Współpraca bazy danych ze stroną internetową, etap III

- Wprowadzenie
- Przeczytaj
- Film samouczek
- Sprawdź się
- Dla nauczyciela

Źródło: domena publiczna.

Język HTML dostarcza wiele gotowych do użycia elementów formularzy, z których można zbudować intuicyjny i wygodny interfejs pośredniczący w wymianie informacji między użytkownikiem a bazą danych. W internecie bardzo często oglądamy formularze rejestracji, logowania, koszyki obsługujące zakupy, wyszukiwarki albo listy wyboru przygotowane właśnie z wykorzystaniem języka HTML.

Programista webowy powinien zapewnić użytkownikowi nie tylko sprawne komunikowanie się z serwerem za pośrednictwem formularza, lecz także zagwarantować bezpieczeństwo przesyłanych danych oraz sprawdzanie poprawności wymienianych przez internet informacji.

W skład serii wchodzą również e-materiały:

- Współpraca bazy danych ze stroną internetową, etap I,
- Współpraca bazy danych ze stroną internetową, etap II,
- Współpraca bazy danych ze stroną internetową, etap IV.

Twoje cele

- Korzystając z języka HTML, skonstruujesz formularz ułatwiający korzystanie z bazy danych.

- Poznasz różnice między dwiema metodami przesyłania informacji z formularza na serwer.
- Unikniesz błędów związanych z brakiem sprawdzenia faktu wysłania informacji z formularza na serwer.
- Rozpoznasz możliwość zaatakowania formularzy przetwarzanych przez skrypt PHP z wykorzystaniem tzw. wstrzykiwania SQL.

Przeczytaj

Pobierz archiwum ZIP z plikami niezbędnymi do wykonania zadań przedstawionych w tym e-materiale:

Plik o rozmiarze 2.71 KB w języku polskim

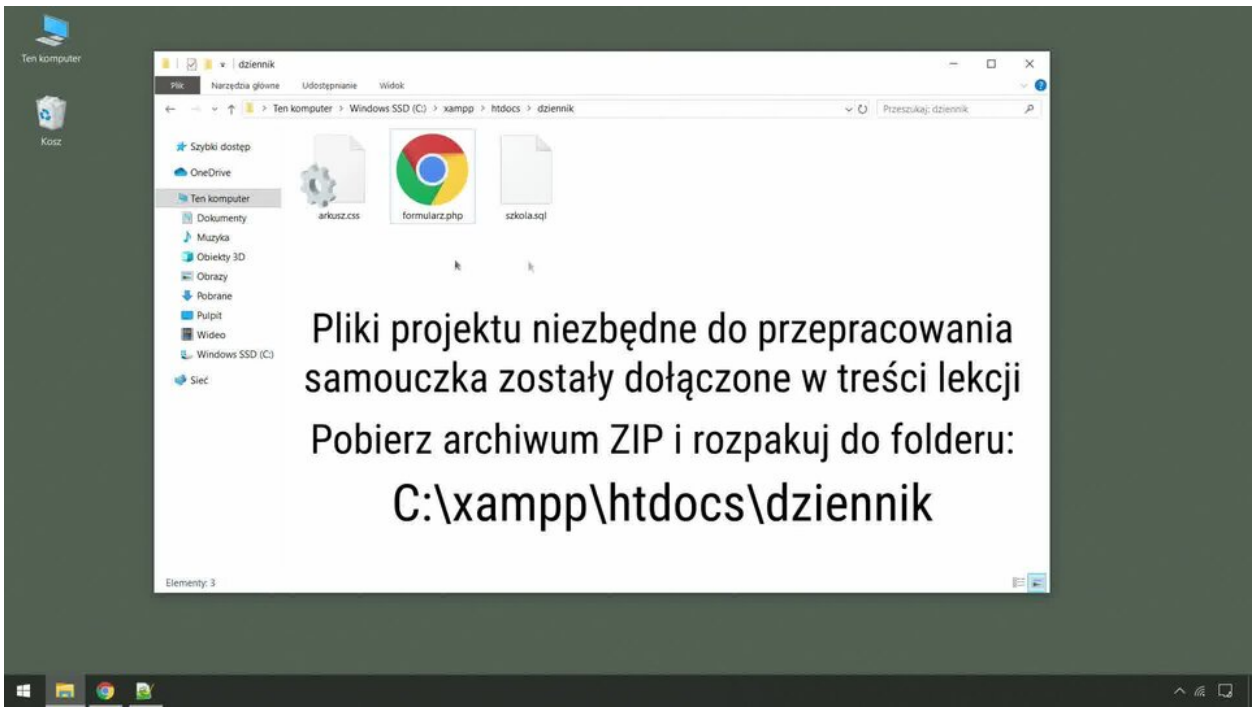
Rozważmy bazę danych obsługującą **internetowy dziennik elektroniczny**. W skład bazy o nazwie *dziennik* wchodzą trzy tabele:

