



Współpraca bazy danych ze stroną internetową, etap I

- [Wprowadzenie](#)
- [Przeczytaj](#)
- [Film samouczek](#)
- [Sprawdź się](#)
- [Dla nauczyciela](#)



Współpraca bazy danych ze stroną internetową, etap I

Źródło: 3844328, dostępny w internecie: pixabay.com, domena publiczna.

Minęły już czasy, gdy przeglądanie internetu polegało jedynie na odwiedzaniu statycznych stron-wizytówek, wyświetlających dla każdego internauty jednakową, niezmienną zawartość. W sieci pragniemy skorzystać z koszyka zamówionych produktów w wirtualnej księgarni, chcemy zobaczyć zdjęcia naszych znajomych po zalogowaniu się do serwisu społecznościowego albo wyświetlić spersonalizowaną listę subskrybowanych przez nas kanałów w usłudze transmitującej internetowe wideo.

Aby współczesna strona internetowa mogła świadczyć internautom takie spersonalizowane usługi, niewątpliwie musi mieć zapewnioną możliwość komunikowania się z bazą danych. Dowiedz się, jak tego dokonać w praktyce.

W skład serii wchodzi również e-materiały:

- [Współpraca bazy danych ze stroną internetową, etap II,](#)
- [Współpraca bazy danych ze stroną internetową, etap III,](#)
- [Współpraca bazy danych ze stroną internetową, etap IV.](#)

Twoje cele

- Skonstruujesz pierwszą stronę internetową skutecznie komunikującą się z bazą danych.

- Dobierzesz środki zapewniające poprawne kodowanie polskich znaków w stworzonej witrynie oraz w obsługującej ją bazie danych.
- Ocenisz różnice występujące pomiędzy językami programowania wykonywanymi po stronie klienta, a tymi realizowanymi zdalnie na serwerze.

