

27.03.2016

Serwer PHP. Skrypty PHP.

Teoria

Protokół HTTP

HTTP (ang. Hypertext Transfer Protocol – protokół przesyłania dokumentów hipertekstowych) to protokół sieci WWW (ang. World Wide Web). Obecną definicję HTTP stanowi RFC 2616. Za pomocą protokołu HTTP przesyła się żądania udostępnienia dokumentów WWW i informacje o kliknięciu odnośnika oraz informacje z formularzy. Zadaniem stron WWW jest publikowanie informacji – natomiast protokół HTTP właśnie to umożliwia. HTTP standardowo korzysta z portu nr 80 (TCP).

Problem bezstanowości.

http jest zaliczany do protokołów **bezstanowych** (ang. stateless) z racji tego, że nie zachowuje żadnych informacji o poprzednich transakcjach z klientem (po zakończeniu transakcji wszystko "przepada"). Pozwala to znacznie zmniejszyć obciążenie serwera, jednak jest kłopotliwe w sytuacji, gdy np. trzeba zapamiętać konkretny stan dla użytkownika, który wcześniej łączył się już z serwerem.

Rozwiązanie problemu, że serwer HTTP nie zachowuje żadnych informacji z ostatniej sesji:

- Najczęstszym rozwiązaniem tego problemu jest wprowadzenie mechanizmu ciasteczek.
- Inne podejścia to m.in. sesje po stronie serwera, ukryte parametry (gdy aktualna strona zawiera formularz) parametry umieszczone w URL-u (jak np. /index.php?userid=3).

Serwer www

Serwer WWW (ang.) web server – program zainstalowany i uruchomiony na serwerze zdalnym (oddalonym do użytkownika). Serwer WWW obsługuje żądania protokołu komunikacyjnego HTTP. Z serwerem WWW łączy się, poprzez sieć komputerową, przeglądarka internetowa, będąca jego klientem, aby pobrać wskazaną stronę WWW.

Serwer WWW może też korzystać z usług innego, równoległe działającego oprogramowania, np. MySQL i PHP, udostępniając wynikowe, dynamicznie utworzone strony WWW, wzbogacone danymi z bazy danych.

Najpopularniejszym serwerem WWW jest Apache. Według badań z grudnia 2011 obsługuje on 65% wszystkich stron internetowych na świecie. Drugie miejsce zajmuje IIS firmy Microsoft z wynikiem 15%.

Statystyki wykorzystania serwerów www (dane z grudnia 2011 według serwisu Netcraft):

Apache	362 267 922	65,22%
Microsoft	82 521 809	14,86%
nginx	49 143 289	8,85%
Google	18 464 148	3,32%

Protokół HTTPS

HTTPS (ang. Hypertext Transfer Protocol Secure) – szyfrowana wersja protokołu HTTP. W przeciwieństwie do komunikacji niezaszyfrowanego tekstu w HTTP klient-serwer, szyfruje go za pomocą protokołu SSL. Zapobiega to przechwytywaniu i zmienianiu przesyłanych danych.

HTTPS działa domyślnie na porcie nr 443 w protokole TCP. Wywołania tego protokołu zaczynają się od https://, natomiast zwykłego połączenia HTTP od http://. Protokół HTTPS jest warstwę wyżej (na transporcie SSL), najpierw następuje więc wymiana kluczy SSL, a dopiero później żądanie HTTP. Powoduje to, że jeden adres IP może serwować tylko jedną domenę lub też tylko subdomeny danej domeny (zależnie od przyznanego certyfikatu).

Czym jest PHP?

PHP to skryptowy język programowania - programy w nim napisane nie są kompilowane do postaci kodu maszynowego zrozumiałego dla procesora, lecz wykonywane przez specjalną aplikację zwaną interpreterem PHP.

PHP został pierwotnie stworzony do wspomagania tworzenia dynamicznych stron WWW, lecz obecnie można w nim pisać także zwyczajne aplikacje dla systemu operacyjnego.

PHP jest projektem open-source. Każdy może pobrać za darmo jego kopię, zainstalować i używać bez żadnych ograniczeń. Do kodu źródłowego zapewniony jest pełny dostęp - jeżeli programista ma odpowiednie zdolności, może go modyfikować do woli oraz nadsyłać własne propozycje zmian do osób nadzorujących projekt. Dzięki takiej wolności PHP rozwija się bardzo dynamicznie, a w Internecie można znaleźć setki modyfikacji oraz dodatkowych modułów.

Mimo otwartości kodu, nad rozwojem oficjalnej wersji projektu czuwa firma Zend Company z Izraela założona przez twórców PHP. Zapewnia ona dodatkowe narzędzia i opiekę prawną, a także wyznacza kierunki rozwoju projektu.

Bardzo istotną cechą PHP jest skalowalność. Może go używać z powodzeniem zarówno początkujący programista, dzięki uproszczeniom składni, jak i ekspert, który znajdzie tutaj wszystkie zaawansowane narzędzia niezbędne do napisania rozbudowanej aplikacji. PHP znaleźć można wszędzie: od prywatnych stron domowych przez gry internetowe aż do potężnych witryn korporacji lub portali. Opanowanie języka jest proste także dlatego, iż Internet pełen jest przewodników oraz artykułów pokazujących, jak w praktyce wykorzystywać wiele z jego możliwości.

Konkurencyjne języki skryptowe dla stron WWW: ASP, JSP.

PHP jest wykonywany po **stronie serwera**. Oznacza to, że PHP nie jest interpretowany (przetwarzany) przez program do oglądania Internetu, lecz przez specjalny program na serwerze!

HTML, JavaScript w wersji wykonywanej po **stronie klienta**. Jak działa interpretator PHP? To proste - przerabia instrukcje PHP na HTML i JavaScript w wersji do interpretacji po

stronie klienta, które trafiają na Twój komputer i wiesz co się dzieje. A przy okazji - Wikibooks i jej siostrzany projekt - Wikipedia - są napisane w PHP. Jak PHP współpracuje ze stroną WWW?

PHP jest językiem **server-side**, tj. **pracuje po stronie serwera WWW**. Przeciwnieństwem są języki **client-side** **pracujące po stronie przeglądarki użytkownika** (np. JavaScript w wersji wykonywanej po stronie klienta). Aby wykorzystywać go na własnej stronie, musisz upewnić się, że twój serwer WWW ma zainstalowaną jego obsługę. Zanim przejdziemy dalej, należy zrozumieć zasadę, na jakiej PHP generuje dynamiczne strony WWW.

Kiedy wpisujemy adres w przeglądarce internetowej, żądanie wyświetlenia strony kierowane jest do serwera HTTP zwanego także serwerem WWW. Jeśli stwierdzi on, na podstawie rozszerzenia pliku, że dany dokument zawiera kod PHP, kieruje do interpretera żądanie przetworzenia podanego pliku. Interpreter wyszukuje w jego treści tzw. wstawki PHP wplecione w statyczny kod HTML i zastępuje je wynikiem ich przetworzenia. Utworzony kod HTML jest zwracany serwerowi, a ten wysyła go z powrotem do internauty.

W tym procesie kod PHP nigdy nie opuszcza serwera. Internauta zawsze otrzyma wyłącznie utworzony przez PHP kod HTML. Oto przykład. Jeśli mamy plik PHP o następującej treści:

```
<html>
<body>
<?php
    echo 'Podaj login';
?>
</body>
</html>
```

To internauta zobaczy jedynie dokument o takiej treści:

```
<html>
<body>
Podaj login
</body>
</html>
```

Cały PHP zniknie, a na jego miejscu pojawi się utworzony przez niego kod HTML.

Dzięki pracy po stronie serwera, PHP idealnie nadaje się do tworzenia złożonych aplikacji zarządzających dużymi ilościami danych: forami dyskusyjnymi, systemami zarządzania treścią, sklepami internetowymi. Generują one odpowiedni kod HTML dla przeglądarki, a w momencie kiedy internauta przegląda stronę, PHP już zakończył nad nią swą pracę. Jest to bardzo istotne, ponieważ wszelkie dalsze reakcje na poczynania użytkownika należy albo pozostawić przeglądarce, albo obsłużyć je za pomocą języka JavaScript.

W PHP stworzono m.in. aplikację MediaWiki, za pomocą której podręcznik ten jest oficjalnie dostępny w ramach projektu Wikibooks. PHP zarządza tutaj pobieraniem treści żądanej strony z bazy, sformatowaniem jej, a w przypadku kliknięcia na opcję "Edytuj" - dodaniem nowej wersji tekstu. JavaScript pracujący po stronie przeglądarki użytkownika ułatwia edycję tekstu, obsługując przyciski automatycznie wstawiające niektóre rodzaje formatowania oraz

znaki narodowe. Jest to dobry przykład współpracowania tych dwóch języków w naprawę dynamicznej aplikacji internetowej.

Historia projektu

Pierwsza wersja PHP, rozpowszechniana pod nazwą PHP/FI ("Personal Home Page/Forms Interpreter"), została stworzona przez Rasmusa Lerdorfa w roku 1994 jako zestaw skryptów Perla służący do monitorowania internautów odwiedzających jego witrynę. Gdy ruch stał się zbyt duży, przepisał je w języku C, dodając przy tym nowe opcje. Niedługo później ludzie zaczęli prosić go o możliwość użycia tych narzędzi na swoich stronach, zatem 8 czerwca 1995 roku autor udostępnił publicznie kod źródłowy (PHP Tools 1.0). Już kilka miesięcy później projekt przekształcił się w załączek znanego obecnie języka programowania, gdy został połączony z innym narzędziem Rasmusa Lerdorfa - "Form Interpreter", które dało drugi człon nazwy. W 1997 roku pojawiło się PHP/FI 2.0, posiadające wtedy kilka tysięcy aktywnych użytkowników na całym świecie oraz obsługujące 50 tys. domen. Co ciekawe, wersja ta spędziła większość "życia" na beta testach. Oficjalne wydanie było tylko jedno i ukazało się w listopadzie 1997 roku.

W 1997 roku projektem zainteresowali się dwaj izraelscy programiści: Zeev Suraski i Andi Gutmans. Odkryli oni, że PHP/FI ma zbyt małe możliwości na potrzeby aplikacji eCommerce, którą tworzyli na uniwersytecie. Zdecydowali wtedy, że przepiszą kod PHP całkowicie od nowa, korzystając z pomocy już istniejącej społeczności PHP. W czerwcu 1998 roku ogłosili PHP 3.0 jako następcę PHP/FI, którego dalszy rozwój został wtedy zatrzymany. Był to wielki krok naprzód. PHP 3.0 posiadało całkowicie nową architekturę, która znacznie zwiększała wydajność. Pojawiły się w niej załączki programowania obiektowego, ale najważniejszą cechą aplikacji była jej modularność. Użytkownicy mogli rozszerzać teraz funkcjonalność języka poprzez dodawanie nowych modułów.

Krótko po wydaniu PHP 3, w zimie 1998 Zeev Suraski oraz Andi Gutmans jeszcze raz zabrali się za przepisywanie kodu źródłowego PHP, korzystając z doświadczeń nabytych przy pracach nad poprzednią wersją. Za główne cele obrali poprawienie modularności oraz wydajności złożonych aplikacji. Choć dotychczasowa wersja potrafiła sobie z nimi poradzić, nie była jednak stworzona do tego celu i przegrywała przez to z innymi rozwiązaniami.

W połowie roku 1999 ukazał się oficjalnie Zend Engine, nowy silnik języka skryptowego, wokół którego niedługo później zaczęto budować PHP 4. Jego nazwa to kompromisowe połączenie imion twórców projektu. Nowa, oparta o niego wersja PHP, ukazała się w maju 2000 roku. Tak jak poprzednio, był to potężny krok naprzód. Programiści mieli do dyspozycji teraz wiele nowych narzędzi, konstrukcji językowych oraz bezpieczniejszy system wejścia/wyjścia. Od strony administracyjnej pojawiło się oficjalne wsparcie dla wielu nowych serwerów. Przez cztery lata od chwili wydania ukazały się trzy kolejne edycje tej wersji oznaczone numerami: 4.1, 4.2 oraz 4.3. W każdej z nich odczuwalne było zwiększenie bezpieczeństwa, szybkości działania oraz możliwości. W 2004 roku obsługiwały one łącznie 20% wszystkich domen sieciowych. Również daleko po premierze PHP 5, "czwórka" była bardzo chętnie wykorzystywana przez administratorów, ze względu na dużą stabilność.

W 2002 roku Zeev Suraski oraz Andi Gutmans ponownie rozpoczęli znaczącą modernizację silnika PHP mającą na celu dodanie do tego języka modelu obiektowego z prawdziwego zdarzenia. W lutym 2003 ukazała się pierwsza wersja alpha nowej wersji PHP oznaczonej numerem 5.0.0. Stabilna wersja ukazała się prawie półtora roku później, w lipcu 2004 roku.

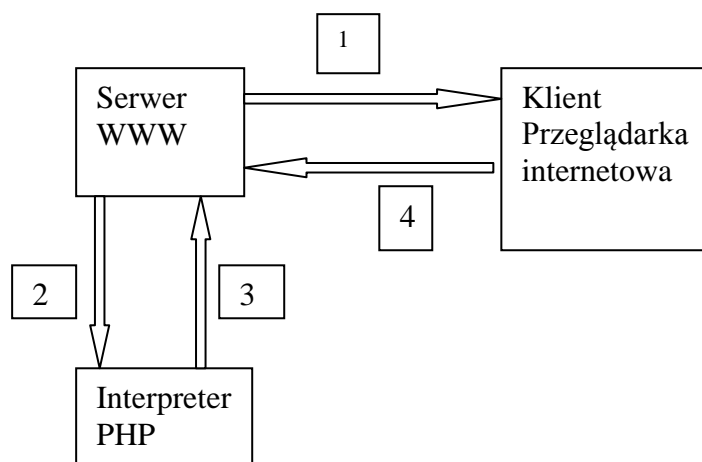
Nowości sprawiły, że PHP zaczął konkurować z innymi rozwiązaniami server-side jak równy z równym. Pojawił się całkowicie nowy model programowania obiektowego, przez co niestety została utracona część kompatybilności z poprzednimi wersjami PHP w niektórych skryptach. Jest to spowodowane zmianą sposobu reprezentacji obiektów. Przebudowano także wiele modułów, w tym do obsługi XML'a i komunikacji z bazą danych, czyniąc je bardziej przyjaznymi dla programistów.

W połowie roku 2005 zaczęły pojawiać się oficjalne sygnały, że rozpoczęto wstępne prace nad PHP 6. Obecnie publicznie dostępne są codzienne snapshoty rozwojowego repozytorium kodu źródłowego, które można ściągnąć i przetestować. Głównym celem jest dalsze dążenie do ujednolicenia projektu, wprowadzenia kolejnych możliwości wymaganych przez złożone projekty (m.in. pełne wsparcie Unicode czy system cache'owania kodu). Usuwane są też kolejne archaiczne rozwiązania pochodzące jeszcze z czasów PHP/FI oraz PHP3, co w przypadku najstarszych skryptów ponownie spowoduje problemy z kompatybilnością.

Serwer PHP(inaczej interpreter PHP)

Jest to program zainstalowany i uruchomiony na komputerze zdalnym (oddalonym od nas), który rozumie polecenia języka PHP. Interpreter otrzymuje dokument w HTML i wyszukuje w nim wstawek w PHP. Po znalezieniu wycina je zmienia na kod HTML następnie wstawia i przesyła plik, który jest w języku HTML.

Schemat wyświetlania stron z użyciem Serwera PHP



1-Przeglądarka żąda przesłania do niej strony

2-Serwer przesyła stronę w języku PHP do Interpretera (serwera) PHP

3-Interpreter po wykonaniu kodu PHP i przesyła stronę w postaci HTML do serwera WWW

4-Strona WWW w postaci HTML trafia do przeglądarki użytkownika

Localhost

127.0.0.1 to adres IPv4 komputera lokalnego. Odpowiadająca mu nazwa mnemoniczna to **localhost**.

Komunikacja z 127.0.0.1 oznacza wymianę informacji wewnątrz jednego hosta.

Zgodnie z RFC 3330 komunikacja w jednym komputerze możliwa jest poprzez adresy mieszczące się w puli 127.0.0.0/8. Jeżeli wystąpi odwołanie do adresu 127.x.x.x, gdzie za x podstawimy liczby od 0 do 255 (1 do 255 w przypadku ostatniego bajta), to zawsze odnosi się do hosta lokalnego. Odpowiada to masce sieciowej 255.0.0.0.

::1 to adres IPv6 komputera lokalnego. Zgodnie z RFC 3513 urządzeniu za localhost opowiada zakres adresów ::1/128

Loopback to wirtualne urządzenie sieciowe, które ma tylko jedno wejście. Często przypisuje się do niego adres 127.0.0.1. W systemie operacyjnym loopback zasadniczo nie różni się od prawdziwego urządzenia sieciowego (np. karty sieciowej). Dziś wiele aplikacji wymaga obsługi protokołu IPv4 i urządzenia typu loopback nawet wtedy, jeżeli fizyczna sieć IPv4 nie jest dostępna i nie byłaby wykorzystywana. Poszczególne programy wykorzystują techniki sieciowe do wymiany informacji w jednej maszynie bez wykorzystywania takich urządzeń jak np. modem.

STRONY STATYCZNE I DYNAMICZNE

Strony statyczne nie zmieniają zawartości lub wyglądu przy każdym wywołaniu. Aby dokonać zmian na stronie programista strony musi manualnie zmienić kod HTML strony.

Strony dynamiczne, w przeciwieństwie do stron statycznych, generowane są na bieżąco przez serwer HTTP na podstawie zmiennych i parametrów przekazanych przez przeglądarkę internetową.

Strony dynamiczne dostosowują swoją zawartość i/lub wygląd w zależności od interakcji z użytkownikiem lub środowiska w którym zostały wywołane (miejsce, czas, modyfikacje w bazie danych itp.).

Zmiany zawartości strony mogą odbywać się w dwojaki sposób.

Sposób 1

Można dokonać tych zmian **po stronie użytkownika**, na jego komputerze (tzw. client-side) przez użyciu takich języków skryptowych jak:

- JavaScript,
- JScript,
- Actionscript

dokonujących zmian bezpośrednio na elementach DOM (Document Object Model - sposób reprezentacji złożonych dokumentów XHTML i XML w postaci modelu zorientowanego obiektowo).

Sposób 2

Jest to kompilacja dynamicznej zawartości na serwerze (tzw. server-side) **po stronie serwera** wykorzystując języki programowania takie jak

- PHP,
- ASP,
- Perl,
- Coldfusion,
- JSP,
- Python.

W skomplikowanych serwisach internetowych zazwyczaj wykorzystywane są obie metody jednocześnie. Korzystanie z technologii dynamicznego generowania stron WWW ma sens

wówczas, gdy oparty na nich serwis internetowy jest bardzo rozbudowany i często odwołuje się do zmiennych danych.

Prawdziwa moc i nieograniczona funkcjonalność stron dynamicznych ujawnia się dopiero podczas współpracy z serwerami baz danych, gdzie przechowywane są elementy niezbędne do wygenerowania pojedynczej strony WWW - przede wszystkim teksty i grafiki. W dużym uproszczeniu można powiedzieć, że dynamiczna strona to kompozycja dwóch składowych: szablonu decydującego o formatowaniu i zmiennych decydujących o zawartości.

Każda z tych technologii posiada swoje wady i zalety. Technologie można bez trudu łączyć w celu uzyskania złożonego efektu.

Zadanie1

Zapisz w zeszycie/sprawozdaniu (przepisz pytanie i udziel odpowiedzi)

- 1)Co to jest serwer WWW. Rodzaje, podaj procenty. Z jakim oprogramowaniem może współpracować.
- 2)Sposoby wykonywania dynamicznych stron WWW
- 3)Co musimy mieć aby wykonywać strony PHP na naszym komputerze.
- 4)W jaki sposób uruchamiamy program w PHP
`http://localhost/~konto/index.php`
- 5)Schemat wyświetlania stron z użyciem PHP
- 6)Historia PHP
- 7)Jaki adres IP ma komputer lokalny, jaki jest mnemonik tego adresu IP.
- 8)Co to jest serwer PHP?

Zadanie2

Temat: Sprawdzanie czy działa serwer PHP.

Wykonaj:

1. Uruchom Webserv (krasnala)
2. Wpisz w okienku adres przeglądarki adres <http://localhost/> a następnie `http://127.0.0.1`
3. Napisz w części opisowej (sprawozdaniu), co oznacza i co można sprawdzić poprzez wpisanie takiego adresu.

Zadanie3

Temat Załóż konto o nazwie z3 dla serwera PHP

Rozwiązanie

- a)dla Webserv w folderze **httpd-users** załóż folder o nazwie z3
- b)(Dla krasnala→Kliknij prawym pióro Apache znajdujące się na pasku zadań następnie Ustawienia→Konta)

Zadanie4

Temat: Budowa skryptu w PHP. Komentarze w PHP. Uruchamianie skryptu.

Wykonaj:

- 1)Zapisz teorię do zeszytu/pracy
 - Budowa skrypt w PHP
 - Komentarze w PHP
- 2)Załącz konto na serwerze PHP o nazwie Z4
- 3)Na to konto nagraj plik PHP o nazwie `Z4_nazwisko_ucznia.php`, który będzie zawierał komentarz jednoliniowy, w tym komentarzu nazwisko ucznia oraz komentarz wieloliniowy, który będzie zawierał dzisiejszą datę.
- 4)Uruchom skrypt.

Teoria:

Budowa skrypt w PHP.

```
<?php treść skryptu?> lub trochę inaczej
```

```
<?php  
Instrukcje języka PHP  
?>
```

Komentarze w PHP

```
// komentarz jednolinijkowy (sposób pierwszy)  
# komentarz jednolinijkowy (sposób drugi)
```

```
/*  
Komentarz  
Wielolinijkowy  
*/
```

Zadanie5

Temat: Sprawdzanie konfiguracji serwera PHP z użyciem skrypt PHP

1. Wpisać następując treść zadania w nowym pliku zapisanym w notatniku.

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

nazwa pliku **index.php** (opcje nagrywania: wszystkie pliki, kodowanie UTF-8) nagrany w folderze **z5**

-uruchom stronę poprzez (takie uruchomienie powoduje, że nie trzeba będzie wpisywać nazwy pliku startowego czyli index.php)

http://localhost/~nazwakonta/
~nazwakonta to z5

2. Umieść listing programu w pracy sprawozdaniu.

Instrukcja echo

Instrukcja, która wyświetla napis :
echo("treść napisu");

instrukcja, która wyświetla napis i przechodzi do nowego wiersza
echo("treść napisu < br>");

Instrukcja wyświetli wynik działania sumowania liczb 2 i 3.
echo(2+3);

Dodatkowe spacje na ekranie

poprzez

Instrukcja print

a) łączenie tekstów → użycie kropki

Znaczenie znaku . (kropka)

Kropka sumuje ciągi znaków.

Np.

```
print("<b>Grafika:</b> " . "php.gif <br>");
```

instrukcja wyświetli pogrubione słowo Grafika oraz doda tekst php.gif i przejdzie do nowej linii.

b) użycie instrukcji print/echo do wykonania instrukcji HTML na przykładzie wyświetlania grafiki

```
print("<img src=\"obrazek.gif\">");
```

znaczenie znaku \" (ukośnik i cudzysłów)

w HTML było by:

```

```

a w PHP

```
print("<img src=\"obrazek.gif\">");
```

czyli

jeżeli przed cudzysłowem występuje \ to interpreter PHP traktuje cudzysłów jako znak sterujący lub element składni. Gdy by nie było znaku \ to cudzysłów traktowany będzie jako znak cudzysłowu.

c) użycie instrukcji print/echo do wykonania instrukcji CSS na przykładzie formatowania tekstu

```
print("<p style=\"font-size: 40px; color: black;\"><br>Dziki melanz<br></p>");
```

Zadanie6

Temat: Wykonanie wizytówki.

Zapisz w zeszycie/części opisowej: (przepisz pytanie i udziel odpowiedzi)

- Instrukcję wyświetlającą tekst (patrz wyżej, dokładny opis z przykładami).
- Jak uzyskujemy dodatkową spację.

Napisz program, który wyświetli twoje dane w formie wizytówki z ramką (ze znaków dostępnych z klawiatury) wizytówka zawiera: nazwisko imię, klasa, fałszywy adres oraz fałszywy telefon całość otoczona znakami z klawiatury.

Nazwij wizytowka_nazwisko_sluchacza.php np. wizytowka_kowalski.php.

Uruchom go poprzez `http://127.0.0.1/~z6/wizytowka_nazwisko_sluchacza.do` odświeżania po poprawkach użyj klawiszy `ctrl+F5`.

Umieść listing programu w pracy końcowej (sprawozdaniu) oraz wynik działania programu. Jest to dokument Worda pod nazwą Twoje nazwisko.

Koniec zadania 6

Uwaga:

Do testowania programów możesz użyć następującej strony:

<http://writecodeonline.com/php/>

Zadanie6a

Napisz program:

- wyświetlający, co najmniej cztery pliki graficzne z opisami odpowiadającymi grafice. Grafiki umieść w ramkach o szerokościach równych liczba_liter_imienia. Wielkość grafik zdefiniowana poprzez parametry:
height="200+dzień urodzenia "
width="200+miesiąc urodzenia "
Opis umieść pod grafiką. Całość ma tworzyć śmieszna przyzwoita historyjkę. Pliki pobierz z Internetu. Opisy wymyśl sam. Jak umieścić grafiki patrz wyżej odnośnie instrukcji **print**.

rozmieszczenie grafik oraz tekstów im odpowiadających jest następujące:















Obliczenie	Który układ
(Numer_w_grupie) mod 4=1	Układ1
(Numer_w_grupie) mod 4=2	Układ2
(Numer_w_grupie) mod 4=0	Układ3
(Numer_w_grupie) mod 4=3	Układ4

Układ1		
Grafika1+opis (pod grafiką)		
	Grafika2+opis (pod grafiką)	
		Grafika3+opis (pod grafiką)
	Grafika4+opis (pod grafiką)	
Układ2		
	Grafika1+opis (pod grafiką)	
Grafika2+opis (pod grafiką)		
	Grafika3+opis (pod grafiką)	
		Grafika4+opis (pod grafiką)
Układ3		
		Grafika1+opis (pod grafiką)
	Grafika2+opis (pod grafiką)	
Grafika3+opis (pod grafiką)		
		Grafika4+opis (pod grafiką)
Układ4		
	Grafika1+opis (pod grafiką)	
Grafika2+opis (pod grafiką)		
		Grafika3+opis (pod grafiką)
	Grafika4+opis (pod grafiką)	

Do centrowania grafik użyj <center>.....</center>

Do wyrównywania użyj align=left/right

- zastosuj kolor tła dla całej strony-numer koloru to numer z tabeli zaliczeniowej

Nazwa	HEX	Kolor
1) silver	#C0C0C0	
2) gray	#808080	
3) maroon	#800000	
4) red	#FF0000	
5) purple	#800080	
6) fuchsia	#FF00FF	
7) green	#008000	
8) lime	#00FF00	
9) olive	#808000	
10) yellow	#FFFF00	
11) navy	#000080	
12) blue	#0000FF	
13) teal	#008080	
14) aqua	#00FFFF	

- zastosuj cztery kolory czcionek oraz cztery różne ich wielkości oraz cztery rodzaje czcionek (możesz stosować instrukcję CSS).

koniec zadania 6a.

Zmienne w PHP

Każdą **zmienną** w PHP zapisuje się, poprzedzając jej nazwę znakiem dolara "\$". Wielkość liter w nazwie zmiennej jest rozróżniana.

Nazw zmiennych dotyczą te same reguły, co innych rodzajów nazw w PHP. Poprawna nazwa zmiennej zaczyna się od litery lub znaku podkreślenia "_", po których może wystąpić dowolna ilość liter, cyfr lub znaków podkreślenia. Jako wyrażenie regularne, można to zapisać tak: '[a-zA-Z_\x7f-\xff][a-zA-Z0-9_\x7f-\xff]*'

Informacja: W naszym rozumieniu, litery to znaki a-z, A-Z i bajty od 127 do 255 (0x7f-0xff).

Informacja: \$this to specjalna zmienna, której nie można definiować.

Zadanie7

Temat: Wpisz i przetestuj następujący skrypt, obliczający sumę dwóch liczb.

Zapisz skrypt o nazwie nazwisko_słuchacza_suma.php np. kowalski_suma.php i listing wstaw do pracy oraz dołącz plik z tym skryptem do pracy.

Wpisz treść przykładu zmieniając nazwy zmiennych np. \$a na \$_cztery_litery_nazwiska \$a_kowa

Umieść listing programu w pracy końcowej (sprawozdaniu)

```
<?php
echo("obliczanie sumy dwóch zmiennych a=1 b=3 <br>");
$a=1;
$b=3;
$c=$a+$b;
echo("c=");
echo($c);
?>
```

Zadanie8

Temat: Napisz program obliczający sumę odwrotności kwadratów liczb x1 i x2
wynik zapisz w zmiennej suma_odrotności nie zapomnij o znaku \$

Zapisz skrypt o nazwie nazwisko_słuchacza_odw.php np. kowalski_odw.php i listing wstaw do pracy oraz dołącz plik z tym skryptem do pracy.

Zmienne środowiskowe

Dają dostęp do rozmaitych danych dostarczanych przez przeglądarkę gościa wizytującego nasz serwis, a także przekazują wiele danych o parametrach samego serwera.

Zmienne środowiskowe dostępne są poprzez superglobalną tablicę \$_SERVER, która może być dostępna w każdym skrypcie.

Niektóre ze zmiennych środowiskowych:

\$_SERVER['SERVER_NAME']

Nazwa hosta (np. localhost)

\$_SERVER['SERVER_SOFTWARE']

Oprogramowanie serwera, np. Apache

\$_SERVER['SERVER_PROTOCOL']

Protokół, jakim jest przesyłana strona

\$_SERVER['DOCUMENT_ROOT']

Katalog główny, w którym działa skrypt

\$_SERVER['HTTP_REFERER']

Strona oglądana jako poprzednia, z której przeniosłeś się tutaj

\$_SERVER['HTTP_USER_AGENT']

Twoja przeglądarka

\$_SERVER['REMOTE_ADDR']

Twój numer IP (Internet Protocol)

\$_SERVER['SERVER_ADMIN']

Adres e-mail administratora serwera
\$_SERVER['SERVER_SIGNATURE']
"Podpis" serwera, który jest wyświetlany na stronach generowanych przez serwer (np. niektóre komunikaty błędów)

Zadanie9

Temat: Skrypt podaje pewne informacje korzystając ze zmiennych systemowych.

- Uruchom następujący skrypt.
- Zapisz skrypt o nazwie nazwisko_słuchacza_zmienne.php np. kowalski_zmienne.php i listing wstaw do pracy oraz dołącz plik z tym skryptem do pracy

```
<?php
echo("twoje IP= ");
echo($_SERVER["REMOTE_ADDR"]);
echo("<br>");
echo("twoje port= ");
echo($_SERVER["REMOTE_PORT"]);
echo("<br>");
echo("twoja przeglądarka= ");
echo($_SERVER["HTTP_USER_AGENT"]);
?>
```

Zadanie10

Temat: Wczytywanie danych w formularzu z użyciem PHP

- Wpisz następujący skrypt z użyciem notatnika.
- Nazwij go nazwisko_słuchacza_okienko. kowalski_okienko (będziesz musiał zmienić nazwę skryptu po *action*)
- Uruchom program.
- Umieść listing przykładu w pracy końcowej (sprawozdaniu).
- Zmienne lic1 oraz lic2 powinny mieć być tak zmienione, aby miały cztery litery Twojego nazwiska

```
<form action="index.php" method="get">
1: <input name=lic1> <br>
2: <input name=lic2> <br>
<input type=submit value=ok>
</form><br>
<?php
echo $_GET[lic1]+$_GET[lic2];
?>
```

Zadanie11

Temat: Napisz skrypt obliczający sumę odwrotności kwadratów dwóch liczb powiększonych o jeden posługując się poprzednim skryptem,

Wskazówka do obliczenia.

Przed obliczeniami zwiększ wartości zmiennych o jeden. Wpisywanie danych(liczb) w formularzu. Nazwy zmiennych muszą zawierać w swojej nazwie Twoje nazwisko np. \$x1_kowalski. W poprzednim zmienne miały nazwy lic1 i lic2. Przycisk do wprowadzania danych powinien mieć OK-twoje_nazwisko a zmienne lic1 i lic2 cztery litery nazwiska.

Uruchom skrypt.

Umieść listing programu w pracy końcowej (sprawozdaniu)

Koniec zadania 11

Instrukcja **if else**

```
<?php
    $a = 9;          // przypisujemy wartość zmiennej $a
    if ($a%2 > 0) // reszta z dzielenia przez 2
    {
        echo "Liczba nieparzysta";
    }
    else // brak reszty z dzielenia przez 2
    {
        echo "Liczba parzysta";
    }
?>
```

Zadanie12

Temat: Wpisz skrypt do symulacji rzutu monetą.

Nagraj ten plik pod nazwą gen_nazwisko_słuchacza.php np. gen_kowalski.php. Uruchom go poprzez `http://127.0.0.1/~twoje_inicjały/_nazwisko_słuchacza.php` do odświeżania po poprawkach użyj klawiszy `ctrl+F5`.

Zmienne powinny mieć nazwy z czterema literami Twojego nazwiska.

Zapisz w części opisowej: (przepisz pytanie i udziel odpowiedzi)

- sposób tworzenia komentarza jednoliniowego
- sposób tworzenia komentarza wieloliniowego
- inicjowania generatora
- używanie generatora
- instrukcja `if else`

```
<?php
//inicjalizacja generatora liczb losowych
//na podstawie aktualnego czasu w milisekundach
srand((double)microtime()*1000000);
$wypadlo=rand(1,2); #wylosuj liczby naturalne od 1 do 2
/*
sprawdzenie wyniku
przyjmujemy arbitralnie
1 to orzeł
2 to reszka
```



```
*/  
  
if ($wypadlo==1){echo('wypadł orzeł');}  
    else  
        {echo('wypadła reszka');}  
?>
```

Koniec zadanie 12

Instrukcja **switch case**

```
<?php  
$zmienna = 'banan';  
switch ($zmienna) {  
    case "apple":                // sprawdza czy $zmienna == 'apple' itd  
        echo "to jest jabłko";  
        break;  
    case "banan":  
        echo "to jest banan";  
        break;  
    case "wiśnia":  
        echo "to jest wiśnia";  
        break;  
}  
?>
```

Zadanie13

Napisz skrypt do symulacji rzutu kostką do gry. Nagraj ten plik pod nazwą `kostka_nazwisko_sluchacza.php` np. `kostka_kowalski.php`. Uruchom go poprzez `http://127.0.0.1/~twoje_konto/kostka_nazwisko_sluchacza.php` do odświeżania po poprawkach użyj klawiszy `ctrl+F5`. Nazwy zmiennych muszą zawierać w swojej nazwie Twoje nazwisko np. `$wypadlo_kowalski`.

Składnia instrukcji `rand(min,max)` np. `rand(20,50)` losowanie liczb naturalnych z przedziału `<20,50>` czyli 20,21,23,.....,49,50

Do rozpatrzenia przypadku ile oczek wypadło użyj **case**.

Napis na ekranie np. Wypadły trzy oczka

Umieść listing programu w pracy końcowej (sprawozdaniu)

Pętla for

```
<?php  
for($i=0;$i<10;$i++)  
{  
    // instrukcje do wykonania  
    // z każdą iteracją  
}  
?>
```

Pierwszym elementem w nawiasie jest przypisanie zmiennej iteracyjnej początkowej wartości. Drugim jest warunek końcowy. Trzecim wskazujemy, jak ma przebiegać zmiana wartości zmiennej, zazwyczaj jest to inkrementacja lub dekrementacja.

Zadanie14

Temat: Wpisz skrypt symulacji 26 rzutów kostką do gry. Wykonaj wydruki w następujący sposób:

Rzut 1: wypadła szóstka //pogrubienie

Rzut 2: wypadła czwórka

.....

Nagraj ten plik pod nazwą kostka_nazwisko_słuchacza_26.php np. kostka_kowalski_26.php. Uruchom go poprzez `http://127.0.0.1/~twoje_inicjały/kostka_nazwisko_słuchacza_26.php` do odświeżania po poprawkach użyj klawiszy `ctrl+F5`.

Umieść listing w pracy.

