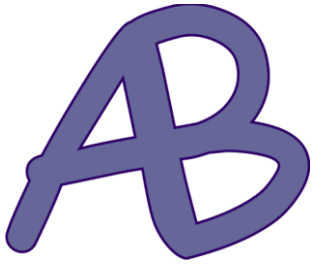


III. PHP - pliki

dr Artur Bartoszewski
UTH Radom

Pliki

Pliki

Otwieranie pliku:

```
$zmienna_plikowa = fopen("nazwa", "tryb");
```

- musimy pamiętać o ścieżce dostępu do naszego pliku,
- Jeżeli pracujemy z plikiem przechowywanym na naszym serwerze możemy otwierać go w dowolnym trybie,
- jeżeli tryb otwierania jest ustawiony na ”r” czyli tylko do odczytu, to za nazwę pliku może służyć adres URL do pliku.
- należy uważać, by nie stworzyć pliku o nazwie, która już istnieje – istniejący plik zostanie wtedy trwale usunięty w miejsce nowo stworzonego.

Zamykanie pliku (nie jest konieczne):

```
fclose($zmienna_plikowa);
```

Pliki – tryby otwarcia

r	(read) Otwiera tylko do odczytu; umieszcza wskaźnik pliku na jego początku. Plik musi istnieć, inaczej zostanie zwrócony błąd.
r+	(read and write) Otwiera do odczytu i zapisu; umieszcza wskaźnik pliku na jego początku. Plik musi istnieć.
w	(write) 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+	(read and write): Otwiera do odczytu i zapisu; umieszcza wskaźnik pliku na jego początku i obcina plik do zerowej długości. Jeśli plik istnieje, to jego zawartość zostanie skasowana. Jeśli plik nie istnieje, zostanie utworzony nowy.
a	(append) Otwiera tylko do zapisu; umieszcza wskaźnik pliku na jego końcu. Jeśli plik nie istnieje to próbuje go utworzyć.
a+	(read and append) Otwiera do odczytu i zapisu; umieszcza wskaźnik pliku na jego końcu. Jeśli plik nie istnieje to próbuje go utworzyć.
x	(exclusive create) 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.
x+	Tworzy i otwiera plik do 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.

Pliki

Ważne jest, aby uwzględniać obsługę błędów przy operacjach na plikach.
Możemy użyć funkcji takich jak file_exists() czy fopen() wraz z instrukcjami warunkowymi, aby obsługiwać sytuacje, w których operacje na plikach nie powiodą się.

Sprawdzenie czy plik istnieje:

```
if (file_exists("nazwa_pliku"))
{} // Tutaj możesz otworzyć plik
else
{}
```

```
if (!file_exists("nazwa_pliku"))
{
    return;
}

// Tu możesz otworzyć plik
```

Sprawdzenie czy plik został otwarty:

```
<?php
$plik = fopen("nazwa_pliku.txt", "r");

if ($plik) {
    echo "Plik został otwarty pomyślnie.";
    // Tu możesz wykonywać operacje na otwartym pliku
    fclose($plik);
} else {
    echo "Nie udało się otworzyć pliku.";
}
?>
```

Jeżeli fopen zwróci NULL. To oznacza, że nie udało się otworzyć pliku z podaną ścieżką lub zadanym trybem dostępu.

Pliki

Wskaźnik pliku określa, skąd w pliku mają być odczytywane dane lub gdzie mają być zapisywane. Przesuwa się on automatycznie dalej po każdej procedurze odczytania określonej porcji danych z pliku.

fseek() - pobiera trzy argumenty:

- uchwyt do pliku,
- przesunięcie,
- (opcjonalny) rodzaj przesunięcia.

Dostępne są trzy rodzaje przesunięcia:

- **SEEK_SET** – ustawia wskaźnik na pozycję określoną poprzez przesunięcie **(domyślne),**
- **SEEK_CUR** – ustawia wskaźnik na pozycję równą aktualnej i uwzględnia przesunięcie,
- **SEEK_END** – ustawia wskaźnik na koniec pliku i uwzględnia przesunięcie. Jeśli chcemy ustawić wskaźnik w konkretnej odległości (liczby znaków) od końca zbioru, przesunięcie musi być ujemne.

rewind(), pobiera jako argument uchwyt do pliku i przesuwa wskaźnik na jego początek.