Przeczytaj

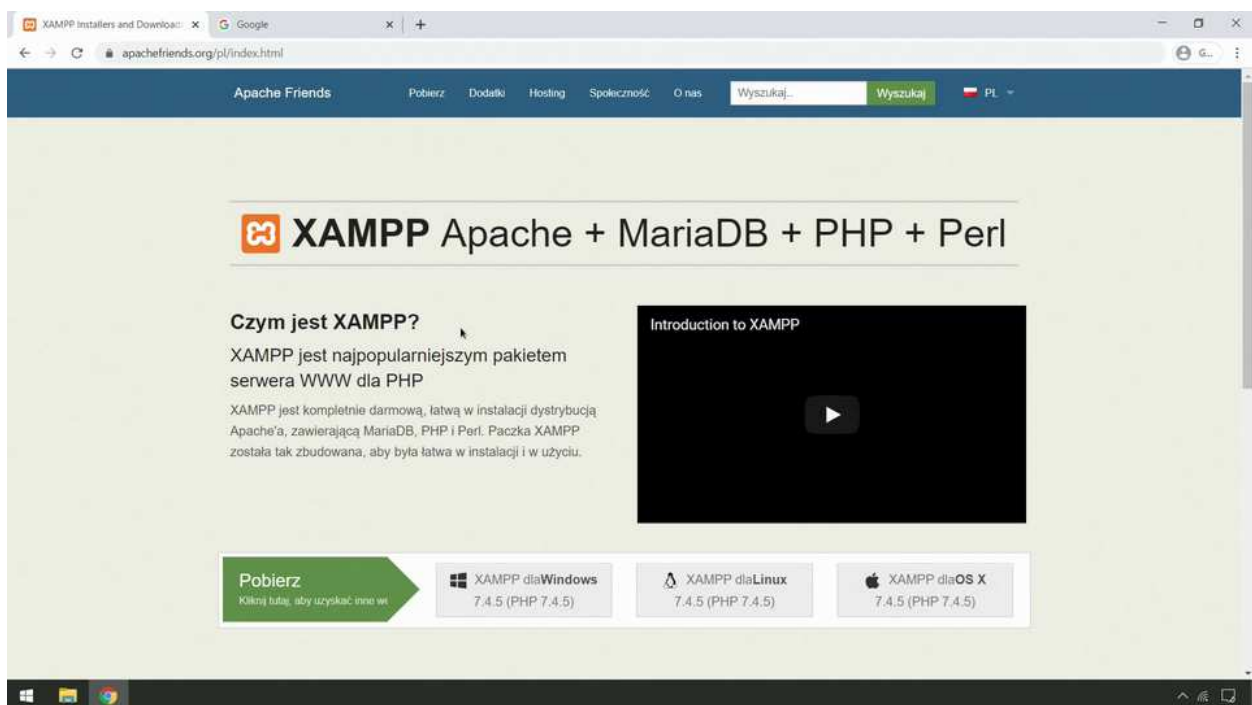
Instalacja pakietu XAMPP

Do zrealizowania współpracy **witryny internetowej** z bazą danych konieczny jest **dostęp do serwera HTTP** – najpopularniejsze obecnie oprogramowanie serwerowe używane przez firmy hostingowe to **Apache2** oraz **Nginx**.

Na potrzeby projektowania nowej strony lub nauki tworzenia aplikacji internetowych postępujemy nieco inaczej – instalujemy tzw. serwer lokalny, czyli taki, który **widoczny jest tylko dla naszej karty sieciowej** pod adresem: `http://127.0.0.1` lub jego aliasem słownym: `http://localhost`

Istnieje wiele gotowych pakietów, zapewniających możliwość łatwej instalacji zarówno serwera lokalnego, jak również interpretera PHP współpracującego z bazami danych MySQL – do najpopularniejszych rozwiązań tego typu należy pakiet **XAMPP** (wieloplatformowy) oraz **WAMPServer** (tylko na Windows). W naszych lekcjach używać będziemy pierwszego rozwiązania.

Proces instalacji, konfiguracji oraz instrukcję użycia pakietu pobranego z witryny fundacji Apache Friends przedstawiono w tutorialu:



Film dostępny pod adresem </preview/resource/Rb5vwfMkOfQae>

Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Film przedstawiający instalację oprogramowania XAMPP

Język komunikacji witryny z bazą danych

Pierwszym językiem programowania, z którym będziemy mieli styczność w kontekście zapewnienia współpracy strony internetowej z bazą danych, jest **PHP**. Jest to pełnoprawny, skryptowy język programowania stworzony przez **Rasmusa Lerdorfa**. Twórca PHP wykorzystał w jego składni mnóstwo elementów znanych już wcześniej w językach C/C++ oraz BASH (skrypty powłoki systemowej Unix/Linux).

Ciekawostka

To właśnie zapożyczenie z języka **BASH** sprawiło, iż w zmiennych tworzonych w PHP pojawia się charakterystyczny operator **\$** poprzedzający nazwę pojemnika na dane:

