

Biblioteka SimpleXML dla języka PHP pozwala na szybkie stworzenie obiektu z dokumentu XML, a następnie jego wygodne przetwarzanie.

### Przykład 1

```
<?php
```

```
$plik = file_get_contents("samochody.xml");  
$xml = new SimpleXMLElement($plik);  
  
foreach($xml as $samochod) {  
    echo $samochod->id."<br/>";  
    echo $samochod->marka."<br/>";  
    echo $samochod->model."<br/>";  
    echo $samochod->rok."<br/>";  
    echo $samochod->pojemnosc."<br/>";  
    echo $samochod->typ_silnika."<br/>";  
    echo $samochod->liczba_poduszek."<br/>";  
    echo $samochod->abs."<br/>";  
    echo $samochod->esp."<br/>";  
}
```

W tym przykładzie za pomocą funkcji `file_get_contents()` wczytywana jest do zmiennej `$plik` cała zawartość dokumentu XML `samochody.xml`. Ze zmiennej `$plik` budowany jest obiekt `SimpleXMLElement`, który reprezentuje cały dokument XML. Następnie za pomocą pętli `foreach` można odczytać zawartość każdego elementu potomnego tego dokumentu.

## Zadanie 1

Proszę zmodyfikować powyższy przykład w ten sposób, żeby dane wyświetlane były w postaci tabeli HTML. Co drugi wiersz tej tabeli powinien mieć kolor szary.

Komórki tabeli zawierające dane typu tak/nie powinny mieć, odpowiednio, kolor tła zielony albo czerwony.

## Przykład 2

```
<?php

$plik = file_get_contents("samochody.xml");
$xml = new SimpleXMLElement($plik);

$pdo = new PDO('mysql:host=localhost;dbname=test', 'root',
'');

foreach($xml as $samochod) {
    $stmt = $pdo->prepare("
        INSERT INTO samochody
        (id, marka, model, rok, pojemnosc, typ_silnika,
        liczba_poduszek, abs, esp)
        VALUES
        (:id, :marka, :model, :rok, :pojemnosc, :typ_silnika,
        :liczba_poduszek, :abs, :esp)
    ");
    $wynik = $stmt->execute([
        'id' => $samochod->id,
        'marka' => $samochod->marka,
        'model' => $samochod->model,
        'rok' => $samochod->rok,
        'pojemnosc' => $samochod->pojemnosc,
        'typ_silnika' => $samochod->typ_silnika,
        'liczba_poduszek' => $samochod->liczba_poduszek,
        'abs' => $samochod->abs,
        'esp' => $samochod->esp,
    ]);
}
```

W powyższym przykładzie, po wczytaniu zawartości pliku XML, w pętli **foreach**, tworzone są dynamicznie (przy pomocy *prepared statements*) zapytania typu INSERT (dodające rekord do bazy).

## Zadanie 2

Proszę zmodyfikować powyższy przykład w ten sposób, żeby skrypt sprawdzał w dokumentach marki.xml i modele.xml, jaka marka oraz model odpowiadają danym id, a następnie do bazy wstawiał ich nazwę.

*Uwaga:* Do powyższego zadania potrzebna będzie tabela samochodów, której kolumny marka oraz model są typu tekstowego (VARCHAR). Tabele taką można stworzyć za pomocą następującego zapytania:

```
CREATE TABLE samochody2 (  
    id int(11) NOT NULL auto_increment,  
    marka varchar(100) NOT NULL,  
    model varchar(100) NOT NULL,  
    rok int(11) NOT NULL,  
    pojemnosc float NOT NULL,  
    typ_silnika enum('benzyna','diesel') NOT NULL,  
    liczba_poduszek int(11) NOT NULL,  
    abs enum('tak','nie') NOT NULL,  
    esp enum('tak','nie') NOT NULL,  
    PRIMARY KEY (id)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

### Przykład 3

```
<?php
```

```
$plik = file_get_contents("samochody.xml");  
$xml = new SimpleXMLElement($plik);  
$csv = '';
```

```
foreach($xml as $samochod) {  
    $csv .= $samochod->id.";" ;  
    $csv .= $samochod->marka.";" ;  
    $csv .= $samochod->model.";" ;  
    $csv .= $samochod->rok.";" ;  
    $csv .= $samochod->pojemnosc.";" ;  
    $csv .= $samochod->typ_silnika.";" ;  
    $csv .= $samochod->liczba_poduszek.";" ;  
    $csv .= $samochod->abs.";" ;  
    $csv .= $samochod->esp."\n" ;  
}
```

```
file_put_contents("przyklad3.csv", $csv);
```

W powyższym przykładzie, dane, po wczytaniu z pliku XML, są dołączane, w formacie CSV, do zmiennej \$csv. Na końcu, za pomocą funkcji **file\_put\_contents**, dane te wpisywane są do pliku tekstowego.