz w miesiącu specjalny program wysyła najnowszą wersję FAQ na grupę, aby każdy mógł się z nim zapoznać, w tym także nowe osoby, które dopiero zaczynają swoją przygodę z PHP.

1.3 Kto tworzy FAQ?

Najprościej byłoby odpowiedzieć krótko - życie grupy. Tak, tak, to właśnie Wy zadając pytania na `pl.comp.lang.php` tworzycie to FAQ. Istnieje natomiast pewna grupa mniej lub bardziej znanych ludzi, którzy z całych sił starają się, aby to FAQ wyglądało jak najlepiej. Wymieniają oni uwagi dotyczące struktury i zawartości FAQ na specjalnie do tego celu stworzonej liście dyskusyjnej (nie chcemy zaśmiecać grupy niepotrzebnymi postami). Oto oni (kolejność alfabetycznie nazwiskami):

Adam Gołębiowski
Wojciech Jukowski

Leszek Krupiński (koordynator projektu)
Adam Major
Metempsychoza
Lech Rychliński
Paweł Szczepański

1.4 Czy FAQ jest rozwijane?

Cały czas zbieramy nowe pytania i opracowujemy odpowiedzi na nie. Jeśli masz propozycje pytań i odpowiedzi, bądź sugestie odnośnie już zamieszczonych, zachęcamy do ślania listów na adres php@faq.pl.

1.5 Czy mogę umieścić FAQ na swojej stronie?

Tak, pod warunkiem, że zamieszczona zostanie informacja o pochodzeniu FAQ i miejscu, gdzie się oryginalnie znajduje. FAQ jest udostępniane w wielu formatach, m. in. XML, więc można wpasować je w design swojej strony. Należy się jednak postarać, aby FAQ było udostępniane w możliwie najświeższej wersji.

Rozdział 2

Grupa dyskusyjna pl.comp.lang.php

2.1 Czego dotyczy grupa?

Grupa jest przeznaczona na dyskusje o programowaniu w języku PHP - instalacja/konfiguracja interpretera, niuanse języka, współpraca z bazami danych, pomocne narzędzia i dodatki, efektywne metody programowania, itd. (<http://www.usenet.pl/opisy/pl.comp.lang.php>) Jest to bardzo dobre miejsce do dyskusji na temat "wyższości" jednych rozwiązań (algoritmów) nad innymi.

Pytanie wykraczające poza PHP (np. konfiguracja serwera WWW, wybór języka czy też dowodzenie wyższości jednego języka ponad PHP) powinno trafiać na grupę pl.comp.www.server-side. Problemy poruszające tzw. client-side powinny być wysłane na grupę pl.comp.www.

Listy z pytaniami odnośnie baz danych winne być wysłane na pl.comp.bazy-danych.

Opisy innych grup (nie tylko komputerowych) znajdziesz pod adresem <http://www.usenet.pl/opisy/>.

2.2 Jakie zasady obowiązują na pl.comp.lang.php?

Przypominamy, iż zgodnie z opisem grupy "Zabronione są pytania, na które odpowiedzi można znaleźć w dokumentacji. Zabronione jest także wysyłanie listów zawierających jakiegokolwiek binaria (programy, zdjęcia, itp.), jakiegokolwiek spam oraz dyskusje nie związane z tematem grupy. Nie jest to także miejsce na ogłoszenia reklamowe, nawet jeśli są związane z tematem grupy." Dodatkowo dyskutantów obowiązują takie zasady jak na każdej innej grupie dyskusyjnej - zachęcamy do zapoznania się z Netykietą (<http://www.netykieta.prv.pl/>). Jedynym odstępstwem jest to, że dużo częściej stosuje się zwrot RTFM (patrz <http://php.faq.pl/klucz/RTFM>), ponieważ większość problemów rozwiązuje przeczytanie manuala, od którego wskazane jest zacząć korzystanie z PHP (adres dokumentacji można znaleźć w rozdziale 3.10 - http://php.faq.pl/klucz/php_dok).

2.3 Gdzie można znaleźć archiwum grupy?

Grupa powstała 16 sierpnia 2001. Serwery newsowe przechowują wiadomości z ostatniego miesiąca. Archiwalne posty z grupy pl.comp.lang.php można przeglądać korzy-

stając z serwisów <http://groups.google.com/groups?q=pl.comp.lang.php>, <http://niusy.onet.pl/niusy.html?t=aktualne&gr>
lub <http://www2.gazeta.pl/usenet/704354,27013.html?group=pl.comp.lang.php>

2.4 Jakie napisać zapytanie SQL, żeby...

Takie pytania należy zadawać na grupie dyskusyjnej pl.comp.bazy-danych. Można także sprawdzić w FAQ tej grupy (<http://www.dbf.pl/faq/>) czy nie padła już odpowiedź na Twoje pytanie.

2.5 Raz po raz widzę w odpowiedzi skrót RTFM lub NTG. Co one oznaczają?

RTFM - Read The Friendly (ew. Fucking) Manual. Jeśli dostałeś taką odpowiedź, to znaczy że odpowiedź na Twoje pytanie można znaleźć czytając podręcznik PHP (gdzie go znaleźć -> <http://php.faq.pl/punkt/3.10>).

NTG - Not This Group, Nie Ta Grupa. Twoje pytanie nie pasuje do tematyki grupy pl.comp.lang.php - zazwyczaj razem z taką odpowiedzią podawana jest nazwa grupy, na której należy zadać to pytanie.

Znaczenie innych stosowanych częściej lub rzadziej akronimów znajdziesz na <http://www.republika.pl/krzywish/skroty>

2.6 Dlaczego nikt nie odpowiada na moje pytanie?

Może być kilka powodów, dlaczego nie dostałeś(aś) odpowiedzi:

- minęło zbyt mało czasu, aby ktoś zdążył przeczytać Twój list i wysłać odpowiedź.
- twój list był nie na temat
- twoje pytanie było tak "banalne", że nikomu nie chciało się na nie odpowiadać
- pytanie było zbyt trudne i nikt na nie nie zna odpowiedzi.

2.7 Grupa pl.comp.lang.php nie spełnia moich oczekiwań. Gdzie jeszcze można podyskutować o PHP?

Z polskojęzycznych grup poświęconych PHP i okolicom istnieją:

pl.comp.www.server-side - grupa poświęcona wszelkim sprawom związanym z otoczeniem serwerowym służącym do "serwowania" stron WWW a więc językom server-side (jak PHP, ASP, Perl, Python, JSP i inne), serwerom HTTP (Apache, IIS itp.).

pl.comp.bazy-danych - grupa poświęcona bazom-danych. Jeśli programując w PHP korzystasz z baz danych warto tam zajrzeć aby dowiedzieć się o sprawach niezwiązanych z samym PHP a raczej często spotykanym "jak skonstruować zapytanie do bazy danych aby..."

Z obcojęzycznych:

Tutaj jest już większy wybór. Praktycznie każdy kraj ma własną grupę związaną z PHP (*.php) więc trzeba poszukać.

Odnosnie forum dyskusyjnego, to polecamy polskie <http://www.ygreg.com/forum/> oraz angielskojęzyczne <http://www.phpbuilder.com/forum/>

Jeśli preferujesz odpowiedzi na żywo możesz odwiedzić kanały IRC (Internet Relay Chat). Polskojęzyczne #phppl oraz #php.pl oraz angielskojęzyczny #php w sieci IRCNET, bądź też #php, #php-gtk w sieci EFNET.

Rozdział 3

Podstawy PHP

3.1 Co to jest PHP? Co oznacza skrót PHP?

PHP (skrót od "PHP: Hypertext Preprocessor") jest wtrąconym w HTML językiem skryptowym wykonywanym po stronie serwera. Znaczy to mniej więcej tyle, że kod PHP jest wstawiany w normalne strony WWW, natomiast specjalny program wykonuje instrukcje zawarte pomiędzy `<?php a ?>` (względnie pomiędzy `<? a ?>`). Pierwsza wersja powstała w roku 1994 (jej autorem był Rasmus Lerdorf). Od wprowadzenia wersji 2 do Rasmusa przyłączały się nowe osoby tworząc PHP Group of core developers. Wersja PHP 3 w krótkim czasie stała się bardzo popularna (zastosowania PHP 3 zaskoczyły samych autorów). Nad rozwojem wersji 4 pracował sztab ludzi, efekty ich pracy możesz na bieżąco podziwiać.

3.2 Do czego potrzebne jest PHP?

PHP jest przydatne do tworzenia dynamicznie generowanych stron WWW, gdzie zachodzi potrzeba wyświetlania różnych danych zależnie od wyboru użytkownika, bądź zawartości bazy danych. Pozwala również wykonywać wiele czynności które na pozór mają mało wspólnego ze stronami internetowymi (np. działanie na plikach, symulacja działania przeglądarki, pobieranie oraz wysyłanie informacji do bazy danych czy nawet tworzenie nowych obrazków czy dokumentów PDF oraz wiele, wiele więcej).