Odszukaj w Internecie i zapisz w sprawozdani:

-sposób tworzenia pętli w PHP, opis słowny oraz przykłady zastosowania (co najmniej trzy).

```
<?php
srand((double)microtime()*1000000);
for ($i=1;$i<=26;$i++)
{
    $wypadlo=rand(1,6);
    echo("rzut nr $i: ");
    echo("<b>$wypadlo</b><br>");
}

?>
```

Zadanie14a

Wpisz skrypt symulacji 1000000 rzutów moneta. Wyświetl tylko ile razy wylosowana była reszka oraz orzeł. Nagraj ten plik pod nazwą moneta_milion_nazwisko_słuchacza_26.php np. moneta_milion_kowalski_26.php. Uruchom go poprzez

`http://127.0.0.1/~twoje_inicjały/moneta_milion_nazwisko_słuchacza_26.php` do odświeżania po poprawkach użyj klawiszy `ctrl+F5`.

Umieść listing w pracy.

Zadanie15

Napisz skrypt symulacji 20+numer_z_dziennika rzutów monetą. Nagraj ten plik pod nazwą moneta_nazwisko_słuchacza_rzuty.php np. moneta_kowalski_rzuty.php. Uruchom go poprzez `http://127.0.0.1/~twoje_inicjały/moneta_nazwisko_słuchacza_rzuty.php` do odświeżania po poprawkach użyj klawiszy `ctrl+F5`.

Skryp ten umieść na płycie (nośniku) po nazwą zad6_twoje_nazwisko.php oraz umieść listing w opisie.

Nazwy zmiennych muszą zawierać w swojej nazwie Twoje nazwisko np.

\$wypadlo_kowalski.

Wygląd ekranu

Nazwisko słuchacza

1 rzut to orzeł

2 rzut to reszka

.....

.....

Umieść listing przykładu w pracy końcowej (sprawozdaniu)

Definiowanie funkcji (podprogramy) w PHP

Funkcja ma **nazwę**. Zapisywana jest po słowie kluczowym `function`.

Nazwa funkcji ma takie same wymagania jak nazwy zmiennej czyli:

- litery (język angielski) i cyfry,
- znak podkreślenia,
- nie może zaczynać się od cyfry.

Argumenty funkcji.

Argumenty funkcji zapisywane są w nawiasach okrągłych () jest to lista nazw. Kolejne argumenty oddzielane są przecinkami. Funkcja może być bez argumentowa wtedy w nawiasach () nie umieszczamy nazw argumentów.

Ciało funkcji.

Kod funkcji nazywanej również ciałem funkcji umieszczana jest w nawiasach klamrowych { }.

return

Powoduje przerwanie działania funkcji i zwrócenie wartości przez funkcję dla zmiennej lub wyrażenia występującej po słowie **return**.

wartości domyślne funkcji

wartości domyślne mogą być na liście parametrów funkcji na ostatnie pozycji listy
`funkcja($szerokosc,$dlugosc=100)`

dyrektywa global

Umożliwia wykorzystanie zmiennej globalnej w podprogramie.

Zmienne lokalne i globalne

lokalne-definiowane wewnątrz funkcji i są widoczne tylko wewnątrz funkcji.

globalne- deklarowane wewnątrz programu głównego

Zadanie16a1

Temat: Definiowanie funkcji bezparametrowej w PHP.

Wykonaj:

Uruchom przykład dopisując pętlę for tak aby wywołać linię poziomą
ile_razy=10+(nazwisko reszta_z_dzielenia 3)

```
<?php
function linia()
{
    echo('<hr style="color:red;background:red;border:0px;height:1px;" />');
}
// tutaj pętla wywołująca linię wielokrotnie
?>
```

Zadanie16a2

Temat: Definiowanie funkcji z bezparametrowem w PHP.

Zmień poprzedni przykład tak, aby funkcja **linia(\$kolor)** miała parametr. Zamiast **red** w ciele funkcji zamień na **\$kolor**. Wywołaj funkcję **linia(\$kolor)** z pięcioma z różnymi parametrami tak, aby otrzymać 5 różnych funkcji.

Zadanie16a3

Temat: Definiowanie funkcji z bezparametrowem domyślnym w PHP.

Wykonaj:

zmień parametr domyślny na 50%+liczba_liter_nazwiska%

```
<?php
function linia($kolor,$szerokosc="100%")
{
    echo('<hr
style="color:'.$kolor.';background:'.$kolor.';border:0px;height:1px;width:'.$szerokosc.'" />');
}
linia('green');
linia('blue','50%');
?>
```

Zmień poprzedni przykład tak, aby funkcja **linia(\$kolor)** miała parametr.

Zadanie16

Temat: Definiowanie funkcji w PHP

Treść skryptu.

Wpisz skrypt wykonujący znajdowanie minimalnego elementu ze zbioru dwóch liczb.

Nagraj ten plik pod nazwą z_16_min_nazwisko_sluchacza.php np. min_kowalski.php.

Uruchom go poprzez http://127.0.0.1/~Z16/min_nazwisko_sluchacza.php do odświeżania po poprawkach użyj klawiszy ctrl+F5.

Nazwy funkcji musi być zmieniona na minimum_twoje_nazwisko(.....)

```
<?php
function minimum($a,$b)
{
```

```

    return 0.5*($a+$b-abs($a-$b));
}
echo('minimum(2,3)='.minimum(2,3).'

```

Zadanie16a

Temat: Definiowanie funkcji w PHP

Treść skryptu.

Wpisz skrypt wykonujący znajdowanie maksymalnego elementu ze zbioru dwóch liczb.

Wskazówka:

Zmień wzór w return ???????;

Nagraj ten plik pod nazwą z16a_max_nazwisko_słuchacza.php np. min_kowalski.php.
 Uruchom go poprzez http://127.0.0.1/~Z16a/max_nazwisko_słuchacza.php do odświeżania po poprawkach użyj klawiszy ctrl+F5.
 Nazwy funkcji musi być max_twoje_nazwisko(.....)

Zadanie17

Temat: Definiowanie funkcji w PHP

Treść skryptu.

Napisz skrypt obliczania pola powierzchni oraz objętości dowolnego prostopadłościanu..

Nagraj ten plik pod nazwą obj_nazwisko_słuchacza.php np. obj_kowalski.php. Uruchom go poprzez http://127.0.0.1/~twoje_inicjały/obj_nazwisko_słuchacza.php do odświeżania po poprawkach użyj klawiszy ctrl+F5.

Skryp ten zapisz pod nazwą zad8_twoje_nazwisko.php oraz umieść listing w opisie.

Nazwy zmiennych muszą zawierać w swojej nazwie Twoje nazwisko np.

\$wypadlo_kowalski.

Wygląd ekranu

Nazwisko słuchacza

V(10,20,5)=obliczona liczba(objętość)→1000 cm³

P(14,23,6)=obliczona liczba(pole_powierzchni)→1088 cm²

Zadanie17a

Rekurencja-zachodzi wtedy gdy jakaś funkcja wywołuje samą siebie.

Temat: Rekurencyjne obliczanie silni.

$$n! = \begin{cases} 1 & \text{dla } n = 0 \\ n(n-1)! & \text{dla } n > 0 \end{cases}$$

```
<?php
function silnia($n)
{
    if ($n==0) return 1;
        else
            return silnia($n-1)*$n;
}
$i=6;
echo($i.'!='.silnia($i));
?>
```

Zadanie17b

Temat: Definiowanie funkcji w PHP

Napisz program rekurencyjnego znajdowania wartości ciągu Fibonacciego.
Który element mamy obliczyć wczytujemy z klawiatury.

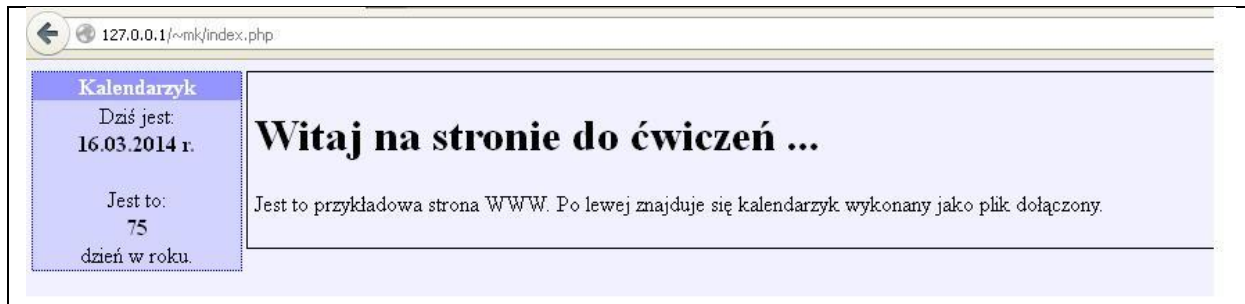
Przykładowy wygląd ekranu:

Podaj, który element ciągu Fibonaciego, który należy obliczyć=6
FIBO(6)=8

$$F_n = \begin{cases} 0 & \text{dla } n = 1; \\ 1 & \text{dla } n = 2; \\ F_{n-1} + F_{n-2} & \text{dla } n > 2. \end{cases}$$

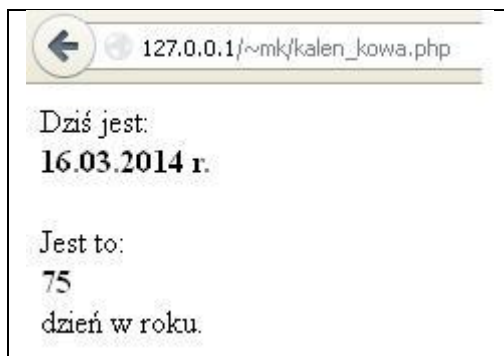
Zadanie18

Temat: Strona posiadająca kalendarz (patrz poniżej).



Etap 1

- Wykonaj skrypt php wypisujący dzisiejszą datę oraz ile dni upłynęło od początku roku.
- Będziesz musiał uzupełnić formaty daty dla miesięcy i roku (trzecia linia skryptu), jakie znaczniki zastosować patrz lista znaczników poniżej.
- Nagraj skrypt pod nazwą kalen_cztery_pierwsze_litery_nazwiska.php np. kalen_kowa.php
- Uruchom skrypt a otrzymasz następujący efekt:



```
<?php
// d->oznacza dzień w formacie dwie cyfry
echo('Dziś jest: <br /><b>'.date("d.tutaj_wpisz_miesiąc_dwie_cyfry.rok_4cyfry").' r.</b>');
    echo('<br /><br />');
    echo('Jest to: <br /><b>'.(date("z")+1));
    echo('</b><br /> dzień w roku.');
    echo('<br />');
?>
```

Funkcja date()

Oczekuje jednego argumentu. Argumentem jest ciąg znaków.

Lista znaczników dla funkcji date()

- d - Dzień miesiąca, 2 cyfry od 01 do 31
- D - Tekstowy opis angielskiej nazwy dnia, trzy litery
- j - Dzień miesiąca bez zer wiodących
- l (mała litera 'l') - Pełen angielski opis dnia tygodnia
- N - Liczbowa forma dnia tygodnia, zgodna z normą ISO-8601 (dodana w PHP 5.1.0)
- S - Angielski przyrostek porządkowy dla dnia miesiąca, 2 litery

w - Liczbowa forma dnia tygodnia
 z - Dzień roku (Zaczynając od 0)
 W - Numer tygodnia w roku, zgodny z normą ISO-8601, Tygodnie rozpoczynają Poniedziałki (dostępne od PHP 4.1.0)
 F - Pełen angielski opis, dnia miesiąca, taki jak January czy March
 m - Liczbowa forma miesiąca, od 01 do 12
 M - Krótki, angielski opis miesiąca, trzy litery
 n - Liczbowa forma miesiąca, bez zer wiodących
 t - Ilość dni w danym miesiącu
 L - Informacja o tym, czy rok jest przestępnym 1- przestępny 0-nie przestępny
 o - Numer roku, zgodny z normą ISO-8601. Zwraca to taką samą wartość jak Y, z takim wyjątkiem, że numer tygodnia ISO (W) należy do poprzedniego lub następnego roku, niż rok użyty w tym miejscu. (dodane w PHP 5.1.0)
 Y - Pełna liczbowa forma roku, 4 cyfry np. 2014
 y - Dwie cyfry reprezentujące rok
 a - Pora dnia - dwie małe litery (przed/po południu) (ang. Ante/Post meridiem)
 A - Pora dnia - dwie duże litery (przed/po południu) (ang. Ante/Post meridiem)
 g - Godzina, w formacie 12-godzinny, bez zer wiodących
 G - Godzina, w formacie 24-godzinny, bez zer wiodących
 h - Godzina, w formacie 12-godzinny, z zerami wiodącymi
 H - Godzina, w formacie 24-godzinny, z zerami wiodącymi
 i - Minuty z zerami wiodącymi
 s - Sekundy, z zerami wiodącymi
 e - Identyfikator strefy czasowej (dodano w PHP 5.1.0)
 I (duże i) - Informacja o tym, czy czas jest letni
 O - Różnica z czasem Greenwich (GMT) w godzinach
 P - Różnica z czasem Greenwich (GMT) z dwukropkiem pomiędzy godzinami i minutami (dodano w PHP 5.1.3)
 T - Skrót dla strefy czasowej
 Z - Różnica dla strefy czasowej w sekundach. Wyrównanie to jest zawsze ujemne dla stref położonych na zachód od południka 0, oraz dodatnie dla tych leżących na wschód od niego.
 c - Data w standardzie ISO 8601 (dodana w PHP 5)
 r - Data sformatowana zgodnie z RFC 2822
 U - Sekundy liczone od ery UNIX-a (1 stycznia 1970 00:00:00 czasu Greenwich - GMT)

Etap2) Wykonanie strony index.php na, której osadzony będzie plik kalen_kowa.php z użyciem instrukcji include();

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pl">
<head>
  <meta http-equiv="Content-type" content="text/html; charset=iso-8859-2" />
  <title>Kalendarz</title>
</head>
<body>
  <?php include('kalen_kowa.php'); ?>
</body>
</html>
  
```

Etap3) Uzupełnienie strony index.php na o style

```
.....</title>
<style type="text/css">
    body { następujący kolor tła f1f1ff;}
    #menu { jaka definicja stylu patrz tabela }
    #main { jaka definicja stylu patrz tabela}
    #kalendarz { jaka definicja stylu patrz tabela}
</style>
```

Nazwa ID	Co realizuje
#menu	{szerokość:150 pikseli; obramowanie:1 piksel w postaci kropek niebieskich; ułożenie:na lewo; kolor tła:#d3d3ff; rozmieszczenie tekstu:centrowany;}
#main	{margines lewy:155pikseli; obramowanie:1 piksel obramowanie w postaci linii; margines zewnętrzny z czterech stron: 5 pikseli;}
#kalendarz	{ rozmieszczenie tekstu:centrowany; kolor tła:#9595ff; kolor liter: biały; grubość czcionki: pogrubiona;}

Etap4)

Uzupełnij o następujące ID

```
.....
</head>
<body>
    <div id="menu">
        <div id="kalendarz">
            Kalendarzyk
        </div>
        <?php include('kalen_kowl.php'); ?>
    </div>
    <div id="main">
        <h1>Witaj na mojej stronie...</h1>
        <p> Przykładowa strona WWW. Po lewej znajduje się kalendarzyk – nasz
            pierwszy skrypt PHP.</p>
    </div>
</body>
</html>
```

Zadanie19

Temat: Modyfikacja strony poprzedniej tak aby strona wyświetlała aktualne imieniny.

Tablice:

Teoria

Kalendarzyk Dziś jest: 04.04.2014 r. Jest to: 94 dzień w roku. Imieniny: Benedykta, Izodory, Wacławy	Witaj na mojej stronie... Przykładowa strona WWW. Po lewej znajduje się kalendarzyk - nasz pierwszy skrypt PHP.
---	---

Etap1

Wykonanie pliku PHP zawierającego tablicę w której będą zapisane aktualne imieniny.

```
<?php
$imiona[0]='Masława, Mieczysława, Mieszka';
$imiona[1]='Bazylego, Makarego, Narcyzy';
$imiona[2]='Arlety, Danuty, Lucjana';
.....tablica imion jest na końcu instrukcji dokonaj kopiowania.....
$imiona[363]='Eugeniusza, Katarzyny, Sabiny';
$imiona[364]='Mariusza, Melanii, Sylwestra';
?>
```

Nazwij go: imieniny_trzy_pierwsze_nazwiska.php np. imieniny_kow.php

Etap2

Do pliku imieniny_trzy_pierwsze_nazwiska.php np. imieniny_kow.php dołącz algorytm określania czy rok jest przestępny czy nie. Patrz schemat blokowy poniżej tych opisów. Jest to konieczne ponieważ luty ma 28 lub 29 dni. Do określania czy rok jest przestępny

użyj funkcję:

```
=date("L");
```

gdzie:

L - Informacja o tym, czy rok jest przestępnym 1- przestępny 0-nie przestępny

Do określenia jaki jest dzień roku użyj funkcji:

```
=date("z");
```

gdzie:

z - Dzień roku (Zaczynając od 0)

Opis schematu blokowego poniżej:

58 - to 28 lutego; 59 - to 29 lutego; 60 - to 1 marca;

do 58 dnia (28 lutego) wszystko jest OK

gdy jednak dzień jest > 58 to...

jeżeli rok NIE jest przestępny, to luty ma 28 dni i musimy

przesunąć się o jeden do PRZODU (przeskoczyć ten 29)

gdy jest PRZESTĘPNY, czyli luty ma 29 dni przeskoku nie ma

Etap3

Uzupełnij plik kalen_kowa.php o wyświetlanie imienin

```
<?php
```

```
// d->oznacza dzień w formacie dwie cyfry
```

```
echo('Dziś jest: <br /><b>'.date("d.tutaj_wpisz_miesiąc_dwie_cyfry.rok_4cyfry").' r.</b>');
```

```
    echo('<br /><br />');
```

```
    echo('Jest to: <br /><b>'.(date("z")+1));
```

```
    echo('</b><br /> dzień w roku.');
```

```
    echo('<br />');
```

```
    echo('Imieniny:<br />');
```

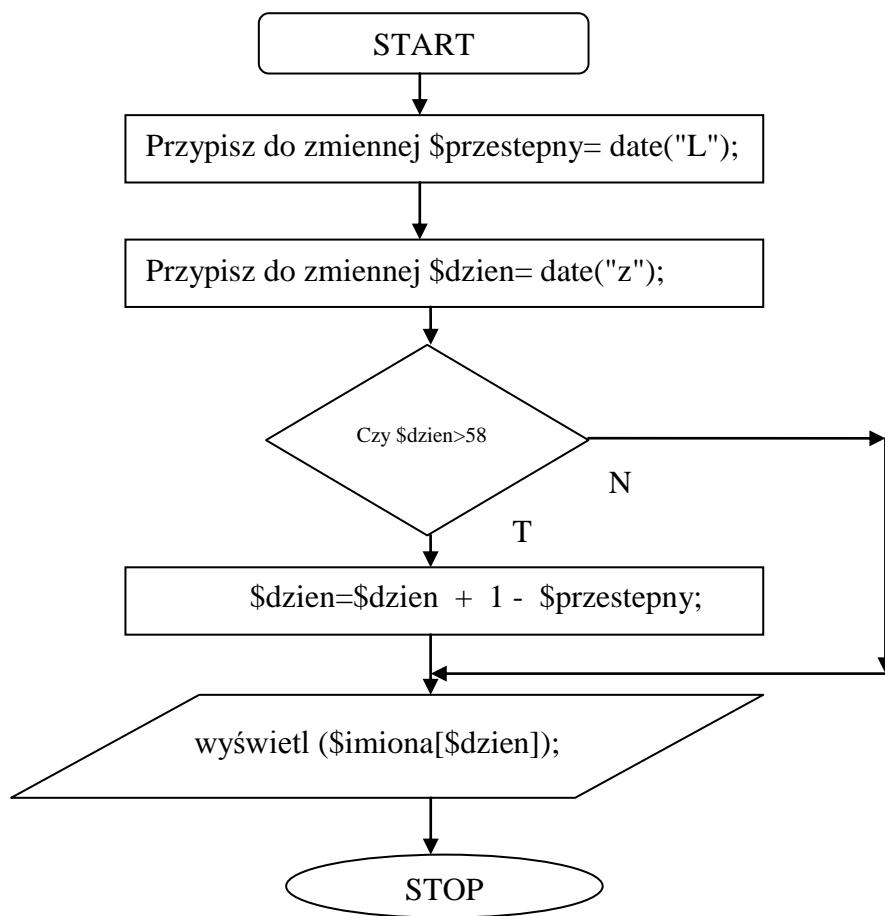
```
    echo('<b>');
```

```
    include('imieniny_kow.php');
```

```
    echo('</b>');
```

```
    echo('<br /><br />');
```

```
?>
```



Zadanie 19a

Temat: Użycie tablic asocjacyjnych jednowymiarowych do wyszukiwania numeru telefonu na podstawie imienia i nazwiska.

Tablice asocjacyjne są to tablice, w których zamiast indeksów liczbowych używa się identyfikatorów znakowych (kluczy):

isset(\$zmienna) - zwraca prawdę, jeżeli zmienna istnieje.

empty(\$zmienna) - zwraca prawdę, jeżeli zmienna ma wartość pustą (np. NULL, 0 albo pusty ciąg tekstowy).

Wykonaj:

a)uruchom przykład poniższy:

```
<?php
    $tel['Jan Nowak']='601-123-456';
    $tel['Piotr Kowalski']='602-321-654';
    $tel['Joanna Kwiatkowska']='603-456-321';
    $osoba='Jan Nowak';
    if (isset($tel[$osoba])) echo("$osoba : Numer telefonu: $tel[$osoba]");
    else
    echo('<span style="color:red;">Brak osoby w bazie!</span>');
?>
```

b)dopisz siebie do bazy

c)zmień treść zadania tak aby pytanie o telefon następowało w formularzu

- imię i nazwisko osobno,
- pola do wpisywania równe
- napis **Brak takiej osoby w bazie!** nie ukazywał się przed wpisaniem danych.

Piotr Kowalski : Numer telefonu: 602-321-654

Imię:

Nazwisko:

Zadanie 19b

Temat: Użycie tablic asocjacyjnych dwuwymiarowych do wyszukiwania numeru telefonu na podstawie imienia i nazwiska.

```
<?php
$tel['Jan Nowak']['telefon']='601-123-456';
$tel['Jan Nowak']['adres']='ul. Piękna 12';
$tel['Piotr Kowalski']['telefon']='602-321-654';
$tel['Piotr Kowalski']['adres']='ul. Światowa 13a';
$tel['Joanna Kwiatkowska']['telefon']='603-456-321';
$tel['Joanna Kwiatkowska']['adres']='ul. Maniaka PHP 1/3';
.....
?>
```

Wykonaj:

- Uzupełnij twoje dane w tablicy dwuwymiarowej (nazwisko i imię prawdziwe reszta fałszywe ale prawdopodobne).
- pola do wpisywania rozmieszczone równo,
- napis **Brak takiej osoby w bazie!** nie ukazywał się przed wpisaniem danych,
- wypisz dane wg przykładu (poniżej),

Jan Nowak : Numer tel.: 601-123-456 Adres: ul. Piękna 12
--

Zadanie 19c

Temat: Tworzenie tablic z użyciem funkcji **array**. Modyfikacja zadania 19a.

Wykonaj:

Zmień Zadanie19a tak, aby wykorzystać definiowanie tablicy wg opisu 2

Opis 1

definiowanie tablicy jednowymiarowej:

```
$t=array("Iwa", "Kasia ", "Basia ");
```

tablica zawiera trzy imiona o indeksach 0 1 2

Opis 2

definiowanie tablicy gdzie **indeksami** są *imiona* a **wartościami** *nazwiska*:

```
$t=array("Iwa"=>"Kowalska", "Kasia "=>"Nowak", "Basia "=>"Kwiatkowska");
```

Zadanie 19d1

Temat: Użycie tablicy do wylosowania motta określonej wielkości tablicy.

Instrukcja Losowa

```
rand(dolna_granica,gorna_granica)
```

losowanie liczb całkowitych z przedziału <dolna_granica,gorna_granica>

Dopisz jedno przysłowie i zwiększ zakres losowania.

<pre><?php \$motto[0]='Trzeba wiedzieć co się mówi, a nie mówić co się wie.'; \$motto[1]='Lepszy straszny koniec niż strach bez końca.'; \$motto[2]='Nie wystarczy mówić do rzeczy, trzeba mówić do ludzi.'; \$motto[3]='Znamy tysiące sposobów zabijania czasu, jednak nikt nie wie jak go wskrzesić.'; \$los=rand(0,3); echo(\$motto[\$los]); ?></pre>

Zadanie 19d2

Temat: Użycie tablicy do wylosowania motta dowolnej wielkości tablicy.

funkcja count

funkcja **count(\$nazwa_tablicy)**. Zwraca ona ilość elementów w tablicy podanej jako parametr.

Przykład2

```
<?php
$motto[]='Trzeba wiedzieć co się mówi, a nie mówić co się wie.';
$motto[]='Lepszy straszny koniec niż strach bez końca.';
$motto[]='Nie wystarczy mówić do rzeczy, trzeba mówić do ludzi.';
$motto[]='Znamy tysiące sposobów zabijania czasu, jednak nikt nie wie,
jak go wskrzesić.';

$los=rand(0,count($motto)-1);
echo('<i>'.$motto[$los].'/i>');
?>
```

Zadanie 19d3

Temat: Losowanie hasła xx-znakowego dla znaków ASCII od 65 do 90.

Gdzie:

xx=8+miesiąc urodzenia

Potrzebne funkcje:

strlen(napis)	Zwraca długość napisu. Gdzie napis jest argumentem tej funkcji
chr(liczba)	Zwraca znak (liczby, litery, @#) wg kodów ASCII po podaniu numeru <0,255>

Lista kroków losowania hasła xx znakowego.

KROK1 : przypisz zmiennej \$los_kow pusty znak ciąg znaków

KROK2 : powtarzaj losowanie znaków ASCII od 65 do 90 i łączenie w jeden ciąg

\$los_kow tak długo dopóki ciąg \$los_kow jest mniejszy od 10

KROK3: wypisz losowy dziesięcioznakowy ciąg znaków

Zadanie 19d4

Temat: Losowanie xx liczb zapisanych do tablicy i znalezienie minimalnego elementu.

Gdzie:

$xx = 12 + \text{miesiąc urodzenia}$

Wykonaj:

1)Uruchom przykład dla xx, zmień nazwę tablicy na trzy pierwsze litery

Twoje_nazwiska.

2)Dopisz część programu obliczającego rozpiętość tablicy (jest to różnica między elementem maksymalnym a minimalnym).

Potrzebne funkcje:

shuffle(tablica)→losowo rozmieszcza elementy istniejącej tablicy

array_slice(tablica, indeks_pierwszego_elementu, indeks_ostatniego_elementu)

→kopiowanie (wycinanie) elementów tablicy od pewnego indexu do pewnego indexu.

print_r(tablica)→eleganckie wyświetlanie tablic

```
<?php
$t=range(1,100);
shuffle($t);
$t=array_slice($t,0,10);
$min=101;
for($i=0;$i<count($t);$i++)
{
    if ($t[$i]<$min)
    {
        $min=$t[$i];
    }
}

echo('<pre>');
echo("&<b>Tablica: </b>\n");
print_r($t);
echo("\nMinimum: <b>$min</b>");
echo('</pre>');
?>
```

Sortowanie tablic

PHP oferuje cały zestaw funkcji służących do sortowania tablic. Są to:

- asort()
- arsort()
- ksort()
- rsort()
- sort()

uasort()
usort()
uksort()

Większość funkcji (oprócz trzech ostatnich) przyjmuje jeden parametr: zmienną zawierającą tablicę do posortowania. Żadna z funkcji nie zwraca żadnego wyniku.

asort() sortuje tablice asocjacyjne zachowując przypisanie kluczy do wartości

```
<?php
$owoce = array ("d"=>"mango", "a"=>"papaja", "b"=>"banan", "c"=>"aronia");
asort ($owoce);
reset ($owoce); // Funkcja ta powoduje powrót do pierwszego elementu tablicy
while (list ($klucz, $wartosc) = each ($owoce)) {
    echo "$klucz = $wartosc";
}
?>
```

arsort() sortuje w odwrotnej kolejności tablice asocjacyjne zachowując przypisanie kluczy do wartości. Funkcja prawie identyczna jak poprzednia, tyle że dane sortowane są “od tyłu”.

ksort() sortuje tablice asocjacyjne według kluczy. Powyższy przykład po podmianie funkcji asort na ksort powinna dać poniższy wynik:

```
a = papaja
b = banan
c = aronia
d = mango
```

rsort() sortuje zwykłe tablice (nie asocjacyjne) w odwróconej kolejności

sort() sortuje zwykłe tablice (nie asocjacyjne) w kolejności alfabetycznej

uasort() funkcja sortująca tablice asocjacyjne za pomocą zdefiniowanej przez użytkownika funkcji porównującej elementy (nazwa funkcji jest podawana za pomocą drugiego parametru)

usort() funkcja sortująca zwykłe tablice za pomocą funkcji zdefiniowanej przez użytkownika

uksort() funkcja sortująca tablice asocjacyjne według klucza za pomocą funkcji zdefiniowanej przez użytkownika.

W trzech ostatnich funkcjach sortujących trzeba jako drugi parametr podać funkcję porównującą elementy tablicy. Jak definiuje się funkcje opisane jest w jednym z następnych rozdziałów. Funkcje takie pobierają 2 argumenty. Zwracane jest 0 jeśli argumenty są sobie równe, -1 jeśli pierwszy argument jest mniejszy od drugiego a 1 jeśli jest większy. Tworzenie ciągów z tablic i odwrotnie

zamianę ciągów na tablice i odwrotnie.

Zamiana ciągu na tablicę jest bardzo przydatna jeśli zachodzi potrzeba wyciągnięcia jakiegoś fragmentu danych z ciągu.

Założmy że odczytaliśmy z pliku z danymi linię z logu zapisanego przez licznik WWW: "12/11/2000;19:23:33;Netscape Navigator;192.168.1.1". Jak widać dane rozdzielone są średnikami.

Do rozdzielania ciągów na tablicę służy funkcja **explode()**. Jako

pierwszy parametr trzeba do niej podać znak lub dłuższy ciąg który oddziela kolejne pola, **drugi** ciąg do rozdzielania.

Opcjonalnie można podać **trzeci** argument, który oznacza maksymalną liczbę pól – jeśli jest ich więcej niż ta liczba, to ostatnie pole będzie zawierało wszystkie pozostałe pola.

Funkcja zwraca tablicę zawierającą kolejne pola.

Przykład

Rozdzielenie danych separowanych średnikiem

```
<?php
$dane = "12/11/2000;19:23:33;Netscape Navigator;192.168.1.1";
$tablica = explode(";", $dane);
?>
```

Rozszerzona wersja funkcji explode: split().

Różni się ona tym, że zamiast prostego ciągu znaków rozdzielających pola, akceptuje ona wyrażenia regularne.

Złączenie pól tablicy w jeden ciąg, w którym pola oddzielone są jakimś znakiem (lub kilkoma).

Do tego służy funkcja implode(). Jako pierwszy parametr podawany jest ciąg za pomocą którego "sklejane" są elementy tablicy, a jako drugi właśnie tablica do posklejania. Zwracany jest ciąg zawierający posklejane elementy. Jako przykład zastosowania może posłużyć właśnie zapisywanie danych o użytkowniku w aplikacji licznika odwiedzin – tablica zawiera dane o odwiedzającym, a potrzebny jest ciąg pooddzielany średnikami.

Łączenie danych z tablicy

```
<?php
$dane = implode(";", $tablica);
?>
```

chr(number)

Funkcja zwraca znak odpowiadający liczbie "number" wg. tablicy ASCII.

```
$a=chr(100);
```

//\$a przyjmie wartość d

crypt (string string [, wzorzec])

Funkcja zwraca ciąg zakodowany przy użyciu standardowego Unix'owego algorytmu DES lub innego dostępnego w systemie.

```
$haslo = crypt('MojeHaslo'); //wygenerowane w ten sposób hasło możemy trzymać np. w bazie danych.
```

/*sprawdzanie poprawności hasła. Wcześniej wygenerowane hasło posłuży jako wzorzec*/

```
if (crypt($haslo_uzytkownika, $haslo) == $haslo){
    echo "Hasło jest poprawne";
}
```

//\$haslo_uzytkownika jest hasłem podanym np. w formularzu logowania
explode(separator, string)
Funkcja rozбивa ciąg "string" w miejscu wystąpienia "separatora" i zwraca tablicę złożoną z poszczególnych części ciągu.

```
$a="janek:kasia:basia:stasio";  
$b=explode(":", $a);  
echo $b[0];  
//janek  
echo $b[1];  
//basia  
echo $b[2];  
//kasia  
echo $b[3];  
//stasio
```

levenshtein(string1, string2)
Funkcja zwraca liczbę operacji (dodania, usunięcia, zmiany znaku) potrzebnych do przekształcenia ciągu "string2" do postaci ciągu "string1".

```
$a="stefan";  
$b="stefcio"  
$c=levenshtein($a, $b);  
//$c przyjmie wartość 3 bo:  
//1: zmiana długości ciągu $b do długości ciągu $a  
//2: zmiana litery "c" w ciągu $b na literę "a"  
//3: zmiana litery "i" w ciągu $b na literę "n"  
Więcej o algorytmie i funkcji Levenshteina można przeczytać na:  
http://www.merriampark.com/ld.htm.  
nl2br(string)  
Funkcja wstawia przed znakami nowej linii znaczniki <br /> i zwraca nowy ciąg.
```

```
$a="Moj pierwszy\ntekst";  
$echo nl2br($a);  
Zwróci:  
Moj pierwszy <br />  
tekst
```

W wersjach wcześniejszych niż 4.0.5 funkcja zwraca ciąg ze znacznikami
.
ord(string)
Funkcja zwraca liczbę odpowiadającą pierwszemu znakowi ciągu wg. tablicy ASCII.

```
$a=ord("defghij");  
//$a przyjmie wartość 100  
strpos(string, znak)
```

Funkcja zwraca pozycję pierwszego wystąpienia znaku w ciągu.

```
$a="moj ciąg tekstowy";  
$b=strpos($a, "j");  
//$b przyjmie wartość 2
```

strRpos(string, znak)

Funkcja zwraca pozycję ostatniego wystąpienia znaku w ciągu.

```
$a="moj ciąg tekstowy";
```

```
$b=strRpos($a, "o");
```

```
//$b przyjmie wartość 14
```

strchr(string, znak)

Funkcja odcina i zwraca tekst od miejsca pierwszego wystąpienia znaku.

```
$a="moj ciąg tekstowy";
```

```
$b=strchr($a, "g");
```

```
//$b przyjmie wartość "g tekstowy"
```

strstr(string, znaki)

Funkcja odcina i zwraca tekst od miejsca pierwszego wystąpienia znaków.

```
$a="moj ciąg tekstowy";
```

```
$b=strstr($a, "ci");
```

```
//$b przyjmie wartość "ciąg tekstowy"
```

strlen(string)

Funkcja zwraca ilość znaków w ciągu.

```
$a=strlen("moj ciąg tekstowy");
```

```
//$a przyjmie wartość 17
```

strtoupper(string)

Funkcja zmienia wszystkie znaki w ciągu na duże.

```
$a=strtoupper("Moj Ciąg tekstowy");
```

```
//$a przyjmie wartość "MOJ CIĄG TEKSTOWY"
```

strtolower(string)

Funkcja zmienia wszystkie znaki w ciągu na małe.

```
$a=strtolower("Moj Ciąg TEKSTOWY");
```

```
//$a przyjmie wartość "moj ciąg tekstowy"
```

substr(string, od, ile)

Funkcja wycina i zwraca z ciągu "string" ilość znaków równą "ile" zaczynając od pozycji "od". Jeżeli "od" jest ujemne funkcja zacznie odcinać znaki licząc od końca ciągu "string". Parametr "ile" jest opcjonalny i jeśli zostanie pominięty funkcja wytnie wszystkie znaki zaczynając od pozycji "od" do końca ciągu. Jeśli parametr "ile" jest ujemny funkcja wytnie wszystkie znaki zaczynając od pozycji "od" do pozycji "ile" licząc od końca.

```
$a="moj ciąg tekstowy";
```

```
$b=substr($a, 4, 5);
```

```
//$b przyjmie wartość "ciąg "
```

```
$b=substr($a, 4);
```

```
//$b przyjmie wartość "ciąg tekstowy"
```

```
$b=substr($a, 4, -2);  
//$b przyjmie wartość "ciąg teksto"  
  
$b=substr($a, -4, -2);  
//$b przyjmie wartość "to"  
ucfirst(string)  
Funkcja zmienia pierwszy znak ciągu na duży, jeśli znak był literą.  
  
