

Język PHP umożliwia tworzenie plików tekstowych na dowolnej platformie, na której może zostać uruchomiony. Zawartość tych plików może być dowolna - w ten sposób można stworzyć na przykład pliki XML.

W celu automatycznego stworzenia pliku tekstowego na dysku oraz umieszczeniu w nim pewnej zawartości, należy skorzystać z funkcji `file_put_contents()`.

Uwaga: Pliki tekstowe powstające w wyniku działania poniższych skryptów PHP proszę otwierać innym edytorem, niż windowsowy Notatnik, gdyż nie interpretuje on poprawnie znaków nowej linii („\n”).

### Przykład 1

```
<?php
$dane = [];
$dane['imie'] = 'Jan';
$dane['nazwisko'] = 'Kowalski';
$dane['data_urodzenia'] = '1987-01-20';
$dane['miejsce_urodzenia'] = 'Sopot';
$dane['ulica'] = 'Newelska';
$dane['nr_domu'] = '6';
$dane['nr_mieszkania'] = '1';
$dane['kod_pocztowy'] = '00-123';
$dane['miasto'] = 'Warszawa';

$string = '';

foreach($dane as $klucz => $wartosc)
    $string .= "$klucz - $wartosc\n";

file_put_contents("przyklad1.txt", $string);
```

W tym przykładzie tworzona jest tablica asocjacyjna `$dane`, a następnie za pomocą pętli `foreach` wartości wszystkich elementów tablicy są wpisywane do zmiennej `$string`. Na końcu zawartość tej zmiennej jest zapisywana do pliku `przyklad1.txt`

### Zadanie 1

Proszę o zmodyfikowanie przykładu 1 w ten sposób, żeby był generowany poprawny plik XML.

### Przykład 2

```
<?php
$string = '';

$pdo = new PDO('mysql:host=localhost;dbname=test', 'root', '');
foreach($pdo->query('SELECT * FROM samochody ORDER BY marka, model') as
$wiersz) {
    $string .= "id - $wiersz[id]\n";
    $string .= "marka - $wiersz[marka]\n";
    $string .= "model - $wiersz[model]\n";
    $string .= "rok - $wiersz[rok]\n";
    $string .= "pojemnosc - $wiersz[pojemnosc]\n";
    $string .= "typ_silnika - $wiersz[typ_silnika]\n";
    $string .= "liczba_poduszek - $wiersz[liczba_poduszek]\n";
    $string .= "abs - $wiersz[abs]\n";
    $string .= "esp - $wiersz[esp]\n";
}

file_put_contents("przyklad2.txt", $string);
```

W tym przykładzie pobierane są dane z tabeli samochody, a następnie tworzony jest plik tekstowy przyklad2.txt, w którym są one umieszczane.

### Zadanie 2

Proszę zmodyfikować przykład 2 w ten sposób, żeby pobierane były również nazwy marek i modeli samochodów (złączenie tabeli samochody z tabelami marki i modele). Dodatkowo plik wynikowy powinien być poprawnym plikiem XML.

### Przykład 3

```
<?php
$uchwyt = fopen("przyklad3.csv", "r");

while (($data = fgetcsv($uchwyt, 1000, ";")) !== FALSE)
    echo $data[0]. " - ".$data[1]. " - ".$data[2]. "<br/>";

fclose($uchwyt);
```

W tym przykładzie odcytujemy, a następnie wyświetlamy w oknie przeglądarki, zawartość pliku CSV (gdzie separatorem pól jest średnik). Po otwarciu pliku do odczytu za pomocą funkcji `fopen()`, w pętli `while` wczytujemy kolejne wiersze pliku. Na danych pobranych z każdego wiersza, wywoływana jest funkcja `fgetcsv()`, która automatycznie tworzy tablicę, rozdzielając pola przy użyciu podanego separatora (w tym przypadku średnik).

### Zadanie 3

Proszę o uzupełnienie przykładu 3 o kod, pozwalający na stworzenie poprawnego pliku XML z danych odczytanych z pliku CSV.