- *klasa*, o atrybutach: `id` (INT), `nazwa` (TINYTEXT),
- *uczen*, o atrybutach: `id` (INT), `Nazwisko` (TINYTEXT), `Imie` (TINYTEXT), `id_klasy` (INT),
- *wychowawca*, o atrybutach: `id` (INT), `imie` (TINYTEXT), `nazwisko` (TINYTEXT), `id_klasy` (INT).

Możemy wyróżnić następujące **powiązania** (relacje) istniejące między atrybutami:

- atrybut `klasa.id` (klucz główny) pozostaje w relacji do pola `uczen.id_klasy` (klucz obcy),
- atrybut `klasa.id` (klucz główny) pozostaje w relacji do pola `wychowawca.id_klasy` (klucz obcy).

Strukturę tabel składowych, powiązań pomiędzy atrybutami oraz opis plików źródłowych wchodzących w skład projektu pokazano w filmie:



Film dostępny pod adresem /preview/resource/RujZoBU3PHNTD

Film samouczek przedstawia współpracę bazy danych ze stroną internetową.

Formularze w języku HTML

Formularze służą użytkownikom do **wprowadzania danych wejściowych** do bazy danych (za pośrednictwem **interfejsu** witryny) w celu ich **przetworzenia**. Przykładem może być **formularz rejestracji**, który pozwala wprowadzić nowego użytkownika do bazy albo **formularz logowania**, dzięki któremu sprawdzana jest zgodność identyfikatora (loginu) i hasła ze wzorcami umieszczonymi w bazie danych.

Cały opis formularza zamykamy w parze znaczników `<form></form>`, zaś **pomiędzy** nimi umieszczamy **dowolny zestaw** predefiniowanych w języku HTML **kontrolek formularza**. Są one różne; mogą to być: pola tekstowe, przyciski, pola do zaznaczenia, listy wyboru itd.

```
<form action="index.php" method="post">
    <!-- Tutaj wybrane kontrolki formularza -->
</form>
```

Atrybut `action` określa jednoznacznie, **który plik** zajmie się **przetwarzaniem danych** otrzymanych z formularza. Jako wartość podajemy **ścieżkę względną** do przygotowanego **skryptu PHP**. Przeznaczenie atrybutu `method` omówiono niżej.

Metody przesyłania informacji z formularza na serwer

Atrybut `method` w znaczniku `<form>` określa sposób dostarczenia informacji z formularza na serwer. Istnieją dwie klasyczne metody komunikacji z serwerem:

- `method="get"` – dane zostaną wysłane w postaci **jawnej** – wartości podane w formularzu zostaną **dołączone („doklejone”) do adresu witryny** pokazywanego w pasku adresu przeglądarki;
- `method="post"` – dane zostaną wysłane w postaci **niejawnej** – wartości podane w formularzu **nie zmieniają adresu** witryny w przeglądarce.

> **Ciekawostka**
>
> W przypadku **braku deklaracji** atrybutu `method` wewnątrz znacznika `<form>` wykorzystana zostanie metoda `get`, ponieważ jest to **wartość domyślna** w standardzie HTML.

Pole tekstowe, etykieta pola

Wiele kontrolek formularzy określamy za pomocą znacznika `<input>` (ang. „wejście"). **Rodzaj elementu**, którego chcemy użyć, deklarujemy, używając atrybutu `type` (typ definiowanej kontrolki). Klasyczne pole tekstowe, znane z systemu Windows albo ze stron internetowych, wstawimy do formularza w następujący sposób:

```
<input type="text" name="klasa">
```

Ważnym elementem standardu HTML, o którym nie należy zapominać w formularzu, jest **etykieta pola** – do jej zdefiniowania służą znaczniki `<label> </label>`. Jest to nie tylko słowny opis przeznaczenia pola – etykieta **reaguje na kliknięcie użytkownika** – powiązana z nią kontrolka formularza zyska aktywność. W przypadku pola tekstowego kliknięcie etykiety spowoduje **pokazanie klawiatury** na ekranie urządzenia mobilnego oraz **ustawienie kursora wewnątrz pola**. A zatem prawidłowo powiązany z kontrolką znacznik `<label>` sprzyja tzw. dostępności formularza.