$a=ucfirst("moj ciąg tekstowy");  
//$a przyjmie wartość "Moj ciąg tekstowy"  
ucwords(string)  
Funkcja zmienia pierwszy znak każdego wyrazu ciągu na duży, jeśli znak był literą.  
  
$a=ucwords("moj ciąg tekstowy");  
//$a przyjmie wartość "Moj Ciąg Tekstowy"
```

Zadanie 19d5

Temat: Operacje na tekstach. Wyświetlanie napisu w pionie.

```
<?php  
$tekscik='ZSE Gdańsk';  
  
for($i=0;$i<strlen($tekscik);$i++)  
{  
    echo($tekscik[$i].<br />');  
}  
  
?>
```

Zadanie 19d6

Temat: Wczytaj wyraz(ciąg znaków) i sprawdź czy wyraz jest palindromem.
Wczytanie z użyciem formularza oraz przycisku sprawdź. Komunikaty podawane przez program patrz poniżej.

Wyraz kajak jest Palindromem. Działanie programu:

TAK wyraz „kajak” jest Palindromem

NIE wyraz „owca” nie jest Palindromem

Zadanie 19d7

Temat: Napisz program, który po wczytaniu tekstu (bez polskich liter) zamieni pierwszą literę małą na duże i między litery wpisze spację..
np. mama zamieni na M_a_m_a (_ oznacza spację)

Zadanie 20

Temat: Różne wersje językowe stron WWW w technologii PHP z użyciem stałych.

Gdy chcemy wykonać stronę WWW gdzie będzie możliwość przełączania wersji językowych można użyć stałych.

Opis define()

- Stałą można zdefiniować używając funkcji define().
- Raz zdefiniowana stała nie może być zmieniona ani usunięta.
- Stałe mogą zawierać jedynie dane boolean, integer, float oraz string
- Stałe nie mogą być typu resource.
- Wartość stałej otrzymujemy, używając jej nazwy.
- W przeciwieństwie do zmiennych, nazw stałych nie poprzedzamy znakiem \$.
- Możesz także użyć funkcji constant() do odczytania wartości stałej, jeśli chcesz wygenerować jej nazwę dynamicznie.
- Aby uzyskać listę wszystkich zdefiniowanych stałych, użyj get_defined_constants().
- Stałe i zmienne globalne operują w różnych przestrzeniach nazw. Oznacza to, że na przykład TRUE i \$TRUE oznaczają co innego.
- Chcąc sprawdzić, czy dana stała istnieje, użyj funkcji defined().
- Stałe mogą zostać zdefiniowane jedynie przy użyciu funkcji define() a nie poprzez zwykłe przypisanie;
- Stałe mogą być definiowane i używane wszędzie, niezależnie od zasięgu zmiennych;

Wykonaj:

Etap1

utwórz plik o nazwie wersja_jezykowa.php

```
<?php
include('polski.php');
?>
<form action="nazwa_tego_pliku.php">
<?php echo T_IMIE ?>:<br />
<input type="text" name="imie" /><br />
<?php echo T_OPIS ?>:<br />
<textarea name="opis" cols="30" rows="4"></textarea><br />
<input type="submit" value="<?php echo T_WYSLIJ ?>" />
</form>
```

Etap2

utwórz plik o nazwie polski.php

```
<?php
define('T_IMIE','Imię');
define('T_OPIS','Opis');
define('T_WYSLIJ','Wyślij');
?>
```

Etap3

utwórz plik o nazwie angielski.php odpowiednik pliku polski.php

Etap4

Przetestuj wersję angielską poprzez zamianę include('polski.php'); na include('angielski.php');

Zadanie21

Temat: Modyfikacja strony tak aby można było przełączać wersje językowa na ikonę flagi (flaga polska i angielska). Który język domyślny wybierz sam.

nr	Język
1	Angielski
2	Węgierski
3	Portugalski
4	Francuski
5	Suahili
6	Hiszpański
7	Holenderski
8	Grecki
9	Turecki
10	Włoski
11	Czeski
12	Słowacki
13	Ukraiński
14	Kaszubski

Etap1

Ściągnij ikony flag np.



Etap2

Wykorzystaj pliki polski.php angielski.php

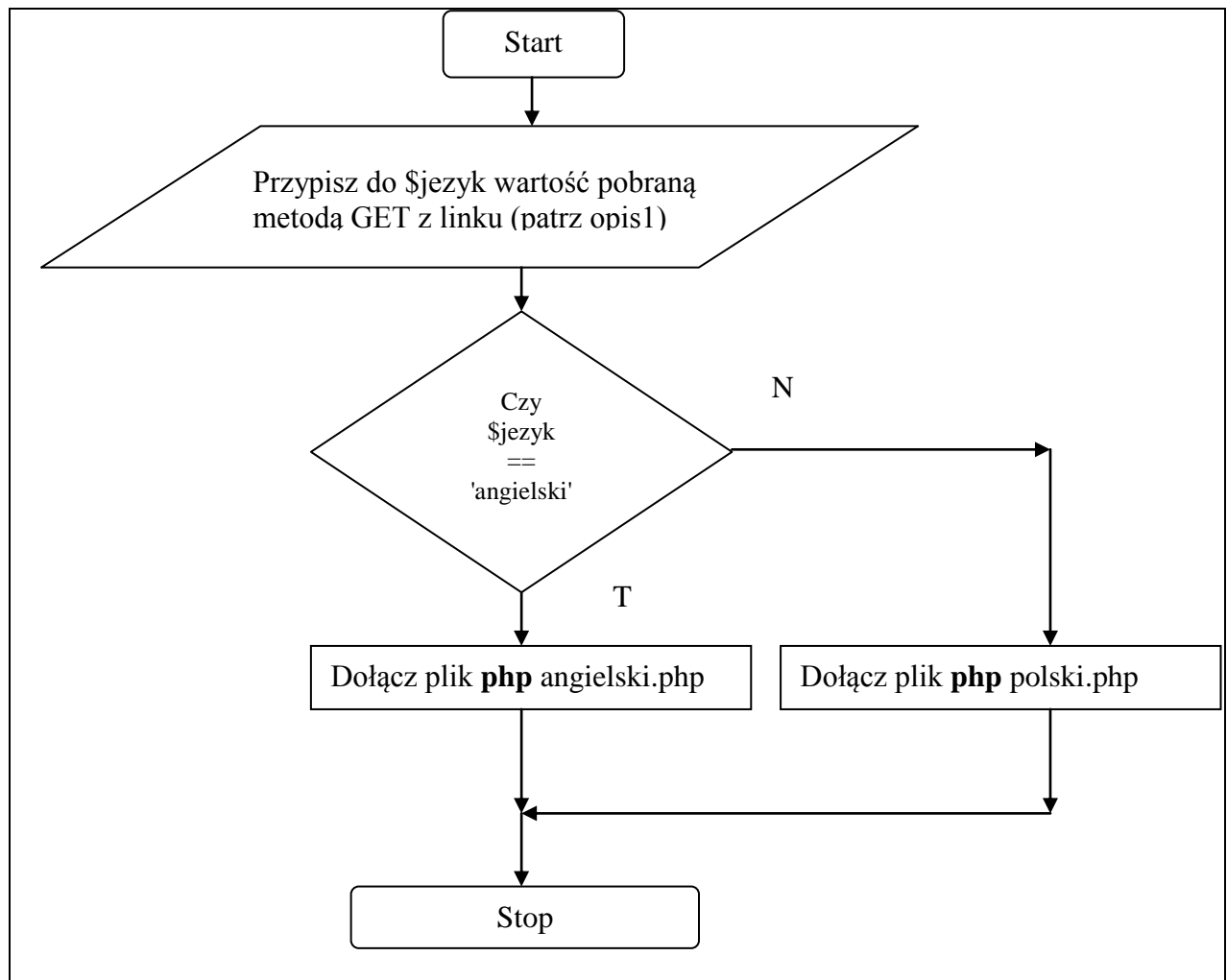
Etap3

Poniższą wstawkę php z zadania poprzedniego wykonaj wg algorytmu(na nowej stronie):

```
<?php
include('polski.php');
?>
```

Uwaga:

W zadaniu mogą być tylko dwie instrukcje include. Ponieważ zmiana wersji językowej wiąże się z stałymi a tych nie można zmieniać podczas działania programu.



opis1

`$jezyk = $_GET['jezyk_kow'];`

gdzie:

`jezyk_kow` jest nazwą zmiennej przekazywanej przez link. Zmienna `jezyk_trzy_pierwsze` litery nazwiska ucznia.

Etap4

Wykonaj przed formularzem linki (flaga polska i amerykańska) przekazujące wartości do strony;

``
i drugi link → wykonaj sam

Gdzie po znaku ? jest zapisana nazwa zmiennej a po = wartość zmiennej przekazywanej przez adres do strony.

Etap5

do `<form >` dodaj metodę GET

Etap6

Zmień domyślną stronę językowa, czyli jak u Ciebie była polska wersja językowa to zmień na angielską lub odwrotnie.

Zadanie21a

Temat: Blokowanie adresu IP.

```
<?php
    ob_start();
    $blokowane=array('192.168.30.2');
    if (in_array($_SERVER["REMOTE_ADDR"],$blokowane))
    {
        echo('<div style="color:red;">Dostęp do tej strony został dla Ciebie
zablokowany.</div>');
    }
    else
    {
        header('Location: http://www.zse.gda.pl');
    }
    ob_end_flush();
?>
```

Zadanie21b

Temat: Grafika w PHP. Wyświetlenie grafiki o nazwie arbuz.jpg

Uwaga: Gdy używasz w php instrukcji header to pliki nagrywaj w kodowaniu **UTF-8 bez BOM**

Użycie grafiki w PHP najprostsze jest z wykorzystaniem biblioteki GD,

```
<?php
header("Content-type: image/jpeg");
$image = imagecreatefromjpeg("arbuz.jpg");
imagejpeg($image);
?>
```

imagecreatefromjpeg()→przechowuje obrazy w pamięci we własnym formacie, stanowiącym generalnie dwuwymiarowy obraz rastrowy o określonej głębi kolorów. Tego typu obiekt - w powyższym przykładzie zapisany pod zmienną **\$image** - jest podstawową jednostką pracy z obrazami w GD.

imagecreatefrompng() - import z pliku w formacie PNG,
imagecreatefromgif() - import z pliku w formacie GIF,
imagecreatefromgd() - import z pliku własnego formatu biblioteki GD (starsza wersja),
imagecreatefromgd2() - import z pliku własnego formatu biblioteki GD2 (nowsza wersja),
imagecreatefromwbmp() - import z pliku BMP (format Windows).

Zadanie21b1

Temat: Grafika w PHP. Wyświetlenie różnych figur graficznych grafik.

Wykonaj:

- 1) Dorysuj przekątne kwadratu w kolorze czerwonym,
- 2) Do rysunku dodaj pięć kół (wypełnionych) olimpijskich w odpowiednich kolorach i nie nachodząc na siebie.
- 3) Zwiększ wielkość czcionki „ZSE Gdańsk jest OK.” oraz kolor czcionki. Użyj funkcji `imagefttext()`.
- 4) Dodaj Twoje nazwisko dużą czcionką (Arial, 30) pisane w pionie.

Uruchom przykład:

```
<?
header("Content-type: image/jpeg");
$rysunek = imagecreate(300,300);
$kolorbialy= imagecolorallocate($rysunek, 255, 255, 255);
$niebieski= imagecolorallocate($rysunek, 0, 0, 255);
$orange=imagecolorallocate($rysunek,220,210,60);
imagefill($rysunek, 0, 0, $niebieski);
imageline($rysunek,10, 10, 10 , 100 , $kolorbialy);
imageline($rysunek,10, 100, 100 , 100 , $kolorbialy);
imageline($rysunek,100, 100, 100 , 10 , $kolorbialy);
imageline($rysunek,100, 10, 10 , 10 , $kolorbialy);
imagefilledrectangle($rysunek,30,30,50,50,$orange);
imagefilledellipse($rysunek,60,180,50,50,$orange);
imagestring($rysunek,20,80,80,'ZSE Gdańsk jest OK',$orange);
        imagejpeg($rysunek);
?>
```

Opis funkcji graficznych dla PHP

Podstawy

Dynamiczną grafikę wstawia się do dokumentu tradycyjnie, używając znacznika ``. Jedyną różnicą jest to, że atrybut `src` wskazuje na plik z rozszerzeniem PHP. Możliwe jest nawet uruchamianie skryptu tworzącego grafikę z parametrami podanymi jako zmienne GET. Oto, jak należy wstawić przykładowy obrazek do dokumentu:

```
<?
echo"<img src=\"licznik.php?wizyty=\".$licz.\"\" />";
?>
```

Content-type w nagłówku, tak aby przeglądarka mogła odpowiednio zinterpretować plik. Dla przykładu:

```
<?
header("Content-type: image/png");
?>
```

W zależności od założonego typu pliku należy ustawić odpowiedni Content-type na `image/png`, `image/gif` lub `image/jpeg`. Następnie należy utworzyć nowy, pusty obrazek za pomocą

imagecreatefrompng()→Można też utworzyć nowy obrazek na podstawie wzorca za pomocą funkcji. Jest to bardzo dobra metoda do tworzenia np. przycisków na stronie. Wystarczy utworzyć jeden obrazek z tłem i później stworzyć tylko skrypt wstawiający tekst z parametru GET na gotowe tło. Dla przykładu:

```
$img=imagecreatefrompng("buttony/podklad1.png");
```

funkcja imagecolorallocate()→służy do deklarowania koloru która pobiera cztery argumenty. Należy podać kolejno zmienną-obiekt z obrazkiem oraz wartości składowe: czerwoną, zieloną i niebieską wybranego koloru (od 0 do 225). Funkcja zwraca zmienną-identyfikator koloru.

Dla przykładu:

```
<?
$orange=imagecolorallocate($img,220,210,60);
$white=imagecolorallocate($img,225,225,225);
$black=imagecolorallocate($img,0,0,0);
?>
```

funkcji imagecreate()→Zwraca ona uchwyt do obrazka. Jako parametry tej funkcji trzeba podać wysokość i szerokość obrazka.

Dla przykładu:

```
$img=imagecreate(200,200);
```

imagefill()→funkcja służy do wypełniania zwartych powierzchni. Należy podać cztery argumenty, zmienną id obrazka, współrzędną początkową x (left), współrzędną początkową y (top) oraz id koloru. Dla przykładu:

```
<?
imagefill($img,0,0,$black);
?>
```

imageline()→narysowanie linii. Argumenty, które należy podać, to obrazek, x_początkowe, y_początkowe, x_koncowe, y_koncowe, kolor.

Dla przykładu:

```
<?
imageline($img,120,120,190,190,$white);
?>
```

imagerectangle()→rysowanie kwadratu

argumenty, które podajemy

obrazek, x_początkowe, y_początkowe, x_koncowe, y_koncowe, kolor, przy czym x_początkowe i y_początkowe to współrzędne lewego górnego wierzchołka kwadratu, a x_koncowe i y_koncowe to współrzędne prawego dolnego wierzchołka.

imagefilledrectangle()→rysowanie wypełnionego kwadratu.

Dla przykładu:

```
<?
imagerectangle($img,10,10,30,30,$orange);
imagefilledrectangle($img,30,30,50,50,$orange);
?>
```

imageellipse()→rysowanie elipsy.

Argumenty:

(obrazek, x_srodka, y_srodka, x_szerokosc, y_wysokosc, kolor).

imagefilledellipse() →rysowanie wypełniona elipsę.

Dla przykładu:

```
<?
imageellipse($img,180,20,20,20,$orange);
imagefilledellipse($img,20,180,20,20,$orange);
?>
```

imagepolygon()→rysowanie nieregularny kształt.

Argumenty, które podajemy, to imagepolygon(obrazek, tablica_wierzchołkow, ilosc_wierzchołkow, kolor).

Tablica wierzchołków to tablica przygotowana według poniższego wzorca:

```
<?
$wartosci=array(
50,50,// punkt 1 (x, y)
90,60,// punkt 2 (x, y)
80,100,// punkt 3 (x, y)
40,60,// punkt 4 (x, y)
);
?>
```

Pamiętajmy, żeby zmienna liczba_wierzchołkow miała taką samą wartość jak liczba, którą zadeklarowaliśmy w tablicy.

imagefilledpolygon()→ wypełniony wielobok

wielobok wypełniony z obrysem w innym kolorze, uzyskujemy, wywołując dwa razy tworzenie wieloboku o tych samych wierzchołkach: np. raz białego z wypełnieniem, a drugi raz pomarańczowego. Dla przykładu:

```
<?
imagefilledpolygon($img,$wartosci,4,$white);
imagepolygon($img,$wartosci,4,$orange);
?>
```

imagearc()→część okręgu.

Argumenty

(obrazek, x_srodka, y_srodka, szerokosc, wysokosc, kat_poczatka, kat_konca, kolor).

Łuk jest zawsze rysowany od poziomu zgodnie z ruchem wskazówek zegara.

Dla przykładu:

```
<?
imagearc($img,120,120,100,100,90,360,$orange);
?>
```

imagefilledarc() → łuki wypełnione

Oprócz parametrów opisanych przy imagearc() musimy również na końcu dodać jeszcze styl wypełnienia. Cztery dostępne style to:

IMG_ARC_PIE – tworzy wypełniony łuk, połączony ze środkiem okręgu,
IMG_ARC_CHORD – tworzy wypełniony trójkąt, początek i koniec łuku
są połączone prostą,
IMG_ARC_NOFILL – tworzy sam łuk bez wypełnienia,
IMG_ARC_EDGED – użyty pojedynczo wygląda tak samo jak
IMG_ARC_PIE.

Dla przykładu:

```
<?
imagefilledarc($img,40,30,100,100,30,120,$orange,IMG_ARC_PIE);
?>
```

Style można łączyć operatorem + (tworząc w ten sposób nowe style). Najlepszym przykładem na to jest połączenie IMG_ARC_NOFILL oraz IMG_ARC_EDGED, dzięki któremu otrzymujemy sam łuk i linie łączące jego końce ze środkiem (bez wypełnienia).

imagechar() → dodanie do obrazka tylko jeden znak.

argumenty to imagechar(obrazek, wielkosc, x_poczatkowe, y_poczatkowe, znak, kolor). Wielkość to cyfra od 1 do 5 – 1 to najmniejsza, 5 to największa.

Dla przykładu:

```
<?
imagechar($img,3,80,80,'P',$orange);
?>
```

imagecharup() → rysuje znak pionowo.

imagestring() → wyprowadzanie ciągu znaków.

Parametry, które podajemy, są takie same jak w wypadku imagechar(), tylko zamiast pojedynczego znaku podaje się cały ciąg.

Dla przykładu:

```
<?
imagestring($img,5,80,80,'PHP jest OK',$orange);
?>
```

Aby dodać tekst pionowo, wystarczy (podobnie jak w wypadku jednego znaku) użyć funkcji **imagestringup()**.

imagefttext() → Styl ciągu znaków
argumenty, które trzeba podać do funkcji:
(obrazek, wielkosc, kat, x_poczatkowe, y_poczatkowe, kolor,
plik_z_czcionka, tekst)

Wielkość to liczba od 5 do 150 reprezentująca wielkość w pikselach.

Kąt to kąt nachylenia tekstu.

Wyrażenie 0 stopni to tekst pisany normalnie – poziomo. Tym razem jednak tekst jest obracany o dany kąt przeciwnie do ruchu wskazówek zegara.

Zmienne x_poczatkowe i y_poczatkowe to współrzędne lewego (dolnego) wierzchołka tekstu. W parametrze plik_z_czcionka należy zaś podać adres do pliku .TTF z czcionką True Type. Czcionki True Type to czcionki używane w systemie – wystarczy skopiować taki plik z folderu C:\WINDOWS\FONTS i wgrać go na własne konto ze stroną. Dla przykładu:

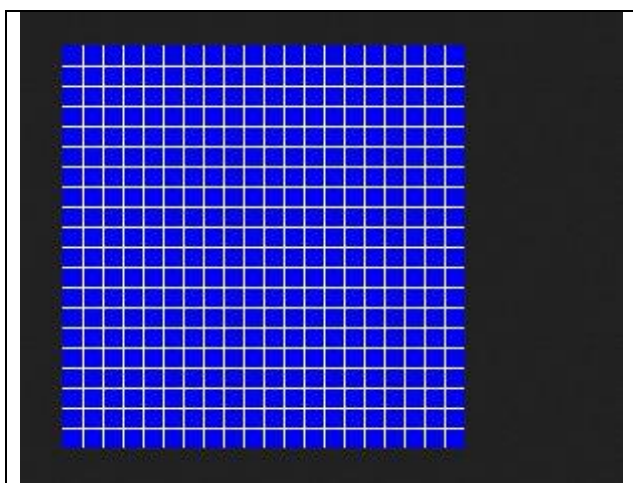
```
<?
imagefttext($img,10,315,30,
30,$orange,"folder/acmesa.ttf","czcionka A.C.M.E - test");
imagefttext($img,10,20,30,
140,$white,"folder/bluehigb.ttf","czcionki True Type");
?>
```

Zadanie21c

Temat: Grafika w PHP. Kratka.

Wskazówka: Zdefiniuj obszar rysowania o kolorze → **patrz tabela**.

Z użyciem pętli narysuj linie w kolorze → **patrz tabela** pionowe i poziome.



nr	obszar rysowania	linie w kolorze	Odległość między liniami	Ilość linii
1	niebieski	biały	15 pikseli	30
2	zielony	żółty	15 pikseli	29
3	czerwony	niebieski	16 pikseli	27
4	niebieski	biały	17 pikseli	26
5	zielony	żółty	18 pikseli	25
6	czerwony	niebieski	19 pikseli	24
7	niebieski	biały	20 pikseli	23
8	zielony	żółty	21 pikseli	22
9	czerwony	niebieski	22 pikseli	21
10	niebieski	biały	23 pikseli	20
11	zielony	żółty	24 pikseli	19
12	czerwony	niebieski	25 pikseli	18
13	niebieski	biały	26 pikseli	17
14	zielony	żółty	27 pikseli	16
15	czerwony	niebieski	28 pikseli	15

Zadanie21c1

Temat: Grafika w PHP. **Licznik** graficzny. Plik licznik.php pobierający liczbę odwiedzin z parametru wizyty i wyświetlający graficznie liczbę odwiedzin.

- 1)Najpierw należy ustawić odpowiednią wartość Content-type.
- 2)Później kopiujemy zawartość zmiennej wizyty z tablicy _GET do zmiennej \$wizyty,
- 3)w zmiennej \$dl zapisujemy długość ciągu zmiennej \$wizyty.
- 4)Kolejno tworzymy obrazek o odpowiedniej szerokości (liczba cyfr*20)
- 5)definiujemy kolory.
- 6)Pokrywamy tło na czarno
- 7)w pętli for wyświetlamy po kolei każdą cyferkę, używając funkcji imagefttext().
- 8)Na koniec wysyłamy gotowy obrazek, używając imagepng().

Dla przykładu:

```
<?php
header("Content-type: image/png");
$wizyty=$_GET['wizyty'];
$dl=strlen($wizyty);
$img=imagecreate(($dl*20),40);//szerokość zależna od liczby cyfr
$orange=imagecolorallocate($img,220,210,60);    //deklaracja
kolorów
$black=imagecolorallocate($img,0,0,0);
imagefill($img,0,0,$black);                    //wypełnienie tła
for($i=0;$i<$dl;$i++)
{
    imagefttext($img,15,0,$i*20+5,27,$orange,"
C:\WINDOWS\Fonts\Arial.ttf",
    substr($wizyty,$i,1));
}
imagepng($img);                                //wysłanie obrazka do
przeglądarki
?>
```

Aby teraz wstawić to do dokumentu, należy użyć znacznika . Dla przykładu:

```

```

a tam, gdzie jest wpisane 12345, wpisać nazwę zmiennej z liczbą do wyświetlenia.

Zadanie21d

Temat: Grafika w PHP. Wczytać z użyciem formularza 3 lub 4 lub 5 liczby (patrz tabela). Na tej podstawie narysować wykres słupkowy. Każdy słupek innym kolorem. Skalowanie wg wielkości → najwyższy słupek połowę ekranu niezależnie od wczytanej wielkości, podpisy wielkościami liczb pod słupkami. Wielkości wczytywane są dodatnie.

Algorytm skalowania:

Krok1

Znajdujesz maksimum z wczytanych wielkości i zapisujesz do zmiennej max.

Krok2

Obliczasz wysokość słupków ze wzoru

$$\text{Wysokosc_słupka1} = (\text{wielkość_słupka} * \text{rozdzielczość_po_y}) / \text{max}$$

Krok3

Obliczasz wysokości wszystkich pozostałych słupków.

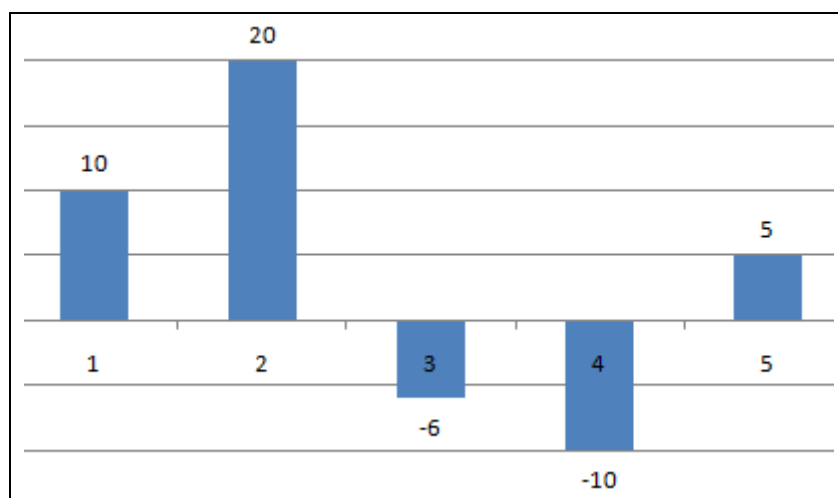
Koniec algorytmu

Nr	Ilość słupków	Słupki w kolorze ważna kolejność
1	3	czerwony, niebieski, żółty
2	4	żółty, czerwony, fioletowy, czerwony
3	5	niebieski, żółty, czerwony, fioletowy, czerwony
4	3	niebieski, żółty, czerwony,
5	4	fioletowy, żółty, czerwony, pomarańczowy
6	5	niebieski, fioletowy, żółty, czerwony, pomarańczowy
7	3	niebieski, czerwony, fioletowy
8	4	niebieski, żółty, czerwony, fioletowy
9	5	fioletowy, żółty, czerwony, niebieski, pomarańczowy
10	3	czerwony, pomarańczowy, niebieski
11	4	fioletowy, czerwony, pomarańczowy, niebieski
12	5	żółty, fioletowy, czerwony, pomarańczowy, niebieski
13	3	fioletowy, żółty, czerwony,
14	4	fioletowy, czerwony, pomarańczowy, niebieski
15	5	fioletowy, żółty, czerwony, pomarańczowy, niebieski

Zadanie21e

Temat: Treść tak sama ale wielkości wczytywane są dodatnie lub ujemne.

Dla wartości ujemnych otrzymujemy. Wykres:



Numerowanie słupków nie jest konieczne, liczby nad słupkami lub pod słupkami, linie poziome nie są potrzebne, słupki kolorowe (jak w tabeli).

Teoria:

Metody przekazywania parametrów do innych stron:

- metoda "GET"
- metoda "POST".

Opis metody GET

Metoda GET używa się kiedy parametrów jest niewiele. Dzieje się tak ponieważ parametry przekazuje się za pomocą adresu URL

np. <http://www.coś.pl/strona.php?parametr1=wartość1¶metr2=wartość2>

którego długość jest ograniczona. Należy też pamiętać że parametry są widoczne w pasku adresu przeglądarki, więc tej metody nie należy używać jeśli przekazywane są np. hasła.

Adres od parametrów oddzielany jest za pomocą **znaku zapytania**, a kolejne pary **parametr=wartość** za pomocą znaku **ampersand ("&")**. Tą metodę można wykorzystać także przy przekazywaniu parametrów przez odnośnik (zobacz dalej).

Dane przesłane metodą GET trafiają do tablicy \$_GET (lub \$HTTP_GET_VARS w wersjach starszych niż 4.1.0)

Opis metody POST

Metoda POST do przekazywania parametrów wykorzystuje nagłówek zapytania – wystarczy wiedzieć, że metoda ta umożliwia przekazywanie dużo większych parametrów, a także że parametrów nie widać w pasku przeglądarki.

Dane przesłane metodą POST to tablica \$_POST (lub \$HTTP_POST_VARS).

Dane z metody Tablice \$_GET i \$_POST są **superglobalne**. Oznacza to, że są widoczne w każdym miejscu kodu PHP bez konieczności użycia składni globals.

<http://gajdaw.pl/bd/podstawy/print.html>

Zagadnienie baz danych w PHP-rozwiazanie calkowite.

1. Bazy danych na potrzeby WWW→wprowadzenie.

Portale internetowe udostępniają często tak duże ilości informacji, że trudno sobie wyobrazić ich działanie bez baz danych.

- 1)Wyszukiwarki,
- 2)katalogi stron WWW przechowują informacje o znanych witrynach w bazach danych,
- 3)Słowniki internetowe stanowią w istocie interfejs do baz danych zawierających tłumaczenia,
- 4)Sklepy on-line prezentują produkty będące rekordami baz danych,
- 5)Biblioteki wirtualne,
- 6)fora dyskusyjne,
- 7)listy ogłoszeń,
- 8)katalogi plików muzycznych.

Witryny internetowe, w których skrypty dynamicznie generują strony WWW na podstawie zapytań klientów i zawartości baz danych sprawdzają się z kilku powodów.

Do najważniejszych należy oddzielenie warstwy prezentacyjnej od danych. Zadania związane z dodawaniem, usuwaniem i modyfikowaniem informacji wykonujemy nie ingerując w kod serwisu. W ten sposób raz przygotowana witryna może być wzbogacana bez konieczności modyfikacji kodu.

2. Oprogramowanie

Tworzenie serwisów WWW wykorzystujących bazy danych wymaga instalacji odpowiedniego oprogramowania. Oprócz pary **Apache** (serwer stron WWW) oraz **PHP** (język skryptowy służący do tworzenia dynamicznych stron WWW) konieczny będzie serwer bazy danych. Oprogramowanie **MySQL** stanowi implementację SQL-owej bazy danych i jest nazywane w żargonie silnikiem bazy danych.

Dodatkowym pakietem ułatwiającym pracę z bazami danych MySQL jest pakiet **phpMyAdmin** (patrz: <http://www.phpmyadmin.net>).

Jednym ze sposobów uruchamiania **phpMyAdmin** jest: należy uruchomić przeglądarkę internetową i po wpisaniu adresu localhost przejść do skryptu index.php zawartego w folderze phpMyAdmin.

3. Tworzenie i usuwanie baz danych

Bazy danych możemy tworzyć na trzy sposoby:

- 1) wykorzystując pakiet **phpMyAdmin**,
- 2) wydając polecenia **SQL w wierszu** poleceń programu MySQL,
- 3) pisząc **skryptu php**.

Metoda 1

Pakiet **phpMyAdmin** zawiera na stronie głównej formularz zatytułowany "Utworzenie nowej bazy danych". Wpisanie nazwy bazy w wierszu wprowadzania danych i naciśnięcie guzika "Utworzenie" spowoduje wykonanie odpowiedniego zapytania SQL. Nowo utworzona baza będzie natychmiast widoczna w menu z lewej strony okna. W celu usunięcia nowo utworzonej bazy danych wystarczy wybrać bazę danych z menu dostępnego w lewej części okna, a następnie kliknąć hiperłącze "Usunięcie bazy danych" dostępne w dolnej części okna.

Metoda 2

Druga metoda operowania bazami danych wymaga uruchomienia programu mysql. W wierszu poleceń należy wydać następującą komendę:

```
C:\mysql\bin\mysql.exe -u root
```

ewentualnie zmieniając lokalizację pliku mysql.exe. Po uruchomieniu wiersza poleceń SQL możemy przystąpić do wydawania komend. Polecenie CREATE DATABASE służy do utworzenia nowej bazy danych. Po znaku zachęty piszemy:

```
CREATE DATABASE bazaosob;
```

Polecenie to spowoduje utworzenie bazy danych o nazwie "bazaosob". Bazę danych możemy usunąć wydając polecenie:

```
DROP DATABASE bazaosob;
```

Metoda 3

Trzecią metodą tworzenia baz danych jest napisanie odpowiedniego skryptu php.

funkcja mysql_connect → połączenia z serwerem baz danych.

```
$link = mysql_connect("localhost", "", "")  
or die("BŁĄD: nie można się połączyć z serverem MySQL");
```

funkcję mysql_query → wykonanie komendy SQL

```
$result = mysql_query("CREATE DATABASE bazaosob")  
or die("BŁĄD: nie można utworzyć bazy danych");
```

funkcja mysql_close → zamykanie połączenia z bazą danych.

```
mysql_close($link);
```

4. Tworzenie tabel i dołączanie rekordów

Pierwszym krokiem zmierzającym do wypełnienia naszej bazy danych jest określenie jej struktury. Strukturę bazy definiujemy tworząc odpowiednie tabele. Nasze pierwsze bazy danych będą zawierały jedną tabelę. Tabela ta będzie umożliwiała przechowywanie imion i nazwisk.

Metoda1

Narzędzie phpMyAdmin umożliwia tworzenie tabeli za pomocą formularza. Po wybraniu bazy danych osoby z lewego menu, wypełniamy dwa pola określające nazwę tabeli oraz liczbę pól. Po naciśnięciu przycisku "Wykonanie" ujrzymy formularz służący do określenia typów pól oraz ich atrybutów. Pierwsze pole nazywamy "imie" i ustalamy jego typ na VARCHAR. Dodatkowo ustalamy atrybut długość na 50. Drugie pole nazywamy "nazwisko", jego typ ustalamy na VARCHAR zaś długość na 40. Dodatkowo zaznaczamy opcję PODSTAWOWY. W ten sposób każdy rekord naszej tabeli będzie zawierał imię będące napisem co najwyżej 50-cio literowym oraz nazwisko będące napisem co najwyżej 40-sto literowym. Naciśnięcie przycisku "Zachowanie" spowoduje utworzenie tabeli.

Usunięcie tabeli wykonujemy klikając hiperłącze "Usunięcie". Jest ono widoczne obok nazwy tabeli po wybraniu bazy danych "bazaosob" z lewego menu pakietu phpMyAdmin.

Metoda2

Tworzenie tabeli z poziomu konsoli mysql sprowadza się do wydania zapytania CREATE TABLE. Należy pamiętać, by po utworzeniu tabeli poleceniem CREATE DATABASE wydać dodatkowo polecenie USE ustalające aktywną bazę danych. Zatem po poleceniu:

```
CREATE DATABASE bazaosob;
```

wydajemy polecenie

```
USE osoby;
```

A następnie polecenie

```
CREATE TABLE osoby (  
    imie VARCHAR (50) not null,  
    nazwisko VARCHAR (40) not null,  
    PRIMARY KEY (nazwisko)  
);
```

Powyższe polecenie powoduje utworzenie tabeli o strukturze identycznej jak poprzednio. W celu usunięcia tabeli należy wydać polecenie:

```
DROP TABLE osoby;
```

Metoda3

Tworzenie tabeli z poziomu skryptów php polega na wywołaniu funkcji mysql_query (ewentualnie mysql_db_query) i przekazaniu zapytania SQL poznanego poprzednio. Tak

więc po połączeniu z bazą danych funkcją `mysql_connect` oraz po ewentualnym utworzeniu bazy danych wywołujemy funkcję `mysql_db_query`:

```
$result = mysql_db_query("bazaosob", "  
CREATE TABLE osoby (  
    imie VARCHAR (50) not null  
    nazwisko VARCHAR (50) not null,  
    PRIMARY KEY (nazwisko))  
")  
or die("BŁĄD: nie można utworzyć tabeli");
```

Natomiast do usunięcia tabeli możemy użyć funkcji `mysql_db_query` w następujący sposób:

```
$result = mysql_db_query("bazaosob", "DROP TABLE osoby ")  
or die("BŁĄD: nie można utworzyć tabeli");
```

5. Dodawanie rekordów

Metoda1

Wykorzystując narzędzie phpMyAdmin wybieramy z menu bazę danych "bazaosob". Hiperłącze umożliwiające dodawanie rekordów do tabeli nazywa się "Dodawanie" i jest umieszczone obok nazwy tabeli w ciągu opcji: "Przeglądanie", "Wybór", "Dodanie", "Właściwości", "Usunięcie", "Wyczyszczenie".

Metoda2

Z poziomu konsoli MySQL rekordy dodajemy wydając polecenia `INSERT INTO` o następującej składni:

```
INSERT INTO osoby(imie, nazwisko) VALUES ('John', 'Lenon');
```

Dołączenie kilku rekordów możemy wykonać jednym zapytaniem oddzielając kolejne wartości przecinkiem:

```
INSERT INTO osoby VALUES ('Paul', 'McCartney'), ('Ringo', 'Star');
```

Metoda3

W języku php dołączanie danych wykonujemy ponownie wykorzystując funkcję `mysql_query` z odpowiednio ustalonym argumentem:

```
$result = mysql_query(  
    "INSERT INTO osoby(imie, nazwisko) VALUES ('John', 'Lenon')"  
    or die("Query failed");
```

Użycie pliku wsadowego SQL do pracy z bazą danych

Skrypt `tworzenie_bazy_bazaosob.php` zawiera listing pełnego skryptu w języku php tworzącego bazę danych "bazaosob". Polecenia wydawane w języku SQL z poziomu konsoli MySQL można zapisywać w pliku i wykonywać wsadowo. Plik `tworzenie_bazy_bazaosob.sql` zawiera kompletną listę poleceń. Następująca komenda:

```
C:\mysql\bin\mysql.exe -u root < tworzenie_bazy_bazaosob.sql
```

wydana z poziomu wiersza poleceń spowoduje wykonanie wszystkich poleceń zapisanych w pliku.

6. Wyświetlanie zawartości bazy danych na stronie WWW

Skrypt PHP będzie wyświetlał wszystkie rekordy umieszczone w tabeli "osoby".

a) Kod skryptu rozpoczynamy od połączenia z bazą danych za pomocą funkcji **mysql_connect**.

b) Następnie ustalamy bazę danych, z którą będziemy pracowali. Zadanie to realizuje funkcja **mysql_select_db**, której parametrem jest nazwa bazy danych.

c) Następnie zadajemy zapytanie SQL, którego rezultatem będzie zawartość tabeli osoby:

"SELECT * FROM osoby". Wyniki zwrócone przez zapytanie przetwarzamy w pętli while.

Funkcja **mysql_fetch_array** zwraca jako wynik kolejny rekord będący rezultatem wydanego zapytania SQL. Parametrem funkcji **mysql_fetch_array** jest wynik zwrócony przez funkcję

mysql_query. Zmienna **\$row** będąca wynikiem zwróconym przez funkcję

mysql_fetch_array jest **tablicą asocjacyjną**, która zawiera kolejne pola danego rekordu. Dla

przykładu pierwszym rekordem zwróconym przez funkcję **mysql_fetch_array** jest rekord, w

którym pole imię ma wartość 'John' zaś nazwisko - 'Lenon'. Zatem tablica **\$row** przyjmie

następujące wartości:

```
$row['imie'] == 'John'
```

oraz

```
$row['nazwisko'] == 'Lenon'
```

d) Instrukcja **echo** umieszczona wewnątrz pętli **while** wysyła odpowiedni wiersz tabeli HTML do klienta.

W przypadku, gdy już wszystkie rekordy zostały odczytane, funkcja **mysql_fetch_array** zwraca wartość logiczną **false**, co spowoduje zakończenie pętli.

e) Na koniec zwalniamy zasoby zaalokowane przez wywołanie zapytania SQL oraz zamykamy połączenie z bazą danych.

Kompletny skrypt został przedstawiony na listingu poniżej.