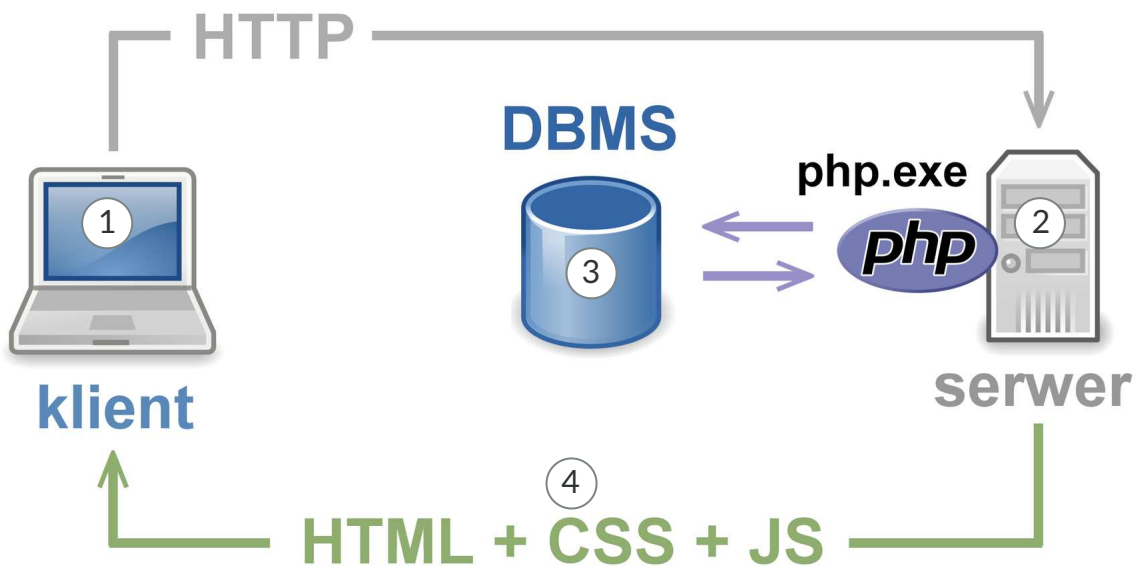
```
1 $imie = "Rasmus";  
2 $nazwisko = "Lerdorf";
```

Jeżeli podczas dotychczasowych zajęć w szkole poznałeś już konstrukcje językowe oraz składnię języków C/C++, to bardzo komfortowo i szybko odnajdziesz się także w PHP. Najważniejsza różnica pomiędzy pisaniem programów wykonywalnych i skryptów polega na tym, iż kod PHP wykonywany jest tylko przez procesor serwera, a nie lokalnego komputera.

Rasmus Lerdorf

Nie stworzyłem języka PHP, który znamy dzisiaj. To dziesiątki, jeśli nie setki ludzi opracowało PHP. Ja zwyczajnie byłem pierwszą osobą na świecie, która nad nim pracowała.

Wszystkie pliki PHP (w tym zawierające instrukcje komunikowania się z bazą danych i wykonywania zapytań SQL) **znajdują się tylko na serwerze**. Gdy się nad tym głębiej zastanowimy, jest to logiczne – skoro baza danych zawiera dane osobowe internautów (np. historię zamówień dokonanych w sklepie internetowym), to takie wrażliwe informacje muszą być przechowywane jedynie po stronie serwera. Wszystkie technologie webowe, które realizowane są zdalnie, określamy mianem **back-endu**.



1

Klient wpisuje adres witryny w przeglądarce, generując tzw. żądanie (ang. HTTP request).

2

Na dysku serwera znajduje się żądany przez klienta skrypt – kod źródłowy PHP znajdujący się w dokumencie zostaje przetworzony przez program php.exe znajdujący się również na dysku twardym serwera.

3

W razie potrzeby skrypt PHP komunikuje się z bazą danych, która również znajduje się na dysku serwera.

4

Po przetworzeniu całego kodu PHP wysłana zostaje odpowiedź (ang. HTTP response) jako pliki HTML, CSS i JS, które odpowiednio wyrenderuje przeglądarka na komputerze klienta.

Komunikacja w architekturze klient-serwer

Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Mianem **front-endu** określa się zaś technologie webowe, których kody źródłowe wykonywane są przez procesor na komputerze użytkownika witryny (czyli w praktyce przez jego przeglądarkę internetową). Pliki te muszą trafić do klienta przed ich wykonaniem, co w praktyce określimy jako proces wczytywania się witryny.

Rozszerzenia PHP

W celu połączenia witryny z bazą danych z użyciem PHP możemy skorzystać z trzech dostępnych w tym języku rozszerzeń (bibliotek) komunikacyjnych. Taką dedykowaną konkretnemu zadaniu bibliotekę, w nomenklaturze programistów określamy mianem [interfejsu API](#). Taki interfejs może być **proceduralny** albo **obiektowy**.

