

Operacje na plikach

Odczyt pliku tekstowego

Do odczytywania danych z plików w języku php jest dostępnych dziesięć funkcji. W zależności od sytuacji, niektóre z nich mogą być wygodniejsze od innych.

1. Podział funkcji odczytu ze względu na ilość odczytywanych danych:
 - a. Funkcje, które służą do przeczytania całego pliku pojedynczym wywołaniem:

```
readfile()
file()
file_get_contents()
parse_ini_file()
```

- b. Funkcje, które odczytują jedynie fragment:

```
fread()
fgets()
fgetc()
fgetss()
fscanf()
fgetcsv()
```

2. Podział funkcji ze względu na uwzględniane wykonywane konwersje.
 - a. Funkcje, które zwracają bajty odczytane z pliku bez wykonywania jakichkolwiek przekształceń:

```
readfile()
file_get_contents()
fread()
fgets()
fgetc()
```

- b. Funkcje , które przekształcają dane pochodzące z pliku i zwracają je w przetworzonej postaci:

```
file()
parse_ini_file()
fgetss()
fscanf()
fgetcsv()
```

Przetwarzanie zawartości pliku na tablicę – funkcja file()

```
file(nazwa_pliku)
```

Funkcja **file()** odczytuje podany plik, po czym w odczytanym tekście oddziela wiersze. Wynikiem działania funkcji jest tablica, której elementami są kolejne wiersze pliku.

Przykład 2

Stwórz plik.txt z zawartością:

```
Witaj Świecie
Piękny dzisiaj dzień
A ja uczę się bardzo przydatnych rzeczy :)|
```

A w pliku php zastosuj polecenie:

```
print_r(file("plik.txt"));
```

Zadanie

1. Zmodyfikuj plik wcześniejszy tak aby wyświetlić zawartość pliku plik.txt za pomocą instrukcji **foreach**

Przykład 3 – sprawdzanie czy dany wyraz istnieje w pliku

```
<?php
//Za pomocą funkcji file wczytujemy zawartość pliku do tablicy, gdzie każda linia to nowy index
$file = file('plik.txt');
//Deklarujemy zmienną z szukany wyrazem
$szukaj = 'Witaj';
//Wykonujemy pętlę tyle razy, ile jest elementów w tablicy (czyli tyle ile jest linii w pliku)
for($i = 0; $i <= count($file); $i++)
{
    //Sprawdzamy, czy w obecnie sprawdzanej linii jest szukany wyraz
    if(ereg($szukaj,$file[$i]))
    {
        //Jeśli szukany wyraz istnieje wyświetlamy tę linię wraz z znakiem przejścia do nowej linii
        echo $file[$i]."<br />\n";
    }
}
?>
```

Zadanie

2. Do sprawdzenia wystąpień wyrazu użyliśmy funkcji **ereg()**. Można było również wykorzystać inne funkcje np.: **strstr()**,**strpos()**,**stristr()**,**strchr()**. Poczytaj o nich na stronie:

http://www.w3schools.com/php/php_ref_string.asp

i spróbuj je zastosować.

3. Napisz skrypt do wyświetlania na stronie losowych cytatów z pliku plik.txt

```
rand (dolny_zakres, górny_zakres);  
echo $cytat[rand(0,sizeof($cytat)-1)];
```

4. Sprawdź działanie funkcji **readfile()**. Funkcja zwraca również liczbę. Co to jest za liczba?
5. Jaką widzisz różnicę między funkcją **file()** a **readfile()**?

Otwarcie pliku – funkcja fopen()

```
fopen (nazwa_pliku, tryb otwierania);
```

To co zostanie zwrócone przez funkcję przypisuje się do zmiennej, co pozwala na operowanie zawartością pliku.

Zestawienie trybów otwarcia

'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).