3.3 Jakie są największe różnice pomiędzy PHP3 a PHP4?

Tworząc PHP 4 autorom przeświecało kilka głównych celów:

- nowa implementacja musi być szybka
 - musi być modułowa
 - musi zostać zachowana kompatybilność z PHP 3
 - musi być łatwo rozszerzalne.
- nowemu silnikowi nadano nową nazwę - Zend Engine

Główne różnice to:

- rozszerzony moduł API
- ulepszona współpraca z serwerami WWW

- poprawiony podświetlacz składni
- wbudowana obsługa sesji HTTP
- buforowanie wyjścia
- zwiększone możliwości konfiguracji
- zliczanie referencji

Więcej informacji znajdziesz na stronie <http://www.php.net/> oraz <http://www.zend.com/>

3.4 Skąd wziąć PHP?

PHP można pobrać ze strony głównej projektu: <http://pl.php.net/downloads.php>, a także z wielu mirrorów. Użytkownikom systemu Windows przypominamy, iż zostały przygotowane dwie skompilowane wersje binarne dla owej rodziny systemów operacyjnych. Pragniemy również zwrócić uwagę, iż wersje różnią się m.in. ilością dołączonych bibliotek.

Można także zajrzeć na stronę <http://snaps.php.net/> gdzie znajdują się stabilne oraz dopiero rozwijane migawki pakietu tworzone na podstawie zasobów CVS co kilka godzin.

3.5 Jak skompilować PHP?

Kompilacja PHP wygląda tak samo jak w przypadku każdego programu używającego pakietu automake. A więc w przypadku Linuksa będąc w katalogu zawierającym źródła PHP należy kolejno wydać polecenia:

```
./configure (+ew. opcje, opis tych opcji można znaleźć tu: http://php.faq.pl/klucz/configure)
make
make install
```

Ostatnie polecenie musi być wydane z konta administratora.

Po udanej kompilacji i instalacji należy zrestartować serwer WWW.

Do kompilacji, oprócz kompilatora języka C, potrzebne są jeszcze programy Make i Lex (oba dostępne są w każdej dystrybucji Linuksa).

3.6 Jakie opcje można podać przy kompilacji PHP?

Pełną listę opcji, jakie można podać przy kompilacji PHP, można znaleźć w dokumentacji PHP, w tym także w polskim tłumaczeniu.

<http://pl.php.net/manual/pl/configuration.php#ini.sect.general>

Informacja gdzie można znaleźć dokumentację PHP znajduje się w punkcie 3.10.

3.7 Czy PHP ma plik konfiguracyjny? Co można w nim ustawić?

PHP posiada plik konfiguracyjny. Można w nim ustawić najważniejsze opcje dotyczące pracy PHP, bezpieczeństwa, sesji, baz danych i ogólnej wydajności PHP. Dokładny opis można znaleźć w podręczniku PHP pod adresem <http://pl.php.net/manual/pl/configuration.php> Zobacz także pytanie 3.25.

3.8 Z jakim serwerem WWW może współpracować PHP?

Współpracuje z serwerami: Apache, Caudium, IIS/PWS, iPlanet, Netscape, OmniHT-TPd, Oreilly Website Pro, Xitami.

Oraz z większością serwerów obsługujących CGI, niestety wtedy PHP jest dostępne tylko i wyłącznie w wersji CGI.

3.9 Z jakimi bazami danych współpracuje PHP?

W chwili obecnej PHP współpracuje z następującymi bazami danych: Adabas D, dBase, Empress, FilePro (read-only), IBM DB2, Informix, Ingres, InterBase, Front-Base, mSQL, MS-SQL, MySQL, ODBC, Oracle (OCI7 i OCI8), PostgreSQL, Solid, Sybase, Velocis, Unix dbm.

3.10 Skąd można czerpać informacje o PHP w Internecie?

Podstawowym źródłem informacji o PHP jest manual - podręcznik napisany przez twórców PHP bardziej lub mniej szczegółowo opisujący poszczególne funkcje. Można go czytać on-line na stronie <http://pl.php.net/manual/pl/>, a także pobrać na dysk w różnych formatach ze strony <http://pl.php.net/download-docs.php>.

Istnieje także wiele kursów on-line dotyczących podstaw korzystania z PHP. Można je znaleźć między innymi pod adresami:

- <http://ygreg.com/>

- <http://www.leon.w-wa.pl/texts/kurs/>

3.11 Czy są jakieś książki o PHP?

Książek jest multum (na polskim i zagranicznym rynku) i raczej trudno polecić jakąś nie znając poziomu programowego czytelnika oraz jego zainteresowań. Warto rozemnieć się na paru stronach poświęconych tematowi i wyrobić sobie własne zdanie na temat przydatności niektórych pozycji:

Księgarnie:

<http://www.helion.pl/> (ok. 9 pozycji)

<http://www.lideria.pl/> (PHP - ok. 15 pozycji)

<http://www.techniczna.com.pl/> (PHP - ok. 20 pozycji)

Recenzje i opisy:

http://www.e-marketing.pl/lit_webmaster.php

Książki po angielsku:

<http://www.hotscripts.com/PHP/Books/> (ok. 28 pozycji)

Powyższe linki to na pewno dobry początek do dalszych poszukiwań ...

A jeśli na prawdę nie chce ci się odwiedzać tych stron to mówiąc w dużym skrócie i uproszczeniu:

dla początkujących polecana jest "Biblia PHP4" wydawnictwa Helion, a dla bardziej zaawansowanych "PHP: Tworzenie Aplikacji" ;-)

3.12 Gdzie można znaleźć skrypt, który...

Największe archiwum skryptów PHP (aplikacji, bibliotek) to http://www.hotscripts.com/PHP/Scripts_and_Programs/, a także <http://www.zend.com/codex.php>. Z kolei największą biblioteką klas PHP jest serwis <http://www.phpclasses.org/>. Warto też pamiętać o PEAR (<http://pear.php.net/>), repozytorium klas, które jest dołączane do dystrybucji PHP. Wiele skryptów można znaleźć także w centrach oprogramowania OpenSource: <http://freshmeat.net/>, <http://sourceforge.net/> i <http://www.opensourcirectory.org/>

3.13 Jakie serwisy Internetowe możecie polecić osobom znającym PHP?

Temat rzeka !!! Można zacząć od poniższych linków ale pamiętajcie, że lista ta jest mocno subiektywna:

<http://php.spinacz.pl/>

<http://www.ygreg.and.pl/>

<http://php.pd.pl/> - to samo co: <http://www.php.computerzone.pl/>

<http://php.jest-ok.pl/>

<http://www.scrypty.com/>

<http://www.php.pl/>

<http://pl.php.net/> - zobacz też od razu: <http://pl.php.net/links.php>

<http://www.abczone.com.pl/indexphp.php>

<http://www.phpbuilder.com/>

http://www.devshed.com/Server_Side/PHP/

<http://www.weberdev.com/>

<http://hotscripts.com/PHP/>

<http://www.phpwizard.net/>

<http://www.zend.com/>

<http://www.php-scripts.com/>

Ponadto polecamy:

- przeszukać archiwa grupy gdzie padały podobne pytania,
- czytać stopki co poniektórych grupowiczów w poszukiwaniu ciekawych adresów,
- zadać pytanie wyszukiwarce w stylu: +serwisy +PHP +skrypty -sex ;-))

3.14 W jakim edytorze pisać skrypty PHP?

Skrypty PHP można pisać w dowolnym edytorze, który ma możliwość zapisywania danych do czystego tekstu. Istnieją jednak edytory tekstu, które mają wbudowane

wspomaganie dla edytowania skryptów PHP, np. kolorowanie składni, możliwość definiowania makr czy autoindentację. Poniżej znajduje się lista edytorów, które takie wspomaganie posiadają.

Edytory niezależne od platformy:

- jEdit (<http://jedit.org/>)
- SciTE (Scintilla) (<http://www.scintilla.org/SciTE.html>)
- VIm (<http://vim.sourceforge.net/>) - pod adresem <http://www.leon.w-wa.pl/texts/vim-php.php> znajduje się artykuł na temat pisania kodu PHP przy użyciu VIMa
- phpmole (<http://phpmole-ide.sourceforge.net/>)

Edytory pod Windows:

- UltraEdit (<http://ultraedit.com/>)
- HomeSite (<http://www.macromedia.com/homesite/>)
- EditPlus (<http://editplus.com/>)
- EzHTML (<http://ezhtml.bydnet.com.pl/>)
- EDHTML (<http://edhtml.binboy.org/>)
- PHPEd (<http://www.soysal.com/PHPEd/>)
- Crimson Editor (<http://crimsoneditor.com/>)
- PHP Coder (<http://www.phpide.de/>) (program daleki jest od doskonałości - niepolecany)
- PHPEdit (<http://www.phpedit.net/>)
- Pajaczek (<http://www.creamsoft.com.pl/pajaczek/>) - produkt polski
- Zend Studio - Personal Edition (wersja 2.0.1 dostępna za darmo) - <http://www.zend.com/>

Edytory pod Linuksa (wszystkie do znalezienia przez <http://freshmeat.net/>):

- Quanta
- BlueFish
- Scream
- nedit (<http://nedit.org/>)
- mcedit - edytor wbudowany w Midnight Commandera, potrzeba jednak pobrać plik z podświetleniami - <http://php.faq.pl/php3.syntax>, przegrać go do katalogu `/usr/lib/mc/syntax/` a w pliku `./cedit/Syntax` dopisać:

```
> file ..\*\*.php(3|4|s)??$ PHP\sScript
include php3.syntax
```

Liste edytorów wspierających PHP można znaleźć również na stronie: <http://phpeditors.linuxbackup.co.uk/index.php>

3.15 Jak napisać skrypt, który...

Zasadniczo sprawa wygląda tak:

- siadasz przed kartką
- rozpisujesz problem
- analizujesz możliwości rozwiązania
- w manualu sprawdzasz, czy są tam funkcje które są ci niezbędne
- piszesz

Jeśli nie bardzo wiesz jak coś napisać, podpatrz jak to jest zrobione u konkurencji. W ostateczności spytaj na grupie dyskusyjnej.