Jeżeli mamy interfejs **proceduralny**, to oznacza to, że operacje bazodanowe realizowane są przez odpowiednio przygotowane i wywołane funkcje. Każda funkcja wykonuje unikalne, dające się jednoznacznie wyróżnić działanie związane z bazą danych (np. ustanawia połączenie, wykonuje kwerendę).

Jeśli zaś interfejs API jest **obiektowy**, to oznacza to, że operacje bazodanowe realizowane są przez skonstruowane z użyciem klas obiekty oraz metody (czyli funkcje wewnątrz klas), wywoływane na rzecz tychże stworzonych obiektów.

Dostępne interfejsy API dedykowane komunikacji PHP z bazą danych MySQL:

- `mysql` – stara biblioteka istniejąca w PHP już od wersji 2.0, obecnie nie zaleca się jej stosowania. Od wersji PHP 5.5 to rozszerzenie jest zdeprecjonowane (jego użycie wygeneruje ostrzeżenie), zaś od wersji PHP 7.0 zostało całkowicie wycofane ze standardu języka. Jest to API proceduralne, brak interfejsu obiektowego;
- `mysqli` – nowoczesna i w pełni funkcjonalna biblioteka – litera **i** w nazwie pochodzi od słowa *improved*, co znaczy: *ulepszona, poprawiona*. API w pełni aktualne, współczesne, zalecane do użycia w nowych projektach. Oferuje do dyspozycji zarówno interfejs proceduralny, jak i obiektowy;
- PDO – akronim od PHP Data Objects – w pełni obiektowa biblioteka obsługująca komunikację z bazą danych. Największą zaletą PDO jest obsługa innych silników bazodanowych aniżeli MySQL.

Przyjrzyjmy się także **porównaniu tych rozszerzeń** w postaci tabelarycznej:

Rozszerzenie (API)	mysql	mysqli	PDO
Wprowadzono w wersji PHP	2.0	5.0	5.0
Wsparcie dla rozszerzenia	zdeprecjonowane od PHP 5.5, usunięte od PHP 7.0	aktywnie wspierane	aktywnie wspierane
Zalecane dla nowych projektów	nie	tak	tak
Interfejs proceduralny	tak	tak	nie

Rozszerzenie (API)	mysql	mysqli	PDO
Interfejs obiektowy	nie	tak	tak
Użycie różnych silników SQL	nie	nie	tak

Nawiązanie połączenia z bazą danych

W naszych pierwszych skryptach PHP użyjemy rozszerzenia **mysqli** w wersji **proceduralnej**. Połączenie z bazą danych nawiązujemy w następujący sposób:

```
1 $c = mysqli_connect("localhost", "root", "", "nazwa_bazy") or die
```

Wewnątrz funkcji użyto kolejno czterech parametrów:

- **host** (adres komputera przechowujący bazę), na serwerze lokalnym jest to adres `127.0.0.1` albo `localhost`;
- **login użytkownika MySQL** (musi posiadać stosowne uprawnienia), na serwerze lokalnym jest to użytkownik o nazwie `root`;
- **hasło dostępowe użytkownika MySQL** (na serwerze lokalnym jest one puste);
- **nazwa bazy danych**, z którą chcemy nawiązać połączenie.

Ponieważ nawiązanie połączenia może się zwyczajnie nie udać (choćby dlatego, że podany użytkownik lub jego hasło jest niepoprawne lub baza danych nie istnieje), to pojawił się na końcu funkcji dodatkowy zapis `or die()`. Tłumaczymy taki zapis wprost jako: *lub zabij skrypt*. W razie wykrycia błędu połączenia dalsze wykonanie skryptu PHP zostanie wstrzymane, zaś w witrynie pojawi się komunikat umieszczony w nawiasie.

Ważne!

Połączenie z bazą danych musiało zostać nazwane – u nas jest to zmienna PHP o nazwie `$c` – od słowa: connection. Identyfikator połączenia jest niezbędny, ponieważ w skrypcie może zostać zdefiniowane więcej niż jedno połączenie!