Istnieją **dwa sposoby przypisania etykiety** do kontrolki formularza:

- Wewnątrz etykiety znajduje się zarówno opis pola, jak i sam znacznik `<input>`:

```
<label>
    Podaj nazwę klasy: <input type="text" name="klasa">
</label>
```

- Etykieta zamknięta jest przed polem tekstowym, ale pojawia się dodatkowo atrybut `for` (ang. „dla"), określający, **dla którego elementu jest to etykieta**. Wartością atrybutu `for` jest identyfikator, podany także w atrybucie `id` wewnątrz znacznika `<input>`. Obydwa identyfikatory oczywiście **muszą być identyczne**: właśnie na ich podstawie dochodzi do **logicznego powiązania** pomiędzy kontrolką formularza i jego etykietą:

```
<label for="klasa">Podaj nazwę klasy:</label>
<input type="text" name="klasa" id="klasa">
```

Przyciski podsumowujące formularz

Istnieją **trzy podstawowe sposoby** definiowania przycisków w języku HTML. Jednak **tylko dwa** umożliwiają **podsumowanie** formularza:

- Klasyczny przycisk **dokonujący operacji submit**, czyli podsumowania całego formularza – dane **zostaną wysłane** na serwer. Napis na przycisku określamy atrybutem `value` (ang. „wartość"): `<input type="submit" value="Wyślij">`

- Podwójny znacznik `<button></button>` zamiast pojedynczego znacznika `<input>`. **Również nastąpi podsumowanie** formularza. W tej wersji napis na przycisku umieszczamy po prostu pomiędzy tagami: `<button>Wyślij</button>`

> **Ciekawostka**
>
> Dlaczego dano nam możliwość zapisywania przycisku w takiej uwspółcześnionej wersji, skoro zdarzenie `submit` i tak następuje dla klasycznego przycisku? Wynika to z faktu, że znacznik `<button>` **jest podwójny** – ma tag **otwierający** oraz **zamykający**, w przeciwieństwie do pojedynczego `<input>`. To z kolei sprawia, że pomiędzy znacznikami `<button></button>` można w języku HTML wskazać dodatkowy element (na przykład grafikę – `<img>`).