Zamknięcie pliku po zakończeniu pracy - funkcja fclose()

```
fclose ($zmienna);
```

Przykład

Odczytaj fragment pliku plik.txt za pomocą funkcji **fread()**

```
echo "<br><br>";
$file = fopen("plik.txt", "r");
echo fread($file, "10");
fclose($file);
```

Zapisywanie danych w pliku – funkcje fwrite(), fputs()

```
fwrite ($zmienna, wprowadzane dane);
fputs ($zmienna, wprowadzane dane);
```

Przykład – wpisanie tekstu do pliku tekstowego

1. Wpisywanie tekstu do pliku tekstowego, ustawiamy dla funkcji **fopen** tryb otwarcia w, w sytuacji gdy nie będzie istniał plik test.txt zostanie utworzony.

```
<?php
$file = fopen("test.txt", "w");
echo fwrite($file, "Hello World!");
fclose($file);
?>
```

2. Wprowadzimy tekst do pliku zadanie.txt za pomocą funkcji **fputs()**

```
$plik=fopen("zadanie.txt", "w+");
Fputs($plik, "<pre> Dobry programista to czlowiek,\n
Ktory spojrzy w obie strony \n
Zanim przejdzie przez jednokierunkowa ulice.</pre>");
fclose($plik);
```

Sprawdź zawartość pliku zadanie.txt. W następnym kroku wykorzystamy pętlę **while** w celu wyświetlenia zawartości pliku na stronie.

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

W wypadku gdy chcemy odczytać zawartość pliku, należy go otworzyć, następnie trzeba zastosować pętlę **while**, zawierając warunek sprawdzający, czy został osiągnięty koniec pliku (**eof**). Wewnątrz pętli należy umieścić funkcję **fgets()**. Jej zadaniem jest odczytanie zawartości pliku wiersz po wierszu. Przyjmuje ona dwa parametry: zmienna wskazującą na plik oraz wartość określającą liczbę bajtów pliku do przeczytania.

```
while(!feof($zmienna)) {
    fgets($zmienna, wartość);
}
```

Przykład - Czyszczenie w pliku z pustych wierszy

Zmień zawartość pliku plik.txt:

Witaj Świecie

Piękny dzisiaj dzień

Lubie poniedziałki

jeszcze jedno zdanie

```
<?php
$plik=file('plik.txt') ;
for($i = 0; $i <= count($plik); $i++)
{ if(trim($plik[$i]))
{
    $dane[]=$plik[$i];
}
}
$plikotworz=fopen('plik_nowy.txt','w');
for ($i=0;$i<count($dane);$i++)
{
    fputs($plikotworz, $dane[$i]);
}
fclose($plikotworz);
?>
```

Wewnątrz pętli tworzymy instrukcję warunkową, która sprawdza, czy linia jest pusta. Funkcja **trim()** wycina wszystkie spacje przed i za tekstem oraz usuwa znaki sterujące. Jeśli linia nie jest pusta wówczas wpisuje ją do nowej tablicy **\$dane[]**. Otwieramy nasz plik do zapisu za pomocą funkcji **fopen()** i pierwszym parametrem jest nowy plik. Funkcja **fputs()** umieszcza w pliku plik_nowy.txt treść z pliku plik.txt. po zakończeniu pętli należy zamknąć plik za pomocą polecenia **fclose()**.

Inne funkcje

Poniżej znajdziesz przykłady zastosowania funkcji:

file_exists() - funkcja o wartości logicznej - sprawdza, czy podany plik istnieje na dysku

filesize() - funkcja sprawdza wielkość pliku

filemtime() zwraca czas ostatniej modyfikacji pliku

Zapoznaj się z innymi funkcjami dotyczącymi plików w php

http://www.w3schools.com/php/php_ref_filesystem.asp

Przykład – wielkość pliku

filesize() – sprawdza wielkość pliku, który znajduje się na serwerze na którym znajduje się nasza strona

```
<?php
    echo filesize('plik.txt');
?>
```

Sprawdzanie wielkości pliku na innym serwerze

Aby sprawdzić wielkość pliku na innym niż nasz serwer musimy użyć **file_get_contents()**, która odczytuje plik jako ciąg znaków i stosując funkcję **strlen()** otrzymamy rozmiar pliku.

```
$z=file_get_contents('http://asg.g4.net.pl/bingo.pdf');
echo strlen($z);
```

Przykład - sprawdzanie czy plik istnieje

file_exists(plik);

Przykład

```
if (file_exists("plik.txt"))
echo "plik istnieje";
else "nie ma takiego pliku" ;
```

Przykład – data ostatniej modyfikacji

filemtime()

```
$data=filemtime('plik.txt');
$dataformat=date('Y-m-d H:i', $data);
echo $dataformat;
```

Zadanie

Napisz skrypt który wpisze do pliku liczby parzyste i podzielne przez 3 z przedziału od 1 do 1000, a następnie przeczyta zawartość pliku i wyświetli go w oknie przeglądarki, w osobnych liniach.