```
<?php
```

```
$link = mysql_connect("localhost", "", "")  
or die("Could not connect");
```

```
mysql_select_db("bazaosob")  
or die("Could not select database");
```

```
$query = "SELECT * FROM osoby";  
$result = mysql_query($query)  
or die("Query failed");
```

```
while ($row = mysql_fetch_array($result)) {  
    echo "<TR><TD>" . $row["imie"] .  
        "</TD><TD>" . $row["nazwisko"] .  
        "</TD></TR>\n";  
}
```

```
mysql_free_result($result);  
mysql_close($link);
```

```
?>
```

7. Przykład: baza danych hoteli

Temat: Strona WWW prezentująca listę hoteli.

a) Pierwszym krokiem jest przygotowanie skryptu tworzącego bazę danych. Baza danych dotycząca hoteli będzie zawierała jedną tabelę.

Każdy rekord tabeli będzie zawierał dane dotyczące jednego hotelu (nazwę, adres, adres strony WWW itd.)

Tabelę tworzymy w następujący sposób:

```
$result = mysql_db_query($nazwabazydanych, "
CREATE TABLE thotele (
    nazwa VARCHAR (150) not null ,
    kod VARCHAR (15) not null,
    miejscowosc VARCHAR (50) not null,
    ulica VARCHAR (150) not null,
    numerdomu VARCHAR (15) not null,
    telefon VARCHAR (15) not null,
    www VARCHAR (100) not null,
    PRIMARY KEY (nazwa, miejscowosc, telefon),
    INDEX (nazwa, miejscowosc, telefon)
)
")
or die("BŁĄD: nie można utworzyć tabeli");
```

b) Wpisanie danych dotyczących hoteli.

Dane hoteli zapisane w pliku hotele.txt dołączane są do bazy danych skryptem hotele-dodawanie-rekordow.php. Skrypt ten odczytuje zawartość pliku, przetwarza odczytane wartości i dołącza je do bazy danych stosując opisaną wcześniej metodę wykorzystującą funkcje mysql_query.

c) Kod skryptu prezentującego w postaci strony WWW informacje o dostępnych hotelach.

```
$nazwabazydanych = "dbhotele";
```

```
$link = mysql_connect("localhost", "", "")
or die("BŁĄD: nie można się połączyć z serverem mySQL");
```

```
mysql_select_db ($nazwabazydanych)
or die("BŁĄD: wybór bazy danych");
```

```
$zapytanie = "SELECT * FROM thotele";
```

```
$result = mysql_query($zapytanie);
while ($wiersz = mysql_fetch_array($result)) {
    echo '<TABLE class="client">';
    if (trim($wiersz['nazwa']) != "") {
        echo "<TR><TD class='nazwa'>{$wiersz['nazwa']}</TD></TR>";
    }
    echo "<TR><TD>{$wiersz['kod']} {$wiersz['miejscowosc']}</TD></TR>";
    echo "<TR><TD>{$wiersz['ulica']} {$wiersz['numerdomu']} </TD></TR>";
    echo "<TR><TD>tel. {$wiersz['telefon']}</TD></TR>";
    echo "<TR><TD><A href='\"{$wiersz['www']}\">{$wiersz['www']}</A></TD></TR>";
```



```
    echo '</TABLE>';  
}  
mysql_close($link);
```

Zadanie22a

Temat: Założenie bazy z użyciem skryptu mysql-od struktury do wypisania danych.

Zadanie22b

Temat: Założenie bazy z użyciem skryptu mysql-od struktury do wypisania danych.

Zadanie22c

Temat: Założenie bazy z użyciem skryptu mysql-od struktury do wypisania danych.

Uchwyt do obsługi pliku

Uchwyt jest to konstrukcja informatyczna, która umożliwi komunikację ze wskazanym przez nas plikiem, umożliwia wykonywanie różnych operacji na pliku.

Nazwa	Opis
mysql_affected_rows();	Zwraca liczbę zmodyfikowanych wierszy przez poprzednie zapytanie SQL.
mysql_close(\$link)	Zamyka połączenie o uchwycie \$link.
mysql_connect(\$h,\$u,\$p)	Nawiązuje połączenie z serwerem pod adresem \$h, logując się na użytkownika \$u z hasłem \$p. Funkcja zwraca uchwyt do połączenia.
mysql_create_db(\$n)	Tworzy bazę danych o nazwie \$n.
mysql_drop_db(\$n)	Usuwa bazę danych o nazwie \$n.
mysql_error()	Zwraca komunikat o ostatnio powstałym błędzie.
mysql_escape_string(\$s)	Przygotowuje ciąg \$s do użycia w zapytaniu MySQL.
mysql_fetch_array(\$r)	Pobiera wiersz odpowiedzi związanych z uchwycem \$r.
mysql_insert_id()	Zwraca wartość AUTO_INCREMENT wygenerowaną przez ostatnie zapytanie (jeśli taka była).
mysql_num_rows(\$r)	Zwraca liczbę wierszy w uchwycie wyników \$r.
mysql_query(\$r)	Wysyła zapytanie \$r do serwera MySQL. Zwraca uchwyt do wyników.
mysql_select_db(\$n)	Służy do wybrania bazy danych \$n.

Zadanie22

Temat: Założenie bazy z użyciem phpMyadmin oraz skrypt do jego obsługi.

1)Użyj phpMyAdmin.

Uruchom go poprzez:

a)krasnala

uruchomienia krasnala, następnie kliknij prawym zielone piór w zasobniku na dole po prawej stronie na pasku zadań i teraz wybierz WWW i phpMyAdmin.

użytkownik→root

hasło→krasnal

b)Webserv

uruchomienia Webserv, następnie kliknij prawym w zasobniku na dole po prawej



stronie na pasku zadań
użytkownik→root
hasło→Enter (bez hasła)

i teraz wybierz phpMyAdmin.

2)Założ bazę danych

Będziesz musiał podać:

- Nazwę bazy
- Nazwę tabeli
- Ilość pól bazy

a)**Założ bazę danych** o nazwie nazwisko_słuchacza_php (bez polskich liter małymi literami).

→o nazwie tabeli kontakty_nazwisko_słuchacza np. kontakty_kowalski (bez polskich liter małymi literami).

→o trzech polach:

numer (tinyint(4)),

nazwisko(text),

waga (int(11))

b)**Wpisz** będąc w PHPAdmin do niej pięć rekordów z sensowymi danymi.

c)dokonaj eksportu poleceń zakładania bazy danych i wpisywania danych

do pliku SQL o nazwie zad22_kowalski.sql

d)na końcu skryptu dopisz **SELECT** wyświetlania wszystkich danych z bazy, uruchom skrypt w mysql (zmień nazwe bazy danych w treści skryptu),

e)Po założeniu bazy wpisz następujący skrypt (treść patrz poniżej). Będziesz musiał zmienić nazwę bazy oraz nazwę tabeli w treści skryptu poniżej.

Nagraj ten skrypt do folderu z22 pod nazwą zad22_twoje_nazwiski.php o

następującej ścieżce dostępu c:\WebServ\httpd-users\z22\

Skrypt uruchom poprzez:

http://127.0.0.1/~nazwa_konta/zad22_twoje_nazwiski.php

Po sprawdzeniu działania skryptu wykonaj:

1)umieść listing w zeszycie (pracy semestralnej).

2)Wykonaj wytlumaczenia wszystkich instrukcji użytych w skrypcie w formie tabeli:

	komenda PHP	Wytłumaczenie
1	<?	początek skryptu

2	\$user= 'root';	przypisanie do zmiennej \$user wartości root→użytkownik, który loguje się do bazy to root
3	\$pass= '';	przypisanie do zmiennej \$pass wartości →użytkownikiem, który loguje się do bazy ma hasło puste (bez hasła)

Skrypt do wpisania. Obsługuje on bazę zapisaną powyżej.

```
<?php
$user= 'root';
$pass= '';
$host = 'localhost';
$base = 'kowalski'; //tutaj nazwa twojej bazy
$con=mysql_connect($host,$user,$pass);
mysql_select_db($base);
$ret = mysql_query("SELECT * FROM `tabela_kowalski` order by numer",$con);
//tutaj nazwa twojej tabeli
echo "<br>";
while($row = mysql_fetch_assoc($ret))
{
    echo ' numer: '.$row['numer'].' Nazwisko: '.$row['nazwisko'].' Waga: '.$row['waga'];
    echo '<br />';
}
if($_GET['sub'] == 'ok')
{
mysql_query('INSERT INTO tabela_kowalski (numer, nazwisko, waga)
VALUES('.$_GET['num'].' , "'.$_GET['nazw'].'" , '.$_GET
['waga'].''))or die("problem z dodaniem wpisu");
    header("Location: z22.php"); // odświeżenie strony
}
?>

<form action="" method="get">
Numer:<input type="text" name="num"><br>
Nazwisko:<input type="text" name="nazw"><br>
Waga:<input type="text" name="waga"><br>
<input name="sub" type="submit" value="ok">
</form>
```

Inny sposób wyświetlania rekordów z bazy:

```
while ($row = mysql_fetch_array($ret))
{
    $y= count($row);
    for($i=0;$i<$y;$i++ )
    {
        echo( $row[$i]." " );
    }
    echo "<br>";
}
```

Zadanie23

Temat: Modyfikacja struktury bazy z użyciem phpMyadmin oraz modyfikacja skryptu-
kasowanie rekordów o określonym ID.

Dodaj nowe pole o nazwie ID przy użyciu PHPAdmin do bazy z zadania
poprzedniego.

wyberz:

opcja Struktura

- Dodaj nowe pole 1
- Na początku tabeli

teraz przycisk Wykonanie

Serwer: localhost ▶ Baza danych: z22 ▶ Tabela: kontakty_kryniewski

Struktura Przeglądanie SQL Szukaj Dodanie Eksport Operacje Wyczyść

	Pole	Typ	Atrybuty	Null	Domyślnie	Dodatkowy	Działanie
<input type="checkbox"/>	numer	tinyint(4)		Nie	0		
<input type="checkbox"/>	nazwisko	text		Nie			
<input type="checkbox"/>	waga	int(11)		Nie	0		

↑ Zaznaczenie wszystkich / Odznaczenie wszystkich Zaznaczone:

Widok do wydruku Propozycja struktury tabeli

Dodanie nowego pola: 1 ☐ Na końcu tabeli ☒ Na początku tabeli ☐ Po numer

wyberz:

nazwa pola

ID

Typ

INT

Długość/Wartość

4

Atrybuty

UNSIGNED

Null

not null

Dodatkowy

auto_increment

zaznacz pole radio

podstawowy

Server: localhost ▶ Baza danych: z22 ▶ Tabela: kontakty_kryniewski

























Pole	Typ	Długość/Wartości*	Atrybuty	Null	Domyślnie**	Dodatkowy
ID	INT	4	UNSIGNED	not null		auto_increment



Zachowanie Lub Pola: 1 Wykonanie

teraz przycisk Wykonanie i potem Zachowanie

Otrzymasz następujący stan struktury bazy danych.



Struktura Przeglądanie SQL Szukaj Dodanie Eksport Operacje Wyczyszczenie

Pole	Typ	Atrybuty	Null	Domyślnie	Dodatkowy	Działanie
<input type="checkbox"/> ID	int(4)	UNSIGNED	Nie		auto_increment	     
<input type="checkbox"/> numer	tinyint(4)		Nie	0		     
<input type="checkbox"/> nazwisko	text		Nie			     
<input type="checkbox"/> waga	int(11)		Nie	0		     

↑ Zaznaczenie wszystkich / Odnaczenie wszystkich Zaznaczone:  

Widok do wydruku Propozycja struktury tabeli

Dodanie nowego pola: 1 Na końcu tabeli Na początku tabeli Po ID Wykonanie

Indeksy:				
Nazwa klucza	Typ	Moc	Działanie	Pole
PRIMARY	PRIMARY	5	 	ID

Utworzenie indeksu dla 1 kolumn Wykonanie

Wykorzystanie przestrzeni:	
Typ	Wykorzystanie
Dane	116 bajtów
Indeks	2 048 bajtów
Sumarycznie	2 164 bajtów

Statystyka rekordów:	
Cecha	Wartość
Format	zmienny
Rekordów	5
Długość rekordu	23

Uzupełnij pole ID liczbami od 1 do liczby ile masz numerów.

Zadanie23a

Temat: Modyfikacja zadania poprzedniego-zastosowanie tabel do wyświetlania danych oraz formularzy.

[Użycie tabeli możesz uzyskać w tym miejscu.](#)

Dopisz się do bazy

ID	Numer	Nazwisko	Waga
63	84	Jan	78
64	86	Ham	76
65	23	Greg	67
66	57	Falkowski	99
67	45	Kowalski-Pokojski	74

Numer:	<input type="text"/>
Nazwisko:	<input type="text"/>
Waga:	<input type="text"/>

Zadanie23b

Temat: Modyfikacja zadania poprzedniego w celu kasowania rekordu na podstawie ID wczytanego z formularza.

Wyrażenia regularne

Wyrażenie regularne to nic innego, jak szablon ciągu. Definiując wyrażenie możemy w dowolny sposób określić format stringa. Narzędzie bardzo przydatne w przypadku sprawdzania poprawności wpisywanych tekstów, jak np. adresu e-mail lub danych osobowych. Wykorzystując wyrażenia regularne możemy również wyszukać oraz podmienić wszystkie podciągi znajdujące się w tekście, które pasują do wpisanej formuły.

W tej lekcji poznamy podstawy konstrukcji wyrażeń regularnych oraz funkcje sprawdzającą występowanie szablonu w ciągu znaków. Zaczniemy od prostego przykładu:

```
<?php

// konstrukcja wyrażenia regularnego
$wyrażenie = '/^[a-z]$/' ;

// preg_match() sprawdza występowanie wyrażenia w ciągu
if(preg_match($wyrażenie, $tekst))
    echo("Tekst zawiera tylko małe litery bez polskich znaków.");
else
    echo("Tekst zawiera dodatkowe znaki.");

?>
```

Przeanalizujmy powyższy kod. Do zmiennej \$wyrażenie przypisaliśmy nasz wzorec w postaci wyrażenia regularnego (omówienie konstrukcji za chwilę). Na chwilę obecną sprawdza on jedynie, czy ciąg zawiera wyłącznie małe litery bez polskich znaków. Następnie funkcja preg_match(\$wyrażenie, \$tekst) sprawdza, czy w ciągu \$tekst znajduje się szablon \$wyrażenie. Zwraca TRUE, jeżeli dopasuje szablon lub FALSE, jeśli nie.
Tworzenie wyrażeń regularnych

Na podstawie wyrażenia wpisanego w poprzednim przykładzie omówię ogólne zasady. Wyrażenie regularne musi znaleźć się pomiędzy ukośnikami "/ ... /". W przypadku przypisania do zmiennej nie możemy zapomnieć o apostrofach! Następnie każde wyrażenie regularne należy rozpocząć znakiem ^ oraz zakończyć znakiem \$. Są to tak zwane znaki początku i końca.

Między tymi granicami wpisujemy żadaną formułę. W naszym przypadku były to małe litery od a do z. By wpisać przedział znaków wykorzystujemy nawiasy kwadratowe []. Możemy między nimi wpisać dowolne przedziały, np.: [a-eA-E0-7], wtedy wzorcem będą tylko takie ciągi, które zawierają w sobie jedynie małe i wielkie litery od a do e oraz cyfry od 0 do 7.

Tym zakończę wprowadzenie do wyrażeń regularnych. W kolejnej lekcji poznamy dodatkowe wzorce formatowania szablonu, takie jak zaprzeczenia, ilość znaków i wiele innych. Zapraszam na następną lekcję!

Zadanie23c

Temat: System rejestracji do konkursu informatycznego.

Po uruchomieniu strony otrzymujemy możliwości:

1)panel założenia konta

Wpisujemy w formularzu:

- Login
- Hasło
- Imię
- Nazwisko
- Szkoła
- Adres
- Telefon

2)panel logowania

Po zalogowaniu (podajemy login i hasło→wpisywane w formularzu) zostajemy przeniesieni do strony z zadaniami.

Rozwiązanie:

1)W PHPAdmin załóż bazę danych o nazwie **logowanie_nazwisko** i tabeli o nazwie **tabela_nazwisko**. Pola utwórz taka aby pasowały do formularza (typ i nazwa) dodaj pole Id z autokrementacją. Zastanów się na optymalizacja typu pól oraz ich długością. Telefon Int (9)

Metoda porównywania napisów to utf8_polish_ci

Aby pole było kluczem podstawowym to:

- wybierz typ pola INT
- zaznacz w kolumnie A_I „ptaszka”
- wybierz w polu Atrybuty UNSIGNED.

2)na stronie wykonaj dwa linki:

*Zakładanie konta (nazwa pliku zakładanie_nazwisko.php)

*Logowanie do strony z zadaniami (nazwa pliku logowanie_nazwisko.php)

Oba linki będą uruchamiać pliki PHP

3)Zakładanie konta:

napisz skrypt w PHP do zakładania konta czyli zapamiętywania danych z formularza. Użyj metody POST.

Zadanie23d

Temat: Zmiana zadania poprzedniego tak aby można było zmieniać hasło.

1)System nie pozwoli założyć konta o istniejącym loginie.

2)Po zalogowaniu na stronie z zadaniami jest przycisk „Zmień hasło” i następujący formularz:

Login:	
Stare hasło	
Nowe hasło	
Potwierdź hasło	

3)Formularz będzie sprawdzany co do złożoności haseł. Przy braku silnego hasła zostaniesz o tym poinformowany. Użyj funkcji preg_match

Zadanie23e

Temat: Zmiana zadania poprzedniego tak aby były dwa rodzaje kont (administracyjne, zwykły użytkownik).

Rozwiązanie:

- 1) Dodaj pole do bazy w którym będziesz trzymał informację czy konto jest administracyjne, czy zwykłe (rozwiązanie zaproponuj sam).
- 2) Utwórz jedno konto o uprawnieniach administratora. Konto to będzie mogło nadawać uprawnienia administratora po podaniu loginu konta, któremu chcesz nadać uprawnienia administracyjne. Konto to będzie mogło teraz nadawać innym uprawnienia administracyjne.
- 3) Będziesz mógł wyświetlić loginy kont zwykłych użytkowników oraz loginy administratorów.
- 3) Konta o uprawnieniach administracyjnych będą mogły kasować użytkowników.

Opcje zakładania bazy.

UNSIGNED – oznacza liczbę bez znaku (l. dodatnie)

NOT NULL – pole nie może być puste

DEFAULT ,x' – wartość domyślna, jeżeli pole nie zostanie uzupełnione przez użytkownika, to zostanie wypełnione wartością domyślną

AUTO_INCREMENT – autoinkrementacja (wartość o jeden wyższa) wartości kolejnych krotek (rekordów)

PRIMARY KEY – określa klucz główny

warto jeszcze określić parametry dla tabeli:

ENGINE=MyISAM – mechanizm składowanie danych, MyISAM jest domyślnym dla MySQL, ew. możemy pominąć ten fragment.

DEFAULT CHARSET=utf8 – kodowanie bazy danych

COLLATE=utf8_polish_ci – metoda porównywania napisów

Metody składowania tabel

MyIsam - podstawowy, domyślny mechanizm składowania MySQL. Jest bardzo dobry i uniwersalny. Nie wiesz co wybrać? - wybieraj to. Ponadto posiada najwięcej narzędzi do naprawiania i sprawdzania tabel oraz tabele mogą być kompresowane i obsługują wyszukiwanie pełnotekstowe.

Memory (HEAP) - mechanizm składowania, który zamiast na dysku twardym przechowuje dane w pamięci RAM, w związku z czym po ew. wyłączeniu serwera dane te kasują się bezpowrotnie. Dane są ponadto hashowane. Idealny do tymczasowego przechowywania danych, np. sesji.

InnoDB - obsługują transakcje, co jest najistotniejszą różnicą charakteryzującą ten typ mechanizmu składowania. Ponadto działają szybciej niż MyISAM i obsługują klucze obce.

Przykład użycia InnoDB

```
CREATE TABLE `obiekty`  
(  
  `id_obiektu` int(10) unsigned NOT NULL auto_increment,  
  `id_autora` int(10) unsigned NOT NULL,  
  `nazwa` char(100) NOT NULL,  
  PRIMARY KEY (`id_obiektu`)  
)  
ENGINE=InnoDB DEFAULT CHARSET=latin2;
```

Transakcje w bazach danych

Czym w zasadzie są transakcje?

Przykład:

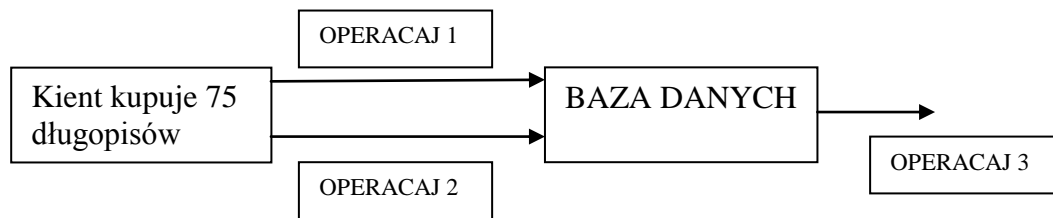
System zarządzający inwentarzem w hurtowni.

Gdy klient kupuje 75 sztuk długopisów, do bazy danych idą zapytania aktualizujące stan kasy, w której kupował oraz zmieniające liczbę długopisów w inwentarzu.

W przypadku gdy długopisów jest już bardzo mało, system automatycznie wysła zamówienie do producenta długopisów na kolejne 2000 długopisów.

Gdyby w połowie wykonywania tych operacji system straciłby zasilanie to np. stan kasy zostałby poprawiony, ale stan długopisów nie zgadzałby się już z rzeczywistością. Mogłoby dojść potem do sytuacji, że kolejny klient chce zakupić 200 długopisów, a w inwentarzu

brakuje 50, ponieważ nie było dostawy długopisów. Dlaczego? Ponieważ przez awarię system nie wykrył, że długopisów jest już mało i zamówienie nie zostało wysłane do producenta.



OPERACJA 1 → zmienia stan kasy o wartość zakupionych długopisów (zmiana wartości pola w bazie danych np. `kasa_posiada`)

OPERACJA 2 → zmienia stan długopisów w magazynie (zmiana wartości pola w bazie danych np. `stan_długopisow`)

OPERACJA 3 → przygotowanie formularza zamówienia do producenta (hurtownia kupuje długopisy), zmiana wartości pól w bazie danych w których zapisane jest jakie zamówienie zostało wystawiane.

Te trzy operacje muszą się wykonać jedna po drugiej. Zablokowanie operacji na bazie nazywamy transakcją

Definicja

Transakcje grupują wyrażenia (instrukcje) SQL, traktując tą grupę jako jedno.

Transakcja może składać się z jednego zapytania lub z wielu, to zależy od charakteru zadania do wykonania.

Transakcje składają się z etapów:

* **Rozpoczęcie transakcji, inaczej inicjalizacja** - po rozpoczęciu transakcji WSZYSTKIE kolejne zapytania SQL są traktowane jako część instrukcji transakcji, następuje ich zgrupowanie do momentu zamknięcia lub anulowania transakcji.

* **Wykonanie transakcji** - wykonują się operacje na bazie danych wchodzące w skład transakcji najczęściej polecenie SQL

* **Zatwierdzenie transakcji** - zapisanie na trwałe wszystkich zmian dokonanych przez instrukcję transakcji. Jeśli nie będziesz kończył transakcji odpowiednio szybko, będzie dochodziło do blokowania całej aplikacji. Poprzez odpowiednio szybko rozumiem sytuację kiedy transakcja jest otwarta TYLKO do momentu, kiedy nie będzie pewne, że jej wynik będzie pozytywny. Nie należy zwlekać z zatwierdzeniem transakcji.

* **Anulowanie transakcji**, lub też wycofanie - kończy transakcję i porzuca wszystkie zmiany jakich miały dokonać instrukcje. Przydaje się w instrukcjach warunkowych aplikacji internetowej. Np. transakcja jest zatwierdzana jeśli przyjdzie potwierdzenie wpłaty pieniędzy na konto bankowe, w przeciwnym wypadku jest anulowana.

Transakcje mają następujące zasady:

ACID,

Atomicity (atomic) →,

Consistency,

Isolation,

Durability - Atomowość,
Spójność,
Izolacja,
Trwałość.

Oznacza to, że każda transakcja wykona się w całości, lub wcale. Podczas jej wykonywania dwie równoległe transakcje nie widzą wprowadzanych przez siebie zmian (zależy od poziomu izolacji) do momentu zatwierdzenia transakcji. Zmiany nie są też widoczne dla pozostałych użytkowników tej bazy danych. Po jej wykonaniu nie zostaną naruszone zasady integralności bazy. A cały system udostępni nam nienaruszone dane po nagłej awarii systemu.

Transakcje można używać samodzielnie, ale ich zastosowanie najczęściej idzie w parze z zastosowaniem kursorów w procedurach składowych.

Procedury składowe są udoskonaleniem struktury języka SQL, są osobnymi programami przetwarzającymi dane na poziomie bazy danych.

W teorii transakcji istnieje też pojęcie **punkt pośredni lub punkt zapisu** (save point), działające na tej samej zasadzie. Chodzi o zapisanie stanu transakcji do danego punktu i możliwość cofania się do tego momentu, wewnątrz transakcji.

To jak zapisywanie stanu gry w czasie wykonywania konkretnej misji w grze komputerowej.

W MySQL mamy do dyspozycji standardowe procedury czyli:

START TRANSACTION (lub BEGIN) - rozpoczynanie transakcji

COMMIT (lub END) - zatwierdzanie

ROLLBACK - anulowanie

SAVEPOINT - punkty pośrednie

ROLLBACK TO SAVEPOINT - cofnięcie do punktu pośredniego

RELEASE SAVEPOINT - usuwa punkt pośredni o podanej nazwie z obecnej transakcji

Typy blokad

Blokady dzielą się na poziome.

Są blokady na poziomie tabel
na poziomie wierszy.

I tutaj wynika pierwsze zagrożenie lub raczej problem w transakcjach.

Bo choć blokady na poziomie tabel można spotkać w większości (we wszystkich?) silników bazodanowych, to blokady wierszy tylko w tych bardziej zaawansowanych technologicznie. Blokowanie CAŁEJ TABELI na czas wykonywania transakcji jest okropnie niewydajne. A co jeśli z tabeli korzysta kilku, albo o zgrozo, kilkunastu użytkowników, to nie będą mogli oni dokonać nic w całej tabeli do czasu, aż nie zakończy się tamta pierwsza transakcja. Teraz już widzisz jak ważne jest zamykanie transakcji kiedy już tylko można tego dokonać.

Blokowanie wierszy jest o wiele lepsze. Ponieważ blokuje tylko jeden rekord i jest mniejsze prawdopodobieństwo zastoju, lub w przypadku źle napisanych aplikacji internetowych, blokady całej strony.

Cofanie transakcji.

Jak już dobrze wiesz transakcje umożliwiają cofanie się do punktu pośredniego lub do stanu przed rozpoczęciem grupy zapytań.

Jak ta część mechanizmu współgra? Otóż na potrzeby tego projektu wynaleziono takie

pojęcie jak dziennik transakcji. Nie jest on niczym innym jak dziennikiem logów aka zmian. Dzięki temu dziennikowi możliwe jest cofanie transakcji JUŻ ZATWIERDZONYCH. Dzienniki taki przechowują zapis wszystkich transakcji zazwyczaj od momentu ostatniego backupu bazy. Jak to działa?

Założmy że mamy backup bazy z przed 3 dni i nastąpiła awaria, dajmy na to dysku. W takim wypadku musimy przywrócić stan z przed 3 dni.

Przywrócenie stanu z przed awarii jest banalnie proste po przywróceniu kopii bazy. Musimy potem tylko przywrócić zapisane transakcji w dzienniku transakcji, który na nowo wykona zapisane zapytania SQL.

Jeśli awaria nastąpiła z naszej winy, to następnym krokiem powinno być czynność zapobiegająca powtórzeniu się sytuacji.

PUNKTY PRZYWRACANIA

Podczas wykonywania transakcji możemy określić również punkty zapisu, tzw. **SAVEPOINT**. Po słowie kluczowym **SAVEPOINT** używamy identyfikatora.

SAVEPOINT nazwa_punktu_przywracania;

Dzięki zdefiniowaniu identyfikatorów punktów zapisu możemy cofnąć zmiany przeprowadzane przez transakcję do określonego za pomocą identyfikatora punktu. Aby cofnąć zmiany do zdefiniowanego wcześniej punktu **SAVEPOINT**, używamy polecenia:

ROLLBACK TO SAVEPOINT nazwa_punktu_przywracania;

Aby usunąć punkt przywracania używamy polecenia:

RELEASE SAVEPOINT nazwa_punktu_przywracania;

KONTROLA TRANSAKCI

Podczas wykonywania transakcji możliwy jest podział ich na mniejsze części za pomocą słowa kluczowego **SAVEPOINT**. Całość transakcji można nadal wycofać używając słowa kluczowego **ROLLBACK**.

Transakcje na przykładzie PHP

```
<?php
//Łączenie z bazą danych.
$pdo = new PDO('mysql:host=host;dbname=nazwaBazy', 'Uzytkownik', 'Haslo');
//Ustawienie poziomu raportowania błędów.
$pdo->setAttribute( PDO::ATTR_ERRMODE, PDO::ERRMODE_WARNING );
//rozpoczynanie transakcji
try
{
$pdo->query('start transaction');
//kto i co kupuje
$ klient = 2;
$przedmiot = 1;
//pierwsze zapytanie zabierające pieniądze z konta.
$stmt = $pdo->prepare('UPDATE users.money SET users.money=users.money-shop.cost
FROM users, shop WHERE shop.id='.$przedmiot.' AND users.id='.$klient.'');
$stmt->execute();
$stmt->closeCursor();
unset($stmt);
//drugie zapytanie dodające przedmiot.
$stmt = $pdo->prepare('INSERT INTO items(what, owner) VALUES('.$przedmiot.',
```

```

'$.klient.'));
$stmt -> execute();
$stmt -> closeCursor();
//bez błędów - zatwierdzenie.
$pdo -> query('commit');
die('powiodło się');
}
catch(PDOException $e)
{
//W razie jakiegokolwiek błędu, wyskoczy wyjątek i od razu razem z nim transakcja zostanie
wycofana.
$pdo -> query('rollback');
//Wyświetlenie błędu.
die('Nie powiodła się! Błąd:'. $e->getMessage());
}
?>

```

Zadanie23e1

Temat: Treść polecenia: Baza danych, zakładanie, wpisywanie danych, obsługa transakcji.

Masz trzy możliwości rozwiązania zadania.

1	Wykonaj dla całości skrypt mysql
2	Wykonaj skrypt mysql z użyciem PHPAdmin dla zakładania bazy i wpisywania danych. Dokonaj eksportu skryptu do pliku tekstowego opcją Eksport w PHPAdmin i dopisz resztę instrukcji w skrypcie.
3	Całość napisz w PHP.

1)założenie bazy o nazwie **hurtownia_nazwisko**, tabela o **nazwie tabela_hurt_nazwisko** następujących polach:

id_nazwisko (trzy litery) (klucz podstawowy)

imie_nazwisko (trzy litery)

nazwisko_nazwisko (trzy litery)

wiek_nazwisko (trzy litery)

Aby widoczne były polskie znaki użyj DEFAULT CHARSET=utf8.

wymuszeniu silnika InnoDB (tzw składowanie tabel)

2)Wpisz dane

id	imie	nazwisko	wiek
1	Jan	Nowak	24
2	Ewa	Werner	30
3	Marcin	Tracz	33
4	Łucja	Kowalska	26
5	Paweł	Nowak	23

3)wyświetl wszystkie dane z bazy

4)Zmodyfikuj dane w tabeli tak, aby wiek wszystkich pracowników o id większym od 2 zmienić na 18. Obejmij tą instrukcję transakcją.

START TRANSACTION;

UPDATE pracownicy SET wiek = 18 WHERE id > 2;

Uwaga: zmień nazwę tabeli z *pracownicy* na Twoją nazwę oraz nazwy pól.

- 5) Wyświetl wszystkie dane z bazy
- 6) Wycofaj zmianę, stosując słowo kluczowe ROLLBACK.
- 7) Wyświetl wszystkie dane z bazy. Czy dane powróciły do
- 8) Zakończ transakcję.
- 9) Obejmij transakcją instrukcję dopisywania nowego pracownika (szóstego). Dopisz siebie i zatwierdź transakcję.
- 10) Wyświetl wszystkie dane z bazy

Zadanie23e2

Treść polecenia: Baza danych, która będzie obsługiwać transakcje wraz z punktami przywracania. Posługujemy się bazą z zadania poprzedniego

- 1) Obejmij transakcją instrukcję:
zmień imiona wszystkich pracowników o id>1 na Marcin.
START TRANSACTION;
UPDATE *pracownicy* SET imie = 'Marcin' WHERE id > 1;
- 2) Zdefiniuj punkt przywracania o nazwie *nazwisko_punkt_1*.
SAVEPOINT *punkt_pierwszy*;
- 3) Obejmij transakcją instrukcję:
zmień imiona wszystkich pracowników o id>1 na nazwiska Zagłoba.
- 4) Zdefiniuj punkt przywracania o nazwie *nazwisko_punkt_2*.
- 5) Wyświetlamy zawartość tabeli:
- 6) Cofamy zmiany do poprzednio zdefiniowanego punktu przywracania:
ROLLBACK TO SAVEPOINT *punkt_pierwszy*;
- 7) Wyświetlamy zawartość tabeli
- 8) Całość transakcji można nadal wycofać używając słowa kluczowego ROLLBACK. Zrób to. Przekonasz się, że zawartość tabeli wróci do stanu przed rozpoczęciem transakcji.

Zadanie23e3

Treść polecenia: Wykonywanie transakcji bankowych z użyciem TRANSAKCJI.

Wykonaj

- 1) Użyj panelu logowania z poprzednich zadań.
- 2) Zmień bazę dodając pola:
Numer_konta_nazwisko
Stan_konta_nazwisko
- 3) Wykonaj formularz przelewu czyli:

????????????????????????????????

- 4) Po transakcji przelewu będzie przycisk ZATWIERDŹ

Grafika w bazie MySQL za pomocą funkcji z biblioteki PHP_GD2.

W bazie MySQL istnieje specjalny typ pola, służący do przechowywania obiektów binarnych. Pole typu BLOB, bo o nim tu mowa, ma aż 4 odmiany mogące przechowywać obiekty o różnych wielkościach:

- TINYBLOB (1 B),
- BLOB (64 kB),
- MEDIUMBLOB (16 MB),
- LONGBLOB (4 GB),

Zadanie23f

Temat: Baza danych do przechowywania grafik z użyciem w całości w PHP.

Tabela w MySQL zawierająca 5 pól:

- pole klucza typu INT,
- pole typu MEDIUMBLOB na zawartość obrazka,
- pole typu CHAR w celu przechowania typu MIME tego obrazka,
- pole typu CHAR do przechowywania jego nazwy
- pole typu INT do przechowywania jego rozmiaru.

Rozwiązanie:


```
<?PHP
```

```
//Nawiązujemy połączenie z serwerem bazy MySQL  
$conn=mysql_connect("localhost","root","")  
or die ('Błąd połączenia z bazą MySQL: '.mysql_error());  
// "localhost","user","hasło" - -> wpisz takie aby uzyskać połączenie  
// z bazą na Twoim kompie na administratora
```

```
//Tworzymy nową bazę  
$result1=mysql_query("create database if not exists naszabaza",$conn)  
or die('Błąd podczas tworzenia nowej bazy: '.mysql_error());  
// "naszabaza" - -> nazwa bazy nazwisko_obrazki
```

```
//Wybieramy do pracy bazę którą przed chwilą stworzyliśmy  
$result2=mysql_select_db("naszabaza",$conn)  
or die ('Błąd podczas wyboru bazy: '.mysql_error());
```

```
//Tworzymy zapytanie tworzące tabelę "obrazy_nazwisko",  
//która będzie składać się z pięciu pól:  
// "ID_obrazy_nazwisko" typu INT, "obrazek_nazwisko" typu MediumBlob,  
// "typ_nazwisko" typu Char, "nazwa_nazwisko" typu Char i
```

```
"rozmiar_nazwisko" typu // INT.  
//zmień poniżej nazwy pól
```

```
$sql1='Create table obraz (
    ID_obrazy Int UNSIGNED NOT NULL AUTO_INCREMENT,
    obrazek MediumBlob NOT NULL,
    typ Char(60) NOT NULL,
    nazwa Char(255) NOT NULL,
    rozmiar Int UNSIGNED NOT NULL,
    UNIQUE (ID_obrazy),
    Primary Key (ID_obrazy))
ENGINE = MyISAM;';
```

```
//NOT NULL zabezpiecza bazę przed dodawaniem pustych lub //niekompletnych  
rekordów.
```

```
//UNIQUE to określenie pola które jest unikatowe.
```

```
//Wykonujemy zapytanie $sql1  
$result3=mysql_query($sql1,$conn)  
or die ('Błąd wykonania zapytania 1: '.mysql_error());
```

```
?>
```

W ten oto sposób utworzyliśmy tabelkę z polem MEDIUMBLOB.

Zadanie23g

Temat: Twórzmy w HTML'u formularz do wyboru i zapisu grafiki.