Pliki – odczyt z pliku

Odczyt po kolei po jednym znaku - fgetc()

Zwraca jeden odczytany znak lub false, jeśli natrafimy na koniec zbioru.

```
while (false  !== ($char = fgetc($file))) {
    echo "Aktualny znak to $char <br />";
}
```

Ważne jest to, że aby sprawdzić, czy funkcja oddała wartość false, należy użyć operatora = = = lub ! = =, ponieważ przy użyciu = = lub != wartości „„ (pusty string), lub 0 dzięki zamianie typów zmiennych w PHP też będą interpretowane jako false.

Pliki – odczyt z pliku

Odczyt linijka po linijce fgets().
Jako parametry należy podać uchwyt do pliku i opcjonalnie maksymalną długość odczytanej linii.

```
$plik = fopen("plik.txt","r");
while(!feof($plik))
{
    $wiersz = fgets($plik);
}
```

Pliki – odczyt z pliku

Odczyt całego pliku (za jednym razem) - fread()
argumentami są:

- uchwyt do pliku
- liczba znaków, jaka ma być odczytana.

Często jako liczbę znaków podaje się całkowitą długość pliku, obliczoną funkcją filesize(nazwa_pliku).

```
$plik = fopen("plik.txt", "r");
$dane = fread($plik, filesize("plik.txt"));
```

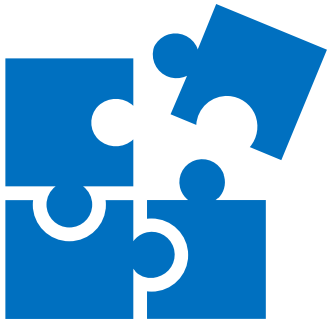
Lub krócej:

```
$dane = fread(fopen("plik.txt", "r"), filesize("plik.txt"));
```

file_get_contents() - funkcja zwróci całą zawartość pliku |(podobnie jak poprzednia)
jako argument podajemy nazwę pliku – tym razem nie jest to uchwyt, czyli
pliku nie trzeba otwierać, używając fopen()

```
$dane = file_get_contents("plik.txt");
```

Przykład do wykonania



Wypisanie na ekran zawartości pliku „dane.txt”

```php
<?php
$plik = @fopen("dane.txt", "r");
if (!($plik)) {
    echo "BŁĄD: Nie da się otworzyć pliku.";
} else
{
    while (!(feof($plik)))
    {
        $wiersz = fgets($plik);
        echo $wiersz , "<br>";
    }
    fclose($plik);
}
?>
```

Pliki – odczyt całego pliku do tablicy

Skrypt wczytuje tekst z pliku "dane.txt" do tablicy, przyjmując, że każdy wiersz pliku znajduje się w osobnej linii:

```php
<?php

// Ścieżka do pliku
$sciezka_do_pliku = 'dane.txt';

// Wczytaj tekst z pliku do tablicy
$tablica_tekstu = file($sciezka_do_pliku, FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);

// Wyświetl zawartość tablicy (do celów testowych)
print_r($tablica_tekstu);

?>
```

file() - funkcja czyta plik wiersz po wierszu i zapisuje każdy wiersz jako element tablicy.
Opcje **FILE_IGNORE_NEW_LINES** i **FILE_SKIP_EMPTY_LINES** pomagają pozbyć się znaków nowej linii na końcu każdego wiersza oraz pomijają puste linie.

Pliki – odczyt z pliku

Przeniesienie zawartości pliku do tablicy, gdzie każdy wpis odpowiada jednej linii z pliku - funkcja file()
W parametrze wystarczy podać nazwę pliku.

