

Bazy danych 2023: lista zadań nr 6

27 luty 2023

Rozwiązaniami zadań z tej listy są skrypty w PHP wyświetlające żądane wyniki w postaci stron WWW (oraz, tam gdzie zadanie tego wymaga, formularze HTML umożliwiające wysłanie do skryptu parametrów jego wykonania). Jeśli zamierzasz pisać aplikację w innej technologii niż PHP, zadania z tej listy (i następnej) *możesz* rozwiązać również w niej, o ile mają one w niej sens. Estetyka rozwiązań (czyli np. style CSS) jest nieistotna (poza zad. 3), znaczenie ma struktura generowanych dokumentów (np. elementy HTML).

Rozwiązania najłatwiej tworzyć umieszczając skrypty w podkatalogu `public_html` w swoim katalogu domowym (tożsamym na stacjach roboczych i na serwerze `panoramx`). Efekt działania skryptów można będzie wtedy sprawdzić, wchodząc na stronę

```
http://panoramx.ift.uni.wroc.pl/~[nazwa_uzytk]/[nazwa_skryptu].php
```

Żeby serwer zinterpretował skrypt, **muszą zostać ustawione właściwe prawa dostępu do niego**, np. poleceniem

```
$ chmod a+r [nazwa_skryptu].php
```

To, jak zachowuje się serwer w wypadku błędów w skryptach, zależy od jego konfiguracji. `panoramx` nie wyświetla żadnych komunikatów o błędach, co na etapie tworzenia skryptów jest dosyć niefortunne. Aby zmienić to zachowanie, można (np. na początku skryptu) umieścić następujące instrukcje:

```
error_reporting(E_ALL);  
ini_set('display_errors', 1);
```

Nie zmieni to jednak nic w odniesieniu do błędów składniowych, których wystąpienie powoduje, że skrypt w ogóle nie jest interpretowany i zwracany jest pusty dokument. Wykryć taką sytuację, oraz zlokalizować błędy, można interpretując skrypt w linii poleceń:

```
$ php [nazwa_skryptu].php
```

Zadania można oczywiście rozwiązywać na dowolnym innym serwerze obsługującym PHP.

Zad. 1. Zadeklaruj zmienną `$samples` jako tablicę kilku napisów. Użyj funkcji `implode()`, aby wyświetlić listę numerowaną, w której treści kolejnych punktów będą stanowić kolejne elementy tablicy `$samples`. (Operatorem konkatencji w PHP jest kropka.)

Zad. 2. Napisz funkcję sortującą przez wstawianie tablicę podaną jako argument. Wyświetl wynik jej działania na przykładowej tablicy.

Zad. 3. Wygeneruj paletę kolorów RGB, w której każda z trzech składowych koloru przyjmuje jedną z sześciu wartości: 00, 33, 66, 99, cc, ff. Paleta może mieć postać np. tabeli $(6 \cdot 6) \times 6$ o zmiennym kolorze tła komórek, w której co kolumnę zmienia się składowa R, co wiersz – składowa G, co 6 wierszy – składowa B.

Zad. 4. Napisz funkcję przyjmującą dwa argumenty: liczbę dni miesiąca oraz numer dnia tygodnia, na który przypada pierwszy dzień tego miesiąca. Funkcja ma zwracać „kalendarz” takiego miesiąca, czyli tablicę, której elementami są tablice reprezentujące jego kolejne tygodnie: mają one zawierać dokładnie siedem napisów – z dziesiętnymi numerami dni lub pustych dla komórek przed rozpoczęciem lub po końcu miesiąca.

Zad. 5. Napisz funkcję, która przyjmuje jako argument tablicę tablic napisów (np. zwróconą przez funkcję z poprzedniego zadania) i wyświetla jej zawartość w postaci tabeli. Możesz sprawdzać, jaka jest (maksymalna) długość „wiersza” podanej tablicy lub założyć, że wynosi ona siedem. W tym drugim przypadku wyświetl także wiersz nagłówkowy z nazwami dni tygodnia. W rozwiązaniu użyj iteratora `foreach` oraz funkcji `implode()`. (Zamiast `foreach` możesz użyć funkcji `array_map()` i własnej funkcji do przetworzenia indywidualnych wierszy.)

Wskazówka (zwłaszcza) dla osób chcących umieścić wyświetlanie formularza i wyniku w jednym skrypcie: superglobalne zmienne `$_POST` i `$_GET` zawsze są zainicjalizowane na początku wykonania skryptu (tj. funkcja `isset()` będzie zwracać prawdę), niezależnie od tego, czy dane z formularza zostały wysłane. Żeby to ustalić, należy sprawdzić, czy te zmienne są *puste* (tj. użyć funkcji `empty()`).

Zad. 6. Stwórz formularz, który pozwoli użytkownikowi na podanie metodą GET liczby n , oraz skrypt, który pobiera jej wartość i wyświetla tabliczkę mnożenia od 0 do n : kwadratową tabelę, której komórka w wierszu i i kolumnie j zawiera liczbę ij .

Zad. 7. Stwórz formularz, który pozwoli użytkownikowi na podanie metodą POST imienia i nazwiska (w osobnych polach typu `text`) oraz wybranie zainteresowań, ulubionych przedmiotów i roku studiów. Listy zainteresowań i przedmiotów mają mieć po co najmniej pięć pozycji i umożliwiać wielokrotny wybór (za pomocą elementów `select multiple` albo rodzin pól typu `checkbox`), zaś lista lat studiów ma umożliwiać tylko wybór jednokrotny (za pomocą elementu `select` albo rodziny pól typu `radio`).

Stwórz skrypt, który pobiera wysłane w ten sposób dane i wyświetla je w postaci zdania typu „Student $[numer\ roku]$ roku $[imię, nazwisko]$ lubi zajęcia z $[wybrane\ zajęcia]$, a w wolnym czasie uprawia $[wybrane\ zainteresowania]$ ”. Odpowiednio obsłuż sytuację, w której użytkownik nie wybierze żadnych zainteresowań lub przedmiotów.

Zad. 8. Stwórz formularz z pewną liczbą pól typu `text`. Stwórz skrypt, który pobiera wysłane przy jego użyciu dane, i wyświetla łączną długość (funkcja `strlen()`) napisów wprowadzonych przez użytkownika w polach formularza. Kod powinien działać niezależnie od liczby pól formularza. Wskazówka: użyj iteratora `foreach` (dla zmiennej `$_GET` lub `$_POST`). Możesz też użyć funkcji `array_map()` (oraz `array_sum()`).