3.16 Korzystam ze skryptu X, który nie chce działać. Co zrobić?

1. Sprawdź prawa dostępu do pliku
2. Sprawdź czy pliki mają odpowiednie rozszerzenia - przypisane do PHP
3. Sprawdź konfigurację skryptu (jeśli takowa istnieje)
4. Sprawdź czy któryś z komunikatów błędów nie jest opisany w tym FAQ

...

10000. W ostateczności skontaktuj się z autorem lub zapytaj się na grupie pl.comp.lang.php (tylko nie zadawaj pytania "Skrypt X nie działa. Co mam robić?" - sprecyzuj co to znaczy "nie działa": jakie pojawiają się komunikaty, jakie masz wersje serwera WWW, PHP, jaki system operacyjny oraz w jakich okolicznościach pojawia Ci się dany błąd)

3.17 Czy znacie bezpłatne serwery, które obsługują PHP?

<http://free.komrel.net/>

<http://www.xo.pl/>

<http://www.piwko.pl/>

<http://www.unas.cz/>

<http://www.host.sk/>

<http://www4.dk3.com/dk3hosting.html>

<http://www.spaceports.com/freesites/>

<http://www.united.net.kg/>

<http://www.fateback.com/>

<http://www.xoasis.com/>

<http://www.tripod.lycos.co.uk/>

<http://www.breezeland.com/>

<http://www.portland.co.uk/freesubdomainapp.esp>

Sponsoring/ ograniczona tematyka:

<http://webhosting.bootbox.net/freehosting> (wymagany zakup domeny)

<http://www.planet-school.de/webpace/index.htm> (tylko szkoły)

<http://www.gery.pl/>

<http://www.bajo.pl/>

<http://fpp.pl/>

<http://www.fivnet.com/>

Zestawienie zagranicznych serwerów oferujących hosting wraz z PHP bezpłatnie można znaleźć pod adresami:

<http://free-php.cjb.net/>

<http://www.free-php-hosting.com/>

<http://www.free-webhosts.com/power-search.php>

<http://www.freewebspace.net/forums/>
<http://www.25mb.net/signup.php>
<http://www.zingto.com/signup.php>
<http://hostsearch.com/>

3.18 Gdzie mogę się pochwalić moim skryptem PHP?

Można to zrobić na różnych serwisach gromadzących informacje o skryptach PHP:

- <http://hotscripts.com/>
- <http://freshmeat.net/>
- <http://www.zend.com/>

Istnieje także serwis, który gromadzi tylko klasy PHP:

- <http://phpclasses.upperdesign.com/>

3.19 Czy z poziomu PHP można tworzyć grafikę?

Tak. Służy do tego zestaw funkcji opisanych w manualu pod nazwą "Image functions" (<http://pl.php.net/manual/pl/ref.image.php>). PHP jest w stanie tworzyć proste obrazki (rysowanie okręgów, prostokątów, linii), modyfikować istniejące, dodawać tekst (używając wbudowanych czcionek lub fontów TTF), zmieniać rozmiar itp.

3.20 Czy w PHP można tworzyć tylko strony internetowe?

Nie, nie tylko. HTML jest domyślnym typem wysyłanym przez PHP, ale można to łatwo zmienić umieszczając linijkę:

```
> <?php
  header('Content-type: twój_typ');
?>
```

gdzie twój_typ to typ mime pliku który chcesz wysłać do użytkownika (np. image/gif czy text/xml). Dalej już osobiście musisz zadbać, aby wszystkie dane wysyłane do klienta były w formacie jaki został zdefiniowany na początku.

3.21 Czy z poziomu JavaScript można wywołać funkcję PHP?

Nie, nie można. PHP jest językiem wykonywanym po stronie serwera a JavaScript po stronie klienta. A więc jak zaczyna działać JavaScript to PHP już dawno skończył swoje działanie - strona została wysłana do klienta.

3.22 Co to są wyrażenia regularne i do czego służą?

Na ten temat napisano już książki-bible, pośród których najlepszą jest ta, o dość wyszukany tytuł "Wyrażenia regularne", autorstwa Jeffrey'a E. F. Friedl'a. Wyrażenia

regularne to ogromne narzędzie dostarczające możliwości zaawansowanych manipulacji oraz porównań na ciągach znaków. "Regular Expressions" jest implementowane w wielu innych językach (np. JavaScript). Wyrażenia regularne możemy stosować w takich funkcjach PHP jak: `ereg()`, `ereg_replace()`, `eregi()`, `eregi_replace()`, `split()`, `spliti()`.
<http://pl.php.net/manual/pl/ref.regex.php>

Dla ciekawskich można wspomnieć, że powyższe funkcje korzystają z rozszerzonej składni POSIX dla wyrażeń regularnych i nie są to jedyne funkcje PHP działające na wyrażeniach.

Osoby znające Perl'a mogą używać innych funkcji: `preg_*`(). Jest tam stosowana składnia wyrażeń zgodna właśnie z tym językiem, trochę odmienna od tej wykorzystywanej w `ereg()`.

<http://pl.php.net/manual/pl/ref.pcre.php>

Regular Expression Details:

<http://pl.php.net/manual/pl/pcre.pattern.syntax.php>

Przykład:

Proste sprawdzenie poprawności adresu e-mail:

```
> if (! ereg ("^.+@.+\\.+.$", $email))  
    echo 'Nieprawidłowy adres e-mail!';
```

A przekładając to na nasz język:

adres ma zaczynać się (^) od jednego lub więcej (+) dowolnych znaków (.), potem jest małpa (@), znowu jeden lub kilka znaków (.), kropka (\.) i zakończenie stringa (\$) dowolnym znakiem.

Na początek możesz zobaczyć poniższe tutoriale:

<http://www.leon.w-wa.pl/texts/kurs/text.php#regex>

<http://www.phpbuilder.com/columns/dario19990616.php3>

3.23 Czym różni się `include()` i `require()`?

Podstawową różnicą jest to, że przy `require()` plik jest zawsze wstawiany - takie copy/paste.

Przy `include()` plik jest wstawiany tylko jeśli ta instrukcja będzie parsowana.

Przykład: mamy plik `require.inc.php`:

```
> <?php  
    echo 'coś tam';  
    echo 'coś innego';  
?>
```

No i chcemy to wykorzystać

```
> <?php  
    if ($test) {  
        require('require.inc.php');  
    }
```

```
?>
```

zawsze zostanie zamienione na

```
> <?php
  if ($test) {
    echo 'coś tam';
    echo 'coś innego';
  }
?>
```

Natomiast jeśli zamiast `require()` użyjemy `include()`, to plik zostanie wstawiony tylko jeśli zostanie spełniony warunek `$test`. Dlatego też przy instrukcjach warunkowych ma sens tylko stosowanie `include()`.

Dodatkowo trzeba pamiętać, że `require()` jest wykonywane tylko raz. Nie ma więc sensu wstawianie `require()` do pętli, w czasie której zmienia się nazwa pliku do pobrania, ponieważ plik ten zostanie wstawiony tylko za pierwszym razem, a każdej następnej iteracji będzie używany kod wstawiony za pierwszym razem.

3.24 Jakie nowości zostały wprowadzone w PHP 4.1.0?

W PHP 4.1.0 dodanych zostało 7 nowych tablic asocjacyjnych (istnieją one wspólnie z już znanymi tablicami `$HTTP_*_VARS`):

- `$_GET` - zawiera wartości z formularzy przesłanych metodą GET
- `$_POST` - zawiera wartości z formularzy przesłanych metodą POST
- `$_COOKIE` - zawiera ciasteczka HTTP
- `$_SERVER` - zawiera zmienne serwera (np. `REMOTE_ADDR`)
- `$_ENV` - zawiera zmienne środowiskowe
- `$_REQUEST` - zawiera połączone tablice `$_GET`, `$_POST` i `$_COOKIE`
- `$_SESSION` - zawiera zmienne HTTP rozpoznane przez moduł sesji

Tablice te są dostępne z każdego miejsca kodu. Nie trzeba używać instrukcji global aby uzyskać do nich dostęp z wnętrza funkcji. Dotychczas stosowane w PHP tablice `$HTTP_*_VARS` niestety tego wymagały. Mamy oczywiście nadal możliwość wyboru z których zmiennych chcemy korzystać.

Dodatkowa możliwość związana jest z tablicą `$_SESSION`. Dodanie nowego elementu do tej tablicy spowoduje zarejestrowanie nowej zmiennej sesyjnej.

Uwaga: Od wersji 4.2.0 opcja `register_globals` domyślnie jest ustawiona na OFF co powoduje, że zmienne dostępne są tylko w w/w tablicach.

3.25 Gdzie znajduje się plik konfiguracyjny PHP?

W systemach Uniksowych plik ten znajduje się domyślnie w katalogu `/usr/local/lib` i nazywa się `php.ini`. W Windows jest to główny katalog systemu - standardowo

C:\WINDOWS albo C:\WINNT. Lokalizację tego pliku można zmienić przy kompilacji podając parametr `-with-config-file-path=/ścieżka/do/pliku`. Jeśli ścieżka do tego pliku została zmieniona przy kompilacji i trzeba ją znaleźć, to pokazana jest ona w tabelce wyświetlanej przez funkcję `phpinfo()`.