Plik nie musi być otwarty funkcją fopen().

```
$tablica = file("plik.txt");
    foreach($tablica as $wiersz)
    {
        echo "<p>$wiersz</>";
    }
```

Pliki – zapis do pliku

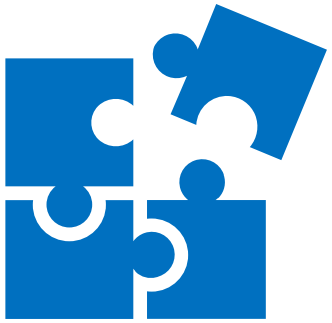
Do zapisu służy funkcja – fwrite().
Spotykany jest też alias do niej – fputs().

```php
// otwieramy plik w trybie umożliwiającym zapis na końcu pliku
$plik = fopen("plik.txt", "a");


// przypisanie zawartości do zmiennej
$zawartosc = "tekst do zapisania w pliku";
fwrite($plik, $zawartosc);
```

fwrite() i fputs() w PHP są praktycznie tożsame i używają się ich w identyczny sposób. Obie funkcje służą do zapisu danych do otwartego pliku

Przykład do wykonania



Program tworzy plik tekstowy z imieniem i nazwiskiem

```php
<?php
$plik = @fopen("dane/imienazwisko.txt", "w");
if (!($plik)) {
    echo „Błąd otwarcia pliku.";
} else {
    echo "Plik z imieniem i nazwiskiem został utworzony.";
    fwrite($plik, "Artur Bartoszewski");
    fclose($plik);
}
?>
```

Pliki – zapis do pliku

Dopisywanie tekstu na początku pliku.

Nie ma bezpośredniej możliwości zapisu na początku czy w środku pliku.

Można dopisywać tylko na końcu pliku lub zastąpić jego aktualną zawartość.

Jeśli więc chcemy coś zapisać na początku już istniejącego zbioru, należy najpierw otworzyć dany plik, odczytać jego zawartość i dopiero wtedy zapisać całość od nowa.

```php
$plik = fopen("plik.txt", "r");
$dane = fread($plik, filesize("plik.txt"));
fclose($plik);
$noweDane = "Coś do dodania na początku pliku";
$noweDane .= $dane;
$plik = fopen("plik.txt", "w");
fwrite($plik, $noweDane);
fclose($plik);
```

Pliki – blokowanie plików

Podczas używania plików w serwisie o dużej liczbie odwiedzin może się zdarzyć, że w tym samym czasie dwa procesy będą próbowały zapisywać coś do pliku. Może to być przyczyną różnych błędów.

Do zapobiegania takim sytuacji służą blokady plików. Istnieją dwa rodzaje blokad:

- **blokada dzielona** – może być założona przez wiele procesów naraz podczas odczytu – uniemożliwia zapis
- **blokada wyłączna** – zakładana podczas zapisu do pliku tylko przez jeden proces.

Aby założyć blokadę, należy użyć funkcji flock()

Pobiera ona dwa argumenty:

- uchwyt do pliku
- rodzaj blokady.

Dostępne rodzaje blokady:

LOCK_SH – zakłada blokadę dzieloną (do odczytu),

LOCK_EX – zakłada blokadę wyłączną (do zapisu),

LOCK_UN – zdejmuje blokadę z pliku.

Jeśli zakładanie blokady się powiedzie, to funkcja zwraca wartość true, w przeciwnym wypadku zwracana jest wartość false.

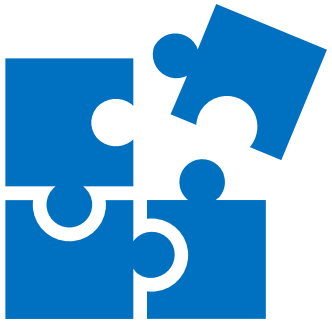
Pliki – blokowanie plików

Przykład blokowania plików:

```
$file = fopen("dane\plik.txt", "w+");
if (flock($file, LOCK_EX)) { // zakładanie blokady wyłącznej
    fwrite($file, "Wartość dopisana do pliku");
    flock($file, LOCK_UN); // zdjęcie blokady
} else {
    echo "Błąd - plik zablokowany";
}
fclose($file)
```

Przykład do wykonania

Licznik tekstowy



```php
<?php
// Program pokazuje działanie prostego licznika tekstowego.
if (!(file_exists("liczba.txt"))) { // nie ma pliku z licznikiem
    $plik = fopen("liczba.txt", "w+");
    fputs($plik, "0");                    // więc go tworzysz
    fclose($plik);
}

$plik = fopen("liczba.txt", "r+");
if (!($plik)) {
    echo "BŁĄD: Nie da się otworzyć pliku.";
} else {
    flock($plik, LOCK_EX);          $ile = fgets($plik);
    $ile++;
    echo "Licznik wskazuje $ile.";
    fseek($plik, 0);
    fputs($plik, "$ile");
    flock($plik, LOCK_UN);
    fclose($plik);
}
?>
```

funkcja fopen() otwiera plik o nazwie „liczba.txt" w trybie zapisu ("w"), co spowoduje utworzenie pustego pliku, jeśli taki nie istnieje. Następnie plik jest natychmiast zamykany za pomocą fclose().