Poprawna obsługa polskich znaków

Już po nawiązaniu połączenia, włączamy obsługę standardu UTF-8 dla przesyłanych danych, używając poniższej funkcji:

```
1 mysqli_set_charset($c, "utf8");
```


Argumentami tej funkcji są kolejno:

- **identyfikator połączenia**, które musi zostać powyżej tej linii poprawnie nawiązane i pozostawać aktualnie aktywne (niezamknięte),
- **nazwa zestawu znaków**, współcześnie używamy standardu UTF-8.

Ważne!

Nazwę standardu wewnątrz funkcji zapisujemy bez myślnika: `utf8`, inaczej niż to ma miejsce w języku HTML, gdzie zazwyczaj stosujemy zapis: `utf-8`.

Dodatkowo, aby zapewnić w całej witrynie poprawne kodowanie polskich znaków (w tym również tekstów wyjmowanych z bazy bądź wkładanych do niej) należy:

- użyć w kodzie źródłowym witryny, wewnątrz znacznika `<html>` atrybutu:
`<html lang="pl">;`
- użyć w kodzie źródłowym witryny, w sekcji `<head>`, ponad tytułem `<title>` znacznika `<meta charset="utf-8">;`
- ustawić kodowanie pliku z rozszerzeniem `.php` na standard UTF-8. Na przykład w edytorze Notepad++ zmiany sposobu kodowania dokumentu dokonujemy wybierając z menu głównego: **Format (Encoding) | Koduj w UTF-8**;
- upewnić się, iż wszystkie pola typu tekstowego (znajdujące się w strukturze tabeli) bazy danych posiadają ustawioną metodę porównywania napisów `utf8_bin` albo `utf8_polish_ci`.

Wykonanie zapytania SQL

Kolejnym krokiem jest zdefiniowanie treści zapytania SQL, które mamy zamiar wykonać na naszej bazie danych. Treść kwerendy zostaje umieszczona w pomocniczej zmiennej, tutaj noszącej nazwę `$q` nawiązującej do angielskiego słowa query. Jednakże rzeczywiste **wykonanie kwerendy** odbywa się dopiero w momencie wywołania funkcji `mysqli_query()`:

```
1 $q = "UPDATE nazwa_tabeli SET nazwa_kolumny='nowa wartość'";
2 mysqli_query($c, $q);
```

Argumentami tej funkcji są kolejno:

- identyfikator połączenia, które musi zostać powyżej tej linii poprawnie nawiązane oraz pozostawać aktualnie aktywne (niezamknięte);
- treść zapytania, tutaj przechowywana w pomocniczej zmiennej `$q`.

Ważne!

Cała treść zapytania została zapisana w cudzysłowie, zaś nowa wartość będąca łańcuchem ograniczona jest apostrofami – użycie cudzysłowu spowodowałoby przedwczesne zakończenie treści całego zapytania i błąd składni!

Zamknięcie aktywnego połączenia

Po zakończeniu wszystkich działań dokonanych w skrypcie na bazie danych, wskazane jest zakończenie połączenia poniższą funkcją – jej jedynym parametrem jest identyfikator zamykanego połączenia:

```
1 mysqli_close($c);
```

Słownik

front-end

ogół technologii webowych (w tym HTML, CSS oraz w wielu zastosowaniach JavaScript), których pliki źródłowe wykonywane są przez przeglądarkę internetową klienta witryny; kody źródłowe zrealizowane w tych technologiach są jawne – każdy użytkownik serwisu może do nich zajrzeć, niezależnie od posiadanych uprawnień

back-end

ogół technologii webowych (w tym PHP, SQL), których pliki źródłowe wykonywane są przez procesor serwera; kody źródłowe zrealizowane w tych technologiach są dla klienta witryny niejawne, może do nich zajrzeć jedynie właściciel serwisu (albo ewentualnie uprawniona przez niego osoba posiadająca dostęp do dysku twardego serwera)