3.26 Jaka jest różnica w zastosowaniu `isset()` i `!empty()` ?

Przeprowadźmy prostą analizę:

Pewna zmienna `$var` może:

- a) nie istnieć,
- b) być pusta (także równać się zero !),
- c) mieć jakąś wartość,

`isset($var)` - daje TRUE gdy `$var` jest b) lub c), FALSE gdy a)

`empty($var)` - daje TRUE gdy `$var` jest a) lub b), FALSE gdy c)

a słownie:

`isset($var)` - daje FALSE jedynie gdy `$var` nie istnieje

`empty($var)` - daje FALSE jedynie gdy `$var` ma jakąś wartość (uważaj na zero!)

Reasumując:

Stosujemy `empty()` jeśli chcemy mieć pewność, że otrzymaliśmy jakąś wartość w `$var` (nie uwzględniamy niestety kłopotliwego przypadku `(int)$var=0` bez względu na to czy zmienna będzie istnieć czy nie.

`Empty()` nie generuje żadnego 'Warning' i sprawdza dwie rzeczy za jednym zamachem.

Rozdział 4

Porady praktyczne

4.1 Co oznacza błąd...

4.1.1 Cannot add header information - headers already sent by...

Oznacza to że przed nagłówkiem wysłano już jakieś dane (np pustą linię przed `<?php` lub coś za pomocą `echo`, `print`). Usuń przyczynę błędu lub w `php.ini` ustaw `output_buffering = On`. Możesz także skorzystać z funkcji `ob_start()` oraz `ob_end_flush()`. Szczegóły dotyczące używania tych funkcji znajdziesz (jak zwykle) w przyjaznym manualu.

4.1.2 `open(/tmp\sess_[...], O_RDWR) failed: m(2)`

Ustaw ścieżkę do katalogu w którym mają być przechowywane sesje. Możesz to zrobić w pliku `php.ini`, sekcja `[Session]`:

```
session.save_path = /dir
```

lub bezpośrednio w skrypcie (koniecznie przez inicjację sesji) za pomocą funkcji `session_save_path("/dir")`. Pamiętaj by katalog `dir` istniał a PHP miało prawo do zapisu w nim.

4.1.3 Unexpected character in input...

W wyrażeniu regularnym nie "neutralizowałeś" backslashem jednego ze znaków specjalnych:

```
(. \ + * ? [ ^ ] $ ( ) { } = ! < > | :)
```

Jeśli jesteś leniwy i nie chce ci się tego robić ręcznie (lub dane składające się na wyrażenie pochodzą z zewnątrz) przepuść wyrażenie przez `preg_quote()`.

4.1.4 Cannot redeclare [class] foo()...

Twoja funkcja lub klasa została już wcześniej zadeklarowana. Sprawdź czy nazwa funkcji nie jest identyczna jak któraś z wbudowanych w PHP (użyj `function_exists()`, `get_defined_functions()`).

Być może dwukrotnie dołączyłeś ten sam plik z zadeklarowaną funkcją; zrezygnuj ze stosowania `include()`, `require()` na rzecz `include_once()`, `require_once()`

4.1.5 Call to undefined function: ...

Taki błąd pojawia się jeśli użyto funkcję, która nie jest zdefiniowana. Jeśli funkcję, którą chcesz użyć, sam zdefiniowałeś, to sprawdź czy nie popełniłeś literówki przy wpisywaniu nazwy (w użyciu bądź w definicji), a jeśli znajduje się w innym pliku to czy plik ten jest dołączany. Jeśli natomiast chcesz użyć funkcję wbudowaną w PHP i jesteś pewny, że nazwa jest dobrze napisana, to najprawdopodobniej nie masz odpowiedniego modułu wkompiowanego w PHP lub dynamicznie dołączonego, np. dla funkcji dotyczących obróbki obrazów niezbędne jest podanie przy kompilacji opcji `-with-gd` (lub dołączenie odpowiedniego modułu).

4.1.6 Failed opening 'foo.php' for inclusion (include_path='')

Klasyczny błąd który mówi sam za siebie. Plik który próbujesz includować nie istnieje, nie masz do niego odpowiednich praw lub podana ścieżka dostępu jest nieprawidłowa.

4.1.7 Maximum execution time of 30 seconds exceeded in...

Najprawdopodobniej wpadłeś w nieskończoną pętlę lub twój skrypt wykonuje czasochłonną operację (np. analizuje logi 20 MB). Jeśli uważasz że trzydzieści sekund to dla Ciebie zbyt mało zmień to za pomocą funkcji `set_time_limit()` lub w `php.ini` dyrektywą `max_execution_time`

4.1.8 First argument to array_*() should/needs to be an array

Zapomniałeś o czymś ważnym, mianowicie o zadeklarowaniu tablicy. Zazwyczaj nie jest to konieczne, lecz przy funkcjach z rodziny `array_*` lepiej o tym pamiętać. Zwykle `$array = array()` przed wywołaniem funkcji i już po problemie.

4.1.9 Supplied argument is not a valid MySQL result resource

Błąd ten oznacza, że zapytanie wydane funkcją `mysql_query()` nie zwróciło prawidłowego wyniku. Może to oznaczać, że albo zapytanie było błędne (komunikat błędu można odczytać wywołując funkcję `mysql_error()`) albo nie zwróciło żadnych rekordów. W tym drugim przypadku, przed odwoływaniem się do wyników należy sprawdzić ilość rekordów zwróconych przez zapytanie korzystając z funkcji `mysql_num_rows()`.

4.2 PHP nie zgłosił żadnego błędu, ale skrypt nie działa poprawnie bądź PHP zawiesza się przy jego wykonywaniu. Co zrobić?

Aby mieć pewność, że PHP na pewno nie zgłasza żadnych komunikatów ustaw w swoich plikach `error_reporting(E_ALL)`. Jeśli PHP się zawiesza to może należy skontrolować poprawność instalacji (albo np. wersję PHP - może jest za stara ;-)) lub dostępność zasobów z jakich korzystasz w skrypcie (baza danych, pliki, sockets, ...)

Kilka możliwych problemów jest opisanych w pliku `install.txt` dołączonym do każdej dystrybucji PHP. W akapicie "Problems?" można znaleźć takie sugestie (w skrócie):
- wykonaj skrypt: `<?php phpinfo(); ?>` i zobacz co otrzymałeś,

- po wykonaniu swojego skryptu zajrzyj do źródła wygenerowanego dokumentu,
- odpal w katalogu PHP komendę: "php -i" i zobacz zwrócone komunikaty. Jeśli otrzymałeś standardowe phpinfo() to problem powinien leżeć po stronie serwera WWW,
- sprawdź prawa dostępu do własnych skryptów oraz takich plików jak php.exe, php4ts.dll, php.ini,

Możesz także zajrzeć na stronę <http://www.php.net/FAQ.php> gdzie znajdziesz odpowiedzi na najczęściej występujące kłopoty użytkowników.

Warto również zobaczyć punkt 4.19 (optymalizacja kodu) w tym FAQ.

4.3 Jak korzystając z PHP ograniczyć dostęp do strony, bądź pozbawić kogoś całkowicie dostępu do witryny?

Możemy skorzystać z autoryzacji HTTP. Jak to zrobić opisane jest w rozdziale 18 w manualu (<http://www.php.net/manual/pl/features.http-auth.php>). Ograniczeniem jest to, że PHP musi działać jako moduł Apache'a, nie jako CGI.

Inny sposób na pozbawienie kogoś dostępu do witryny to odpowiednia konfiguracja serwera WWW. Np. dla Apache to pliki .htaccess i .htpasswd:
<http://httpd.apache.org/docs/howto/auth.html#basicconfig>

4.4 Mam kilka skryptów do wyboru. Który poleć?

Najlepiej jest wybrać taki skrypt, który najbardziej odpowiada Twoim potrzebom. Warto też zwrócić uwagę na sposób instalacji (dołączenia) tego skryptu do swojej strony, popularność skryptu (im bardziej popularny, tym większa szansa na jego rozwój i wychodzenie bardziej dopracowanych wersji), oraz inne czynniki które są zmienne w zależności od naszych wymagań, możliwości serwera, typu strony na jakiej chcemy ten skrypt umieścić, możliwość ewentualnej rozbudowy/lepszego dopasowania do Twoich potrzeb (w tym także czytelności kodu) czy wersji językowej.

Ranking skryptów prowadzi wiele serwisów, np. <http://hotscripts.com/>

4.5 Jak przekierować przeglądarkę na inną stronę?

Można to uzyskać wysyłając odpowiedni nagłówek.

```
> <?php
  header('Location: inna_strona.php');
  exit;
?>
```

Aby funkcja zadziałała to nie może (jeśli `output_buffering=off`) przed jej wywołaniem zostać wysłany (np. za pomocą `echo`) jakikolwiek znak do przeglądarki.

exit; po wywołaniu funkcji header() jest bardzo istotny, ponieważ zapobiega dalszemu wykonywaniu się skryptu, podczas rozpoczęcia przekierowania (które trochę trwa).