Pamiętamy o zablokowaniu pliku na czas wykonania skryptu

Jeżeli plik istnieje otwieramy go za w trybie „r+” czyli do odczytu i zapisu

Pliki – informacje o plikach

W PHP istnieje też kilka funkcji, które zwracają informacje o pliku. Należą do nich:

- fileatime(nazwa_pliku) – zwraca datę i czas ostatniego odczytu pliku podane w formacie timestamp (uniksowym),
- filemtime(nazwa_pliku) – zwraca datę i czas ostatniej modyfikacji pliku podane w formacie timestamp,
- filegroup(nazwa_pliku) – zwraca liczbowy identyfikator grupy, do której należy właściciel pliku,
- fileowner(nazwa_pliku) – zwraca identyfikator właściciela pliku,
- fileperms(nazwa_pliku) – zwraca prawa dostępu do pliku,
- filesize(nazwa_pliku) – zwraca wielkość pliku w bajtach.

Pliki – informacje o plikach

Oprócz tego istnieje też kilka funkcji zwracających wartości true lub false. Są to:

- is_dir(nazwa_pliku) – informuje o tym, czy zbiór jest katalogiem,
- is_executable(nazwa_pliku) – informuje o tym, czy plik jest wykonywalny,
- is_file(nazwa_pliku) – informuje o tym, że plik istnieje i jest zwykłym plikiem,
- is_readable(nazwa_pliku) – informuje o tym, czy plik można odczytać,
- is_writeable(nazwa_pliku) – informuje o tym, czy plik można zapisywać,
- is_uploaded_file(nazwa_pliku) – informuje o tym, czy plik został wysłany z formularza.

Operacje na plikach

Do kopiowanie plików służy funkcja copy(). Zwraca ona wartość true, jeśli plik zostanie poprawnie skopiowany, lub false, jeżeli wystąpi błąd.

```
if (!copy("tymczasowy/plik.txt", "dane/plik.txt"))
   echo "Błąd podczas kopiowania";
```

Aby zmienić nazwę zbioru, należy użyć funkcji o nazwie rename(). Może ona też służyć do kopiowania plików i zwraca true lub false, tak samo jak w wypadku copy().

```
  if (!rename("/tmp/tymczasowy_plik.txt", "/home/user/login/docs/plik.txt"))
       echo "Błąd podczas kopiowania";
```

Do usuwanie plików służy funkcja o nazwie unlink(). Jako parametr wystarczy podać nazwę pliku.

```
unlink("plik.txt");
```

Operacje na katalogach

Aby utworzyć katalog, należy wywołać funkcję mkdir() i jako to:

- nazwa katalogu do utworzenia
- prawa dostępu (np. 0777).

Funkcja zwraca wartości true lub false w zależności od powodzenia operacji.

```
mkdir("nazwa",0777);
```

Do przeglądania zawartości katalogów służy mechanizm pseudoobiektowy. Przeglądanie zaczynamy funkcją dir(nazwa_folderu), która zwraca obiekt-uchwyt do katalogu. Kolejne pozycje z katalogu pobieramy za pomocą metody read(). Pracę z folderem kończymy metodą close().

```
$dir = dir("logs");
while ($entry = $dir->read()) {
    echo "Kolejna pozycja z tego folderu to $entry<br />";
}
$dir->close();
```

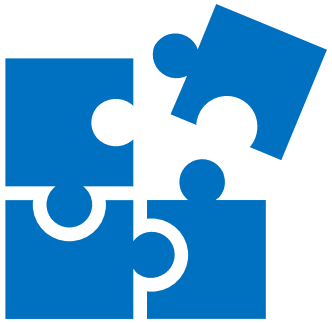
Operacje na katalogach

Bardziej skomplikowane jest usuwanie katalogów. Służy do tego funkcja rmdir(),
Jej argumentem jest nazwa folderu przeznaczonego do usunięcia.
Warunkiem koniecznym jest to, aby usuwany katalog był pusty.