API

zestaw klas, metod, funkcji, zmiennych, parametrów, których aplikacja używa w celu zrealizowania zaplanowanych przez programistę zadań

Film samouczek

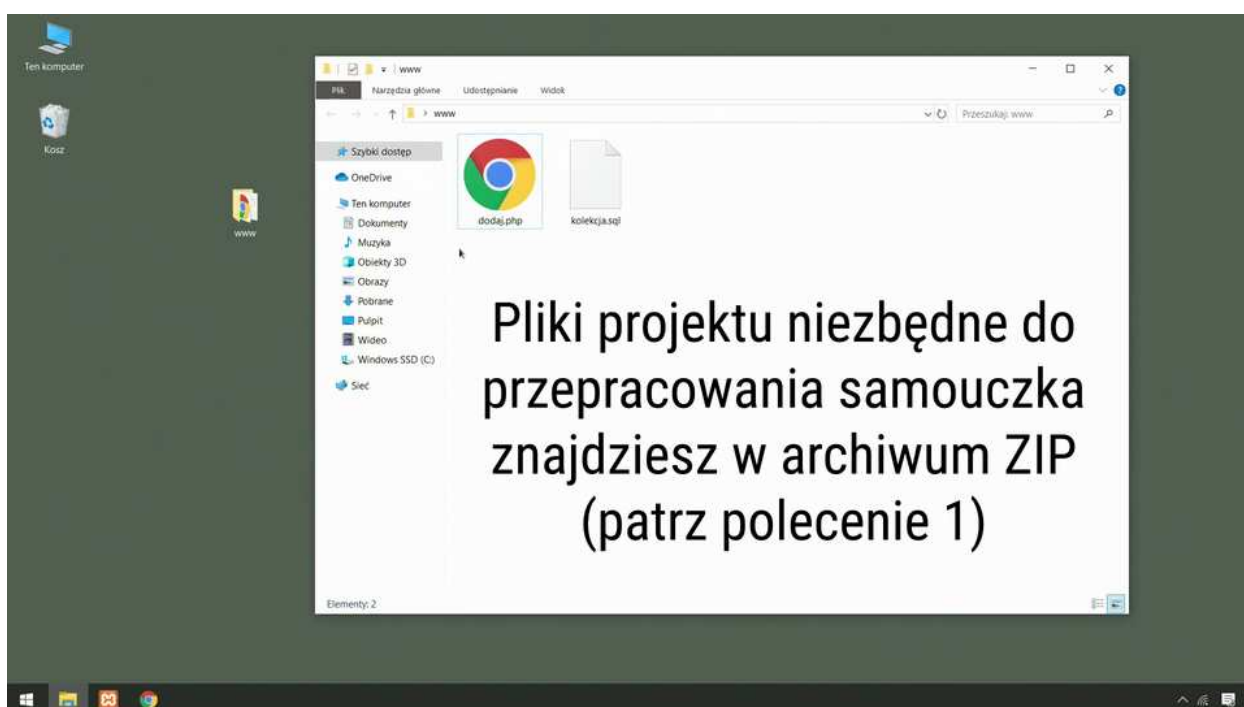
Polecenie 1

Zapoznaj się z filmem instruktażowym i wykonaj kolejno przedstawione czynności na swoim komputerze. Pliki w archiwum .zip, potrzebne do wykonania zadania znajdziesz poniżej.

Plik o rozmiarze 1.30 KB w języku polskim

Ważne!

Pamiętaj, by po każdej zmianie skryptu go zapisać.



Film dostępny pod adresem </preview/resource/R1DDChUXFqKBm>

Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Film-samouczek: Współpraca bazy danych ze stroną internetową, etap pierwszy.

Plik o rozmiarze 386.00 B w języku polskim

Polecenie 2

Zrealizuj poprawną obsługę polskich znaków diakrytycznych w plikach projektu, tak aby nowe rekordy dodawane z użyciem kwerendy INSERT zawierały poprawne polskie litery (pełne wyjaśnienie treści zadania znajdziesz w filmie powyżej).