4.6 Jak zasugerować przeglądarce nazwę pliku do zapisania?

Robi się to za pomocą nagłówka "Content-disposition". Pełne polecenie sugerujące nazwę pliku wygląda tak:

```
> <?php
  header("Content-Type: application/octet-stream");
  header("Content-Disposition: attachment; filename=nazwa_pliku");
?>
```

4.7 Jak skonwertować polskie znaki pomiędzy stronami kodowymi WIN-1250 i ISO-8859-2?

Można to zrobić za pomocą funkcji zamieniającej konkretny znak na inny. Konwersja WIN-1250 => ISO8859-2 wygląda tak:

```
> <?php
$string = strtr($string, "\xA5\x8C\x8F\xB9\x9C\x9F",
  "\xA1\xA6\xAC\xB1\xB6\xBC");
?>
```

A ISO8859-2 => WIN-1250 tak:

```
> <?php
$string = strtr($string, "\xA1\xA6\xAC\xB1\xB6\xBC",
  "\xA5\x8C\x8F\xB9\x9C\x9F");
?>
```

4.8 Jak wysłać SMS z poziomu PHP?

Adresy do bramek naszych operatorów to:

<http://www.text.plusgsm.pl/sms/sendsms.php>

<http://sms.idea.pl/sendsms.asp>

<http://boa.eragsm.pl/sms/sendsms.asp?sms=1>

Są to bezpośrednie namiary na skrypty wysyłające, nie na strony z formularzem.

Pamiętaj także, że zawsze istnieje możliwość, że powyższe bramki przestaną działać.

Sposób na wysłanie SMS'a to, mówiąc ogólnie, wypełnienie formularza i wysłanie go na wskazaną stronę. Wykorzystując funkcję fsockopen() inicjujemy połączenie:

```
> <?php
$socket = @fsockopen($host, 80, $errno, $errstr);
?>
```

i wysyłając na otwarte 'gniazdo' odpowiedni nagłówek (\$header):

```
> <?pre
fputs($socket, $header);
fclose($socket);
?>
```

emulujemy zatwierdzenie formularza.

Cała zabawa polega na odpowiednio skonstruowanym nagłówku z parametrami jakie powinny przyjść z formularza. Niektóre bramki (na pewno Era) stosują dodatkowe zabezpieczenie w postaci konieczności pobrania ciasteczka z id sesji i odczytania ukrytego pola z formularza. Od listopada 2002 bramka Idea wymaga podania tokena przeczytanego z obrazka.

4.9 Czy można ukryć kod PHP?

Tak, można ukryć kod PHP. Służy do tego Zend Encoder (<http://www.zend.com/>). Niestety przyjemność ta trochę kosztuje. Nieco tańsze może okazać się zastosowanie Encodera produkcji Ioncube: <http://www.ioncube.com/encoder/index.php>.

Jest jeszcze phpyteencoder, jednak jak na razie ciężko coś na jego temat powiedzieć.

Istnieje także phpytecompiler dostępny pod adresem <http://pbc.sourceforge.net/>. Niestety jest on dostępny wyłącznie w postaci skompilowanego modułu, a i szyfrowanie kodu odbywa się przez formularz dostępny na stronie projektu. Wygląda też na to, że projekt ten nie jest dalej rozwijany.

Ponadto od czasu do czasu w Internecie pojawiają się plotki o innych darmowych narzędziach tego typu, jednak jak do tej pory nikt z grupowiczów nie widział ich w działaniu, tak więc nic na ten temat nie można dokładnie powiedzieć.

4.10 Jak uzyskać dostęp do danych przesłanych formularzem?

W zależności od metody jaką zostały wysłane (post lub get) poprzez tablice asocjacyjne \$_POST lub \$_GET (w starszych wersjach PHP były to odpowiednio \$HTTP_POST_VARS oraz \$HTTP_GET_VARS). Pliki (dołączone przez input type="file") znajdziesz w tablicy \$_FILES (a w starszych wersjach PHP \$HTTP_POST_FILES).

4.11 Jak uruchomić skrypt w regularnych odstępach czasu?

W systemach Unix, możemy skorzystać z programu cron. Umożliwia on wykonywanie określonych zadań w wybrany przez nas dzień i godzinie.

Postępowanie:

1. Tworzymy skrypt Bashowy, który będzie uruchamiał nasz program w PHP.
2. Tworzymy odpowiednią regułę w cronie aby uruchamiał nasz skrypt.

Gdy PHP jest jako CGI to punkt 1 można pominąć.

ad 1.

```
> #!/bin/bash
lynx -dump http://adres/katalog/plik.php >> out.txt
```

Zapisujemy np. pod nazwą start1
Nadajemy prawa do uruchamiania

```
> chmod 700 start1
```

sprawdzamy czy skrypt bashowy uruchomi nam nasz program w PHP.

```
> ./start1
```

W wyniku zadziałania tego skryptu powinien zostać utworzony plik out.txt zawierający wynik (w postaci TXT) działania skryptu PHP.

Jeśli nie chcemy tworzyć pliku z wynikiem to zamiast » out.txt należy wpisać > /dev/null

ad 2.

aby edytować swoje reguły w cronie należy uruchomić crontab -e zostanie uruchomiony domyślny edytor (zwykle Vi)

następnie wpisać przykładową regułę

```
> 40 11 * * * /home/cos/start1
```

Reguła ta określa że każdego dnia o 11:40 ma być uruchamiany plik start1 mieszczący się w katalogu /home/cos/

więcej informacji:

man cron

man 1 crontab

man 5 crontab

Jeśli dostępne jest php w postaci CGI, do crona można dodać bezpośrednie wywołanie skryptu:

```
> /usr/bin/php -q /home/cos/plik.php
```

Jeśli plik php posiada prawa wykonywalności oraz pierwsza linia tego pliku to:

```
> #!/usr/bin/php
```

to do crona można wpisać już samą nazwę tego pliku, bez jawnego wywołania interpretera PHP, czyli:

```
> /home/cos/plik.php
```

4.12 Słyszałem o sesjach. Co to jest? Jak ich używać? Gdzie znajdę więcej informacji na ten temat?

Sesje to związane ze sobą serie wielu interakcji, dzięki czemu nie tracimy informacji, co być może; o przedmiotem naszych operacji chwile wcześniej, zatem to doskonały sposób do autoryzowania transakcji zawieranych przez konkretnego użytkownika znanego systemowi. Użytkownik loguje się w systemie i zostaje mu przydzielony tzw. identyfikator sesji, który także od tego czasu jest chwilowo składowany na serwerze, posługując się tym unikalnym identyfikatorem system ma pewność, kto w danej chwili buszuje w systemie i zawiera transakcję.

Więcej o sesjach:

<http://pl.php.net/manual/pl/ref.session.php>

<http://www.zend.com/zend/tut/session.php> (ang)

4.13 Jak wysłać list z załącznikiem?

Aby wysłać taki list można skorzystać z biblioteki PEAR (patrz punkt 6.8) lub jednej z wielu innych - można takie znaleźć np. na stronie <http://zend.com/> lub <http://phpclasses.upperdesign.com/>

4.14 Jak sprawdzić poprawność adresu e-mail?

Poniższa funkcja (napisana przez lemming nine) wykonuje dokładne sprawdzanie poprawności adresu e-mail. Sprawdza także czy domena istnieje i posiada prawidłowy rekord MX:

```
> <?php
function verifyEmail($email) {
    $wholeexp = '/^(.+?)@(([a-z0-9\.-]+?)\.[a-z]{2,5})$/i';
```

```

$userexp = "/^[a-z0-9~\!\#\$\%\&\(\)\-\_\+\=\[\]\;\:\'\\"",\.\./]+\$/i";
if (preg_match($wholeexp, $email, $regs)) {
    $username = $regs[1];
    $host = $regs[2];
    if (checkdnsrr($host, MX)) {
        if (preg_match($userexp, $username)) {
            return true;
        } else {
            return false;
        }
    } else {
        return false;
    }
} else {
    return false;
}
}
?>

```

4.15 Czy w PHP można sprawdzić poprawność numeru...?

4.15.1 NIP

```

> <?php
# zwraca 1 gdy NIP jest poprawny

function check_nip($nip)
{
    // tworzenie tablicy wag
    $steps = array(6, 5, 7, 2, 3, 4, 5, 6, 7);

    // wycinanie zbędnych znaków z numeru
    $nip = str_replace('-', '', $nip);
    $nip = str_replace(' ', '', $nip);

    if (strlen($nip) != 10) return 0;

    // tworzenie sumy iloczynów
    for ($x = 0; $x < 9; $x++) $sum_nb += $steps[$x] * $nip[$x];

    $sum_m = $sum_nb % 11;

    if ($sum_m == 10) $sum_m = 0; // *
    if ($sum_m == $nip[9]) return 1;

    return 0;
}

```

```
// sposób wykorzystania
if (check_nip('768-000-24-66')) echo 'NIP poprawny';
?>
```

* Poprawka z dnia 03.09.2004, błąd zauważył Anakin

Szczegółowe informacje na temat algorytmu m.in. na stronie: <http://wipos.p.lodz.pl/zylla/ut/nip-rego.html>

Inny sposób sprawdzenia można zobaczyć na stronie:
<http://anakinek.com/faq/4.15.php>

4.15.2 REGON

```
> <?php
# zwraca 1 gdy REGON jest poprawny

function check_regon($regon)
{
    // tworzenie tablicy wag
    $steps = array(8, 9, 2, 3, 4, 5, 6, 7);

    // wycinanie zbędnych znaków
    $regon = str_replace('-', '', $regon);
    $regon = str_replace(' ', '', $regon);

    if (strlen($regon) != 9) return 0;

    // tworzenie sumy iloczynów
    for ($x = 0; $x < 8; $x++) $sum_nb += $steps[$x] * $regon[$x];
    $sum_m = $sum_nb % 11;

    if($sum_m == 10) $sum_m = 0;
    if ($sum_m == $regon[8]) return 1;
    return 0;
}

// sposób wykorzystania
if (check_regon('590096454')) echo 'REGON poprawny';
?>
```

Szczegółowe informacje na temat algorytmu m.in. na stronie: <http://wipos.p.lodz.pl/zylla/ut/nip-rego.html>

Inny sposób sprawdzenia można zobaczyć na stronie:
<http://anakinek.com/faq/4.15.php>
Również dla REGONu 7-mio cyfrowego.