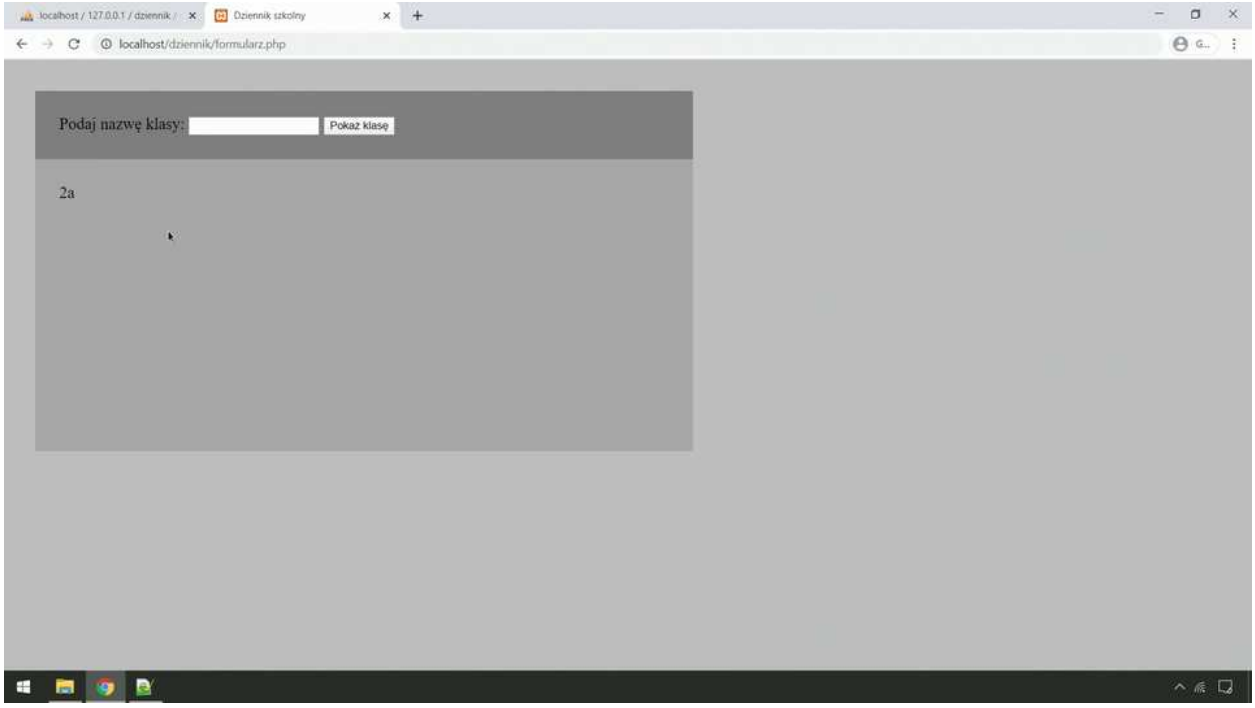
3. Znacznik definiujący „zwykły", „niegorący" przycisk: jego kliknięcie **nie podsumowuje** formularza, tak jak to miało miejsce dla dwóch wcześniejszych kontrolek:

```
<input type="button" value="Kliknij">
```

> **Ciekawostka**
>
> **Nie każdy przycisk** powinien podsumować formularz – zdarzają się sytuacje, w których kliknięcie powinno być obsłużone **jedynie po stronie klienta** (w języku JavaScript) – na przykład przycisk przywołujący w animowany sposób menu główne witryny na ekran.

Wykonanie przykładowego formularza

Pora wykorzystać **w praktyce** zdobyte informacje. W filmie-samouczku zrealizujemy **podstronę dziennika elektronicznego**, umożliwiającą **wypisanie listy uczniów** wskazanej w formularzu klasy:



Film dostępny pod adresem /preview/resource/RzdJmfqdnY3OU

Film samouczek przedstawia współpracę bazy danych ze stroną internetową oraz tworzenie formularzy.

> **Polecenie 1**
>
> Tuż pod tabelą zawierającą wypisanie imion i nazwisk uczniów należących do wybranej klasy powinna się pojawić także informacja (odczytana z bazy danych) o tym, **kto jest wychowawcą** tej grupy uczniów.
>
>

Odczytywanie w skrypcie PHP wartości przysłanej z formularza

Wartość podaną przez użytkownika w formularzu można **odebrać w skrypcie PHP** w specjalnej, **globalnie widocznej** tablicy, w której indeks **jest taki sam**, jak wartość atrybutu `name` danej kontrolki formularza**. W zależności od wybranego w atrybucie `method` sposobu przesyłania informacji, operacja odczytu wygląda następująco:

```php
<!--Kontrolka formularza -->
<input type="text" name="nazwa_pola">

<?php

    // Odczyt, jeśli wybrano dla formularza metodę POST
    $wartosc = $_POST["nazwa_pola"];

    // Odczyt, jeśli wybrano dla formularza metodę GET
    $wartosc = $_GET["nazwa_pola"];

?>
```

> **Ciekawostka**
>
> Zwróć uwagę, że atrybut `name` kontrolki formularza służy właśnie do **przesłania wartości** wyjętej z formularza **do skryptu PHP** na serwerze. Atrybut `id` najczęściej przydaje się z kolei w przypadku chęci **uchwycenia kontrolki w języku JavaScript** (lub ewentualnie do ustanowienia logicznego powiązania elementu z etykietą). Stąd niejednokrotnie spotkamy w formularzach HTML kontrolki mające obydwa wspomniane atrybuty.