Sprawdź się

Pokaż ćwiczenia:   

Ćwiczenie 1



Połącz w pary funkcje biblioteki `mysqli` z zadaniami, które realizują w kodzie źródłowym PHP.

wybranie używanego w połączeniu zestawu znaków

`mysqli_query()`

nawiązanie połączenia z bazą danych

`mysqli_close()`

zamknięcie aktywnego połączenia z bazą danych

`mysqli_connect()`

wykonanie kwerendy SQL w bazie danych

`mysqli_set_charset()`

Ćwiczenie 2



Podaj właściwą kolejność użycia instrukcji, w algorytmie komunikowania się strony internetowej z bazą danych.

`mysqli_query()`



`mysqli_connect()`



`mysqli_set_charset()`



`mysqli_close()`



Ćwiczenie 3



Wstaw we właściwej kolejności argumenty do funkcji `mysqli_connect()`.

```
$polaczenie = mysqli_connect(" ", " ", " ", "  
" ) or die("Błąd połączenia!");
```

admin

m0jetajn3ha5lo

artykuly

78.46.80.114

Ćwiczenie 4



Przypisz wybrane cechy rozszerzenia do odpowiedniej biblioteki PHP.

mysql

interfejs zarówno proceduralny,
jak i obiektowy

niezalecane do nowych
projektów

mysqli

interfejs tylko obiektowy

rozszerzenie wycofane od PHP
w wersji 7.0

PDO

możliwe użycie silników innych
niż MySQL

interfejs tylko proceduralny

Ćwiczenie 5



Uzupełnij brakujące fragmenty algorytmu współpracy strony internetowej z bazą danych.

```
$polaczenie = mysqli_ ("localhost", "root", "", "sklep")  
 ("Error!");
```

```
$zapytanie = "UPDATE ksiazki SET tytul = 'Potop' WHERE id = 145";
```

```
mysqli_query( , );
```

```
mysqli_close($polaczenie);
```

\$zapytanie

connect

or die

set_charset

\$polaczenie

utf8

SELECT

Ćwiczenie 6



Wskaż, co należy wpisać pomiędzy znacznikami, aby w pliku z rozszerzeniem .php umieścić działający kod języka PHP.

☐ `<?php> . . . . . </php>`

☐ `<php> . . . . . </php>`

☐ `<!php . . . . . !>`

☐ `<?php . . . . . ?>`

Ćwiczenie 7



Wstaw odpowiednie frazy w brakujące fragmenty opisu technologii webowych.

Do technologii (realizowanych po stronie klienta) zaliczamy między innymi języki . Kody źródłowe w tych plikach są . Zaś mianem języków określamy technologie realizowane zdalnie, czyli na serwerze. Przykładowe języki zaliczane do tej drugiej kategorii, to między innymi . Kody źródłowe w tych plikach są .

PHP i SQL

back-endowych

front-endowych

utajnione

HTML i CSS

jawne

Ćwiczenie 8



Połącz w pary nazwy funkcji biblioteki `mysqli` z ilością argumentów, których potrzebują do poprawnego wywołania w kodzie PHP.

`mysqli_connect()`

cztery argumenty

`mysqli_query()`

dwa argumenty

`mysqli_close()`

jeden argument

Dla nauczyciela

Autor: Mirosław Zelent

Przedmiot: Informatyka

Temat: Współpraca bazy danych ze stroną internetową, etap I

Grupa docelowa:

Szkoła ponadpodstawowa, liceum ogólnokształcące, technikum, zakres rozszerzony

Podstawa programowa:

Cele kształcenia – wymagania ogólne

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Treści nauczania – wymagania szczegółowe

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych.