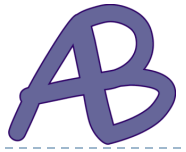
Przykład funkcji rekurencyjnej do usuwania całego katalogu (wraz z plikami i podkatalogami):.

```
function deltree($dirn)
{
    $dir = @dir($dirn);
    while ($entry = $dir->read()) {
        if (is_dir($dirn . '/' . $entry) && $entry != '.' && $entry != '..') {
            @deltree($dirn . '/' . $entry);
        } else if ($entry != '.' && $entry != '..') {
            @unlink($dirn . '/' . $entry);
        }
    }
    $dir->close();
    @rmdir($dirn);
}
```

Przykład do wykonania

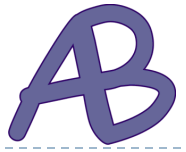


Program Pozwala na wybór towarów, a następnie na wpisanie danych użytkownika.
Po wpisaniu wyświetla informacje o wyborze i użytkowniku. Informacje o wybranych towarach są przechowywane w pliku tekstowym katalogu tymczasowym systemu (serwera , nie klienta)



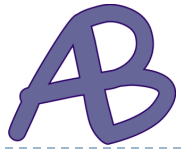
Przykład do wykonania

```php
<?php
  if (!(isset($_POST['nazwapliku']) || isset($_POST['zak1']) || isset($_POST['zak2']))) {
      // nie ma danych
      echo '<form w05p02.php" method="post"><div>';
      echo '<input type="checkbox" name="zak1" value="1" />Towar 1<br />';
      echo '<input type="checkbox" name="zak2" value="1" />Towar 2<br />';
      echo '<input type="submit" value="Wyślij" />';
      echo '</div></form>';
  } else …
```



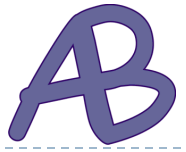
Przykład do wykonania

```
… } else {
    if (isset($_POST['zak1']) || isset($_POST['zak2'])) { // wybrano towar w poprzednim formularzu
        $tempDir = sys_get_temp_dir(); // Pobranie katalogu tymczasowego
        $tempFile = $tempDir . DIRECTORY_SEPARATOR . "plik_tymczasowy.txt";
        // Pełna ścieżka do pliku tymczasowego
        $plik = fopen($tempFile, "w");
        if (isset($_POST['zak1'])) $zak1 = $_POST['zak1'];
        else $zak1 = '';
        if (isset($_POST['zak2'])) $zak2 = $_POST['zak2'];
        else $zak2 = '';
        fputs($plik, $zak1 . "\n");
        fputs($plik, $zak2 . "\n");
        fclose($plik);
        echo '<form action="w05p02.php" method="post"><div>';
        echo "<input type=\"hidden\" name=\"nazwapliku\" ";
        echo "value=\"$tempFile\" />";
        echo 'Podaj imię i nazwisko:<br />';
        echo '<input type="text" name="imienazwisko" />';
        echo '<input type="submit" value="Wyślij" />';
        echo '</div></form>';
    } else { // masz już wszystkie dane
…
```



Przykład do wykonania

```
… } else { // masz już wszystkie dane
                $plik = fopen($_POST['nazwapliku'], "r");
        $zak1 = fgets($plik, 255);
        $zak2 = fgets($plik, 255);
        fclose($plik);
        if ($zak1 > 0) {
            echo 'Wybrano towar 1<br />';
        }
        if ($zak2 > 0) {
            echo 'Wybrano towar 2<br />';
        }
        echo "<br />Zamawiający: " . $_POST['imienazwisko'] . "<br />";
        unlink($_POST['nazwapliku']);
    }
}
?>
```



Literatura

W prezentacji użyto przykładów z książki:
- Żygłowicz Jerzy - PHP - Kompendium wiedzy, Helion

- https://www.php.net