Odebranie wartości z globalnej tablicy `$_POST` lub `$GET` powinno nastąpić po uprzednim **upewnieniu się**, że użytkownik **przesłał formularz** – potrzebna jest do tego dodatkowa **instrukcja warunkowa**. W przeciwnym razie pierwsze wejście użytkownika do dokumentu spowoduje pokazanie następującego ostrzeżenia:

```
Notice: Undefined index nazwa_pola
```

Nie mieliśmy jeszcze **możliwości wpisania czegokolwiek** do pola tekstowego, stąd nie ma sensu dokonywać odczytu wartości, ani tym bardziej łączenia się z bazą danych, aby **nieistniejącej wartości** użyć w kwerendzie! Przyjrzymy się zatem możliwości sprawdzenia instrukcją warunkową sensu dokonywania aktu odczytu.

Sprawdzenie, czy formularz został wysłany

Do zrealizowania **testu podsumowania formularza** przez użytkownika, możemy wykorzystać dwa rodzaje instrukcji warunkowych:

- Sprawdzenie funkcją PHP o nazwie `isset()` (ang. „istnieje, jest ustawione”), czy podany indeks w tablicy globalnej został zarezerwowany w pamięci RAM serwera:

```
1 if (isset($_POST["nazwa_pola"])) { ... }
```

> **Ciekawostka**
>
> Metody PHP `isset()` **nie można zastąpić** popularną funkcją `empty()`, ponieważ indeks w tablicy musi istnieć, ale może **zawierać pustą wartość**. Z tego powodu funkcja `empty()` doskonale nada się do sprawdzenia, czy użytkownik nie zapomniał w ogóle wpisać do pola wartości.

- Weryfikacja wartości indeksu `REQUEST_METHOD` w globalnej tablicy `$_SERVER`. Ta metoda jest **niezależna od ewentualnych pomyłek** programisty w wartościach atrybutu `name`, ponieważ atrybuty te nie zostają w ogóle użyte wewnątrz warunku:

```
1 if ($_SERVER["REQUEST_METHOD"] == "POST") { ... }
```

> **Ciekawostka**
>
> Druga metoda zachowa się poprawnie **tylko dla metody POST**. Zastosowanie jej z wykorzystaniem żądania GET **spełni warunek zapisany w instrukcji** nawet wtedy, gdy **nie przesłano formularza**. Dzieje się tak dlatego, iż nawet zwykłe wejście na stronę internetową bez wypełnienia formularza **stanowi żądanie GET** w protokole HTTP.

Słownik

> **dostępność**
>
> (ang. *accessibility*) ogół zasad projektowania interfejsów komunikacji człowieka z komputerem, których nadrzędnym celem jest zapewnienie komfortowego dostępu do funkcji serwisów internetowych jak największemu gronu osób (w tym także ludziom narażonym na wykluczenie cyfrowe z powodu niepełnosprawności, wieku, braku wykształcenia lub wskutek ograniczeń technologicznych)
> **sanityzacja kodu**
>
> proces mający doprowadzić do wyeliminowania z danych wejściowych możliwie największej liczby znaków sterujących i operatorów, które mogą mieć wpływ na zmianę sposobu działania aplikacji
> **wstrzykiwanie SQL**
>
> (ang. *SQL injection*) metoda ataku na witrynę internetową komunikującą się z bazą danych, polegająca na przemyceniu fragmentu zapytania SQL poprzez odpowiednio spreparowaną wartość wprowadzaną w formularzu