4.15.3 PESEL

```
> <?php
function check_pesel($pesel)
{
    if (strlen($pesel) != 11 || !is_numeric($pesel))
        return 0;
    $steps = array(1, 3, 7, 9, 1, 3, 7, 9, 1, 3);
    for ($x = 0; $x < 10; $x++) {
        $sum_nb += $steps[$x] * $pesel[$x];
    }
    $sum_m = 10 - $sum_nb % 10;
    if ($sum_m == 10)
        $sum_c = 0;
    else
        $sum_c = $sum_m;
    if ($sum_c == $pesel[10])
        return 1;
    return 0;
}
?>
```

Szczegółowe informacje na temat algorytmu PESEL i innych opartych o sumę kontrolną (NIP, REGON itp.) można znaleźć m.in. pod adresem:
<http://wipos.p.lodz.pl/zylla/ut/pesel.html>

Inny sposób sprawdzenia można zobaczyć na stronie:
<http://anakinek.com/faq/4.15.php>

4.16 Jak zarejestrować funkcję callback, która jest metodą w klasie?

```
><?php # przykładowa klasa:
class MyClass {
    function MyMethod($a, $b) {
        return $a.$b;
    }
}

# na przykładzie array_map()
$object = new MyClass;
$array = array_map(array($object, 'MyMethod'), $a, $b);
```

4.17 Co to jest referencja? Do czego jej użyć?

Referencja to odwołanie do obszaru pamięci zmiennej.
Dzięki referencjom skrypt zużywa mniej pamięci, przez co staje się szybszy i bardziej

wydajny.

Zazwyczaj przydają się do dwóch rzeczy:

a) modyfikacji argumentów funkcji bez zwracania ich za pomocą instrukcji return, przydatne jeśli nasza funkcja dokonuje operacji na kilku zmiennych, a nie możemy / nie chcemy zwrócić ich jako tablicy:

```
> <?php
$args = 1;
function foo(&$arg) {
    $arg++;
}

foo($arg);
echo $arg; //zwróci 2
?>
```

UWAGA:

W niektórych skryptach (nawet tych w manualu) można się spotkać z konstrukcją foo(&\$arg). Jest ona przestarzała, działa tylko jeśli w php.ini allow_call_time_pass_reference = On i nie będzie supportowana w przyszłych wersjach interpretera .

b) tworzenia aliasów:

```
> <?php
$var1 = 'ala ma kota';
$var2 =& $var1;
$var2 = 'ala nie ma kota!';
echo $var1; //zwróci nam 'ala ma kota!'
?>
```

Cokolwiek zrobimy z \$var2 w rzeczywistości będzie działaniem na \$var1. Bez znaczka "&" interpreter skopiowałby zawartość \$var1 i przypisał ją do \$var2. W ten sposób mielibyśmy w pamięci dwie zmienne o tej samej zawartości, a przecież nie zawsze jest nam to potrzebne. Na tym przykładzie nie widać wagi problemu, ale wyobraźmy sobie że pod \$var1 znajduje się sporej wielkości plik wczytany przez readfile()...
gdzie więcej przeczytać o referencjach?

<http://www.php.net/manual/en/language.references.php>

4.18 Jak korzystać z plików konfiguracyjnych w stylu php.ini?

4.19 Jak zoptymalizować kod napisany w PHP?

Przed wszystkim wyłącz komputer, prześpij się a następnego dnia spójrz na swoje dzieło trzeźwym okiem. Krok po kroku przeanalizuj wszystkie wykonywane operacje i wczuj się w rolę parsera który jest (podobnie jak człowiek) bardzo leniwym tworem i chciałby zrobić jak najwięcej jak najmniejszym kosztem.

- Podmiana znaków: jeśli wiemy co podmieniamy i gdzie nie używajmy funkcji operujących na wyrażeniach regularnych.

- Nie powtarzaj się! Jeśli wykonujesz jakieś działania a ich wynik będzie potrzebny kilkakrotnie zapisz go w zmiennej i później wykorzystaj. Dotyczy to również operacji wykonywanych przy każdym przebiegu pętli. np: `for($i=0; $i < count($i); $i++)` można zastąpić wywołaniem `foreach` bądź wcześniej zapamiętać ile razy będziemy musieli wykonać pętle.

- Uważaj by nie tworzyć zbyt wielu zmiennych tymczasowych (np aktualny numer indeksu przy przejściu pętli). Pamiętaj, że w PHP dostępne są referencje więc nie zawsze musisz tworzyć kopii zmiennej by użyć jej wewnątrz np jakiejś funkcji. Stworzenie własnej notacji bądź przyjęcie jakiejś określonej pozwoli też nie zgubić się w gąszczu zmiennych.

- Nie daj się skusić na `$array = file()`. To bardzo wygodne, lecz tylko jeśli wiemy co wczytujemy do tablicy i ile tego jest. Domyślnie jeden skrypt PHP ma prawo skonsuować do 8M pamięci, więc nawet prosta analiza logów może skończyć się tragicznie. Znacznie lepszym rozwiązaniem jest zastosowanie pętli `while()` w połączeniu z funkcją `fgets()`.

- Ustaw `error_reporting` na `E_ALL` w `php.ini`. Znaczący problem twierdzą, że dopiero wówczas rozpoczyna się prawdziwa przygoda z PHP :-)

- Ciapki pojedyncze vs. podwójne: gdzie tylko możesz stosuj ciapki pojedyncze, gdyż string nimi objęty jest uwolniony od dalszego zainteresowania parsera, który milcząco zakłada że może zostawić go w spokoju, natomiast gdy ciapki są podwójne parser jest bardziej dociekliwy i analizuje zawartość stringa próbując znaleźć w nim zmienne lub znaki specjalne.

- Nie mieszaj ogórków z dżemem, czyli HTML-a z PHP-em: jeśli nie zdecydowałeś się na korzystanie z szablonów, a co za tym idzie odseparowanie logiki od warstwy prezentacyjnej staraj się aby html był poza zasięgiem znaczników `<?php ?>`

- Pamiętaj, iż jeśli przygotowałeś kilka rozwiązań i nie wiesz które z nich będzie działać szybciej, po prostu to zmierzyc. Patrz też FAQ 6.14.

- Sprawdź, które części twojego skryptu są najwolniejsze i pomyśl jak można by je przyspieszyć. Jeśli np. operujesz na wielu plikach tekstowych to może warto użyć bazy danych. Jeśli twoje zapytania wykonują się za długo to spróbuj znaleźć sposób by je ulepszyć (np zamiast `SELECT kod FROM adresy WHERE miasto='Wroclaw' AND ulica='Szczytnicka'` lepiej napisać `SELECT kod FROM adresy WHERE ulica='Szczytnicka' AND miasto='Wroclaw'` bo mniej mamy ulic 'Szczytnicka' w Polsce niż ulic we Wrocławiu), możesz też użyć polecenia `EXPLAIN` w MySQL'u by zobaczyć jak działa twoje zapytanie.

- Nie próbuj zabijać muchy z armaty - część danych istotnie musi być generowana automatycznie dla każdego wchodzącego na stronę. Niektóre jednak informacje rzadko się zmieniają - np. kto danego dnia ma imieniny, czy całodzienna prognoza pogody. Możesz więc te informacje generować tylko raz na dzień, bądź co najwyżej cyklicznie co

kilka godzin, a nie z każdym wejściem gościa na twoją stronę. Szczególnie wyczulamy na nadmiarowe korzystanie z nie najszybszych bibliotek (np GD). W tym momencie tracąc trochę miejsca na naszym dysku twardym zyskujemy sporo mocy procesora.

- Software: jeśli musisz znacznie przyspieszyć wykonywanie skryptów a nie masz na to czasu możesz użyć programów temu przeznaczonych. Zestawienie kilku popularnych aplikacji znajdziesz na stronie http://php.weblogs.com/php_debugger_cache

- Nie bój się czytelnego formatowania kodu. Fakt że dla maszyny to wszystko jedno (czasem nawet przełknij brak nawiasu zamykającego blok warunkowy), lecz pamiętaj że im czytelniej tym łatwiej wypatrzeć błędy i niedociągnięcia.

Jeśli powyższe wskazówki sprawiły że twój skrypt jest szybki jak burza to wspaniale, a jeśli nie... Cóż, prześpij się jeszcze jedną noc i zacznij wszystko od początku. Wiele przydatnych wskazówek można znaleźć pod adresem <http://phplens.com/lens/php-book/optimizing-debugging-php.php>

4.20 Jak odczytać w PHP rozdzielczość ekranu klienta?

Bezpośrednio nie ma takiej możliwości. Skrypty PHP są wykonywane po stronie serwera, a serwer nic nie wie o komputerze klienta - on tylko wysyła dane.