```
<!-- plik upload1_nazwisko.html -->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-type"
  CONTENT="text/html; charset=iso-8859-2">
</HEAD>
<BODY>

<FORM ACTION="ladowanie_nazwisko.php" METHOD="POST"
  ENCTYPE="multipart/form-data">
  <INPUT TYPE="HIDDEN" NAME="MAX_FILE_SIZE" VALUE="8388608">
  <INPUT TYPE="FILE" NAME="plik" VALUE="" size="30">
  <INPUT TYPE="SUBMIT" VALUE="Wczytaj">
</FORM>

</BODY>
</HTML>
```

Uwagi:

- Atrybut ACTION odnosi się do skryptu PHP, który wczyta plik do bazy,
- Atrybut METHOD jest ustawiony na "POST",
- Dodatkowy atrybut ENCTYPE jest ustawiony na "multipart/form-data", co pozwoli przeglądarce rozpoznać typ pliku i prawidłowo wysłać go do serwera WWW,
- Ukryte pole MAX_FILE_SIZE określa maksymalny rozmiar przesyłanego pliku określony w bajtach. Wartość ta nie może przekraczać wartości ustawionej w pliku PHP.INI przy dyrektywie "upload_max_filesize". Wartość dyrektywy "upload_max_filesize" z kolei nie może przewyższać wartości dyrektywy "post_max_size".
- Pole formularza typu "FILE" służy do wskazania ścieżki do naszego pliku. Pole to musi mieć unikatową nazwę. W przykładzie jego nazwa to "plik".
- Przycisk typu "SUBMIT" służy do inicjacji wysłania danych z formularza.

Zadanie23h

Temat: Skryptu PHP, który wczytuje grafikę do bazy MySQL.

```
<?PHP
//plik ladowanie_nazwisko.php

if (isset($_FILES['plik']))
{
  if ($_FILES['plik']['error']==UPLOAD_ERR_OK)
  {
```

```

$filename=$_FILES['plik']['name']; //Nazwa wysłanego pliku
$filetype=$_FILES['plik']['type']; //Typ wysłanego pliku
$filesize=$_FILES['plik']['size']; //Rozmiar wysłanego pliku
$filesrc=$_FILES['plik']['tmp_name']; //Ścieżka do pliku
//tymczasowego na serwerze

if ($filetype=="image/png" || $filetype=="image/x-png" ||
    $filetype=="image/gif" || $filetype=="image/jpeg" ||
    $filetype=="image/pjpeg")
{

    $plik=fopen($filesrc,"r"); //otwórz ten plik w trybie do odczytu

    $mysqlplik = addslashes(fread($plik,$filesize));
    //odczytaj go do końca, dodaj znaki "\" przed apostrofami
    //i cudzysłowami. Całość zapisz w zmiennej $mysqlplik.

    fclose($plik); //zamykamy plik
    unlink($filesrc); //i kasujemy go, bo już nie jest potrzebny.

    $mysqlfiletype = addslashes($filetype); //dodajemy znaki "\"
    //przed apostrofami i cudzysłowami do wartości zmiennej
    //$filetype, w której został zdefiniowany typ MIME naszego
    //pliku. Wynik zapisujemy w zmiennej $mysqlfiletype.

    $mysqlfilename = addslashes($filename); //dodajemy znaki "\"
    //przed apostrofami i cudzysłowami do wartości zmiennej
    //$filename, w której została zapisana nazwa naszego pliku.
    //Wynik zapisujemy w zmiennej $mysqlfilename.

    //Nawiązujemy połączenie z serwerem bazy MySQL
    $conn=mysql_connect("localhost","user","hasło")
    or die ('Błąd połączenia z bazą MySQL: '.mysql_error());

    //Wybieramy bazę w której mamy tabelę z polem blob
    $result2=mysql_select_db("naszabaza",$conn)
    or die ('Błąd podczas wyboru bazy: '. mysql_error());

    //Tworzymy zapytanie zapisujące do tabeli "obrazy" plik,
    //w naszym przypadku obrazek, jego typ MIME, nazwę
    //oraz rozmiar w bajtach.
    $sql1="INSERT INTO obraz (ID_obraz,obrazek,typ,nazwa,rozmiar)
    VALUES ('','$mysqlplik','$mysqlfiletype','$mysqlfilename','$filesize')";

    //Wykonujemy zapytanie $sql1
    $result1=mysql_query($sql1,$conn)
    or die ('Błąd wykonania zapytania 1: '. mysql_error());

}
else {die("Nieobsługiwany format pliku !!!"); exit;}

```

```

}
else {die("Błąd podczas wysyłania pliku !!!"); exit;}

}
else {die("Nie wskazano pliku do wysłania !!!"); exit;}

?>

```

W tym listingu należy zwrócić uwagę na przedstawione poniżej szczegóły:

- Wynik funkcji "isset(\$_FILES['plik'])", która sprawdza czy podaliśmy ścieżkę do pliku który ma być później wczytany. Jeśli funkcja zwróci wartość "false", będzie to oznaczało że nie wskazaliśmy pliku do wczytania.

- Atrybut "\$_FILES['plik']['error']", który powinien być równy predefiniowanej w PHP stałej "UPLOAD_ERR_OK", z tego atrybutu dowiadujemy się czy upload przebiegł prawidłowo, czy został wczytany właściwy plik, czy jego rozmiar nie przekracza wartości określonej przez element "MAX_FILE_SIZE" itp.

- W naszym przypadku sprawdzamy typ MIME przesyłanego pliku, żeby dowiedzieć się czy jest to akceptowany przez nas format obrazka. Ja ograniczyłem się do formatów JPG, PNG i GIF.

- Dopiero gdy wszystkie powyższe warunki zostaną spełnione, otwieramy w trybie do odczytu przesłany plik znajdujący się na serwerze jako plik tymczasowy o lokalizacji ustalonej przez serwer, a wiadomej nam z atrybutu "\$_FILES['plik']['tmp_name']".

- Odczytu zawartości pliku dokonujemy funkcją "fread()", w której jako pierwszy parametr podajemy wskaźnik do struktury plikowej obrazka, a jako drugi parametr podajemy znany nam z atrybutu \$_FILES['plik']['size'] rozmiar pliku.

- Odczytaną zawartość pliku musimy jeszcze na potrzeby jej użycia w zapytaniu MySQL specjalnie "oszlifować" funkcją "addslashes()", która wstawi nam znaki "\"" przed znaki apostrofu ['], cudzysłowia ["] oraz slash[\\]. Jest to konieczne, bo inaczej PHP niepoprawnie by interpretował te znaki w trakcie wykonywania zapytania zapisującego do bazy zawartość pliku. Po prostu uznałby że zawartość pliku kończy się na jednym z tych znaków.

- Po odczytaniu zawartości pliku, zamykamy go funkcją "fclose()" i kasujemy funkcją "unlink()". Jeśli sami go nie usuniemy, serwer powinien to zrobić za nas zaraz po zakończeniu działania skryptu "upload2.php".

- Dodajemy slashy do zmiennej \$filetype i zapisujemy wynik w zmiennej \$mysqlfiletype. I tu muszę zaznaczyć że jest to zabieg tak na wszelki wypadek, bo nie widziałem jeszcze typu MIME, który w nazwie zawierałby znaki ['],['],[\\].

- Następnie inicjujemy połączenie z serwerem MySQL i wybieramy bazę na której chcemy pracować.

- Potem zostało już tylko wstawienie do tabeli "obrazy" nowego rekordu, który w drugiej komórce będzie zawierał nasz obrazek wczytany tym razem z utworzonej przez nas zmiennej \$mysqlplik, zawierającej zawartość przesyłanego pliku ze wspomnianymi wcześniej znakami "\", w trzeciej komórce będzie zawierał definicję typu MIME wczytywanego pliku również ze wspomnianymi wcześniej znakami "\" jeśli będzie to konieczne, w kolejnej komórce pojawi się nazwa pliku "oszlifowana" funkcją "addslashes()" oraz w ostatniej komórce znajdzie się rozmiar przesyłanego pliku liczony w bajtach.

- Uwaga !!! Znaków "\" potrzebował jedynie sam PHP do wysłania prawidłowego zapytania do serwera MySQL. W bazie MySQL została zapisana oryginalna zawartość obrazka, bez dodanych znaków "\" przed cudzysłowiami, apostrofami i slashami.

Zadanie23i

Temat: Skryptu PHP, który wyświetla obrazek z bazy MySQL.

Sposób 1 wykorzystujący możliwości biblioteki PHP GD2.

```
<?PHP
```

```
$conn=mysql_connect("localhost","user","hasło")  
or die ('Błąd połączenia z bazą MySQL: '.mysql_error());
```

```
//Wybieramy bazę w której mamy tabelę z polem blob  
$result1=mysql_select_db("naszabaza",$conn)  
or die ('Błąd podczas wyboru bazy: '.mysql_error());
```

```
$sql1="select * from obrazy where ID_obrazy='1'";
```

```
$result2=mysql_query($sql1,$conn)  
or die ('Błąd wykonania zapytania 1: '.mysql_error());
```

```
//tworzymy tablicę asocjacyjną $row i wczytujemy do niej  
//dane z wybranego rekordu tabeli. W naszym przypadku  
//jest to rekord 1.  
$row = mysql_fetch_assoc($result2);
```

```
//Z tablicy $row wydobywamy zawartość komórki "obrazek"  
//i zapisujemy ją do zmiennej $grafika.  
$grafika=$row['obrazek'];
```

```
//Na podstawie strumienia danych przechowywanych w polu BLOB,  
//traktowanych w tej chwili jako ciąg znaków tworzymy obrazek  
//w pamięci serwera i przypisujemy go do wskaźnika $obraz.  
$obraz=imageCreateFromString($grafika);
```

```
//Generujemy w przeglądarce obrazek, w tym przypadku JPG.  
imageJPEG($obraz);
```

```
//Uwalniamy z pamięci wskaźnik $obraz.  
ImageDestroy($obraz);
```

?>

Jak widać nie jest to specjalnie skomplikowana metoda. Kryje ona w sobie wiele możliwości, ale niestety ma też pewne wady.

Możliwości jest dużo. Dzięki wykorzystaniu PHP_GD2 można właściwie rysować (!) za pomocą PHP, tworzyć tzw. znaki wodne na obrazkach, zmieniać swobodnie ich rozmiar, ustawiać indeks koloru przeźroczystego, konwertować paletę kolorów do 256, 16 i 2 i wiele, wiele więcej. Dodatkowo odpowiednie nagłówki HTTP są wysyłane automatycznie.

Wad jest zdecydowanie mniej, ale czasem potrafią być dokuczliwe. Chodzi mi przede wszystkim o obsługę tylko plików graficznych w formatach *.JPG, *.GIF, *.PNG i *.WBMP (nie mylić z *.BMP). Istnieją jeszcze 2 inne formaty o rozszerzeniach *.XBM i *.XPM, ale nie warto sobie zawracać nimi głowy.

Generalnie biblioteka ta jest w dwóch wersjach: GD i GD2. Pociąga to za sobą kilka niezgodności. Biblioteka GD nie potrafi wyświetlić obrazka w formacie PNG, a GD2 nie potrafi wyświetlić obrazka w formacie GIF. Obie potrafią wczytywać do pamięci obrazki za pomocą funkcji "ImageCreateFromGIF()" i "ImageCreateFromPNG()", ale pierwsza nie obsługuje funkcji "ImagePNG()", a druga "ImageGIF()". Dodatkowo pierwsza wersja GD, mimo że w pełni obsługuje format GIF nie wczyta prawidłowo animowanego GIF'a. Wersja GD2 nie generuje obrazków GIF więc nie ma nad czym się rozwodzić. Reasumując PHP_GD i PHP_GD2 obsługują tylko obrazy statyczne.

Kolejną niedogodnością jest fakt że określone funkcje szerzej omówione w manualu PHP nie działają poprawnie ze starszymi wersjami PHP. Nie powinno nam to jednak przeszkadzać. Wystarczy pracować na PHP 4.3 i wszystkie funkcje będą działać.

Niektórzy twierdzą, że nie powinno się przechowywać grafiki w bazie, bo grafika znacznie obciąża serwer MySQL. Duża ilość obrazków i innych plików faktycznie potrafi znacznie obciążyć bazę. Czasami jednak trzeba się oprzeć na bazie danych.

Obiecałem omówić zapis do bazy, odczyt z bazy i wyświetlanie grafiki wykorzystując PHP_GD2. Słowa dotrzymałem. W tym artykule na tym koniec :-).

Sposób 2→ tradycyjny.

<?PHP

```
//Nawiązujemy połączenie z serwerem bazy MySQL  
$conn=mysql_connect("localhost","user","hasło")  
or die ('Błąd połączenia z bazą MySQL: '.mysql_error());
```

```

//Wybieramy bazę w której mamy tabelę z polem blob
$result1=mysql_select_db("naszabaza",$conn)
or die ('Błąd podczas wyboru bazy: '.mysql_error());

//Tworzymy zapytanie wybierające z bazy MySQL zapisaną w niej
//zawartość wczytanego wcześniej pliku wraz z jego typem MIME,
//nazwą i rozmiarem.
$sql1="select * from obrazy where ID_obrazy='1'";

//Wykonujemy zapytanie $sql1
$result2=mysql_query($sql1,$conn)
or die ('Błąd wykonania zapytania 1: '.mysql_error());

//tworzymy tablicę asocjacyjną $row i wczytujemy do niej
//dane z wybranego rekordu tabeli. W naszym przypadku
//jest to rekord 1.
$row = mysql_fetch_assoc($result2);

//Z tablicy $row wydobywamy zawartość komórki "obrazek"
//i zapisujemy ją do zmiennej $grafika.
$grafika=$row['obrazek'];

//Z tablicy $row wydobywamy zawartość komórki "typ"
//i zapisujemy ją do zmiennej $typpliku.
$typpliku=$row['typ'];

//Wysyłamy do przeglądarki nagłówek HTTP, informujący
//przeglądarkę że wynik który za chwilę wyświetlimy to
//obrazek w formacie określonym przez zmienną $typpliku.
header("Content-type:$typpliku");

//Drukujemy zawartość zmiennej $grafika,
//czyli wyświetlamy zawartość naszego obrazka
print $grafika;

?>

```

W przypadku powyższego kodu, należy między innymi zwrócić uwagę na sam sposób wyświetlenia obrazka. Mianowicie, najpierw wysyłamy do przeglądarki nagłówek HTTP z definicją typu MIME naszego obrazka, a dopiero potem drukujemy zawartość obrazka. Dzięki wysłaniu nagłówka z definicją określonego typu MIME, przeglądarka prawidłowo zinterpretuje wyświetlane dane będące zawartością obrazka i wyświetli je jako obrazek. W przeciwnym wypadku, gdybyśmy takiego nagłówka nie ustawili, zawartość obrazka zostałaby wyświetlona jako niezrozumiały tekst.

Pokazany tutaj sposób wyświetlania można również zastosować do wszystkich innych typów plików, nie tylko obrazków. Można przykładowo podczas procedury

uploadu pliku do bazy znieść ograniczenie co do typu MIME przesyłanego pliku i już w bazie można zapisywać dosłownie wszystko.

Należy jednak zwrócić uwagę na dość istotny szczegół. W przypadku gdy będziemy próbowali wyświetlić zawartość pliku o rozszerzeniu nieznanym przeglądarce jako element współtworzący stronę WWW (np. *.zip, *.exe, *.notype), przeglądarka rozpozna taki plik oczywiście po definicji typu MIME danego pliku w wysłanym nagłówku i wyświetli okienko w rodzaju "Zapisz jako".

W przypadku gdy będziemy próbowali wyświetlić plik w formacie nieznanym przeglądarce jako element tworzący stronę WWW, ale przeglądarka będzie posiadała określony plugin do jego wyświetlania, wtedy przeglądarka wyświetli zawartość takiego pliku w tym samym oknie ale w środowisku jakie oferuje nam dany plugin. Coś takiego ma miejsce gdy np. otwieramy dokumenty *.PDF. Zwykle uruchamia się wtedy plugin programu Adobe Acrobat Reader.

Można jednak zmusić przeglądarkę do wyświetlania okienka "Zapisz jako" w przypadku każdego formatu pliku, również *.html, *.txt, *.jpg, *.png, *.gif, *.pdf, *.doc, itd. Można to osiągnąć poprzez wysłanie do przeglądarki dodatkowych nagłówków HTTP. Wygląda to następująco:

```
<?PHP
```

```
$conn=mysql_connect("localhost","user","hasło")  
or die ('Błąd połączenia z bazą MySQL: '.mysql_error());
```

```
//Wybieramy bazę w której mamy tabelę z polem blob  
$result1=mysql_select_db("naszabaza",$conn)  
or die ('Błąd podczas wyboru bazy: '.mysql_error());
```

```
$sql1="select * from obrazy where ID_obrazy='1'";
```

```
$result2=mysql_query($sql1,$conn)  
or die ('Błąd wykonania zapytania 1: '.mysql_error());
```

```
//tworzymy tablicę asocjacyjną $row i wczytujemy do niej  
//dane z wybranego rekordu tabeli. W naszym przypadku  
//jest to rekord 1.  
$row = mysql_fetch_assoc($result2);
```

```
//Z tablicy $row wydobywamy zawartość komórki "obrazek"  
//i zapisujemy ją do zmiennej $grafika.  
$grafika=$row['obrazek'];
```

```
//Z tablicy $row wydobywamy zawartość komórki "typ"  
//i zapisujemy ją do zmiennej $typpliku.  
$typpliku=$row['typ'];
```

```
//Z tablicy $row wydobywamy zawartość komórki "nazwa"  
//i zapisujemy ją do zmiennej $nazwapliku.  
$nazwapliku=$row['nazwa'];
```



```
//Z tablicy $row wydobywamy zawartość komórki "rozmiar"
//i zapisujemy ją do zmiennej $rozmiarpliku.
$rozmiarpliku=$row['rozmiar'];

//Wysyłamy do przeglądarki nagłówek HTTP, informujący
//że wynik który za chwilę będziemy chcieli pobrać to
//obrazek w formacie określonym przez zmienną $typpliku.
header("Content-type:$typpliku");

//Wysyłamy do przeglądarki nagłówek HTTP, podający
//przeglądarcie rozmiar pobieranego pliku określony
//przez zmienną $rozmiarpliku.
header("Content-length: $rozmiarpliku");

//Wysyłamy do przeglądarki nagłówek HTTP, wymuszający
//na przeglądarce wyświetlenie okienka "Zapisz jako".
//Dodatkowo w nagłówku tym podajemy domyślną docelową
//nazwę pliku zdefiniowaną przez zmienną $nazwapliku
header("Content-Disposition: attachment; filename=$nazwapliku");

//Drukujemy zawartość zmiennej $grafika,
//ale tym razem nie zostanie ona wyświetlona.
//Będziemy mogli ją zapisać lub otworzyć lokalnie.
print $grafika;

?>
```

Niestety przedstawione tutaj rozwiązanie ma pewną wadę. Nie wszystkie przeglądarki z nim sobie poradzą. W manualu PHP czytamy:

"W Microsoft Internet Explorer 4.01 jest błąd, który uniemożliwia wykorzystanie tego mechanizmu. Nie ma na to rozwiązania. Błąd, który zahacza o ten mechanizm, jest także w Microsoft Internet Explorer 5.5, jednak da się go ominąć aktualizując przeglądarkę poprzez Service Pack 2 lub późniejszy."

Z moich doświadczeń wynika że z tym mechanizmem nie radzą sobie również przeglądarki Opera 6, Mozilla 0.9.7 i Netscape 6 lub ich starsze wersje. Internet Explorer 6, Opera 7 i Mozilla 1.2 radzą sobie doskonale. Nie testowałem przeglądarki Mozilla w wersjach (0.9.7,1.1> oraz Netscape nowszej niż wersja 6. Tyle jeśli chodzi o sposób tradycyjny, w którym wysyłaliśmy do przeglądarki nagłówki HTTP.

Zadanie24a

Temat: Użycie sesji do licznika odwiedzin strony.

Lista kroków dla stworzenia licznika:

KROK1

inicjacja sesji

KROK2

sprawdź instrukcją **isset(\$_SESSION['licznik_nazwisko'])** czy zmienna **licznik_nazwisko** sesyjna istnieje jeśli tak idź do KROK3 jeśli nie do KROK4

KROK3

zwiększ zmienną **licznik_nazwisko** o jeden instrukcją

\$_SESSION['licznik_nazwisko']++;

idź do KROK5

KROK4

przypisz zmiennej **licznik_nazwisko** wartość 0 instrukcją

\$_SESSION['licznik_nazwisko']=0;

KROK5

wyświetl wartość licznika instrukcją

echo('Licznik:'. \$_SESSION['licznik_nazwisko']);

Sesje w PHP

Kontrola sesji udostępnia możliwość:

1. śledzenia poczynañ odwiedzającego, podczas pojedynczej wizyty na naszej stronie. Innymi słowy, umożliwia rozpoznanie, czy użytkownik dopiero co wszedł na stronę, czy przechodzi z innej podstrony serwisu.
2. Dodatkowo, możemy tworzyć i usuwać zmienne, których istnienie jest ściśle powiązane z pojedynczą sesją (wizytą). Ich żywot kończy się wraz z zamknięciem okna przeglądarki lub przejścia na inną stronę.

Rozpoczynanie sesji

```
<?php
```

```
// inicjacja sesji
```

```
session_start();
```

```
?>
```

Tworzenie zmiennych sesji

Po wywołaniu funkcji `start_session()`, PHP udostępnia nam tablicę superglobalną `$_SESSION`. Od tej pory możemy tworzyć tymczasowe zmienne, które będą przechowywane w tej właśnie tablicy. Jej wartości pobierane są przez funkcję `start_session()` za każdym razem, gdy odświeżamy stronę.

przykład utworzenia zmiennej sesyjnej:

```
<?php
```

```
session_start();
```

```
// utworzenie zmiennej sesji
```

```
$_SESSION['zm_sesji'] = 1;
```

```
?>
```

Przechowywanie zmiennych sesji

Domyślnie, zmienne sesyjne są przechowywane na komputerze użytkownika w postaci **cookies**. Jest to niewielki plik tekstowy. Przeglądarka tworzy specjalny folder

na cookies. W pliku cookies zapisany jest unikalny ciąg znaków, tzw identyfikator. Z tym identyfikatorem skojarzone są wszystkie zapisane na serwerze zmienne sesyjne. Po ponownym otwarciu skryptu przeglądarka przesyła identyfikator sesji do serwera , który dzięki temu może odnaleźć związaną z nim określoną zmienną sesyjną.

Zadanie24

Temat: Portal miniksiążka telefoniczna.

Wygląd strony:

| | |
|---|---|
| Miniksiążka telefoniczna | |
| Motto: <i>Trzeba wiedzieć co się mówi, a nie mówić co się wie.</i> | |
| [Szukaj] [Lista] [Informacje] | |
| Strona szukaj | <div>Dziś jest:
18.05.2014 r.

Jest to:
138
dzień w roku.

Imieniny:
Alicji, Edwina,
Eryka</div> |

Etap1

1) Zakładamy folder o nazwie z24

2) Do tego folderu wgrywamy pliki (nazwy zmień jak poniżej):

- imieniny.php → dwa etapy1 i 2 z zadania 19,
- motto.php → przykład2 z zadania 19d,
- kalendarzyk.php → etapy3 z zadania 19 include(zmień nazwę).

Etap2

Wykonaj stronę o nazwie index.php:

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="pl">
<head>
  <meta http-equiv="Content-type" content="text/html; charset=iso-8859-2" />
  <title>Twoje nazwisko-Mini książka telefoniczna</title>
<style type="text/css">
  • Ciało (treść) dokumentu { definicja patrz tabela }
  • Znacznik początku tabeli, Znacznik początku komórka danych w tabeli
  • Klasa o nazwie tab { definicja patrz tabela }
  • Nagłówek o największej wielkości liter { definicja patrz tabela }
  • Nagłówek o drugiej co do wielkości liter { definicja patrz tabela }
  • pseudoklasa linków odwiedzonych { definicja patrz tabela }
  • pseudoklasa linku wskazywanego przez mysz { definicja patrz tabela }
  • Linia rozdzielająca { definicja patrz tabela }
  • Klasa o nazwie motto{ definicja patrz tabela }
</style>
</head>
<body>
  <h1>Miniksiążka telefoniczna</h1>
  <div class="motto">Motto: <?php include('motto.php') ?></div>
  <table align="center" class="tab">
    <tr>
      <td valign="top">
        <!-- menu -->
        <div align="center">
          [ <a href="index.php?pokaz=szukaj">Szukaj</a> ]
          [ <a href="index.php?pokaz=lista">Lista</a> ]
          [ <a href="index.php?pokaz=info">Informacje</a> ]
          <hr />
        </div>
        <!-- treść główna -->
        <?php include('sys.php'); ?>
        <!--koniec treści głównej -->
      </td>
      <td width="120" valign="top">
        <div style="text-align:center;background:#b1ffb1;border:1px solid green;">
          <?php include('kalendarzyk.php'); ?>
        </div>
      </td>
    </tr>
  </table>
</body>
</html>

```

| Opis | Nazwa w HTML (CSS) | definicja | Rodzaj W CSS |
|---|--------------------|---|-----------------------|
| Ciało (treść) dokumentu | body | {kolor tła:#ddffdd; rodzaj czcionki:arial; wielkość czcionki:12pikseli;} | Selektor prosty |
| Znacznik początku tabeli oraz Znacznik początku komórka danych w tabeli | table,? | {rodzaj czcionki:arial; wielkość czcionki:12pikseli;} | Grupowanie selektorów |
| Klasa o nazwie Tab | .??? | {obramowanie: 1piksel z_kropek kolor obramowania→ #00ad00; szerokość:90%; kolor tła:#cffffcf} | Selektor Klasy |
| Nagłówek o największej wielkości liter | ?? | {tekst-pisany:centrowany; kolor tekstu:zielony; tekst-dekorowanu:podkreślenie;} | Selektor Prostý |
| Nagłówek o drugiej co do wielkości liter | ?? | { kolor tekstu:zielony; } | Selektor Prostý |
| pseudoklasa linków odwiedzonych | ?:?????? | { kolor :zielony; } | Selector Pseudoklasyk |
| pseudoklasa linku wskazywanego przez mysz | ?:???? | { Kolor tła:#73ff73 } | Selector Pseudoklasyk |
| Linia rozdzielająca | ?? | { Kolor:zielony; wysokość:1piksel;} | Selektor Prostý |
| Klasa o nazwie Motto | .????? | { tekst-pisany:centrowany; obramowanie: 1 piksel linią_ciągłą białe szerokość: 95%; margines-lewy: 5%; kolor tła: #00cd00; kolor liter:biały; } | Selektor klasy |

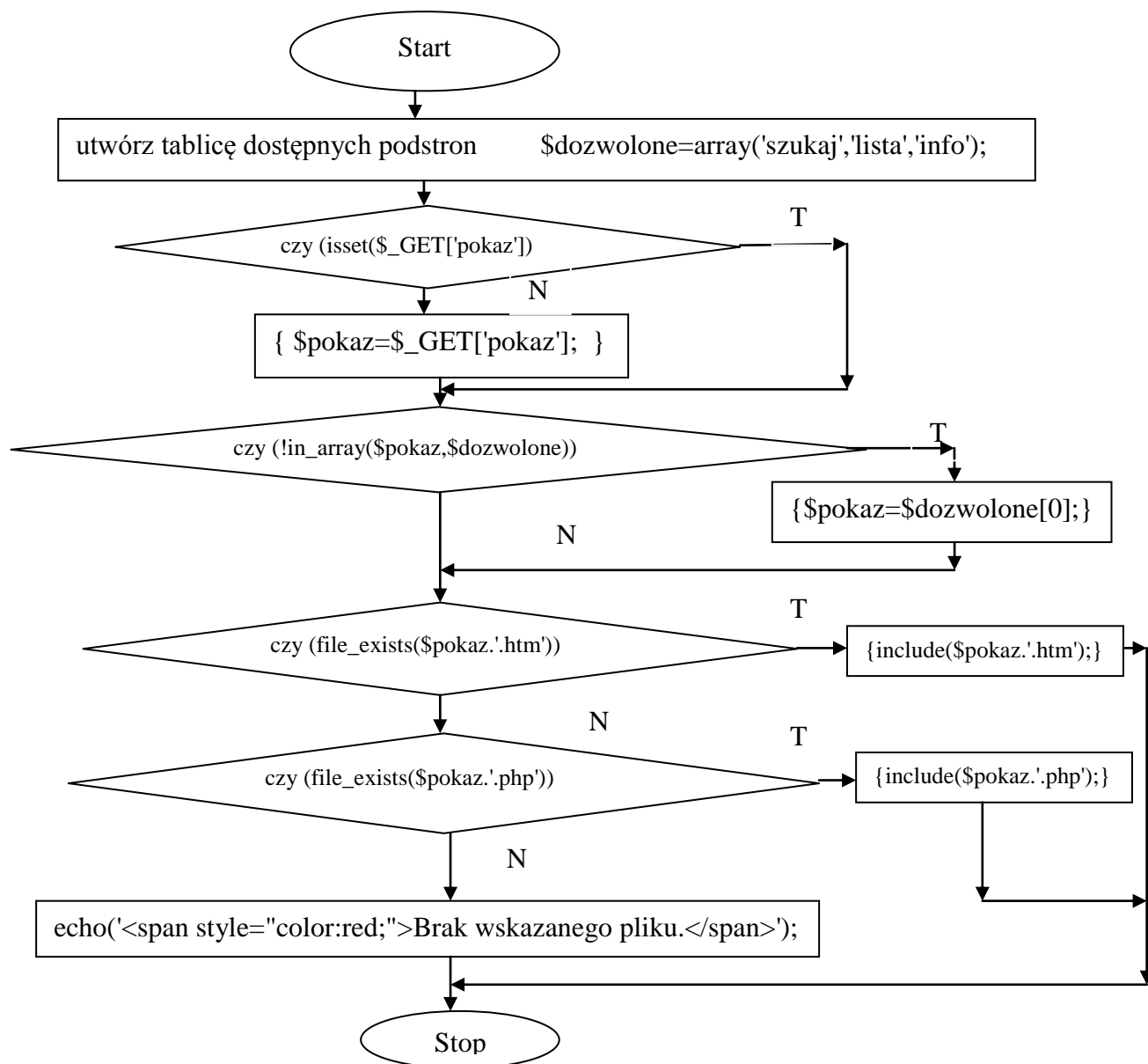
Etap3

Tworzymy plik **sys.php**, który będzie wyświetlał strony naszego portalu. utwórz plik wg schematu blokowego umieszczonego.

Opis algorytmu:

Tworzymy tablice **\$dozwolone** ze wszystkich dostępnych podstron witryny. W zmiennej **\$pokaz** zapamiętujemy przekazane do tego skryptu z linku umieszczonego na stronie index.php wartość parametru **pokaz**. Jeśli jednak nie przekazano tego parametru, w zmiennej **\$pokaz** zapamiętujemy pusty ciąg znaków. Następnie sprawdzamy, czy zmienna \$pokaz zawiera dowolny ciąg znaków(nazwę podstrony). Jeśli tak nie jest to zmiennej **\$pokaz** przypisujemy wartość pierwszego elementu

tablicy czyli **\$dozwolone[0]**. W wyniku takiego działania zmienna **pokaz** będzie miała zawsze przypisaną wartość **szukaj**. Następnie sprawdzamy czy istnieje na dysku plik o nazwie zapisanej w zmiennej **\$pokaz** i rozszerzeniu **.htm** to samo robimy dla pliku o rozszerzeniu **.php**. Jeśli obu plików nie ma to wyświetlamy komunikat o błędzie.



Zadanie24a

Temat: Portal miniksiążka telefoniczna → ciąg dalszy.