Zakres rozszerzony. Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:

4) przygotowując opracowania rozwiązań złożonych problemów, posługuje się wybranymi aplikacjami w stopniu zaawansowanym:

d) projektuje i tworzy relacyjną bazę złożoną z wielu tabel oraz sieciową aplikację bazodanową dla danych związanych z rozwiązywanym problemem, formułuje kwerendy, tworzy i modyfikuje formularze oraz raporty, stosuje język SQL do wyszukiwania informacji w bazie i do jej modyfikacji, uwzględnia kwestie integralności danych, bezpieczeństwa i ochrony danych w bazie,

Kształtowane kompetencje kluczowe:

- kompetencje cyfrowe;
- kompetencje osobiste, społeczne i w zakresie umiejętności uczenia się;
- kompetencje matematyczne oraz kompetencje w zakresie nauk przyrodniczych, technologii i inżynierii.

Cele operacyjne (językiem ucznia):

- Skonstruujesz pierwszą stronę internetową skutecznie komunikującą się z bazą danych.
- Dobierzesz środki zapewniające poprawne kodowanie polskich znaków w stworzonej witrynie oraz w obsługującej ją bazie danych.
- Ocenisz różnice występujące pomiędzy językami programowania wykonywanymi po stronie klienta, a tymi realizowanymi zdalnie na serwerze.

Strategie nauczania:

- konstruktywizm;
- konektywizm.

Metody i techniki nauczania:

- dyskusja;
- rozmowa nauczająca z wykorzystaniem multimediu i ćwiczeń interaktywnych;
- metody aktywizujące.

Formy pracy:

- praca indywidualna;
- praca w parach;
- praca w grupach;
- praca całego zespołu klasowego.

Środki dydaktyczne:

- komputery z głośnikami, słuchawkami i dostępem do internetu;
- zasoby multimedialne zawarte w e-materiale;
- tablica interaktywna/tablica, pisak/kreda;
- telefony z dostępem do internetu.

Przebieg lekcji

Przed lekcją:

1. **Przygotowanie do zajęć.** Nauczyciel loguje się na platformie i udostępnia e-materiał: „Współpraca bazy danych ze stroną internetową, etap I”. Nauczyciel prosi uczniów o zapoznanie się z treściami w sekcji „Przeczytaj”.

Faza wstępna:

1. Wyświetlenie przez nauczyciela tematu i celów lekcji. Określenie wiążących dla uczniów kryteriów sukcesu.
2. **Rozpoznanie wiedzy uczniów.** Uczniowie tworzą pytania dotyczące tematu zajęć, na które odpowiedzą w trakcie lekcji.

Faza realizacyjna:

1. **Praca z tekstem.** Uczniowie przystępują do cichego czytania tekstu e-materiału. Indywidualnie zapoznają się z treścią w sekcji „Przeczytaj”.
2. **Praca z multimediami.** Nauczyciel wyświetla zawartość sekcji „Film samouczek”. Uczniowie oglądają film instruktażowy, realizując kolejno przedstawione czynności na swoim stanowisku komputerowym. Pliki potrzebne do wykonania zadania znajdują się w archiwum ZIP.
3. **Ćwiczenie umiejętności.** Uczniowie wykonują indywidualnie ćwiczenia nr 1-8, a następnie dzielą się wynikami swojej pracy z kolegą lub koleżanką.

Faza podsumowująca:

1. Wybrany uczeń podsumowuje zajęcia, zwracając uwagę na nabyte umiejętności, omawia ewentualne problemy podczas rozwiązywania ćwiczeń.

Praca domowa:

1. Uczniowie wykonują ćwiczenie nr 2 z sekcji „Film samouczek”. Ich zadaniem jest realizacja poprawnej obsługi polskich znaków diakrytycznych w plikach projektu, tak aby nowe rekordy dodawane z użyciem kwerendy INSERT zawierały poprawne polskie litery. Pełne wyjaśnienie treści zadania znajduje się w filmie.

Wskazówki metodyczne:

- Treści w sekcji „Przeczytaj” można wykorzystać jako podsumowanie i utrwalenie wiedzy uczniów.