Można to jednak zrobić w inny sposób. Rozdzielczość ekranu można odczytać przez JavaScript. Jednak tu też nie ma bezpośredniej możliwości odczytania tej wartości przez PHP. Trzeba to jakoś obejść. Istnieją 2 możliwości.

Zmienne z JavaScriptu można przekazać do PHP przez URL. Czyli przez zmianę lokalizacji okna przeglądarki na skrypt PHP z rozdzielczością, co wygląda mniej więcej tak:

```
> <script language="JavaScript">
window.location="index2.php?width="+screen.width+"&height="+screen.height;
</script>
```

Można to też zrobić przez ciasteczka. Z poziomu JavaScriptu trzeba ustawić ciasteczka z odpowiednimi wartościami i przeładować stronę. To robi się tak:

```
> <script language="JavaScript">
document.cookie = "width="+screen.width;
document.cookie = "height="+screen.height;
window.location="index2.php";
</script>
```

I w tym momencie na stronie `index2.php` dostępne są zmienne `$width` i `$height` zawierające odpowiednio szerokość i wysokość ekranu.

4.21 Gdzie można znaleźć bazę danych z datami imiennin?

Bazę danych z imienninami można znaleźć pod adresem <http://www.pomorze.mao.pl/var/imieniny.php>, natomiast przykład jej użycia pod adresem <http://www.pomorze.mao.pl/imieniny/index.php>

4.22 W jaki sposób umieścić dane wewnątrz istniejącego już pliku tekstowego?

W żadnym z języków programowania nie ma możliwości dopisania danych na początku czy w środku istniejącego pliku. Jedyną możliwością jaka istnieje, to napisanie pliku od nowa i zapisanie go pod starą nazwą. W przypadku dopisywania danych na początek pliku będzie to wyglądać tak:

```
> <?php

// wczytanie starych danych

// otwarcie pliku do odczytu
$fp = fopen('plik.txt', 'r');

// odczytanie danych
$stareDane = fread($fp, filesize('plik.txt'));

// zamknięcie pliku
fclose($fp)

// stworzenie nowych danych

$noweDane = "To, co chcesz, żeby było na początku";
$noweDane .= $stareDane;

// zapisanie nowych danych

// otwarcie pliku do zapisu
$fp = fopen('plik.txt', 'w');

// zapisanie danych
fputs($fp, $noweDane);

// zamknięcie pliku
fclose($fp);
?>
```

4.23 Jak zapobiec powtórnemu submitowi formularza?

Propozycje są cztery:

Skrypt przetwarzający dane z formularza po swoim zakończeniu powinien przekierować przeglądarkę na inną stronę. Odświeżenie jej nie wywoła powtórnego submita, jednak cofając się guzikiem back w przeglądarce możemy znów doprowadzić do sytuacji, w której formularz zostanie wywołany ponownie.

Drugie rozwiązanie bazuje na unikalnej zmiennej przekazywanej razem z danymi formularza w polu typu hidden. Tworząc stronę z formularzem zmiennej takiej przypisujemy wygenerowany string, a następnie w momencie przetwarzania wyników sprawdzamy czy taki identyfikator był już przesyłany. Dobrze jest trzymać identyfikatory np. w bazie danych i czyścić je raz na jakiś czas. Dzięki temu klient nawet jeśli cofnie się do strony z formularzem i wykona ponowny submit nie uzyska porządnego efektu. Natomiast, aby ponownie dodać coś od siebie będzie musiał kliknąć odśwież średnio na stronie z formularzem, a to spowoduje wyczyszczenie jego zawartości.

Trzecie rozwiązanie jest analogiczne do drugiego jednak bazuje na cookies/sesjach. Nie zawsze mamy możliwość korzystać z bazy. Tym razem po przesłaniu formularza w przeglądarce klienta ustawiamy cookie zawierające informacje, że dany formularz został już wypełniony. Przykładowo przypisujemy wartość true do cookie o nazwie np. moj_formularz. Skrypt przetwarzający dane dla tego formularza powinien weryfikować obecność cookie o tej nazwie. Opcjonalnie dane te możemy przechowywać w sesji.

Jeśli trzy sposoby nam mało możemy jeszcze weryfikować zmienną HTTP_REFERER. Gdy zawiera ona adres do aktualnej strony wtedy możemy uznać, że nastąpiło odświeżenie. Pamiętać jednak należy, iż przeglądarka nie musi przesłać tej zmiennej w nagłówkach.

4.24 Czemu nie działa mi autoryzacja HTTP?

Aby autoryzacja HTTP działała, PHP musi być uruchamiane jako moduł Twojego serwera WWW (np. apache), a nie jako CGI (samodzielny plik wykonywalny). Aby to zmienić, należy dokonać zmian w pliku konfiguracyjnym serwera www. Cały proces instalacji i konfiguracji PHP jako moduł serwera opisany jest w podręczniku PHP. Jeśli nie masz do tego uprawnień, poproś o to swojego administratora.

4.25 Czemu nie działają mi zmienne globalne po upgrade?

Od wersji 4.2.0 w domyślnej konfiguracji wyłączono ze względów bezpieczeństwa zmienną `register_globals`. Tak więc nie możesz korzystać ze zmiennej `$foo`, która mogło pochodzić z zapytania POST, GET, z cookie, sesji, czy też ze środowiska. Musisz skorzystać z tablic `$_POST`, `$_GET`, `$_COOKIE`, `$_FILES`, czyli np. `$_POST['foo']`. Innym rozwiązaniem jest przywrócenie starego zachowania poprzez edycję pliku konfiguracyjnego `php.ini`. Jeśli nie masz do tego uprawnień, poproś o to swojego admini-

stratora.

Kilka powodów dla których warto jednak korzystać z tablic \$_GET a nie zmiennych globalnych znajdziesz w <http://www.zend.com/zend/art/art-oertli.php#Heading6>

4.26 Jak ukryć parametry w adresie strony?

Aby uzyskać adres w postaci: http://moja.domena.pl/parametr_wartosc/parametr_wartosc niezbędna niestety jest modyfikacja w konfiguracji serwera WWW oraz późniejsze poprawne zinterpretowanie otrzymanych danych. Szerszy opis całego zagadnienia w oparciu o serwer Apache można zleźć tu: <http://php.faq.pl/docs/rewrite.php>

4.27 Jak sprawdzić, które komputery w sieci są włączone?

Najprostszym sposobem na sprawdzenie czy dana maszyna jest dostępna w sieci jest wysłanie do niej tzw. ping'a. Komputery działające odpowiedzą na takie zapytanie natomiast maszyny wyłączone rzecz jasna takowej nie udzielą. W praktyce skrypt realizujący takie zapytanie może wyglądać tak:

```
> <?php
exec('ping -c1 -q tu_ip_hosta', $skan);
preg_match("/([0-9]+)% packet loss/", join(' ', $skan), $wynik);

if ($wynik[1] != '100')
    echo 'On-Line';
else
    echo 'Off-Line';
?>
```

Pierwsza linia wysyła ping do danego komputera, druga linia natomiast za pomocą regexa wyciąga procent pakietów, które zginęły. Jeśli liczba ta wynosi 100 przykładowie pięciokrotnie pingujemy maszynę docelową aby wstępnie wykluczyć problemy z siecią. Skanując jednak sieć lokalną w której mamy pewność przepływu danych spokojnie licznik możemy ustawić na 1 aby zbyt nie wydłużać czasu wykonania skryptu.

4.28 Czemu nie działa mi funkcja mail na localhoscie?

Aby funkcja mail działała poprawnie musisz mieć w przypadku systemu Windows poprawnie wpisaną konfigurację serwera SMTP w pliku php.ini lub w przypadku systemów *nixowych działającego MTA.

4.29 Jak przyspieszyć pracę samego PHP?

Gdy już niewiele możemy wycisnąć z naszego kodu PHP warto zainteresować się akceleratorem PHP.

afterBURNER:

<http://bwcache.bware.it/>

ionCube PHP Accelerator

<http://www.php-accelerator.co.uk/>

APC (Alternative PHP Cache)

<http://apc.communityconnect.com/>

Turck MMCache for PHP

http://www.turcksoft.com/en/e_mmc.htm

Zend Performance Suite (płatny)

<http://zend.com/store/products/zend-performance-suite.php>

4.30 Jak skonwertować pseudo-unicode odczytany z programu Excel do formatu iso-8859-2?

Można to zrobić przy pomocy poniższej funkcji.

```
> <?php
```

```
function xls2iso($string)
```

```
{
```

```
    $zamiana = array("\xC4\x85" => "\xB1", "\xC4\x84" => "\xA1", "\xC4\x87" =>
=> "\xEA", "\xC4\x98" => "\xCA", "\xC5\x82" => "\xB3", "\xC5\x81" => "\xA3", "
"\xC3\xB3" => "\xF3", "\xC3\x93" => "\xD3", "\xC5\x9B" => "\xB6", "\xC5\x9A" =>
=> "\xAF", "\xC5\xBA" => "\xBC", "\xC5\xB9" => "\xAC");
```

```
    $string = strtr($string, $zamiana);
```

```
    return $string;
```

```
}
```

```
?>
```

Rozdział 5

Bezpieczeństwo

5.1 Jak bezpiecznie skonfigurować PHP?

5.2 Jak zabezpieczyć pliki PHP przed dostępem osób niepowołanych?

5.3 Jak zabezpieczyć biblioteki przed odczytem przez WWW?

Pierwszą czynnością jest używanie dla bibliotek (includów) rozszerzeń *.php. Dzięki temu użytkownik nie podejrzewa łatwy sposób kodu źródłowego biblioteki, a jedynie ewentualny "output".