- Dokonaj kopiowania wszystkich plików z zadania poprzedniego do nowego folderu.
- Skasuj z nowego folderu pliki szukaj.htl oraz lista.html
- Zapisujemy część kod z zadania 19b pod nazwą dane.php. Jest to plik z naszymi danymi. Dopisywanie kolejnych osób będzie zwiększało naszą bazę.

```
<?php
$tel['Jan Nowak']['telefon']='601-123-456';
$tel['Jan Nowak']['adres']='ul. Piękna 12';

$tel['Piotr Kowalski']['telefon']='602-321-654';
$tel['Piotr Kowalski']['adres']='ul. Światowa 13a';

$tel['Joanna Kwiatkowska']['telefon']='603-456-321';
$tel['Joanna Kwiatkowska']['adres']='ul. Maniaka PHP 1/3';
?>
```

- Na początku pliku **sys.php** za pomocą instrukcji **include()** dołączamy **dane.php**
- Tworzymy skrypt do wyświetlania danych z dane.php o nazwie **lista.php**

```
<h2>Lista nazwisk w bazie</h2>
<table style="szerokość:90%; obramowanie: 1 piksel linia_ciągła zielona;">
<?php
foreach($tel as $k=>$v)
{
    echo('<tr><td>'.$k.'</td><td>'.$v['telefon'].'</td><td>'.$v['adres'].'</td></tr>');
}
?>
</table>
```

uwaga: Uzupełnij/zmień styl liniowy w drugiej linii tego skryptu.

Opis:

foreach(\$tel as \$k=>\$v)

Pętla foreach ma specyficzne zastosowanie, ponieważ służy wyłącznie do przeglądania zawartości typów złożonych: tablic oraz obiektów.

Pętla foreach w naszym przykładzie umożliwia zwracanie indeksów (w tym przypadku nazw) elementów.

Tablica **\$tel** jest indeksowana następująco **\$tel[nazwisko,telefon]**

czyli do:

- zmiennej **\$k** zostanie przypisany indeks którego wartością będzie konkretne nazwisko
- zmiennej **\$v** zostanie przypisany indeks którego wartością będzie konkretny telefon

=> oznacza oddzieleni indeksów

- tworzymy skrypt o nazwie **szukaj.php**

funkcja **trim(ciąg znaków)**

Usuwanie białych znaków z początku i końca ciągu.

funkcja **strpos(ciąg znaków,znak)**

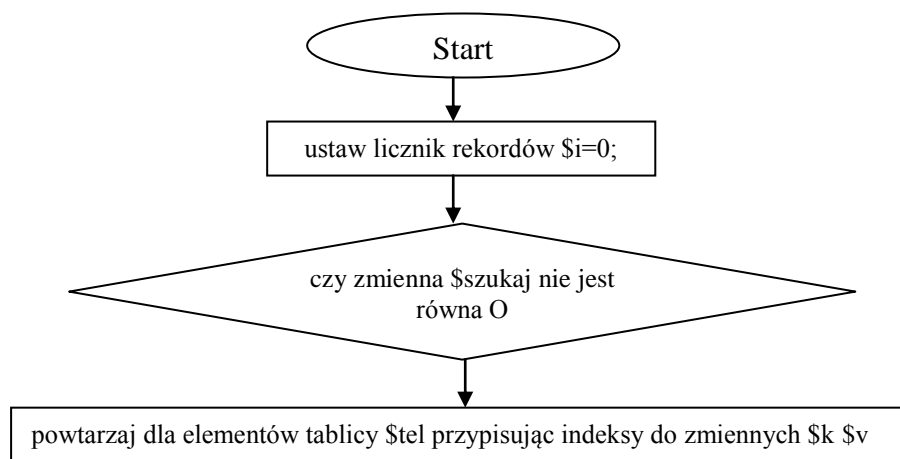
Funkcja ta wyszukuje w tekście (pierwszy argument) pierwsze wystąpienie znaku podanego jako drugi argument.

Opis algorytmu:

Na początku skryptu pobieramy parametr szukaj i przypisujemy go do zmiennej \$szukaj. Przypisanie następuje z użyciem funkcji trim→czyli usuwamy białe znaki. Jeśli parametr szukaj nie został przekazany to do zmiennej \$szukaj zostanie przypisany ciąg pusty. Następnie tworzymy formularz HTML. Formularz posiada okienko edycyjne o nazwie szukaj. W okienku tym z użyciem wstawki PHP wyświetlana jest zawartość zmiennej \$szukaj. W skrypcie zastosowana jest metod GET czyli przekazywanie parametrów z formularza przez linki do pliku index.php. W pliku są dwie wstawki PHP. Druga wstawka wyszukuje osoby wtedy gdy wartość zmiennej \$szukaja nie jest pusta.Wyszukiwanie z użyciem instrukcji foreach oraz funkcji strpos()).

Plik **szukaj.php** znajduje się pod schematem blokowym.

Wyszukiwanie informacji następuje wg algorytmu:



Treść skryptu **szukaj.php**

```
<?php
if (isset($_GET['szukaj'])) $szukaj=trim($_GET['szukaj']); else $szukaj="";
?>

<form action="index.php" method="GET">
<table style="width:60%;border:1px solid green;" align="center">
<tr><td align="center">
    Szukaj: <input type="text" name="szukaj" value="<?php echo $szukaj; ?>" />
    <input type="submit" value="Szukaj" />
</td></tr></table>
</form>

<?php
$i=0;
if ($szukaj!="")
{
    echo('<table style="width:90%;border:1px solid green;" align="center">');
    foreach ($tel as $k=>$v)
    {
        $t=$v['telefon'];
        $a=$v['adres'];

        if (strpos(' '.$t,$szukaj) || strpos(' '.$a,$szukaj) || strpos(' '.$k,$szukaj))
        {
            echo('<tr><td>'.$k.'</td><td>'.$t.'</td><td>'.$a.'</td></tr>');
            $i++;
        }
    }

    echo('</table>');
    if ($i==0) echo('<div style="color:red;text-align:center;font-weight:bold;">Nic nie
znaleziono</div>');
    else echo('<div style="margin-left:5%;width:90%;">Łącznie znaleziono <b>'.$i.'</b>
rekordów.</div>');
}
?>
```

Zadanie24b

Temat: Przesyłanie formularzy metodą GET.

Formularz tworzymy w języku HTML w znacznikach **<form>....</form>**.

atrybuty form:

action→adres pliku, do którego dany formularz ma być wysłany,

metoda→sposób wysłania formularza **GET** i **POST**.

Metoda GET:

cała zawartość formularza przesyłana jest z użyciem **adresu URL**.

Przykład przesyłania danych do pliku:

index.php&imie=Jan&nazwisko=Kowalski

Sprawdzanie czy informacja z adresu URL została przekazana do skryptu PHP.
\$_GET[imie] → wyświetla zawartość zmiennej **imie**

isset(\$zmienna) - zwraca prawdę, jeżeli zmienna istnieje.

Wykonaj:

- 1)Uruchom przykład,
- 2)Dodaj pole do wpisania nazwiska i uruchom przykład wpisz Twoje nazwisko i imię,
- 3)Wykonaj zrzut ekranu z polem Adres przeglądarki, otocz znak & niewielkim czerwonym prostokątem,
- 4)Dołącz zrzut ekranu do strony (tylko pasek adresu-nie całą stronę),
- 5)Poproś nauczyciela w celu sprawdzenia.

```
<html>
<head>
  <meta http-equiv="Content-type" content="text/html; charset=iso-8859-2" />
  <title>Demonstracja metody GET</title>
</head>
<body>

<?php
if (isset($_GET['imie']))
{
  echo('<h1>Witaj ' . $_GET['imie'] . '!</h1>');
  echo('<hr />');
}
?>

<form action="z24b.php" method="get">
  Podaj imie: <input type="text" name="imie" />
  <input type="submit" value="Wyślij" />
</form>

</body>
</html>
```

Wykrywanie nazwy pliku

\$_SERVER["PHP_SELF"]

<form action="<?php echo(\$_SERVER["PHP_SELF"]); ?>" method="get">

Wady stosowania metody GET

- 1)nie można przysyłać dużej ilości pól formularza (zbyt długi adres),
- 2)wiele przeglądarek zapamiętuje adresy i może się zdarzyć, że zapamiętane zostaną hasła,

Zadanie24b1

Temat: Przesyłanie formularzy metodą POST, wykrywanie nazwy pliku.

Wykonaj:

- 1)Zmień poprzednie zadanie aby realizowane było metoda POST,
- 2)dodaj automatyczne wykrywanie nazwy pliku,
- 3)Usuń wyświetlanie zrzutu ekranu.
- 4)Poproś nauczyciela w celu sprawdzenia.

Metoda POST:

cała zawartość formularza przesyłana jest z użyciem **nagłówka pliku przesyłanego z przeglądarki do serwera**.

Cechy metody POST

- 1)nie ma limitu danych,
- 2)nie zmienia się adres URL.

Przesyłanie plików z użyciem POST →upload plików na serwer.

z użyciem tej cechy możemy wykonać:

- 1)wymiana plików przez użytkowników naszej strony,
- 2)tablica ogłoszeń,
- 3)przesyłanie zdjęć w między użytkownikami.

Zadanie24b2

Temat: Upload plików na serwer do domyślnego folderu.

Selektor plików→wybór pliku do przesłania.

```
<form action="..." enctype="multipart/form-data">  
    <input type="file" name="nazwa" />  
</form>
```

gdzie jako "nazwa" należy wpisać krótką nazwę danego pola. Zostanie ona później wysłana wraz z formularzem.

Polecenie wprowadza selektor plików, składający się z dwóch pól:

pola tekstowego, w którym wpisuje się ścieżkę dostępu do pliku,

przycisku, po naciśnięciu którego, można przeszukać dysk, aby odnaleźć plik.

Możliwe jest także podanie dodatkowych atrybutów:

- 1)Długość pola (tekstowego)

```
<input type="file" name="nazwa" size="k" />
```

gdzie jako "k" podaje się długość pola (tekstowego) w ilości znaków, które będą w nim jednocześnie widoczne.

- 2)Blokada pola:

```
<input type="file" name="nazwa" disabled="disabled" />
```

Wstawienie atrybutu disabled="disabled" powoduje zablokowanie pola.

- 3)Filtr typu plików

```
<form action="..." enctype="multipart/form-data" accept="lista typów">  
    <input type="file" name="nazwa" />
```

</form>

Czasami oczekujemy, że użytkownik wyśle nam poprzez formularz konkretny typ pliku - np. obrazek.

Wykonaj:

- 1)Uruchom przykład,
- 2)Zmień nazwę pola ukrytego na nazwisko_plik_maksimum
- 3)Ustaw wielkość przesyłanych plików na 1000+liczba_liter_nazwiska*120
- 4)Poproś nauczyciela w celu sprawdzenia.

Opis:

- 1)dla <input> stosujemy typ **file** o nazwie name w tej zmiennej podajemy nazwe pliku do wysłania,
- 2)ukryte pole o typie hidden i nazwie MAX_FILE_SIZE → stosowane jest aby kontrolować wielkość przesyłanego pliku,
- 3)sprawdzenie czy z formularza przesłany został plik → isset
- 4)instrukcja print_r(\$_FILES['Plik']); (eleganckie wyświetlanie tablic) wyświetla danych z tabeli asocjacyjnej (pięć pól → name, type, tmp_name, error, size) \$_FILES['Plik'], która zawiera dane o przesyłanym pliku

```
<html>
<head>
  <meta http-equiv="Content-type" content="text/html; charset=iso-8859-2" />
  <title>Upload pliku z szkole ZSE Gdańsk</title>
</head>
<body>
<?php
if (isset($_FILES['Plik']))
{
  echo('<pre>');
  print_r($_FILES['Plik']);
  echo('</pre>');
}
?>
<form action="<?php echo $_SERVER["PHP_SELF"]; ?>" method="post"
enctype="multipart/form-data">
  <input type="hidden" name="MAX_FILE_SIZE" value="2000" />
  Wskaż plik do przesłania: <input type="file" name="Plik" />
  <input type="submit" value="Wyślij plik" />
</form>
</body>
</html>
```

Zadanie24b3

Temat: Upload plików na serwer do określonego folderu na serwerze, opisem mechanizmu przesyłu.

move_uploaded_file(nazwa_pliku_przesyłanego,nazwa_pliku_nagranego_z_siezką)

Wykonaj:

- 1)Uruchom przykład
- 2)Założ folder o nazwie nazwisko, do którego będziesz nagrywał przesłane pliki
- 3)Zmień treść skryptu tab aby pliki trafiały tam gdzie trzeba,
- 3)Poproś nauczyciela w celu sprawdzenia.

```
<html>
<head>
  <meta http-equiv="Content-type" content="text/html; charset=iso-8859-2" />
  <title>Upload pliku</title>
</head>
<body>

<?php
if (isset($_FILES['Plik']))
{
  if
(move_uploaded_file($_FILES['Plik']['tmp_name'], './uploads/'.$_FILES['Plik']['name']))
  {
    echo('Plik został przesłany prawidłowo.');
  }
  else
  {
    switch($_FILES['Plik']['error'])
    {
      case 1: echo('Przesyłany plik jest większy niż zezwala konfiguracja serwera.');break;
      case 2: echo('Przesyłany plik jest większy niż zezwala formularz.');break;
      case 3: echo('Przesłano tylko część pliku.');break;
      case 4: echo('Plik nie został przesłany');break;
    }
  }
}
?>

<form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post"
enctype="multipart/form-data">
  <input type="hidden" name="MAX_FILE_SIZE" value="2000" />
  Wskaż plik do przesłania: <input type="file" name="Plik" />
  <input type="submit" value="Wyślij plik" />
</form>
</body>
</html>
```

Zadanie24c

Temat: Analiza formularzy s76-79

Działanie skryptu:

1) Formularz składa się z pięciu pól (cztery jednowierszowe, jedno wielowierszowe),
2) Nius będzie zapisany gdy pola formularza będą wypełnione poprawnie
3) Gdy zmienna \$_POST['autor'] istnieje oznacza to, że formularz został wypełniony i przesłany.

4) Gdy formularz został wypełniony i przesłany sprawdzamy dalsze warunki:

warunek1 → długość tytułu niusa

warunek2 → długość nazwy autora

warunek3 → kontrola poprawności maila

stosujemy funkcję `ereg()`

5) Jeśli podczas sprawdzania danych z formularza wystąpił błąd to w zmiennej **\$blad** zapamiętujemy odpowiedni komunikat. Na początku skryptu do zmiennej **\$blad=' '** wpisujemy pusty ciąg znaków. czyli gdy w zmiennej **\$blad** jest ' ' to oznacza, że formularz został wypełniony prawidłowo,

6) funkcja **info(\$parametr1,\$parametr2)**

\$parametr1 → treść komunikatu

\$parametr2 → kolor komunikatu

7) użyte funkcje standardowe

strlen() → określenie długości tekstu

strpos(\$parametr1,\$parametr2) → znajdowanie tekstu wewnątrz innego tekstu

\$parametr1 → ciąg znaków przeszukiwany

\$parametr2 → podciąg szukany

funkcja zwraca:

gdy podciąg jest znaleziony → liczba od, której zaczyna się szukany podciąg

gdy podciąg nie jest znaleziony → to FALSE

ereg(\$parametr1,\$parametr2) → sprawdza, czy ciąg tekstowy pasuje do wyrażenia regularnego

\$parametr1 → pierwszy parametr przyjmuje wyrażenie regularne,

\$parametr2 → drugi ciąg który ma być porównany z wyrażeniem.

eregi(\$parametr1,\$parametr2) → istnieje także wersja tej funkcji ignorująca wielkość znaków:

Zwracana jest wartość **true** jeśli ciąg pasuje do wyrażenia regularnego, a **false** jeśli nie pasuje.

Wyrażenia regularne

Wyrażenia regularne są to ciągi znaków, które można dopasować do różnych innych ciągów.

Gwiazdka w wyrażeniach regularnych działa niejako 'wstecz' – opisuje dowolną ilość powtórzeń (zero lub więcej) poprzedniego znaku, czyli:

wyrażenie **'a*'** będzie pasowało do **'aaa'**, **'aaaaa'** ale i **'b'**, ponieważ w tym wyrażeniu nie ma ograniczenia że nie może być nic oprócz **'a'**.

Takie ograniczenie można uzyskać dzięki znakom **'^'** i **'\$'**.

Znak ,^' rozumiany jest jako początek linii, a ,\$\$' jako koniec.

przykład1

Tak więc jeśli chcemy znaleźć ciąg, który będzie pusty lub będą w nim same litery ,a', to powinien on wyglądać tak: ,^a*\$'.

przykład2

Natomiast jeśli ,a' musi wystąpić przynajmniej raz, to zamiast znaku ,*' należy użyć znaku ,+' – oznacza on jedno lub więcej powtórzeń ostatniego znaku.

Istnieje też możliwość użycia znaku ,?', który oznacza żadne lub jedno wystąpienie poprzedzającego ten znak atomu, znaku lub zakresu. Można także podać konkretną ilość wystąpień znaku, atomu lub przedziału przez użycie za raz po nich nawiasów klamrowych. Jeśli będzie w nich jedna liczba, to będzie ona określała dokładnie daną liczbę powtórzeń. Jeśli będzie w nich liczba a po niej przecinek, to wyrażenie będzie musiało wystąpić przynajmniej tyle razy ile wynosi podana liczba. Istnieje także możliwość podania w nawiasach klamrowych dwóch liczb oddzielonych przecinkiem, co oznacza powtórzenie wyrażenia minimalnie tyle razy ile pierwsza liczba, a maksymalnie tyle ile druga (włącznie).

Zamiast pojedynczego znaku można podać zakres wyrazów lub całe wyrażenie – tzw. atom. Zakres znaków podaje się w nawiasach kwadratowych. W tych nawiasach można podać pojedyncze znaki jeden po drugim (np. ,[egt],), przedział liter (np. wszystkie małe litery: ,[a-z], – uwaga, zakres ten nie obejmuje polskich znaków diakratycznych), lub połączenie tych dwóch typów (np. ,[a-zA-Ząęłńóśż], określa wszystkie litery). W nawiasach kwadratowych można podawać także symbole. Jeśli w przedziale potrzebny jest znak myślnika, który normalnie używany jest do określania przedziałów znaków, to trzeba umieścić go na pierwszej lub ostatniej pozycji w nawiasach kwadratowych. W nawiasach kwadratowych znak ,^' ma inne znaczenie. Oznacza ono zaprzeczenie, czyli jeśli umieści się je przed znakiem lub zakresem, to ten znak lub zakres nie będzie mógł się pojawić w tym miejscu.

Jeśli znak może być dowolny, to zamiast konkretnej litery, znaku czy przedziału można użyć symbolu kropki (,.'). Przykładowo wyrażenie ,^.*\$' opisuje dowolny ciąg składający się z dowolnej ilości dowolnych znaków.

Atom, czyli wyrażenie zawarte w nawiasach okrągłych, może zawierać (prawie) dowolną ilość znaków, zakresów i innych atomów. Pozwala to na dopasowywanie powtarzających się fragmentów ciągów znaków. Dobrym przykładem jest sprawdzenie, czy dany ciąg znaków jest ścieżką w systemie unixowy. Przykładowy ciąg to ,/usr/local/bin/php'.

Wyrażenie regularne powinno wyglądać tak: ,(/[a-Z])+(/[a-Z\.]*)'. Pierwszy atom określa dowolną ilość katalogów (na początku znak slash a po nim dowolna ilość liter o dowolnej wielkości) po czym nazwa pliku – po atomie nie ma żadnego znaku określającego ilość wystąpień, a więc musi wystąpić dokładnie raz.

Pomiędzy atomami, znakami lub zakresami można postawić znak '|', który oznacza logiczną operację 'OR', czyli po prostu oznacza że może wystąpić jeden lub drugi atom (ten przed znakiem '|' i ten po).

Jeśli w wyrażeniu regularnym musisz podać znak, który byłby interpretowany inaczej niż zamierzamy (np. jeśli musisz znaleźć ciąg zaczynający się od gwiazdki po której jest dowolna ilość liter A, to nie można 'normalnie' podać znaku gwiazdki, ponieważ byłby on interpretowany jako dowolna ilość powtórzeń ostatniego znaku), to należy zamienić ten znak na tzw. 'escape sequence' – po prostu należy przed tym znakiem wstawić znak ukośnika ('\'). Takie znaki to: ',', ')', '[', ']', '{', '}', '\\', '*', '|', '^', '\$', '?'. Jeśli w wyrażeniu trzeba użyć właśnie znaku ukośnika, to należy wpisać dwa ukośniki, jeden po drugim.

Symbol	Znaczenie symbolu
.	Określa dowolny znak z wyjątkiem znaku nowego wiersza
\	Określa, że następny znak ma być traktowany traktowany jako zwykły znak , a nie znak specjalny
*	Oznacza zero lub więcej wystąpień poprzedzającego znaku
?	Oznacza zero lub jedno wystąpienie poprzedzającego znaku
+	Oznacza jedno lub jedno wystąpienie poprzedzającego znaku
^	Początek wiersza
\$	Koniec wiersza

przykład

"^.+@.+\\.+ \$" → kontrola e-maila

Wy tłumaczenie

rozpoczyna się dowolnym znakiem (co najmniej jednym), po którym powinien być @, a po nim ponownie ciąg dowolnych znaków (co najmniej jednoznakowy), kropka i dalszy ciąg znaków (również co najmniej jednoznakowy)

Wykonaj:

- 1)Uruchom przykład
- 2)zmień minimalna długość treści niusa na 8
- 3)zmień maksymalna długość treści niusa na 8*miesiąc urodzenia+100
- 4)Uruchom przykład i poproś nauczyciela w celu sprawdzenia

```
<?php
function info($s,$kolor)
{
    echo ('<div align="center"><p style="color:'. $kolor.'">'. $s.'</p></div>');
}
if (isset($_POST['autor']))
{
    $blad="";
    if (strlen($_POST['tresc'])<10 || strlen($_POST['tresc'])>1024) $blad.='Treść niusa ma nieprawidłową
długość<br>';
    if (strlen($_POST['tytul'])<10 || strlen($_POST['tytul'])>200) $blad.='Tytuł niusa ma nieprawidłową
długość<br>';
    if (strlen($_POST['autor'])<3) $blad.='Żle wprowadzono informacje o autorze<br>';
    if (!ereg ("^.+@.+\\.+ $", $_POST['email'])) $blad.='Podano niepoprawny adres e-mail<br>';
    if ($blad!="") info($blad,'red');
    else
    {
        info('Formularz został zapisany.', 'green');
```

```

return;
}
}
?>
<form action="dodaj.php" method="POST">
<input type="hidden" name="pokaz" value="dodaj" />
<table border="0"><tr><td>Autor:</td><td>
<input size="30" type="text" name="autor" value="<?php echo $_POST['autor']; ?>" /></td></tr>
<tr><td>Adres e-mail:</td><td>
<input size="30" type="text" name="email" value="<?php echo $_POST['email']; ?>" /></td></tr>
<tr><td>Źródło:</td><td>
<input size="30" type="text" name="zrodlo" value="<?php echo $_POST['zrodlo']; ?>" /></td></tr>
<tr><td>Tytuł:</td><td>
<input size="30" type="text" name="tytul" value="<?php echo $_POST['tytul']; ?>" /></td></tr>
<tr><td>Treść:</td><td><textarea cols="40" rows="5" name="tresc">
<?php echo $_POST['tresc']; ?></textarea></td></tr>
<tr><td></td><td><input type="submit" value="Dodaj nowy nius" /></td></tr></table>
</form>

```

Zadanie24c1

Temat: Analiza formularzy. Na podstawie poprzedniego przykładu oraz zastosowania wyrażeń regularnych wykonaj stronę z formularz rejestracji do kursu komputerowego.

Etap 1

Wykonaj stronę w technologii CSS na DIV'ach.

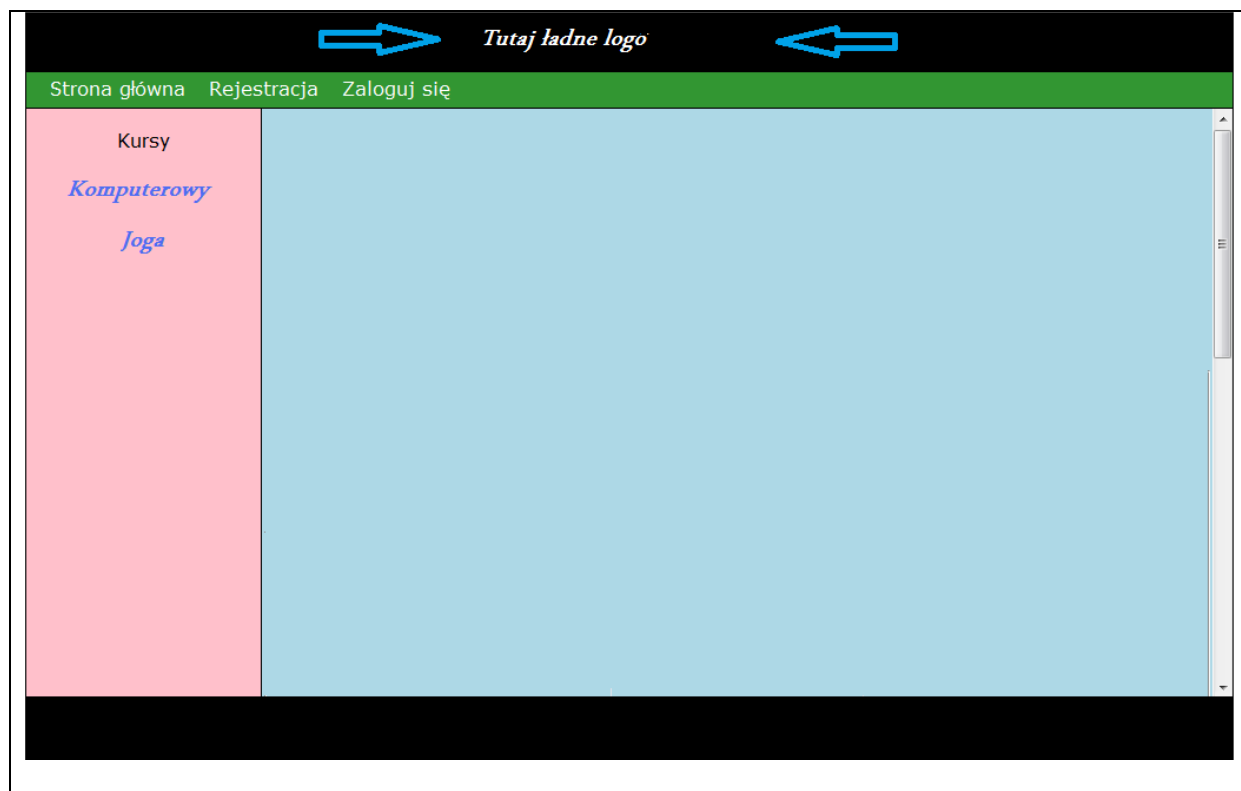
Ułożenie:

(numer_w_dzienniku reszta_z_dzielenie 4)=0	
Zielony pasek	
Różowy Menu	Niebieska część

(numer_w_dzienniku reszta_z_dzielenie 4)=1	
Niebieska część	Różowy Menu
Zielony pasek	

(numer_w_dzienniku reszta_z_dzielenie 4)=2	
Zielony pasek	
Niebieska część	Różowy Menu

(numer_w_dzienniku reszta_z_dzielenie 4)=3	
Niebieska część	Różowy Menu
Zielony pasek	



Wybór opcji

Strona główna powoduje wyświetlenie w niebieskim oknie tekstu „jesteś na stronie głównej”

Rejestracja powoduje wyświetlenie w niebieskim oknie formularza rejestracyjnego.

Etap 2

Wykonanie formularza bez sprawdzania poprawności danych (przykład poniżej)
jakie pola formularza w tabeli pod niebieski przykładem.

Nazwa użytkownika *

Spaces are allowed; punctuation is not allowed except for periods, hyphens, apostrophes, and underscores.

Adres e-mail *

Adres e-mail, koniecznie prawidłowy. Będą nań przesyłane wszystkie wiadomości e-mail od systemu. Adres nie jest udostępniany publicznie i będzie wykorzystywany jedynie w wypadku prośby o przesłanie nowego hasła lub do przesyłania informacji o nowościach czy innych komunikatów.

Dane osobowe

Kraj *

Polska

No kraj trza wpisać :) któryś z 46 europejskich!

Tytuł *

Pan

Facet czy babka

Imię *

No imię :)

Ułożenie pól formularza (ważna kolejność) numerowanie konieczne	
(numer_w_dzienniku reszta_z_dzielenie 4)=3	
1. Nazwisko	
2. Imię	
3. PESEL	
4. Miasto	
5. Ulica	
6. Numer domu	
7. Numer mieszkania	
8. Kod pocztowy	
9. Adres e-mail	
10. Wykształcenie	
11. Wiek	
12. Płeć	
13. Wzrost	
14. Narodowość	
15. Login	
16. Hasło	
17. Nick	
18. Stopień wojskowy	
19. Fax	
20. Telefon	
21. NIP	
(numer_w_dzienniku reszta_z_dzielenie 4)=2	
1. Nazwisko	
2. Imię	
3. PESEL	
4. Miasto	
5. Ulica	

6. Numer domu
7. Numer mieszkania
8. Kod pocztowy
9. Adres e-mail
10. Wykształcenie
11. Wiek
12. Płeć
13. Wzrost
14. Narodowość
15. Nick
16. Login
17. Hasło
18. Stopień wojskowy
19. Fax
20. Telefon
21. NIP

(numer_w_dzienniku reszta_z_dzielenie 4)=1

1. PESEL
2. Miasto
3. Ulica
4. Numer domu
5. Numer mieszkania
6. Kod pocztowy
7. Adres e-mail
8. Stopień wojskowy
9. Wykształcenie
10. Nazwisko
11. Imię
12. Płeć
13. Wzrost
14. Narodowość
15. Login
16. Wiek
17. Hasło
18. Fax
19. Telefon
20. NIP
21. Nick

(numer_w_dzienniku reszta_z_dzielenie 4)=0

1. Imię
2. PESEL
3. Nazwisko
4. Miasto
5. Nick
6. Adres e-mail
7. Wykształcenie
8. Płeć
9. Wzrost
10. Wiek
11. Narodowość

- 12. Hasło
- 13. Stopień wojskowy
- 14. Login
- 15. Fax
- 16. Telefon
- 17. Numer mieszkania
- 18. Ulica
- 19. Kod pocztowy
- 20. Numer domu
- 21. NIP

Etap 3

Dopisanie formatów pod polami (pod okienkiem pola)

np. *W formacie yyyy-mm-dd czyli pole max 10 znakowe*

Sprawdzenie poprawności formularza danych z zakreśleniem ramki na czerwono w przypadku złych danych np.

Dla PESEL jest sprawdzana tylko długość.

Temat:		Należy wpisać temat.
--------	--	----------------------

Etap 4

Dopisanie danych z formularza do bazy.

Etap 5

Dorobienie nowej opcji (obok Rejestracja i Strona główna) Wyświetl dane rejestracyjne użytkowników

Etap 6

Wykonanie możliwości kasowania wybranej osoby z bazy rejestracyjnej.

Etap 7

Sprawdzenie poprawności PESEL

Numer PESEL → algorytm sprawdzania poprawności.

Numer PESEL jest to 11-cyfrowy, stały symbol numeryczny, jednoznacznie identyfikujący określoną osobę fizyczną.

Zbudowany jest z następujących elementów:

- zakodowanej daty urodzenia
- liczby porządkowej
- zakodowanej płci
- cyfry kontrolnej

Przykładowa postać:

4	4	0	5	1	4	0	1	4	5	8
---	---	---	---	---	---	---	---	---	---	---

cyfry [1-6] – data urodzenia z określeniem stulecia urodzenia

cyfry [7-10] – numer serii z oznaczeniem płci

cyfra [10] – płeć

cyfra [11] – cyfra kontrolna

Data urodzenia w formacie

RRMMDD

RR -rok

MM -miesiąc

DD -dzień

dla osób urodzonych w latach 1900 do 1999 – miesiąc zapisywany jest w sposób naturalny
dla osób urodzonych w innych latach niż 1900 – 1999 dodawane są do numeru miesiąca następujące wielkości:

dla lat 1800-1899 - 80

dla lat 2000-2099 - 20

dla lat 2100-2199 - 40

dla lat 2200-2299 - 60

Przykładowo osoba urodzona 14 lipca 2002 roku ma następujący zapis w numerze ewidencyjnym:

0	2	2	7	1	4
---	---	---	---	---	---

Płeć

Informacja o płci osoby, zawarta jest na 10. (przedostatniej) pozycji numeru PESEL.

cyfry 0, 2, 4, 6, 8 – oznaczają płeć żeńską

cyfry 1, 3, 5, 7, 9 – oznaczają płeć męską

Cyfra kontrolna i sprawdzanie poprawności numeru

Jedenasta cyfra jest cyfrą kontrolną, służącą do wychwytywania przekłamań numeru. Jest ona generowana na podstawie pierwszych dziesięciu cyfr. Aby sprawdzić czy dany numer PESEL jest prawidłowy należy, zakładając, że litery **a-j** to kolejne cyfry numeru od lewej, obliczyć wyrażenie

$$1*a + 3*b + 7*c + 9*d + 1*e + 3*f + 7*g + 9*h + 1*i + 3*j$$

Następnie należy odjąć ostatnią cyfrę otrzymanego wyniku od 10. Jeśli otrzymany wynik nie jest równy cyfrze kontrolnej, to znaczy, że numer zawiera błąd.

Uwaga implementacyjna - jeśli ostatnią cyfrą otrzymanego wyniku jest 0, w wyniku odejmowania otrzymamy liczbę 10, podczas gdy suma kontrolna jest cyfrą. Oznacza to tyle, że cyfra kontrolna winna być równa 0 (stąd dobrze jest wykonać na wyniku odejmowania operację modulo 10). W wyniku niezbyt precyzyjnego opisu na stronie MSW ten aspekt jest często pomijany i prowadzi do błędów w implementacji sprawdzania poprawności numeru PESEL.

Przykład dla numeru PESEL 44051401358:

$$1*4 + 3*4 + 7*0 + 9*5 + 1*1 + 3*4 + 7*0 + 9*1 + 1*3 + 3*5 = 101$$

Wyznaczamy resztę z dzielenia sumy przez 10:

$$101:10 = 10 \text{ reszta} = 1$$

Jeżeli reszta = 0, to cyfra kontrolna wynosi 0. Jeżeli reszta \neq 0, to cyfra kontrolna będzie uzupełnieniem reszty do 10, czyli w podanym przykładzie jest to cyfra 9.

$$10 - 1 = 9$$

Wynik 9 nie jest równy ostatniej cyfrze numeru PESEL, czyli 8, więc numer jest błędny.

Zadanie24d

Temat: operacja na plikach s79-84

Dzięki operacjom na plikach można:

- 1)tworzyć statystyki odwiedzin
- 2)przechowywać wyniki ankiet
- 3)wpisy do księgi gości
- 4)umieszczanie niusów

Algorytm zapisu danych do pliku:

- 1)Otwarcie pliku w trybie do zapisu lub otwarcie pliku nieistniejącego, co oznacza jego utworzenie,
- 2)Blokujemy dostęp do tego pliku tak, aby inne skrypty nie mogły z niego w tym samym momencie korzystać,
- 3)Zapisujemy dane do pliku,
- 4)Zdejmujemy blokadę,
- 5)Zamykamy plik, co oznacza zapisanie danych.

Przykład na działanie algorytmu:

```
<?php
$plik=fopen('stats.txt','w');
flock($plik,LOCK_EX);
fputs($plik,$_SERVER['HTTP_REFERER']);
flock($plik,LOCK_UN);
fclose($plik);
?>
```

fopen(\$parametr1,\$parametr2) →otwieranie pliku

\$parametr1→nazwa otwieranego pliku

\$parametr2→tryb w jakim otwieramy plik (np. r+ lub inny)

Gdy otwarcie zakończy się sukcesem do zwracany jest tzw. Uchwyt do obsługi pliku. Uchwyt jest zapisywane w zmiennej. Gdy jest błąd podczas otwierania pliku zostanie zwrócone FALSE.

