**Zapis danych do pliku i odczyt danych z pliku**

Zapis/odczyt danych do/z pliku wykonuje się w kilku etapach:

1. Zapisywanie danych w pliku.
   1. Otwarcie pliku, jeżeli plik nie istnieje należy go utworzyć.
   2. Zapisanie danych w pliku.
   3. Zamknięcie pliku.
2. Odczytywanie danych z pliku.
   1. Otwarcie pliku. Należy rozpoznać czy plik istnieje oraz czy może zostać otwarty.
   2. Odczyt danych z pliku.
   3. Zamknięcie pliku.

Będziemy rozpatrywać **pliki tekstowe** w których zapisujemy i odczytujemy dane typu string.

Zestaw funkcji języka PHP do pracy z plikami znajdziesz tu: <http://www.w3schools.com/php/php_ref_filesystem.asp>

***Zapis danych***

*Otwieranie pliku*

Do otwierania pliku służy funkcja: fopen(nazwa pliku, string tryb). Drugi parametr określa tryb otwarcia pliku. Tryb trzeba dobrać odpowiednio do tego, co się chce z plikiem robić. Możliwe tryby to:

* „r” - plik tylko do odczytu; wewnętrzny wskaźnik pliku umieszczany jest na początku pliku tzn., że odczyt rozpocznie się od początku pliku.
* „r+” - plik do odczytu i zapisu; wewnętrzny wskaźnik pliku umieszczany jest na początku pliku .
* „w” - plik tylko do zapisu; wewnętrzny wskaźnik pliku umieszczany jest na końcu pliku; zawartość pliku jest niszczona (długość pliku jest zmieniana na zero); jeśli plik nie istnieje PHP próbuje go stworzyć.
* „w+” - plik do odczytu i do zapisu; wewnętrzny wskaźnik pliku umieszczany jest na końcu pliku; zawartość pliku jest niszczona (długość pliku jest zmieniana na zero); jeśli plik nie istnieje PHP próbuje go stworzyć.
* „a” - plik tylko do zapisu (dopisywanie na końcu); wewnętrzny wskaźnik pliku umieszczany jest na końcu pliku; jeśli plik nie istnieje PHP próbuje go stworzyć.

Funkcja fopen zwraca false, jeśli pliku nie udało się otworzyć, jeśli plik został otwarty – wskaźnik do pliku. Wskaźnik ten jest wykorzystywany w innych operacjach na pliku.

*Przykład* Plik został otwarty do zapisu. Jeśli taki plik nie istniał, zostanie utworzony.

$fp = fopen("dane.txt", "w");

*Zapisywanie danych w pliku*

Będziemy wykonywać za pomocą funkcji fwrite($wskaznik\_pliku,$string\_do\_zapisania).

*Zamykanie pliku*

Po wykonaniu jakichkolwiek operacji na pliku należy go obowiązkowo zamknąć. Wykonuje się to za pomocą funkcji fclose() zwracającej true w przypadku powodzenia operacji lub false po niepowodzeniu.

*Blokowanie pliku*

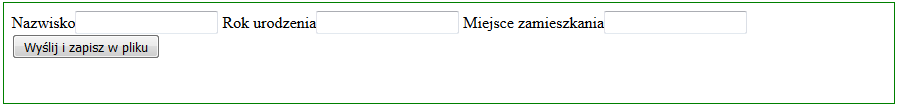
Co będzie, jeżeli kilku użytkowników zechce zapisać dane w tym samym pliku w tym samym czasie? Na pewno nic dobrego z tego nie wyniknie. Istnieje prosty sposób zablokowania pliku dla innych użytkowników w czasie kiedy zapisujemy/odczytujemy dane. Po otwarciu pliku ale przed zapisem/odczytem danych należy zastosować funkcję flock(). Argumentami tej funkcji są wskaźnik do pliku (pokażemy to w ćwiczeniu) oraz parametr określający rodzaj działania i przyjmujący następujące wartości:

* LOCK\_SH lub 1 - blokowanie odczytu. Pozwala na dzielenie pliku z innymi czytającymi.
* LOCK\_EX lub 2 - blokowanie zapisu. Plik wyłączony jest z użytku, nie może być współdzielony z innymi użytkownikami.
* LOCK\_UN lub 3 - zwolnienie istniejącej blokady.

Funkcja flock() zwróci true, jeżeli blokada została prawidłowo założona, w przeciwnym wypadku, zwróci false.

***Ćwiczenie 1*** Wykonaj formularz, w którym użytkownik wpisze swoje nazwisko, rok urodzenia oraz miejsce zamieszkania, skrypt odbierający te dane zapisze je w pliku dane\_osobowe.txt, znajdującym się w tym samym katalogu co plik z formularzem. Operacja powinna zakończyć się informacją o sukcesie lub niepowodzeniu.