Ponadto warto biblioteki składować poza katalogami udostępnianymi przez http (na ogół powyżej public_html). PHP jako język wykonujący się po stronie serwera będzie miał do nich dostęp, a sam serwer http już nie, więc nie będzie w stanie wysłać tych bibliotek.

Ustawienie odpowiednich praw dostępu do pliku jest rzeczą na tyle oczywistą, iż nie trzeba tutaj o tym wspominać.

5.4 Dlaczego należy używać tablic \$HTTP_* zamiast zmiennych globalnych?

Odpowiedź na to pytanie można znaleźć na stronach zend.com. Jest to główna strona ludzi tworzących PHP (patrz punkt 3.3).

<http://www.zend.com/zend/art/art-oertli.php>

Pamiętaj, że tablice \$HTTP_*_VARS są tworzone jeśli w pliku konfiguracyjnym jest ustawiona dyrektywa track_vars.

W skrócie chodzi o to, iż dzięki stosowaniu tablic \$HTTP_*_VARS ty decydujesz skąd spodziewasz się otrzymać odpowiednią zmienną; czy to z POST, czy z GET, czy

np. z COOKIE. Nie dopuszczasz przez to możliwości nadpisania jej wartości z innych źródeł. Zawsze pamiętaj, że nie możesz wierzyć w poprawność danych, które przychodzą do skryptu PHP od użytkowników.

Zobacz na podaną powyżej stronę do rozdziału "Global Variables". Znajdziesz tam opisany przykład dziury w jednym z dużych systemów do obsługi dynamicznych portali.

Rozdział 6

PEAR (PHP Extension and Application Repository)

6.1 Co to jest PEAR?

PEAR (ang. gruszka) to zbiór bibliotek i rozszerzeń PHP będących odpowiednikiem perlowskiego CPAN dzięki którym budowa nawet skomplikowanej (jak na PHP ;-) aplikacji staje się dziecinną igraszka.

6.2 Jak zainstalować PEAR?

Jeśli masz PHP4 to i PEAR też już jest. Jedyne co musisz zrobić to w php.ini podać ścieżkę dyrektywie "include_path", tak aby wskazywała na katalog z PEAR, choć prawdopodobnie nawet to nie będzie konieczne.

UWAGA:

PEAR nie będzie działał z PHP3!

Jeśli nie masz odpowiednich uprawnień, aby zainstalować PEAR globalnie, możesz to zrobić lokalnie (zakładam, że masz możliwość zalogowania się na koncie):

```
>cvs -d :pserver:cvsread@cvs.php.net:/repository
```

Powinieneś ujrzeć następujący tekst:

(Logging in to cvsread@cvs.php.net)

CVS password:

Jako hasło podajesz phpfi Następnie wykonujesz poniższą komendę:

```
>cvs -d :pserver:cvsread@cvs.php.net:/repository co
```

Zostaną wyświetlone informacje na temat aktualnie pobieranych plików. Po skończeniu aktualizacji świeżutka wersja biblioteki PEAR będzie znajdować się w podkatalogu <code>pear</code> bieżącego katalogu.

Nie pozostaje Ci nic innego, jak odpowiednio przerobienie skryptów, w których wykorzystujesz PEAR dodając następujący kod:

```
> $incpath = ini_get("include_path");  
$incpath .= ":/sciezka/do/katalogu/z/pear";  
$incpath = str_replace(":::", ":", $incpath);  
ini_set("include_path", $incpath);
```

Voila! Możesz korzystać z PEAR. Gratulacje! Innym sposobem jest skorzystanie ze skryptu go-pear:

```
>$ lynx -source http://pear.php.net/go-pear | php
```

6.3 Gdzie można znaleźć najnowszą wersję PEAR?

<http://cvs.php.net/cvs.php/pear>

6.4 Gdzie można znaleźć dokumentację do PEAR?

Niestety dokumentacja trochę kuleje, oficjalne początki są na: <http://pl.php.net/manual/en/pear.php> zaś oficjalna strona projektu to <http://pear.php.net/> Wprowadzenie do PEAR: <http://www.phpbuilder.com/columns/sean>

6.5 Dlaczego używać PEAR?

Ponieważ jest dołączana do każdej oficjalnej dystrybucji PHP, dzięki temu masz pewność że aplikacja uruchomiona na serwerze A będzie poprawnie działa na serwerze B. PEAR oszczędza czas który musiałbyś poświęcić na napisanie wielu standardowych procedur (autoryzacja, sockety, obsługa błędów, komunikacja z bazami danych...). Praca z PEAR uczy podstaw efektywnego programowania oraz pisania czytelnego kodu.

6.6 Jak sprawdzić poprawność danych?

Na samą myśl o wyrażeniach regularnych przechodzą cię dreszcze? Jest coś co może pomóc:
<http://vulcanonet.com/soft/index.php?pack=validata>

6.7 Jak porcjować wyniki zapytania?

<http://vulcanonet.com/soft/index.php?pack=pager>

6.8 Jak wysłać maila z załącznikiem lub w formacie HTML?

<http://pear.php.net/manual/en/package.mail.mail-mime.php>

6.9 Jak rozwiązany jest dostęp do baz danych w PEAR?

http://vulcanonet.com/soft/index.php?pack=pear_tut
<http://phpbuilder.org/columns/allan20010115.php3>
<http://www.onlamp.com/pub/a/php/2001/07/19/pear.html>
<http://www.onlamp.com/pub/a/php/2001/05/24/pear.html>

6.10 Jak PEAR obsługuje upload plików?

<http://vulcanonet.com/soft/index.php?pack=uploader>

6.11 Szablony w PEAR

6.11.1 Co to są szablony? Po co ich używać?

<http://www.pckurier.pl/webmaster/2000/marzec/felsztukier/phpaktualizacje.html>

6.11.2 Jak używać szablonów?

szablon.htm:

```
> <HTML>
  <HEAD>
    <TITLE>{TYTUL}</title>
  </head>
  <BODY>
    {TRESC}
  </body>
</html>
```

a teraz skrypt:

```
> <?php
# dołączamy klasę
include_once 'Html/IT.php';
# informujemy gdzie są nasze szablony
$tpl = new IntegratedTemplate('moje_szablony/');
# ładujemy szablon
$tpl->loadTemplateFile('szablon.htm');
$tpl->setVariable('TYTUL', 'My homepage');
$tpl->setVariable('TRESC', 'Cześć, jestem treścią!');
# i do przeglądarki!
$tpl->show();
?>
```

6.12 Jak udawać przeglądarkę?

6.13 Nie mam / Nie chcę używać sendmaila do wysyłania poczty. Co zrobić?

```
> <?php
# zrywamy gruszkę
require_once 'Mail.php';

# twój serwer smtp
$params['host'] = 'smtp.wp.pl';

# dodatkowe parametry połączenia (opcjonalne):
# $params['port'] = 25;
# jeśli serwer wymaga autentykacji:
# $params['auth'] = false;
# $params['username'] = '';
# $params['password'] = '';

# wszelkie nagłówki...
$headers['Subject'] = 'To jest temat';
$headers['From'] = 'kazio@wp.pl';

$body = 'oto jakiś spam....';

# jeśli adresat jest jeden wystarczy
# $recipients = 'user@domena.pl';
$recipients = array('mieciu@wp.pl', 'ziutek@onet.pl');

$mail = Mail::factory('smtp', $params);

if (PEAR::isError($mail)) {
    print $mail->getMessage();
} else {
    # sio w Internet!
    $error = $mail->send($recipients, $headers, $body);
    if (PEAR::isError($error)) {
        print $error->getMessage();
    } else {
        print 'mail został wysłany.';
    }
}
?>
```

6.14 Napisałem dwa różne skrypty/funkcje/algorytmy wykonujące te same zadania. Jak sprawdzić, co jest szybsze?

Skorzystaj z klas Benchmark/Timer.php oraz Benchmark/Iterate.php przykład zastosowania:

<http://www.dclp-faq.de/q/q-code-performance.html> <http://www.onlamp.com/lpt/a/1592>

6.15 Jak wczytać do tablicy nazwy plików zawartych w danym katalogu lub znaleźć określony plik?

```
> <?php
include_once 'File/Find.php';
$obj = new File_Find;

# w ten sposób uzyskujemy dwie tablice odzwierciedlające
# strukturę plików i katalogów zawartych w /home/www
list($katalogi, $pliki) = $obj->maptree('/home/www');

# lecz co zrobić gdy poszukujemy tylko określonych plików?
# (np. *.htm) są na to dwa sposoby:

$pliki = $obj->glob("htm$", '/home/www');

# powyższa metoda przeszukuje tylko aktualny katalog dopasowując
# do wzorca całą jego zawartość, tak więc możliwe staje się
# poszukiwanie zarówno plików jak i katalogów

# druga metoda przeszukuje wszystkie ścieżki w całej strukturze
# katalogów /home/www
$pliki = $obj->search("htm$", '/home/www');
?>
```

Domyślnie mechanizm wyszukiwawczy operuje na POSIX-ach (funkcje `ereg_*`) i jeśli komuś to nie wystarcza z łatwością może przesiąść się na PCRE (funkcje `preg_*`). Wystarczy w metodach `search()` i `glob()` dodać trzeci argument, czyli flagę `'perl'`.