Uchwyt do obsługi pliku

Utworzenie obiektu, który umożliwi nam komunikację ze wskazanym przez nas plikiem.

flock(\$parametr1,\$parametr2) →manipulowanie blokadami

\$parametr1→uchwyt do pliku

\$parametr2→rodzaj blokady

W PHP dostępne są następujące poziomy blokady:

LOCK_SH = 1 - gdy odcytujemy z pliku

LOCK_EX = 2 - gdy zapisujemy do pliku

LOCK_UN = 3 - zdjęcie blokady

Jeśli chcemy założyć blokadę na zapis to wcześniej wszystkie inne blokady muszą być zdjęte. Aby założyć blokadę na odczyt wystarczy aby nie była na nim założona blokada na zapis.

fputs(\$parametr1,\$parametr2) →zapisywanie danych

\$parametr1→uchwyt do pliku

\$parametr2→treść, która chcemy zapisać

fwrite(\$parametr1,\$parametr2,parametr3) →zapisywanie danych

\$parametr1→uchwyt do pliku

\$parametr2→treść, która chcemy zapisać

r'	Otwiera tylko do odczytu; umieszcza wskaźnik pliku na jego początku.
,r+'	Otwiera do odczytu i zapisu; umieszcza wskaźnik pliku na jego początku.
,w'	Otwiera tylko do zapisu; umieszcza wskaźnik pliku na jego początku i obcina plik do zerowej długości. Jeśli plik nie istnieje to próbuje go utworzyć.
,w+'	Otwiera do odczytu i zapisu; umieszcza wskaźnik pliku na jego początku i obcina plik do zerowej długości. Jeśli plik nie istnieje to próbuje go utworzyć.
,a'	Otwiera tylko do zapisu; umieszcza wskaźnik pliku na jego końcu. Jeśli plik nie istnieje to próbuje go utworzyć.
,a+'	Otwiera do odczytu i zapisu; umieszcza wskaźnik pliku na jego końcu. Jeśli plik nie istnieje to próbuje go utworzyć.
,x'	Tworzy i otwiera plik tylko do zapisu; umieszcza wskaźnik pliku na jego początku. Jeśli plik już istnieje, wywołanie fopen() nie powiedzie się, zwróci FALSE i wygeneruje błąd na poziomie E_WARNING. Jeśli plik nie istnieje, spróbuje go utworzyć. To jest równoważne z określeniem flag O_EXCL O_CREAT stosowanym w wywołaniu systemowym open(2).
,x+'	Tworzy i otwiera plik odczytu i zapisu; umieszcza wskaźnik pliku na jego początku. Jeśli plik już istnieje, wywołanie fopen() nie powiedzie się, zwróci FALSE i wygeneruje błąd na poziomie E_WARNING. Jeśli plik nie istnieje, spróbuje go utworzyć. To jest równoważne z określeniem flag O_EXCL O_CREAT stosowanym w wywołaniu systemowym open(2).
b	Dodanie litery b do innych trybów oznacza, że otwierany plik jest zbiorem binarnym(nie tekstowym).

Nazwa	Opis
fclose(\$p)	Funkcja zamyka plik o uchwycie \$p
feof(\$p)	Funkcja zwraca TRUE, kiedy osiągnięty zostanie koniec pliku o uchwycie \$p , w przeciwnym wypadku zwraca FALSE
fgets(\$p,\$i)	Funkcja odczytuje z pliku o uchwycie \$p linię tekstu lub też \$i znaków tej linii, jeśli jest ona dłuższa
file_exists(\$a)	Funkcja zwraca TRUE, jeśli plik o nazwie \$a istnieje. W przeciwnym wypadku zwraca FALSE.
file(\$a)	Funkcja odczytuje plik o nazwie \$a i zwraca tablicę wszystkich linii tekstu z tego pliku.
fileatime(\$a)	Funkcja zwraca datę ostatniego dostępu do pliku o nazwie \$a . Jest to specjalny format daty, który można przekazać jako drugi argument funkcji date() .
filectime(\$a)	Funkcja zwraca datę utworzenia pliku o nazwie \$a
filemtime(\$a)	Funkcja zwraca datę ostatniej modyfikacji pliku o nazwie \$a
filesize(\$a)	Funkcja zwraca liczbę określającą wielkość pliku o nazwie \$a
flock(\$p,\$t)	Funkcja ustawia blokadę \$t dla pliku o uchwycie \$p
fputs(\$p,\$s)	Funkcja zapisuje do pliku o uchwycie \$p ciąg znaków \$s
fread(\$p,\$i)	Funkcja odczytuje \$i bajtów z pliku o uchwycie \$p
fwrite(\$p,\$s)	Funkcja zapisuje do pliku o uchwycie \$p ciąg znaków \$s
mkdir(\$s)	Funkcja tworzy katalog o nazwie \$s
fclose(\$p)	Funkcja zamyka plik o uchwycie \$p
readfile(\$a)	Funkcja odczytuje i wyświetla zawartość pliku o nazwie \$a
unlink(\$a)	Funkcja usuwa plik o nazwie \$a

Zadanie24e1

W

`$_SERVER['HTTP_REFERER']`

Strona oglądana jako poprzednia, z której przeniosłeś się tutaj

Zadanie24e2

W kolejnych wierszach pliku o nazwie „liczby-1.txt” znajdują się liczby zapisane w systemie dziesiętnym. Napisz programy, które dadzą odpowiedzi do poniższych podpunktów:

- a) Ile jest wszystkich liczb?
- b) Ile jest liczb podzielnych przez „5” ?
- c) Jaka liczba jest największa a jaka najmniejsza. Wypisz te liczby?
- d) Ile jest liczb, których liczba cyfr jest liczbą nieparzystą?
- e) Ile jest liczb, których liczba cyfr jest równa „3” a ile liczb, których liczba cyfr jest równa „4”?
- f) Ile jest liczb, których pierwsza cyfra jest równa ostatniej?
- g) Ile jest liczb pierwszych?

```

<?php

$plik = file("liczby-1.txt");
$licznik = 0;
$licznik_podzielne = 0;
$max = 0;
$min = 99999;
$licznik_nieparz = 0;
$licznik_parz = 0;
$licznik_trzy = 0;
$licznik_cztery = 0;
$licznik_rowne_4 = 0;
$licznik_rowne_3 = 0;
$licznik_pierwsze = 0;

$plik = array_merge(array($plik[0]),array_map('intval', array_slice($plik, 1)));
foreach ($plik as $line_num => $line) {
    $licznik++;
    if(strval($line) % 5 == 0)
        $licznik_podzielne++;
    if(strval($line) % 2 != 0)
        $licznik_nieparz++;
    if($line > $max)
        $max = strval($line);
    else if ($line < $min)
        $min = strval($line);
    if(strlen($line) == 3)
    {
        $licznik_trzy++;
        settype($line, "string");
        if($line[0] == $line[2])
            $licznik_rowne_3++;
    }
    if(strlen($line) >= 4)
    {
        $licznik_cztery++;
        settype($line, "string");
        if($line[0] == $line[3])
            $licznik_rowne_4++;
    }

    if(czyPierwsza($line) == true)
        $licznik_pierwsze++;
}

function
czyPierwsza($liczbaKtoraJestPodejrzanaOBycieLiczbaPierwszaLeczMozeSieOkazacZeNiaNieJest) {
    $l_d = 0;

    for ($i = 1; $i <=
$liczbaKtoraJestPodejrzanaOBycieLiczbaPierwszaLeczMozeSieOkazacZeNiaNieJest; $i++)
    {

        if(($liczbaKtoraJestPodejrzanaOBycieLiczbaPierwszaLeczMozeSieOkazacZeNiaNieJest % $i) == 0)

```

```
        $l_d++;  
    }  
    if ($l_d == 2)  
        return true;  
    else if ($l_d != 2)  
        return false;  
}  
echo "a) ".$licznik."<br>";  
echo "b) ".$licznik_podzielne."<br>";  
echo "c) ".$max."; ".$min."<br>";  
echo "d) ".$licznik_nieparz."<br>";  
echo "e) ".$licznik_trzy."; ".$licznik_cztery."<br>";  
echo "f) ".$licznik_rowne_3."; ".$licznik_rowne_4."<br>";  
echo "g) ".$licznik_pierwsze."<br>";  
?>
```

Zadanie25

Temat: Księga gości w oparciu o pliki.

Etap 1

Założenia projektowe:

a) zewnętrzny arkusz styli,

b) nazwa pliku PHP → ksiega_nazwisko_ucznia.php np. ksiega_kowalski.php,

c) nazwy elementów Zwróć uwagę na czarne obramowania elementów.

<input type="text" name="nick_kow" > → trzy pierwsze litery nazwiska nick_kow.

<textarea name="tresc_kow" cols="39" rows="10"></textarea>

→ trzy pierwsze litery nazwiska tresc_kow.

d) kolory, obramowania, teksty kolor i wielkość takie same

Diagrama przedstawia interfejs formularza. Całość jest otoczona zieloną ramką. W środku znajduje się czerwony prostokąt. W jego górnym środku jest napis 'Nick' w białym kolorze. Poniżej niego jest białe pole tekstowe. Poniżej tego pola jest napis 'treść' w niebieskim kolorze. W centrum czerwonego pola znajduje się białe pole tekstowe. Poniżej tego pola jest przycisk 'Dodaj komentarz' w białym kolorze. Na samym dole formularza znajduje się żółta linia z napisem 'tutaj będą wpisy' w czarnym kolorze.

Etap 2

Założenia projektowe:

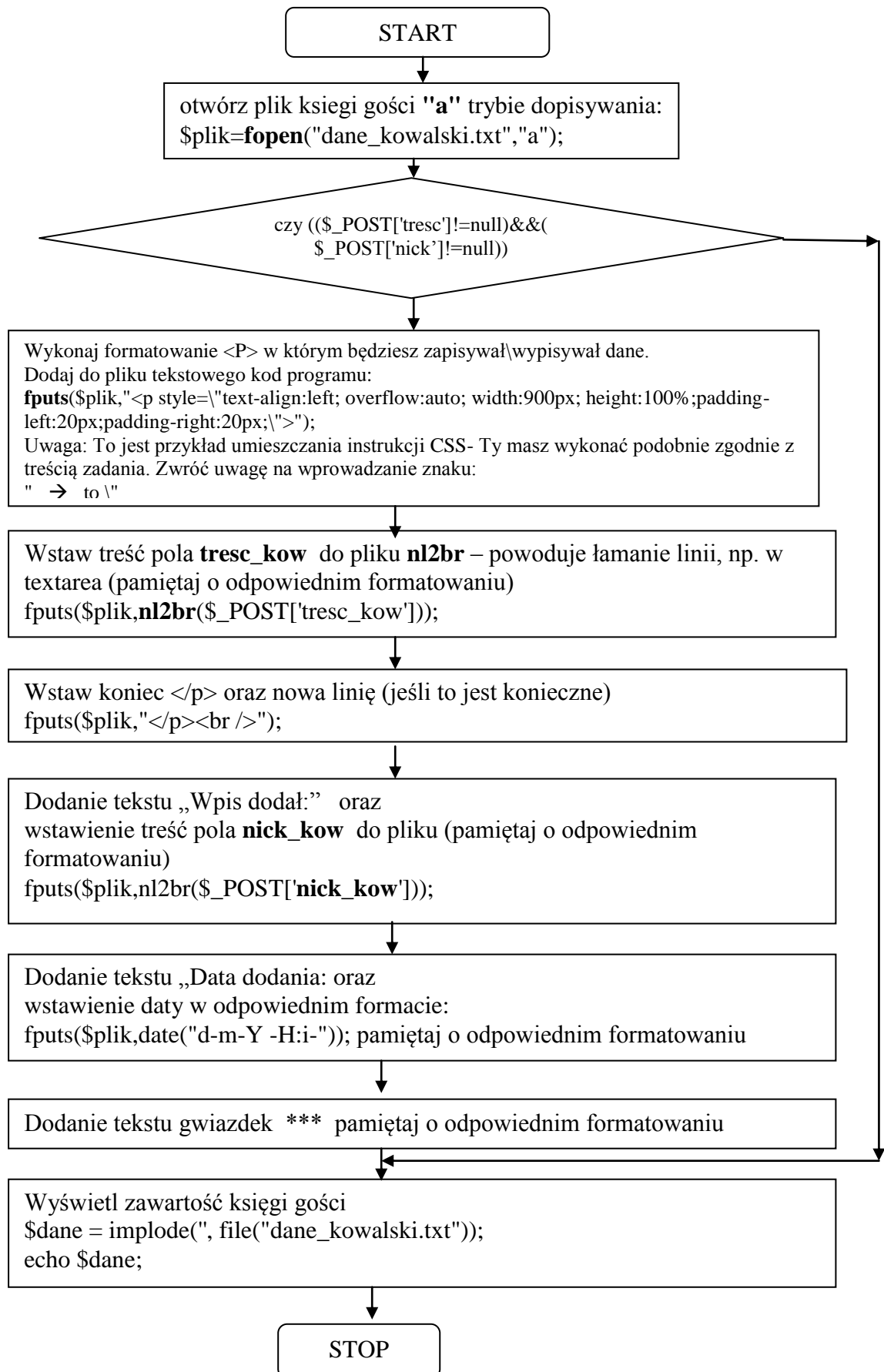
a) nazwa pliku z danymi dane_nazwisko_ucznia.txt np. dane_kowalski.

b) Do pliku tekstowego dołączone są następujące informacje:

- treść komentarza,
- tekst Wpis dodał → Nick osoby (kolor oraz wielkość patrz na przykład),
- tekst Data dodania oraz godzina ((kolor oraz wielkość, format daty oraz godziny patrz na przykład),
- trzy gwiazdki (kolor oraz wielkość patrz na przykład).

Algorytm wpisywania zapisu do księgi gości

if (\$_POST['tresc']!=null) { } → sprawdzenie czy pole o nazwie jest niepuste



Wygląd po wpisaniu się do księgi gości. (parz poniżej)

<div></div>	
<div>Dodaj komentarz</div>	
<div>wpisy</div>	
<div>Pierwszy wpis -programowanie jest OK</div>	
<div>Wpis dodał: Marek Data dodania: 03-06-2014 -08:51- ***</div>	
<div>Drugi wpis- dobrze mi płacą</div>	
<div>Wpis dodał: programista Data dodania: 03-06-2014 -08:53- ***</div>	

Przydatne instrukcje:

```
$plik=fopen("dane.txt","a");
```

\$plik – zmienna plikowa

fopen – otwarcie pliku w trybie dopisywania („a” – dopisywanie)

```
fputs($plik,"<br />"); → zapis do pliku kodu HTML
```

fputs(\$plik,"<div style=\"text-align:center;\">>"); → zapis do pliku kodu HTML , który
zwiera w swojej treści cudzysłowami, przed każdym cudzysłowem wstawiamy backslash „\”

fputs(\$plik,nl2br(\$_POST['tresc'])); → zapis do pliku wartości pól formularza

nl2br – powoduje łamanie linii, np. w textarea

```
fputs($plik,date("d-m-Y -H:i-"));
```

zapis do pliku daty w formacie: np. 14-10-2012 -20:02-

`echo implode(", ", file("dane.txt"));` → wyświetlenie zawartości całego pliku na ekran

```
if ($_POST['nick']!=null && $_POST['tresc']!=null) {
```

```
// wtedy dodaj posta
```

}

instrukcja if, sprawdzająca czy pola nick i treść nie są puste

&& – pascalskie AND

`$_POST['nick']` – odwołanie do tablicy przechowującej dane formularza

== - z C++ równe

!= – z C++ różne

Zadanie26

Temat: Poprawa zadania poprzedniego:

- 1) tak aby wpisy wyświetlały się od najmłodszego wpisu (decyduje data wpisu).
- 2) szyfrowanie danych → wpisać instrukcje do szyfrowania.
- 3)

Zadanie27

Temat: Sondaż z użyciem plików zewnętrznych.

Sondaż dotyczy stosowania diety przez wypełniającego.			
Co tuczy najbardziej?		NICK	Podaj płeć:
<input type="radio"/> Alkohol <input type="radio"/> Czekolada <input type="radio"/> Golonka		Tutaj okienko do wpisania nick	<input type="radio"/> Kobieta <input type="radio"/> Mężczyzna
		Podaj jaka dieta jest najlepsza i dlaczego?	
Procentowe zestawienie co najbardziej tuczy	textarea→wpisujemy odpowiedź o dietę.		Procentowe zestawienie płci
Alkohol - 30%			Kobiet - 60%
Czekolada- 50%			Mężczyzn - 40%
Golonka -20%			

Założenia projektowe:

a)wyświetlanie wpisów pod tabelą sondażową od najmłodszego wpisu (decyduje data wpisu).

b)zewnętrzny arkusz styli,

c)nazwa pliku PHP→sondaz_nazwisko_ucznia.php np. sondaz_kowalski.php,

d)nazwy elementów formularza z trzema literami ucznia/słuchacza,

e)kolory, obramowania, teksty kolor i wielkość dobrać sam,

f)wskazówka → aby obliczyć dane procentowe(co tuczy, płeć) zapisuj w osobnym pliku lub plikach.

g)Do **pliku tekstowego** dołączone są następujące informacje:

- treść komentarza,
- tekst Wpis dodał→Nick osoby (kolor oraz wielkość zaproponuj sam),
- tekst Data dodania oraz godzina ((kolor oraz wielkość, format daty oraz godziny zaproponuj sam),

Zadanie28

Temat: Portal niusy z użycie formularza.

Etap1

Wykonaj plik index.php

-meta polskie litery

-tytuł Twoje nazwisko

-podłączenie arkusza styli o nazwie nazwisko.css

-podłączenie pliku niusy_twoje_nazwisko.php

Etap2

Tworzymy plik nazwisko.css

Opis	Nazwa w HTML (CSS)	definicja	Rodzaj W CSS
Ciało (treść) dokumentu	body	{kolor tła: #afffaf; rodzaj czcionki:arial; wielkość czcionki:12pikseli;}	Selektor prosty
Znacznik początku tabeli oraz Znacznik początku komórka danych w tabel	table,?	{rodzaj czcionki:arial; wielkość czcionki:12pikseli;}	Grupowanie selektorów
Klasa o nazwie Tab	.???	{obramowanie: 1piksel z_kropek kolor obramowania→ #6fff6f; szerokość:90%; kolor tła#8fff8f }	Selektor Klasy
Nagłówek o największej wielkości liter	??	{tekst-pisany:centrowany; kolor tekstu:zielony; tekst-dekorowanu:podkreślenie;}	Selektor Prosty
Nagłówek o drugiej co do wielkości liter	??	{ kolor tekstu:zielony; }	Selektor Prosty
pseudoklasa linków odwiedzonych	?:??????	{ kolor :zielony; }	Selector Pseudoklasyk
pseudoklasa linku wskazywanego przez mysz	?:????	{ Kolor tła: #006100;kolor:biały; }	Selector Pseudoklasyk
Linia rozdzielająca	??	{ Kolor: #004f00; kolor tła: :#004f00; wysokość:1piksel; obramowanie:0 pikseli }	Selektor Prosty
Klasa o nazwie motto	.?????	{ tekst-pisany:centrowany; obramowanie: 1 piksel linią ciągłą białe szerokość: 95%; margines-lewy: 5%; kolor tła: #00cd00; kolor liter:biały; }	Selektor klasy
Pola formularza Pole tekstowe obszar tekstowy	Pole tekstowe, obszar tekstowy	{rodzaj czcionki; wielkość czcionki 12 pikseli; kolor tła :afffaf; obramowanie 1 piksel linią ciągłą o kolorze #004f00; kolor tekstu: #004f00;}	
Klasa o nazwie blad	.????	{kolor: #ff0000;obramowanie: 2 piksele ciągłe#ff0000; tekst centrowany; tekst pogrubiony; szerokość: 80%; margines wewnętrzny: 5 pikseli;}	
Klasa o nazwie ok	.??	{kolor: #00af00;obramowanie: 2 piksele ciągłe##00af00; tekst centrowany; tekst pogrubiony; szerokość: 80%; margines wewnętrzny: 5 pikseli;}	

Etap3

kod pliku **niusy.php** jest:

```
<h1>Aktualności</h1>
<div class="motto">
    Motto: <?php podłączenie pliku motto.php;) ?>
</div>
<table align="center" class="tab">
    <tr>
        <td valign="top">
            <div align="center">
                [ <a href="index.php?pokaz=lista">Lista aktualności</a> ]
                [ <a href="index.php?pokaz=dodaj">Dodaj news</a> ]
                <hr />
            </div>
            <?php podłączenie pliku sys.php; ?>
        <small>
            Wyświetlono:
            <?php podłączenie pliku licznik.php; ?>
        </small>
    </td>
</tr>
</table>
```

Etap4

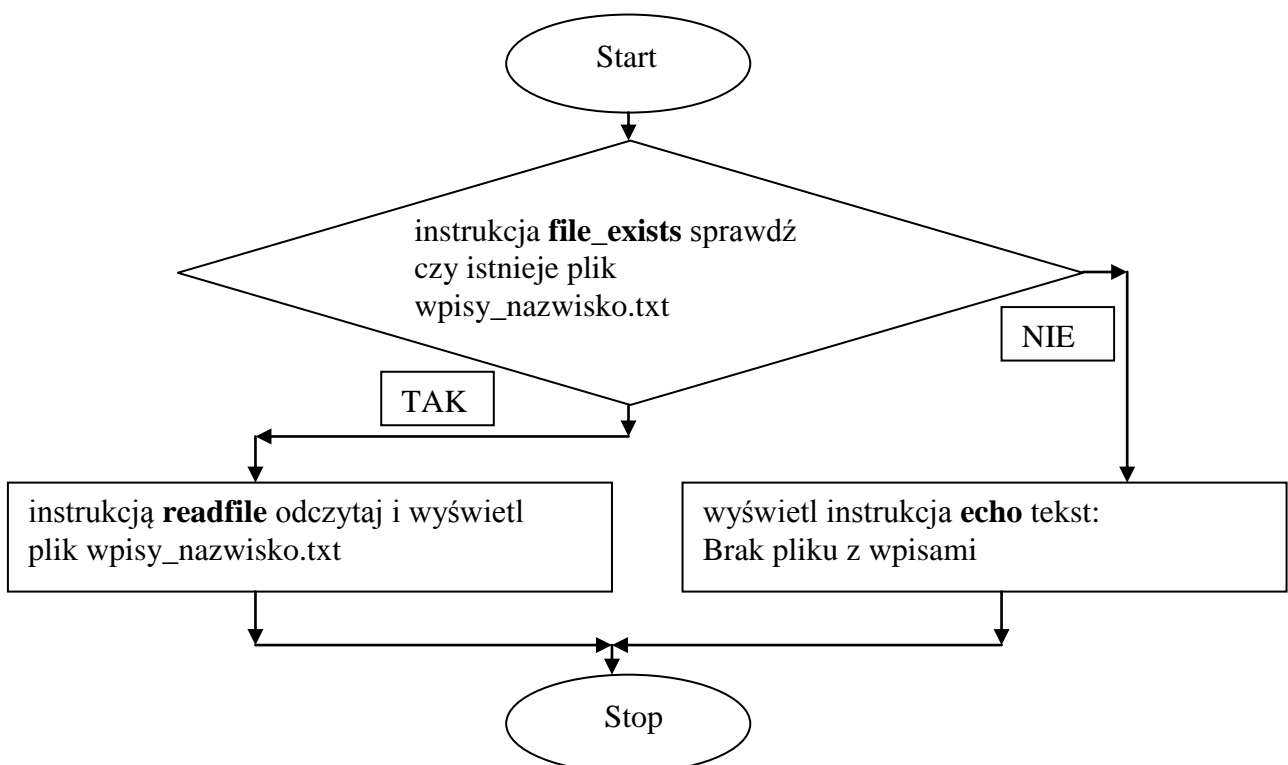
Tworzenie pliku **sys.php**. Zadaniem tego pliku jest obsługa menu, czyli podłączenie :

- skrypt wyświetlający newsy → **lista.php**
- skrypt wyświetlający formularz do wypełnienia → **dodaj.php**

Który skrypt zostanie uruchomiony decyduje zawartość zmiennej \$pokaz

Etap5

skrypt wyświetlający newsy → **lista.php**



Etap6

Skrypt **dodaj.php**

funkcja **info(\$s,\$klasa)**→wyświetlanie komunikatów.

parametr **\$s**→treść komunikatów,

parametr **\$klasa**→określa klasę (CSS), jaka ma zostać użyta do wyświetlania,

klasa **blad**→komunikat o błędach w czerwonej ramce

klasa **ok**→komunikat pozytywny w zielonej ramce

funkcja **konwertuj(\$s)**→konwersja danych z formularza do formatu zapisu do pliku tekstowego czyli usuwanie znaczników HTML z wyjątkiem ****, **<i>**, **<u>** oraz usuwanie znaku \ przed cudzysłowów oraz zamiana znaków cudzysłowów na zapisy **"**;

Inne algorytmy skryptu:

- sprawdzanie czy formularz został przesłany
- konwersja pola formularza
- sprawdzanie poprawności formularza
- jeżeli zmienna \$blad nie zawiera żadnych komunikatów dodajemy nowy wpis do książki
- dodanie nowego wpisu do książki polega na wczytaniu pliku wpis_nazwiska.txt do tablicy (\$dane=file('wpisy.txt'); else \$dane=array();), dołączenie do pliku wpis_nazwiska.txt nowego wpisu, wpis został umieszczony na koniec pliku, przepisanie pliku, tak aby wpisy były od najmłodszego do najstarszego.

```
<h2>Dodaj nowy news</h2>
<?php
function info($s,$klasa)
{
    echo ('<div align="center"><p class="'. $klasa. '">'. $s. '</p></div>');
}

function konwertuj($s)
{
    $s=strip_tags($s,'<b><i><u>');
    $s=stripslashes($s);
    $s=str_replace('"','&quot;',$s);
    return ($s);
}

if (isset($_POST['autor'])) // sprawdzanie czy formularz został przesłany
{
    $_POST['autor']=konwertuj($_POST['autor']); // konwersja pola formularza
    $_POST['email']=konwertuj($_POST['email']);
    $_POST['zrodlo']=konwertuj($_POST['zrodlo']);
    $_POST['tytul']=konwertuj($_POST['tytul']);
    $_POST['tresc']=konwertuj($_POST['tresc']);

    $blad=""; // sprawdzanie poprawności formularza
    if (strlen($_POST['tresc'])<10 || strlen($_POST['tresc'])>1024) $blad='Treść newsa
ma nieprawidłową długość';
```

```

    if (strlen($_POST['tytul'])<10 || strlen($_POST['tytul'])>200) $blad='Tytuł newsa
ma nieprawidłową długość';
    if (! ereg ("^.+@.+\\..+$", $_POST['email'])) $blad='Podano niepoprawny adres e-
mail';
    if (strlen($_POST['autor'])<3) $blad='źle wprowadzono informacje o autorze';
    if ($blad!="") info($blad,'blad');
    else {
        //zapisujemy dane do pliku
        if (file_exists('wpisy.txt')) $dane=file('wpisy.txt'); else $dane=array();

        if ($plik=fopen('wpisy.txt','w'))
        {
            flock($plik,LOCK_EX);
            fputs($plik,"<table border='0'>");
            fputs($plik,"<tr><td><b>$_POST[tytul]</b></td><td
align='right'>".date('d.m.Y H:i:s')."</td></tr>");
            fputs($plik,"<tr><td colspan='2'><i>$_POST[tresc]</i></td></tr>");
            fputs($plik,"<tr><td><b>Autor:</b>&nbsp;<a
href='mailto:$_POST[email]'>$_POST[autor]</a></td><td
align='right'><b>źródło:</b>&nbsp;<$_POST[zrodlo]</td></tr>");
            fputs($plik,"</table><hr />");
            for ($i=0;$i<count($dane);$i++) fputs($plik,$dane[$i]."\r\n");
            flock($plik,LOCK_UN);
            fclose($plik);
            info('Twój news został zapisany.','ok');
        } else info('Wystąpiły problemy z zapisaniem pliku.');
```

```

        include('lista.php');
        return;
    }
} else
{
    $_POST['autor']="";
    $_POST['email']="";
    $_POST['zrodlo']="";
    $_POST['tresc']="";
    $_POST['tytul']="";
    echo('<p>Wypełnij poprawnie poniższy formularz.</p>');
}
?>

<form action="index.php" method="POST">
<input type="hidden" name="pokaz" value="dodaj" />
<table border="0">
<tr><td>Autor:</td><td><input size="30" type="text" name="autor" value="<?php
echo $_POST['autor']; ?>" /></td></tr>
<tr><td>Adres e-mail:</td><td><input size="30" type="text" name="email"
value="<?php echo $_POST['email']; ?>" /></td></tr>
<tr><td>źródło:</td><td><input size="30" type="text" name="zrodlo" value="<?php
echo $_POST['zrodlo']; ?>" /></td></tr>
<tr><td>Tytuł:</td><td><input size="30" type="text" name="tytul" value="<?php

```

```

echo $_POST['tytul']; ?>" /></td></tr>
<tr><td>Treść:</td><td><textarea cols="40" rows="5" name="tresc"><?php echo
$_POST['tresc']; ?></textarea></td></tr>
<tr><td></td><td><input type="submit" value="Dodaj nowy news" /></td></tr>
</table>
</form>

```

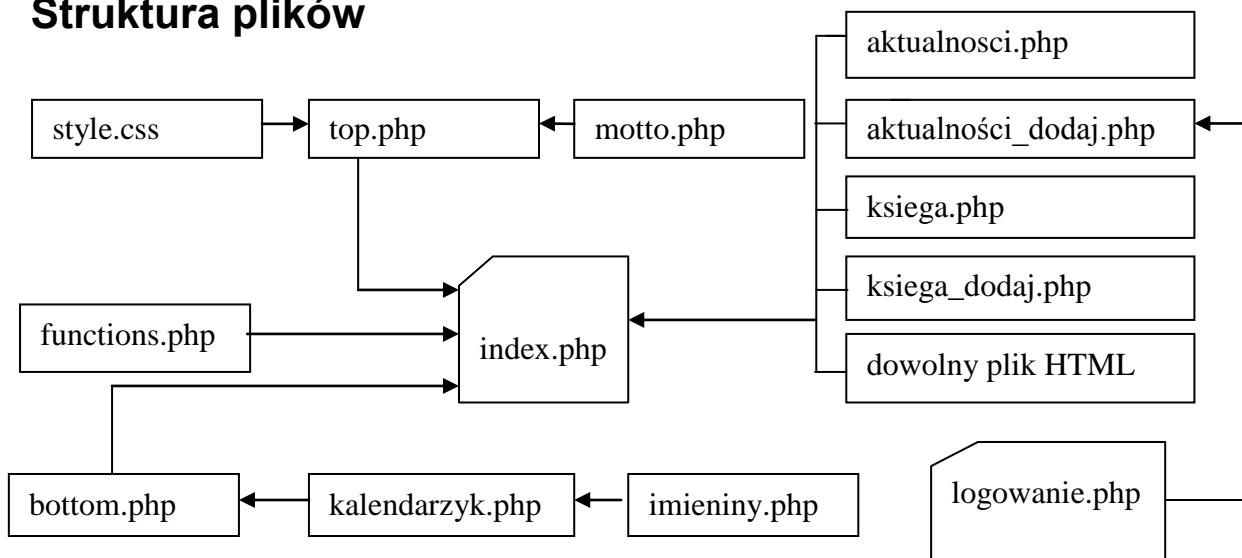
Zadanie29

Zadanie30

Temat: Portal o nazwie Miniportal. Zawiera aktualności (może dodać tylko administrator po zalogowaniu) oraz księgę gości (może dodać każdy) obie usługi będą zapisywane w bazie SQL.



Struktura plików



Etap1

Napisz skrypt o **nazwie nazwisko.sql**

a)stwórz w MySQL bazę o nazwie **portal_nazwisko**

b)stwórz tabelę o nazwie **aktualności_nazwisko**

```
tytul VARCHAR(60),  
tresc BLOB,  
autor VARCHAR(20),  
zrodlo VARCHAR(30),  
data_dodania INT,  
id INT NOT NULL AUTO_INCREMENT,  
PRIMARY KEY(id);
```

c)stwórz tabelę o nazwie **ksiega_nazwisko**

```
opinia BLOB,  
autor VARCHAR(30),  
email VARCHAR(40),  
data_dodania INT,  
id INT NOT NULL AUTO_INCREMENT,  
PRIMARY KEY(id);
```

Etap2

Załącz folder z30 (w miejscu gdzie bedziesz mógł testowac portal)

Plik index.php

Realizuje:

- inicjacja sesji
- dołączenie plik z funkcjami functions.php
- tworzenie zmiennej \$pokaz z wartością pobranej metodą GET
- nawiązanie połączenia z serwerem MySQL
- dołączenie plik z top.php
- dołączenie pliku na podstawie wartości zmiennej **\$pokaz**
 - a)**ksiega.php**→wyświetlenie tabeli (Mysql) **tabela**,
 - b)**ksiega_dodaj.php**→wyswietlanie formularza, który pozwala dopisywać nowe wpisy do Księgi gości. Jest to dopisanie nowego wiersza w tabeli (Mysql),
 - c)**aktualności.php**--.wyświetla aktualności (nisów) z tabeli **aktualności**,
 - d)**aktualności_dodaj.php**→dodawanie newsów oraz sprawdzanie czy użytkownik jest zalogowany (na podstawie analizy sesji), jeśli nie jest zalogowany to uruchomienie pliku **logowanie.php**.
- dołączenie pliku **bottom.php**
- zamknięcie połączenia

```
<?php  
session_start();  
include("functions.php");  
$dozwolone=array('aktualnosci','ksiega','ksiega_dodaj','aktualnosci_dodaj');  
if (isset($_GET['pokaz'])) $pokaz=$_GET['pokaz']; else $pokaz=$dozwolone[0];  
if (isset($_POST['pokaz'])) $pokaz=$_POST['pokaz'];  
if (strpos($pokaz,'.')!==false) $pokaz=$dozwolone[0];  
if (file_exists($pokaz.'.php') && in_array($pokaz,$dozwolone)) $pokaz='.php';
```

```

else $pokaz='html/'.$pokaz.'.htm';
$link = mysql_connect("127.0.0.1","root", "haslo") or die ("Nie można nawiązać
połączenia.");
mysql_select_db("portal");
include('top.php');
if (file_exists($pokaz)) include($pokaz); else info('Nie znaleziono pliku.','blad');
include('bottom.php');
mysql_close($link);
?>

```

Etap3

Do folderu z30 dograj pliki motto.php, imieniny.php i nne(z innych zadań)

Etap4

Utwórz plik **funtions.php**, który będzie zawierał dwie znane funkcje z poprzednich zadań **info()** i **konwertuj()**.

Etap5

Plik **top.php** jest nagłówkiem strony. Zawiera tabelkę z trzema kolumnami.