1. wykonaj formularz jak poniżej:



1. odbierz dane z formularza i przygotuj do zapisu:

$nazwisko=$\_POST['nazwisko'];

$rok\_urodzenia=$\_POST['rok\_urodzenia'];

$miejsce\_zamieszkania=$\_POST['miejsce\_zamieszkania'];

$tekst=$nazwisko.' '.$rok\_urodzenia.' '.$miejsce\_zamieszkania;

1. dokonaj zapisu danych z blokowaniem pliku

@$plik=fopen('dane\_osobowe.txt','w');//otwarcie pliku

if (!$plik)//jeśli błąd podczas otwarcia

{

echo 'Wystąpił błąd podczas otwierania pliku!';

exit;

}

if (!flock($plik, LOCK\_EX))//jeśli błąd podczas blokowania

{

echo 'Wystąpił błąd podczas zakładania blokady pliku!';

fclose($plik);

exit;

}

fwrite($plik,$tekst);//zapis do pliku

flock($plik, LOCK\_UN);//odblokowanie pliku

fclose($plik);//zamknięcie pliku

echo 'Operacja zapisywania danych zakończona sukcesem!';

***UWAGA!*** Przejście do nowego wiersza w pliku to zapis dwóch znaków: **fwrite($plik,”\r\n”)** – koniecznie w cudzysłowie!

***Zadanie 1*** Wykonaj skrypt, który zapisze do pliku o podanej nazwie n liczb losowych z przedziału <p,q>. Każdą liczbę umieść w oddzielnym wierszu. Nazwa pliku, n, p i q jest pobierana z formularza.

***Odczyt danych***

*Odczytywanie całego pliku za pomocą funkcji fread()*

***fread(plik, ile)*** odczytuje tyle bajtów danych z pliku, ile określa drugi parametr funkcji. Jeśli chcemy odczytać cały plik wystarczy jako drugi parametr wpisać *filesize(‘nazwa\_pliku’)*.

***Ćwiczenie 2*** Wykonaj odczyt danych z pliku dane\_osobowe.txt z ćw.1.

1. najpierw sprawdzamy czy plik istnieje, czy można go otworzyć i czy udało się go zablokować

if (!file\_exists('dane\_osobowe.txt'))

{

echo 'Nie znaleziono pliku!';

exit;

}

@$plik=fopen('dane\_osobowe.txt','r');

if (!$plik)

{

echo 'Błąd podczas otwierania pliku!';

exit;

}

if (!flock($plik, LOCK\_SH))

{

fclose($plik);

echo 'Błąd podczas blokowania pliku!';

exit;

}

1. odczytujemy cały plik do jednej zmiennej za „jednym zamachem” i pokazujemy na stronie

$tekst=fread($plik,filesize('dane\_osobowe.txt'));

flock($plik, LOCK\_UN);

fclose($plik);

echo $tekst;

*Odczyt całego pliku z wynikiem w postaci tablicy, za pomocą funkcji file()*

***file()*** - funkcja ta jest bardzo prosta w zastosowaniu i bardzo użyteczna. Sama otwiera plik, odczytuje jego zawartość i zamyka po zakończeniu odczytu. Również wynik który jest tablicą wierszy, jest bardzo wygodny do dalszego przetwarzania.

***Ćwiczenie 3*** Wykonaj odczyt danych z pliku dane\_osobowe.txt z ćw.1 za pomocą funkcji file().

if (!file\_exists('dane\_osobowe.txt'))

{

echo 'Nie znaleziono pliku!';

exit;

}

$tablica\_wierszy=file('dane\_osobowe.txt');

if (count($tablica\_wierszy)==0)

{

echo 'Plik jest pusty!';

exit;

}

foreach($tablica\_wierszy as $wiersz)

echo $wiersz.'<br />';

*Odczyt pliku znak po znaku, za pomocą funkcji fgetc()*

***Ćwiczenie 4*** Wykonaj odczyt danych z pliku dane\_osobowe.txt z ćw.1 za pomocą funkcji fgetc(). Znaki odczytane z pliku wypisz na stronie oddzielając przecinkami. Uwaga! Brak polskich liter.

if (!file\_exists('dane\_osobowe.txt'))

{

echo 'Nie znaleziono pliku!';

exit;

}

@$plik=fopen('dane\_osobowe.txt','r');

if (!$plik)

{

echo 'Błąd podczas otwierania pliku!';

exit;

}

//pominięto blokowanie pliku

while (!feof($plik))//czy osiągnięto koniec pliku?

{

$znak=fgetc($plik);

echo $znak.',';

}

fclose($plik);

*Sprawdzanie loginu i hasła – wejście na stronę (wersja 1)*

***Ćwiczenie 5***

1. Załóż nowy folder o nazwie ***logowanie*** i umieść w nim plik ***pass.txt*** zawierający login i hasło. Login i hasło zapisz w osobnych wierszach. Np. *admin* oraz *admin123*.
2. Wykonaj stronę o nazwie ***zalogowany.html*** i umieść ją w folderze ***logowanie***. Niech na stronie będzie nagłówek h1 o treści: „Witaj! Jesteś zalogowany!”. Strona otworzy się gdy podane w formularzu login i hasło będą zgodne z tymi w pliku ***pass.txt***.
3. Wykonaj stronę ***formularz.php*** zawierającą formularz do wpisania loginu i hasła oraz skrypt.
4. Wstaw kod formularza do pliku ***formularz.html***

<form action="formularz.php" method="post">

Login<input type="text" name="login" />

Hasło<input type="password" name="haslo" />

<input type="submit" name="logowanie" value="Zaloguj" />

</form>

1. Napisz skrypt, który pobierze dane z formularza…

if (!isset($\_POST['logowanie'])) exit;

$login=$\_POST['login'];

$haslo=$\_POST['haslo'];

1. odczyta zawartość pliku pass.txt i wykona sprawdzenie

if (!file\_exists('pass.txt'))

{

echo 'Nie znaleziono pliku!'; exit;

}

$tablica = file('pass.txt');

if (count($tablica)!=2)

{

echo 'Błąd danych w pliku pass.txt!'; exit;

}

if ($login==trim($tablica[0]) && $haslo==trim($tablica[1]))

header('location: zalogowany.html');//przekierowanie do pliku

else echo 'Błędny login lub hasło!';

*Sprawdzanie loginu i hasła – wejście na stronę (wersja 2)*

***Ćwiczenie 6*** Różnica polega na tym aby w pliku pass.txt mieć zaszyfrowane login i hasło.

1. Napisz funkcję zaszyfruj(), która utworzy plik ***pass2.txt*** i zapisze w nim wersje zaszyfrowane loginu i hasła oraz wywołaj ją w pliku formularz.html

function zaszyfruj()

{

$plik=fopen('pass2.txt','w');

fwrite($plik,md5('admin')."\r\n");

fwrite($plik,md5('admin123'));

//md5()-przekształca dowolnie długi ciąg bajtów w liczbę 32 bajtową

fclose($plik);

}

zaszyfruj();//wywołanie funkcji

1. Zaszyfruj funkcją md5() login i hasło wczytane z formularza

$login=md5($\_POST['login']);

$haslo=md5($\_POST['haslo']);

1. Dalej postępuj jak w ćwiczeniu 5

***Przydatne funkcje do operacji na plikach i katalogach***

*Tworzenie nowego katalogu za pomocą funkcji mkdir()*

Funkcja ***mkdir($sciezka,$prawa\_dostepu)*** zwraca true w przypadku powodzenia i false w przypadku kiedy nie uda się utworzyć nowego katalogu. Drugi, opcjonalny argument jest liczbą ósemkową i domyślnie wynosi 0777. To jest katalog ogólnodostępny a więc najmniej bezpieczny dla przechowywanych w nim danych.

*Usuwanie katalogu za pomocą funkcji rmdir()*

Funkcja ***rmdir($sciezka)*** zwraca true, jeżeli operacja przebiegła pomyślnie lub false w przypadku niepowodzenia. Można usunąć tylko pusty katalog.

***Zadanie 2*** Napisz stronę zawierająca formularz z dwoma przyciskami: *Utwórz katalog* i *Usuń katalog*. Skrypt powinien tworzyć katalog o nazwie ***kat1*** w bieżącym folderze (o ile jeszcze nie istnieje) lub usuwać go (o ile istnieje). ***Wskazówka!*** Istnienie katalogów sprawdza się tak samo jak istnienie plików czyli za pomocą funkcji: ***file\_exists().***

*Kopiowanie pliku za pomocą funkcji copy()*

Funkcja ***copy($plik\_kopiowany, $plik\_docelowy)*** zwraca true, jeżeli operacja kopiowania przebiegła pomyślnie lub false w przypadku niepowodzenia. Jeżeli $plik\_docelowy istnieje, to zostanie nadpisany.

*Usuwanie pliku za pomocą funkcji unlink()*

Funkcja ***unlink($sciezka\_pliku)*** zwraca true, jeżeli operacja przebiegła pomyślnie lub false w przypadku niepowodzenia.

***Zadanie 3*** W nowym folderze umieść dowolny plik i nazwij go ***test.txt***. Napisz stronę zawierająca formularz z dwoma przyciskami: *Kopiuj test.txt do kopia\_test.txt* oraz *Skasuj kopia\_test.txt*. Skrypt powinien utworzyć kopię pliku test.txt (o ile taki plik istnieje) oraz kasować plik kopia\_test.txt (o ile taki plik istnieje).

***Zadanie 4*** (dla chętnych) Napisz skrypt, który wykona kopię pliku o podanej nazwie będącego w bieżącym folderze. Kopia ta będzie miała na początku komentarz wieloliniowy z danymi autora skryptu oraz do każdego wiersza będzie na początku dopisany jego numer.