Wyświetlanie menu naszej strony	Wyświetlanie na podstawie zmiennej \$pokaz (np. księga gości)	Wyświetlanie skryptów <ul style="list-style-type: none"> • kalendarzyk.php • imieniny.php • linki do zaprzyjaźnionych stron
---------------------------------	--	--

kopiuujemy skrypt **motto.php** do folderu z portalem

kopiuujemy z ramki poniżej skrypt **top.php** do folderu z portalem

```

<html>
<head>
  <meta http-equiv="Content-type" content="text/html;
    charset=iso-8859-2" />
  <title>MiniPortal</title>
  <link rel="stylesheet" href="style.css" type="text/css" media="all" />
</head>
<body>
<h1 id="tytul">MiniPortal</h1>
<table border="0" cellpadding="0" cellspacing="0" width="95%" align="center">
<tr><td class="motto"><?php include("motto.php");?></td></tr></table>
<table class="tab" align="center">
<tr>
<td width="150" valign="top">
  <!-- menu po lewej -->
  <table class="panel">
    <tr><th class="panel">Menu</th></tr>
    <tr><td style="padding-left:10px;">
      &raquo; <a href="index.php">Aktualności</a><br />
      &raquo; <a href="index.php?pokaz=ksiega">Księga gości</a><br />
      <hr />
      &raquo; <a href="index.php?pokaz=omnie">O mnie</a><br />
    </td></tr>
  </table>
</td><td valign="top" style="padding-left:20px;padding-right:20px;">

```

Etap6

Tworzymy plik bootom.php
zawiera on prawe menu czyli:

- wstawianie pliku kalendarzyk.php oraz pośrednio plik imieniny.php
- linki do stron <http://www.zse.gda.pl> oraz <http://binboy.sphere.pl>
- stopka Copyright © 2014 by Twoje nazwisko
- dopisz do menu link O mnie, oddzielony linia od linków istniejących (tam gdzie są dwa linki)

```
</td><td width="150" valign="top">
<!-- menu po prawej -->
<table class="panel">
<tr><th class="panel">Kalendarz</th></tr>
<tr><td align="center">
<?php include('kalendarzyk.php'); ?>
</td></tr>
</table>
<br />
<table class="panel">
<tr><th class="panel">Polecamy</th></tr>
<tr><td>
&raquo; <a href=" Wisz pierwszy link">Komputer świat Ekspert</a><br />
&raquo; <a href="wpisz drugi link">Binboy at Sphere</a><br />
</td></tr>
</table>
</td>
</tr>
</table>
<table border="0" cellpadding="0" cellspacing="0" width="95%" align="center">
<tr><td class="stopka">Copyright &copy; 2006 by Twoje nazwisko</td></tr></table>
</body>
</html>
```

Etap7

plik ksiega.php

```
<h1>Księga gości</hr /></h1>
<?php
$query='SELECT * FROM ksiega ORDER BY data_dodania DESC LIMIT 10';
$result = mysql_query ($query);
while ($row = mysql_fetch_array($result))
{
    echo('<table border="0">');
    echo('<tr><td><a href="mailto:'. $row['email']. ">'. $row['autor']. '</a></td><td align="right">'.date("d.m.Y H:i:s", $row['data_dodania']). '</td></tr>');
    echo('<tr><td colspan="2" style="text-align:justify;">'. $row['opinia']. '</td></tr>');
    echo('</table><hr />');
}
?>

[ <a href="index.php?pokaz=ksiega_dodaj">Dodaj wpis do Księgi gości</a> ]
```

Etap8

plik ksiega_dodaj.php

```
<?php
if (isset($_POST['autor']))
{
    //dostosowujemy wszystkie pola
    $_POST['autor']=konwertuj($_POST['autor']);
    $_POST['opinia']=konwertuj($_POST['opinia']);
    $_POST['email']=konwertuj($_POST['email']);
    //kontrolujemy poprawność
    $blad="";
    if (strlen($_POST['opinia'])<10 || strlen($_POST['opinia'])>1024) $blad='Treść
opinii ma nieprawidłow± długość!';
    if (! ereg ("^.+@.+\\..+$", $_POST['email'])) $blad='Podano niepoprawny adres e-
mail';
    if (strlen($_POST['autor'])<3) $blad='Źle wprowadzono informacje o autorze!';
    if ($blad!="") info($blad,'blad');
    else
    {
        //dodajemy do tabeli
        mysql_query("INSERT INTO ksiega
VALUES(\"$_POST[opinia]\", \"$_POST[autor]\", \"$_POST[email]\", ".time().", NULL)")
or die(mysql_error());
        info('Twoja opinia została zapisana.', 'ok');
        include('ksiega.php');
        return;
    }
} else
{
    $_POST['autor']="";
    $_POST['email']="";
    $_POST['opinia']="";
}
?>
<h1>Dodaj wpis do Księgi gości<hr /></h1>
<form action="index.php" method="POST">
<input type="hidden" name="pokaz" value="ksiega_dodaj" />
<table border="0">
<tr><td>Autor:</td><td><input size="30" type="text" name="autor" value="<?php
echo $_POST['autor']; ?>" /></td></tr>
<tr><td>E-mail:</td><td><input size="30" type="text" name="email" value="<?php
echo $_POST['email']; ?>" /></td></tr>
<tr><td>Opinia:</td><td><textarea cols="40" rows="5" name="opinia"><?php echo
$_POST['opinia']; ?></textarea></td></tr>
<tr><td></td><td><input type="submit" value="Dodaj opinię" /></td></tr>
</table>
</form>
```

Etap9

plik aktualnosci.php

```
<h1>Aktualności</h1>
<?php
$query='SELECT * FROM aktualnosci ORDER BY data_dodania DESC LIMIT 10';
$result = mysql_query ($query);
while ($row = mysql_fetch_array($result))
{
    echo('<table border="0">');
    echo('<tr><td><b>'. $row['tytul']. '</b></td><td align="right">'.date("d.m.Y
H:i:s", $row['data_dodania']). '</td></tr>');
    echo('<tr><td colspan="2" style="text-align:justify;">'. $row['tresc']. '</td></tr>');
    echo('<tr><td>Dodał: <b>'. $row['autor']. '</b></td><td align="right">Źródło:
<b>'. $row['zrodlo']. '</b></td></tr>');
    echo('</table><hr />');
}
?>
[ <a href="index.php?pokaz=aktualnosci_dodaj">Dodaj nowy news</a> ]
```

Etap10

plik aktualnosci_dodaj.php

- zmień hasło logowania do newsów na twoje inicjały

```
<?php
//jak jest parametr $_POST['haslo'] i jest w nim słowo 'tajne' to logujemy
if (isset($_POST['haslo']) && $_POST['haslo']=='tajne')
$_SESSION['zalogowany']=true;
//jeśli nie jesteśmy zalogowani, to wyświetlamy skrypt logowanie.php
if (!isset($_SESSION['zalogowany']) || $_SESSION['zalogowany']==false)
{
    include('logowanie.php');
    return;
}
//jak jest pole autor, tzn. że przesyłamy formularz
if (isset($_POST['autor']))
{
    //dostosowujemy wszystkie pola
    $_POST['autor']=konwertuj($_POST['autor']);
    $_POST['tytul']=konwertuj($_POST['tytul']);
    $_POST['tresc']=konwertuj($_POST['tresc']);
    $_POST['zrodlo']=konwertuj($_POST['zrodlo']);
    //kontrolujemy poprawność
    $blad="";
    if (strlen($_POST['tresc'])<10 || strlen($_POST['tresc'])>1024) $blad='Treść newsa
ma nieprawidłową długość!';
    if (strlen($_POST['tytul'])<10 || strlen($_POST['tytul'])>200) $blad='Tytuł newsa ma
nieprawidłową długość!';
    if (strlen($_POST['zrodlo'])<2 || strlen($_POST['zrodlo'])>20) $blad='Źródło ma
nieprawidłową długość!';
    if (strlen($_POST['autor'])<3) $blad='źle wprowadzono informacje o autorze!';
    if ($blad!="") info($blad,'blad');
```

```

        else
        {
            //dodajemy do tabeli
            mysql_query("INSERT INTO aktualnosci
VALUES(\"$_POST[tytul]\", \"$_POST[tresc]\", \"$_POST[autor]\", \"$_POST[zrodlo]\", ".
time().", NULL)") or die(mysql_error());
            info('Twój news został zapisany.', 'ok');
            include('aktualnosci.php');
            return;
        }

    } else
    {
        $_POST['autor']="";
        $_POST['tresc']="";
        $_POST['tytul']="";
        $_POST['zrodlo']="";
    }
?>
<h1>Dodaj nowy nius<hr /></h1>
<form action="index.php" method="POST">
    <input type="hidden" name="pokaz" value="aktualnosci_dodaj" />
    <table border="0">
        <tr><td>Tytuł:</td><td><input size="30" type="text" name="tytul" value="<?php
echo $_POST['tytul']; ?>" /></td></tr>
        <tr><td>Treść:</td><td><textarea cols="40" rows="5" name="tresc"><?php echo
$_POST['tresc']; ?></textarea></td></tr>
        <tr><td>Źródło:</td><td><input size="30" type="text" name="zrodlo" value="<?php
echo $_POST['zrodlo']; ?>" /></td></tr>
        <tr><td>Autor:</td><td><input size="30" type="text" name="autor" value="<?php
echo $_POST['autor']; ?>" /></td></tr>
        <tr><td></td><td><input type="submit" value="Dodaj nowy nius" /></td></tr>
    </table>
</form>

```

Etap11

plik logowanie.php

```

<h1>Logowanie<hr /></h1>
<?php
//jak jest pole $_POST['haslo'], tzn. że przesyłaliśmy
//ten formularz i się nie zalogowaliśmy, czyli podaliśmy błędne hasło
if (isset($_POST['haslo'])) info('Podano złe hasło!', 'blad');
?>
<form action="index.php" method="POST">
    <input type="hidden" name="pokaz" value="aktualnosci_dodaj" />
    <table border="0">
        <tr><td>Proszę podać hasło:</td><td><input size="30" type="password"
name="haslo" value="" /></td></tr>
        <tr><td></td><td><input type="submit" value="Zaloguj" /></td></tr>
    </table>

```

</form>

Etap12

plik CSS

```
body {
  background:#ffd7af;
  font-family:arial;
  font-size:12px;
  color:#713800;
}
table, tr, td {
  font-family:arial;
  font-size:12px;
  color:#713800;
}
#tytul {
  text-align:center;
  color:#cf6700;
  text-decoration:underline;
}
.motto {
  text-align:center;
  background:#ff9e3d;
  color:#5f2f00;
  border-top:1px dotted #512800;
  border-left:1px dotted #512800;
  border-right:1px dotted #512800;
}
.stopka {
  text-align:center;
  background:#ff9e3d;
  color:#5f2f00;
  border-bottom:1px dotted #512800;
  border-left:1px dotted #512800;
  border-right:1px dotted #512800;
}
.tab {
  border-left:1px dotted #512800;
  border-right:1px dotted #512800;
  padding:0px;
  margin:0px;
  width:95%;
}
table.panel {
  width:100%;
  border:1px solid #dd6e00;
}
th.panel {
  background:#dd6e00;
```

```
    color:white;
}
a,a:visited {color:#8f4700;}
a:hover {color:#ff870f;}
h1,h2,h3,h4,h5,h6 {color:#dd6e00;}
h1 {font-size:20px;}
hr {border:1px;color:#dd6e00;background:#dd6e00;height:1px;}
input, textarea {background:#ffcc99;border:1px solid #ef7700;color:#ef7700;}
.blad {color: #ff0000;border:2px solid #ff0000; text-align:center;font-
weight:bold;width:80%;padding:5px;}
.ok {color: #dd6e00;border:2px solid #dd6e00; text-align:center;font-
weight:bold;width:80%;padding:5px;}
```


<?php

\$imiona[0]='Masława, Mieczysława, Mieszka';
\$imiona[1]='Bazylego, Makarego, Narcyzy';
\$imiona[2]='Arlety, Danuty, Lucjana';
\$imiona[3]='Anieli, Elżbiety, Tytusa';
\$imiona[4]='Edwarda, Hanny, Szymona';
\$imiona[5]='Kacpra, Melchiora, Baltazara';
\$imiona[6]='Juliana, Lucjana, Walentyny';
\$imiona[7]='Artura, Rajmunda, Seweryny';
\$imiona[8]='Adriana, Alicji, Teresy';
\$imiona[9]='Ady, Jana, Wilhelma';
\$imiona[10]='Feliksa, Honoraty, Marty';
\$imiona[11]='Bernarda, Czesławy, Greta';
\$imiona[12]='Bogumiła, Bogumiły, Weroniki';
\$imiona[13]='Feliksa, Hilarego, Martyny';
\$imiona[14]='Arnolda, Dory, Pawła';
\$imiona[15]='Mascelego, Walerii, Włodzimierza';
\$imiona[16]='Antoniego, Henryki, Mariana';
\$imiona[17]='Beatrycze, Małgorzaty, Piotra';
\$imiona[18]='Erwiny, Henryka, Mariusza';
\$imiona[19]='Fabioli, Miły, Sebastiana';
\$imiona[20]='Agnieszki, Jarosława, Nory';
\$imiona[21]='Dominiki, Mateusza, Wincentego';
\$imiona[22]='Fernandy, Jana, Rajmunda';
\$imiona[23]='Felicji, Roberta, Sławy';
\$imiona[24]='Miłosza, Pawła, Tatiany';
\$imiona[25]='Lutosława, Normy, Pauliny';
\$imiona[26]='Anieli, Juliana, Przemysława';
\$imiona[27]='Agnieszki, Kariny, Lesława';
\$imiona[28]='Franciszka, Konstancji, Salomei';
\$imiona[29]='Martyny, Macieja, Teofila';
\$imiona[30]='Joanny, Ksawerego, Luizy';
\$imiona[31]='Brygidy, Dobrogniewa, Ignacego';
\$imiona[32]='Kornela, Marii, Mirosławy';
\$imiona[33]='Błażeja, Joanny, Telimeny';
\$imiona[34]='Andrzeja, Mariusza, Weroniki';
\$imiona[35]='Agaty, Filipa, Justyniana';
\$imiona[36]='Amandy, Bogdana, Doroty';
\$imiona[37]='Ryszarda, Teodora, Wilhelminy';
\$imiona[38]='Irminy, Piotra, Sylwii';
\$imiona[39]='Bernarda, Eryki, Rajmunda';
\$imiona[40]='Elwiry, Elizy, Jacka';
\$imiona[41]='Bernadetty, Marii, Olgierda';
\$imiona[42]='Czasława, Damiana, Normy';
\$imiona[43]='Grzegorza, Lesława, Katarzyny';
\$imiona[44]='Liliany, Walentyny, Walentego';
\$imiona[45]='Arnolda, Jowity, Georginy';
\$imiona[46]='Danuty, Daniela, Juliany';
\$imiona[47]='Donata, Gizeli, Łukasza';
\$imiona[48]='Konstancji, Krystiana, Sylwany';

\$imiona[49]='Bettiny, Konrada, Mirosława';
\$imiona[50]='Anety, Lehca, Leona';
\$imiona[51]='Eleonory, Lenki, Kiejstuta';
\$imiona[52]='Małgorzaty, Marty, Nikifora';
\$imiona[53]='Damiana, Romana, Romany';
\$imiona[54]='Boguty, Bogusza, Macieja';
\$imiona[55]='Almy, Cezarego, Jarosława';
\$imiona[56]='Bogumiła, Eweliny, Mirosława';
\$imiona[57]='Gagrieli, Liwii, Leonarda';
\$imiona[58]='Ludomira, Makarego, Wiliany';
\$imiona[59]='Lecha, Lutomira';//29
\$imiona[60]='Albina, Antoniny, Radosławy';
\$imiona[61]='Halszki, Heleny, Karola';
\$imiona[62]='Kingi, Maryna, Tycjana';
\$imiona[63]='Adrianny, Kazimierza, Wacława';
\$imiona[64]='Aurory, Fryderyka, Oliwii';
\$imiona[65]='Jordana, Marcina, Róży';
\$imiona[66]='Flicyty, Kajetana, Pauli';
\$imiona[67]='Beaty, Juliana, Wincentego';
\$imiona[68]='Dominika, Franciszki, Katarzyny';
\$imiona[69]='Bożysławy, Cypriana, Marcelego';
\$imiona[70]='Konstantego, Ludosława, Rozyny';
\$imiona[71]='Grzegorza, Justyny, Józefiny';
\$imiona[72]='Bożeny, Krystyny, Marka';
\$imiona[73]='Dalii, Leona, Matyldy';
\$imiona[74]='Delfiny, Longina, Ludwiki';
\$imiona[75]='Izabeli, Henryka, Oktawii';
\$imiona[76]='Reginy, Patryka, Zdyszka';
\$imiona[77]='Edwarda, Narcyza, Zbysława';
\$imiona[78]='Aleksandryny, Józefa, Nicety';
\$imiona[79]='Joachima, Kiry, Maurycego';
\$imiona[80]='Benedykta, Lubomiry, Lubomira';
\$imiona[81]='Bogusława, Jagody, Katarzyny';
\$imiona[82]='Feliksa, Konrada, Zbysławy';
\$imiona[83]='Gabrieli, Marka, Seweryna';
\$imiona[84]='Bolka, Cezaryny, Marioli';
\$imiona[85]='Dory, Olgi, Teodora';
\$imiona[86]='Ernesta, Jana, Marka';
\$imiona[87]='Anieli, Kasrota, Soni';
\$imiona[88]='Marka, Wiktoryny, Zenona';
\$imiona[89]='Amelii, Dobromira, Leonarda';
\$imiona[90]='Balbiny, Kamila, Kornelii';
\$imiona[91]='Chryzamtyny, Grażyny, Zygmunta';
\$imiona[92]='Franciszka, Malwiny, Władysława';
\$imiona[93]='Pankracego, Renaty, Ryszarda';
\$imiona[94]='Benedykta, Izodory, Wacławy';
\$imiona[95]='Ireny, Kleofasa, Wincentego';
\$imiona[96]='Ady, Celestyny, Ireneusza';
\$imiona[97]='Donata, Herminy, Rufina';
\$imiona[98]='Amadeusza, Cezaryny, Juliany';

\$imiona[99]='Mai, Marcelego, Wadima';
\$imiona[100]='Borysławy, makarego, Michała';
\$imiona[101]='Filipa, Izoldy, Leona';
\$imiona[102]='Juliusza, Lubosława, Wiktoryny';
\$imiona[103]='Artemona, Justyny, Przemysław';
\$imiona[104]='Bernarda, Martyny, Waleriana';
\$imiona[105]='Adolfiny, Odetty, Wacława';
\$imiona[106]='Bernarda, Biruty, Erwina';
\$imiona[107]='Anicety, Klary, Rudolfina';
\$imiona[108]='Apoloniusza, Bogusławy, Gościławy';
\$imiona[109]='Alfa, Leonii, Tytusa';
\$imiona[110]='Agnieszki, Amalii, Czecha';
\$imiona[111]='Jarosława, Konrada, Selmy';
\$imiona[112]='Łukasza, Kai, Nastazji';
\$imiona[113]='Ilony, Jerzego, Wojciecha';
\$imiona[114]='Bony, Horacji, Jerzego';
\$imiona[115]='Jarosława, Marka, Wiki';
\$imiona[116]='Marii, Marzeny, Ryszarda';
\$imiona[117]='Sergiusza, Teofila, Zyty';
\$imiona[118]='Bogny, Walerii, Witalisa';
\$imiona[119]='Hugona, Piotra, Roberty';
\$imiona[120]='Balladyny, Lilli, Mariana';
\$imiona[121]='Józefa, Lubomira, Ramony';
\$imiona[122]='Longiny, Toli, Zygmunta';
\$imiona[123]='Jaropełka, Marii, Niny';
\$imiona[124]='Floriana, Michała, Moniki';
\$imiona[125]='Irydy, Tamary, Waldemara';
\$imiona[126]='Beniny, Filipa, Judyty';
\$imiona[127]='Augusta, Gizeli, Ludomiry';
\$imiona[128]='Kornela, Lizy, Stanisława';
\$imiona[129]='Grzegorza, Karoliny, Karola';
\$imiona[130]='Antoniny, Izydory, Jana';
\$imiona[131]='Igi, Mamerta, Miry';
\$imiona[132]='Dominika, Imeldy, Pankracego';
\$imiona[133]='Agnieszki, Magdaleny, Serwacego';
\$imiona[134]='Bonifacego, Julity, Macieja';
\$imiona[135]='Dionizego, Nadziei, Zofii';
\$imiona[136]='Andrzeja, Jędrzeja, Małgorzaty';
\$imiona[137]='Brunony, Sławomira, Wery';
\$imiona[138]='Alicji, Edwina, Eryka';
\$imiona[139]='Celestyny, Iwony, Piotra';
\$imiona[140]='Bazylego, Bernardyna, Krystyny';
\$imiona[141]='Jana, Moniki, Wiktora';
\$imiona[142]='Emila, Neleny, Romy';
\$imiona[143]='Leonejusza, Michała, Renaty';
\$imiona[144]='Joanny, Zdenka, Zuzanny';
\$imiona[145]='Borysa, Magdy, Marii-Magdaleny';
\$imiona[146]='Eweliny, Jana, Pawła';
\$imiona[147]='Amandy, Jana, Juliana';
\$imiona[148]='Augustyna, Ingi, Jaromira';

\$imiona[149]='Benity, Maksymiliana, Teodozji';
\$imiona[150]='Ferdynanda, Gryzeldy, Zyndrama';
\$imiona[151]='Anieli, Feliksa, Kamili';
\$imiona[152]='Gracji, Jakuba, Konrada';
\$imiona[153]='Erazma, Marianny, Marzeny';
\$imiona[154]='Anatola, Leszka, Tamary';
\$imiona[155]='Christy, Helgi, Karola';
\$imiona[156]='Bonifacego, Kiry, Waltera';
\$imiona[157]='Laury, Laurentego, Nory';
\$imiona[158]='Ariadny, Jarosława, Roberta';
\$imiona[159]='Ady, Celii, Medarda';
\$imiona[160]='Anny-Marii, Felicjana, Sławoja';
\$imiona[161]='Bogumiła, Diany, Małgorzaty';
\$imiona[162]='Barnaby, Benedykta, Flory';
\$imiona[163]='Gwidona, Leonii, Niny';
\$imiona[164]='Antoniego, Gracji, Lucjana';
\$imiona[165]='Bazylego, Elizy, Justyny';
\$imiona[166]='Jolanty, Lotara, Wita';
\$imiona[167]='Aliny, Anety, Benona';
\$imiona[168]='Laury, Leszka, Marcjana';
\$imiona[169]='Elżbiety, Marka, Pauli';
\$imiona[170]='Gerwazego, Protazego, Sylwii';
\$imiona[171]='Bogny, Rafaeli, Rafała';
\$imiona[172]='Alicji, Alojzego, Rudolfa';
\$imiona[173]='Pauliny, Sabiny, Tomasza';
\$imiona[174]='Albina, Wandy, Zenona';
\$imiona[175]='Danuty, Jana, Janiny';
\$imiona[176]='Łucji, Witolda, Wilhelma';
\$imiona[177]='Jana, Pauliny, Rudolfiny';
\$imiona[178]='Cypriana, Emanuela, Władysława';
\$imiona[179]='Florentyny, Ligii, Leona';
\$imiona[180]='Pawła, Piotra, Salomei';
\$imiona[181]='Arnolda, Emiliany, Lucyny';
\$imiona[182]='Bogusza, Haliny, Mariana';
\$imiona[183]='Kariny, Serafiny, Urbana';
\$imiona[184]='Anatola, Jacka, Mirosławy';
\$imiona[185]='Aureli, Malwiny, Zygryda';
\$imiona[186]='Antoniego, Bartłomieja, Karoliny';
\$imiona[187]='Dominiki, Jaropełka, Łucji';
\$imiona[188]='Estery, Kiry, Rudolfa';
\$imiona[189]='Arnolda, Edgara, Elżbiety';
\$imiona[190]='Hieronima, Palomy, Weroniki';
\$imiona[191]='Filipa, Sylwany, Witalisa';
\$imiona[192]='Benedykta, Kariny, Olgi';
\$imiona[193]='Brunona, Jana, Wery';
\$imiona[194]='Danieli, Irwina, Małgorzaty';
\$imiona[195]='Kamili, Kamila, Marcelego';
\$imiona[196]='Henryka, Igi, Włodzimierza';
\$imiona[197]='Eustachego, Mariki, Mirelli';
\$imiona[198]='Aleksa, Bogdana, Martyny';

\$imiona[199]='Kamila, Karoliny, Roberta';
\$imiona[200]='Alfreny, Rufina, Wincentego';
\$imiona[201]='Fryderyka, Małgorzaty, Seweryny';
\$imiona[202]='Danieli, Wawrzyńca, Wiktora';
\$imiona[203]='Magdaleny, Mileny, Wawrzyńca';
\$imiona[204]='Sławy, Sławosza, Żelysławy';
\$imiona[205]='Kingi, Krystyna, Michaliny';
\$imiona[206]='Jakuba, Krzysztofa, Walentyny';
\$imiona[207]='Anny, Mirosławy, Joachima';
\$imiona[208]='Aureliusza, Natalii, Rudolfa';
\$imiona[209]='Ady, Wiwiany, Sylwiusza';
\$imiona[210]='Marty, Konstantego, Olafa';
\$imiona[211]='Julity, Ludmiły, Zdobysława';
\$imiona[212]='Ignacego, Lodomiry, Romana';
\$imiona[213]='Jarosława, Justyny, Nadziei';
\$imiona[214]='Gustawa, Kariny, Stefana';
\$imiona[215]='Augustyna, Kamelii, Lidii';
\$imiona[216]='Dominiki, Dominika, Protazego';
\$imiona[217]='Emila, Karoliny, Kary';
\$imiona[218]='Jakuba, Sławy, Wincentego';
\$imiona[219]='Donaty, Olechny, Kajetana';
\$imiona[220]='Izy, Rajmunda, Seweryna';
\$imiona[221]='Klary, Romana, Rozyny';
\$imiona[222]='Bianki, Borysa, Wawrzyńca';
\$imiona[223]='Luizy, Włodzmierza, Zuzanny';
\$imiona[224]='Hilarii, Juliana, Lecha';
\$imiona[225]='Elwiry, Hipolita, Radosławy';
\$imiona[226]='Alfreda, Maksymiliana, Selmy';
\$imiona[227]='Marii, napoleona, Stelii';
\$imiona[228]='Joachima, Nory, Stefana';
\$imiona[229]='Anity, Elizy, Mirona';
\$imiona[230]='Bogusława, Bronisława, Ilony';
\$imiona[231]='Emilii, Julinana, Konstancji';
\$imiona[232]='Bernarda, Sabiny, Samuela';
\$imiona[233]='Franciszka, Kazimiery, Ruty';
\$imiona[234]='Cezarego, Marii, Zygfrieda';
\$imiona[235]='Apolinarego, Miły, Róży';
\$imiona[236]='Bartosza, Jerzego, Maliny';
\$imiona[237]='Belii, Ludwika, Luizy';
\$imiona[238]='Ireneusza, Konstantego, Marii';
\$imiona[239]='Cezarego, Małgorzaty, Moniki';
\$imiona[240]='Adeliny, Erazma, Sobiesława';
\$imiona[241]='Beaty, Racibora, Sabiny';
\$imiona[242]='Benona, Jowity, Szczęsnego';
\$imiona[243]='Cyrusa, Izabeli, Rajmundy';
\$imiona[244]='Belindy, Bronisza, Idziego';
\$imiona[245]='Dionizy, Izy, Juliana';
\$imiona[246]='Joachima, Liliany, Szymona';
\$imiona[247]='Dalii, Idy, Rocha';
\$imiona[248]='Doroty, Justyna, Wawrzyńca';

\$imiona[249]='Beaty, Eugeniusza, Lidy';
\$imiona[250]='Reginy, Marka, Melchiora';
\$imiona[251]='Czcibora, Marii, Serafiny';
\$imiona[252]='Aldony, Jakuba, Sergiusza';
\$imiona[253]='Eligii, Irmy, Łukasza';
\$imiona[254]='Dagny, Jacka, Prota';
\$imiona[255]='Amadeusza, Gwidy, Sylwiny';
\$imiona[256]='Apolinarego, Eugenii, Lubomira';
\$imiona[257]='Bernarda, Mony, Roksany';
\$imiona[258]='Albina, Lolity, RONALDA';
\$imiona[259]='Jagienki, Kamili, Korneliusza';
\$imiona[260]='Franciszka, Lamberty, Narcyza';
\$imiona[261]='Ireny, Irminy, Stanisława';
\$imiona[262]='Januarego, Konstancji, Leopolda';
\$imiona[263]='Eustachego, Faustyny, Renaty';
\$imiona[264]='Darii, Mateusza, Wawrzyńca';
\$imiona[265]='Maury, Milany, Tomasza';
\$imiona[266]='Bogusława, Liwiusza, Tekli';
\$imiona[267]='Dory, Gerarda, Maryny';
\$imiona[268]='Aureli, Kamila, Kleofasa';
\$imiona[269]='Cypriana, Justyny, Łucji';
\$imiona[270]='Damiana, Mirabeli, Wincentego';
\$imiona[271]='Libuszy, Wacławy, Wacława';
\$imiona[272]='Michaliny, Michała, Rafała';
\$imiona[273]='Gerald, Honoriusza, Wery';
\$imiona[274]='Heloizy, Igora, Remigiusza';
\$imiona[275]='Racheli, Sławy, Teofila';
\$imiona[276]='Bogumiła, Gerarda, Józefy';
\$imiona[277]='Edwina, Rosławy, Rozalii';
\$imiona[278]='Flawii, Justyna, Rajmunda';
\$imiona[279]='Artura, Fryderyki, Petry';
\$imiona[280]='Krystyna, Marii, Marka';
\$imiona[281]='Brygidy, Loreny, Marcina';
\$imiona[282]='Arnolda, Ludwika, Sybili';
\$imiona[283]='Franciszka, Loretty, Poli';
\$imiona[284]='Aldony, Brunona, Emila';
\$imiona[285]='Krystyny, Maksa, Serafiny';
\$imiona[286]='Edwarda, Geraldyny, Teofila';
\$imiona[287]='Alany, Damiana, Liwii';
\$imiona[288]='Jadwigi, Leonarda, Teresy';
\$imiona[289]='Ambrożego, Florentyny, Gawła';
\$imiona[290]='Antonii, Ignacego, Wiktora';
\$imiona[291]='Hanny, Klementyny, Łukasza';
\$imiona[292]='Michaliny, Michała, Piotra';
\$imiona[293]='Ireny, Kleopatry, Witalisa';
\$imiona[294]='Celiny, Hilarego, Janusza';
\$imiona[295]='Haliszki, Lody, Przybysława';
\$imiona[296]='Edwarda, Marleny, Seweryna';
\$imiona[297]='Arety, Marty, Marcina';
\$imiona[298]='Ingi, Maurycego, Sambora';

\$imiona[299]='Ewarysta, Lucyny, Lutosławy';
\$imiona[300]='Twony, Noemi, Szymona';
\$imiona[301]='Narcyza, Serafina, Wioletty';
\$imiona[302]='Angeli, Przemysława, Zenobii';
\$imiona[303]='Augustyny, Łukasza, Urbana';
\$imiona[304]='Konrada, Seweryny, Wiktoryny';
\$imiona[305]='Bohdany, Henryka, Tobiasza';
\$imiona[306]='Huberta, Miły, Sylwii';
\$imiona[307]='Albertyny, Karola, Olgierda';
\$imiona[308]='Balladyny, Elżbiety, Sławomira';
\$imiona[309]='Arletty, Feliksa, Leonarda';
\$imiona[310]='Antoniego, Kaliny, Przemiły';
\$imiona[311]='Klaudii, Seweryna, Wiktoriusza';
\$imiona[312]='Anatolii, Gracji, Teodora';
\$imiona[313]='Leny, Lubomira, Natalii';
\$imiona[314]='Bartłomieja, Gertrudy, Marcina';
\$imiona[315]='Konrada, Renaty, Witolda';
\$imiona[316]='Arkadii, Krystyna, Stanisławy';
\$imiona[317]='Emila, Laury, Rogera';
\$imiona[318]='Amielii, Idalii, Leopolda';
\$imiona[319]='Edmunda, Marii, Marka';
\$imiona[320]='Grzegorza, Salomei, Walerii';
\$imiona[321]='Klaudyny, Romana, Tomasza';
\$imiona[322]='Elżbiety, Faustyny, Pawła';
\$imiona[323]='Anatola, Edyty, Rafała';
\$imiona[324]='Janusza, Marii, Reginy';
\$imiona[325]='Cecylii, Jonatana, Marka';
\$imiona[326]='Adeli, Felicyty, Klemensa';
\$imiona[327]='Emmy, Flory, Romana';
\$imiona[328]='Elżbiety, Katarzyny, Klemensa';
\$imiona[329]='Leona, Leonarda, Lesławy';
\$imiona[330]='Franciszka, Kseni, Maksymiliana';
\$imiona[331]='Jakuba, Stefana, Romy';
\$imiona[332]='Błażeja, Margerity, Saturnina';
\$imiona[333]='Andrzeja, Maury, Ondraszka';
\$imiona[334]='Blanki, Edmunda, Eligiusza';
\$imiona[335]='Balbiny, Ksawerego, Pauliny';
\$imiona[336]='Hilarego, Franciszki, Ksawery';
\$imiona[337]='Barbary, Hieronima, Krystiana';
\$imiona[338]='Kryspiny, Norberta, Sabiny';
\$imiona[339]='Dionizji, Leontyny, Mikołaja';
\$imiona[340]='Agaty, Dalii, Sobiesława';
\$imiona[341]='Delfiny, Marii, Wirginiusza';
\$imiona[342]='Anety, Leokadii, Wiesława';
\$imiona[343]='Danieli, Bohdana, Julii';
\$imiona[344]='Biny, Damazego, Waldemara';
\$imiona[345]='Ady, Aleksandra, Dagmary';
\$imiona[346]='Dalidy, Juliusza, Łucji';
\$imiona[347]='Alfreda, Izydora, Zoriny';
\$imiona[348]='Celiny, Ireneusza, Niny';

\$imiona[349]='Albiny, Sebastiana, Zdzisławy';
\$imiona[350]='Jolanty, Łukasza, Olimpi';
\$imiona[351]='Bogusława, Gracjana, Laury';
\$imiona[352]='Beniaminy, Dariusza, Gabrieli';
\$imiona[353]='Bogumiły, Dominika, Zefiryna';
\$imiona[354]='Honoraty, Seweryny, Tomasza';
\$imiona[355]='Bożeny, Drogomira, Zenona';
\$imiona[356]='Dagny, Sławomiry, Wiktora';
\$imiona[357]='Adama, Ewy, Irminy';
\$imiona[358]='Anety, Glorii, Piotra';
\$imiona[359]='Dionizego, Kaliksta, Szczepana';
\$imiona[360]='Fabioli, Jana, Żanety';
\$imiona[361]='Antoniusza, Cezarego, Teofilii';
\$imiona[362]='Dawida, Dionizy, Tomasza';
\$imiona[363]='Eugeniusza, Katarzyny, Sabiny';
\$imiona[364]='Mariusza, Melanii, Sylwestra';

?>

Nazwa ID	Co realizuje
#menu	{ width:150px;border:1px dotted blue;float:left;background:#d3d3ff;text-align:center; }
#main	{ margin-left:155px;border:1px solid;padding:5px; }
#kalendarz	{ text-align:center;background:#9595ff;color:white;font-weight:bold; }

```
$przestepny=date("L");
$dzien=date("z");
if ($dzien>58) $dzien=$dzien+1-$przestepny;
echo($simiona[$dzien]);
```

```
echo("<Script>
history.back()
</Script>");
```

Błąd→PHP: Cannot modify header information - headers already sent by...

Błąd ten występuje w momencie gdy wywoływany skrypt wyśle do przeglądarki jakiejkolwiek dane, a następnie będziemy chcieli przesłać dodatkowe nagłówki (np. przy pomocy funkcji header()).

```
<?php
echo 'Tutaj jest przykładowy tekst';
header('Location: http://www.nowy-adres-strony.pl/');
?>
```

Wywołanie takiego kodu spowoduje wystąpienie błędu:

Warning: Cannot modify header information - headers already sent by
(output started at /home/skrypt.php:3) in /home/skrypt.php on line 5

Oczywiście nasz przykładowy tekst zostanie również wyświetlony gdyż został on przesłany wcześniej.

WAŻNE: wystąpienie błędu spowoduje każdy znak (nawet spacja) wysłany przed wysłaniem nagłówka.

Gdzie szukać problemu

Błąd można zlokalizować analizując komunikat który wyświetlił parser PHP:

Warning: Cannot modify header information - headers already sent by
(output started at /home/skrypt.php:3) in /home/skrypt.php on line 5

Mamy zatem podany plik oraz numer linii, w której zostało rozpoczęte wysyłanie danych.

Można poprawić kod z błędem lub zastosować jedno z rozwiązań podanych poniżej.

Usuwanie błąd - edycja kodu

Wystarczy poprawić kod strony tak żeby żadne nagłówki nie były wysyłane po rozpoczęciu wysyłania danych. Rozwiązanie czasem może być trudne do wykonania, więc można zastosować kolejne rozwiązanie.

1)Usuwanie błąd - buforowanie - edycja kodu

Dzięki buforowaniu mamy pewność, że żadne dane nie zostaną przesłane przed zakończeniem wykonywania skryptu. Buforowanie możemy wykonać na dwa sposoby: edytując skrypt oraz edytując ustawienia PHP.

Edycja kodu polega na skorzystaniu z funkcji `ob_start()` oraz `ob_end_flush()`, które są standardowo dostępne w PHP.

```
<?php
ob_start(); //rozpoczęcie buforowania
echo "Tutaj jest przykładowy tekst";
header('Location: http://www.nowy-adres-strony.pl/');
ob_end_flush(); //zakonczenie buforowania
?>
```

2)Usuwanie błąd - buforowanie - zmiana ustawień PHP

Edycja ustawień PHP polega na włączeniu (wartość On) zmiennej `output_buffering`. Dzięki temu nie będziemy musieli dodawać w pliku funkcji `ob_start()` oraz `ob_end_flush()`. Parser PHP będzie wykonywać buforowanie automatycznie. Dowiedz się u swojego hostingodawcy w jaki sposób możesz zmienić tę zmienną. Na serwerach WebHost.pl zmienna ta jest standardowo włączona.

3)Usuwanie błąd - kodowanie UTF-8

Jeżeli zapisujemy pliki w kodowaniu UTF-8, to trzeba pamiętać żeby na początku pliku nie występował znacznik BOM. Znacznik ten będzie powodował wyświetlanie błędu PHP. Edytory tekstu jak np. Notepad++ posiada możliwość zapisu bez tego znacznika.

```
<style type="text/css">
  body { background:#ddffdd;font-family:arial;font-size:12px;}
  table,td { font-family:arial;font-size:12px;}
  .tab { border:1px dotted #00ad00;width:90%;background:#cfffcf}
  h1 { text-align:center;color:green;text-decoration:underline;}
  h2 { color:green;}
  a,a:visited { color:green;}
  a:hover { background:#73ff73}
  hr { color:green;height:1px;}
  .motto { text-align:center;border:1px solid white; width:95%;margin-left:5%;
          background:#00cd00;color:white;}
</style>
```