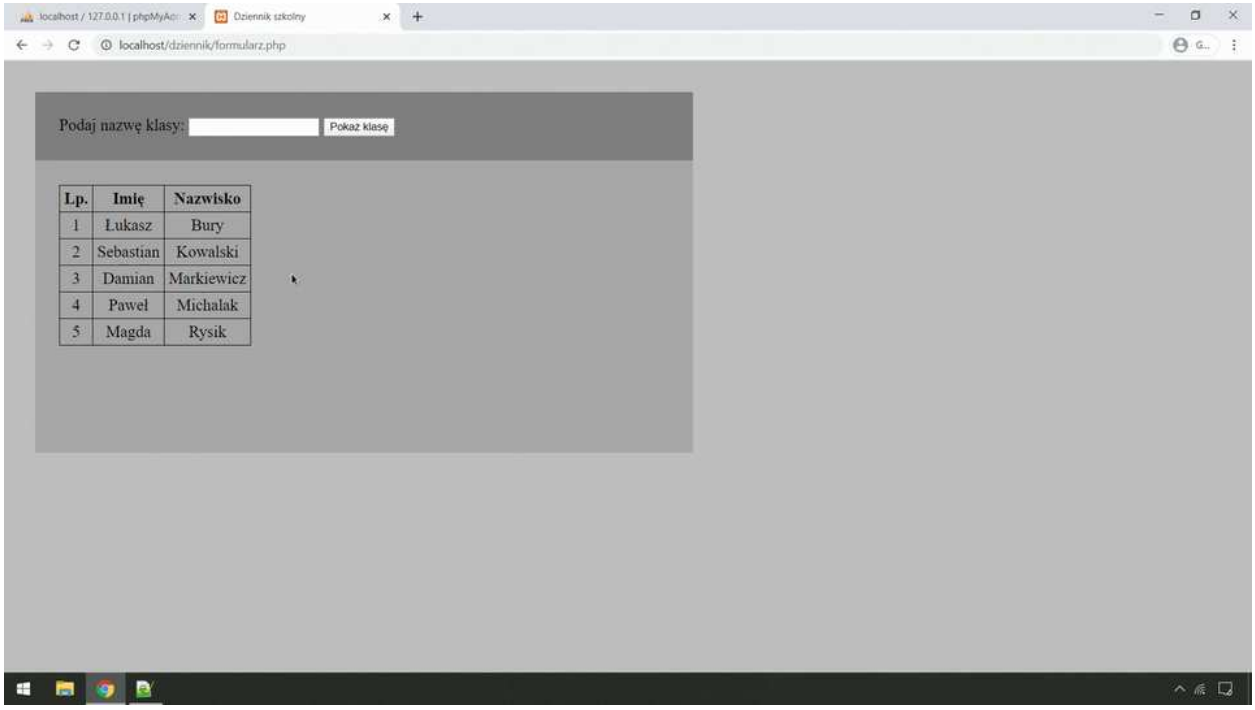
Film samouczek

Wstrzykiwanie SQL

Niezwykle ważne jest, aby podczas projektowania formularzy używanych na stronach internetowych rozumieć **mechanikę przeprowadzania ataku** tzw. wstrzykiwania SQL. Jest to podatność korzystającej z bazy danych strony internetowej na atak, polegający na **przemyceniu fragmentu zapytania SQL** poprzez wartość kontrolki formularza.

Do przeprowadzenia ataku tego rodzaju wykorzystuje się **znak apostrofu** oraz **operator komentarza** w języku SQL. Szczegóły mechaniki przeprowadzania ataku na formularz przedstawiono w filmie.

> **Polecenie 1**
>
> Zapoznaj się z filmem.



Film dostępny pod adresem /preview/resource/R14kb4v0gO2Wa

Film samouczek przedstawia działanie ataku na stronę internetową SQL injection.

Plik o rozmiarze 507.00 B w języku polskim

Atak powiedzie się tylko w przypadku niedostatecznej tzw. sanityzacji wartości pobranej z formularza. **Zasada ogólna**, której powinni w kontekście bezpieczeństwa danych przestrzegać wszyscy programiści webowi, brzmi: **nigdy nie ufaj danym wejściowym wprowadzonym przez użytkownika**.

Sprawdź się

Pokaż ćwiczenia:

Ćwiczenie 1

Wskaż, której z instrukcji warunkowych należy użyć, by upewnić się, że formularz HTML został wysłany na serwer.

- ◯ `if (empty($_POST["nazwa_pola"]))`
- ◯ `if (isset($_POST["nazwa_pola"]))`
- ◯ `if ($_SERVER["REQUEST_METHOD"] == "GET")`
- ◯ `if ($_POST["nazwa_pola"] == "")`

Ćwiczenie 2

Uporządkuj we właściwej kolejności etapy realizowania poprawnej obsługi formularza.

- zamknięcie aktywnego połączenia z bazą danych
- nawiązanie połączenia z bazą danych
- sprawdzenie, czy przesłano formularz
- ustawienie obsługi polskich znaków
- sprawdzenie, czy podana w polu wartość jest pusta
- wykonanie zapytania i włożenie rekordów do `$result`
- umieszczenie treści zapytania w zmiennej `$q`
- operacja fetchowania i wypisanie wyjętych rekordów

Ćwiczenie 3

Wstaw do skryptu w odpowiednich miejscach funkcje zapewniające poprawną obsługę formularza.

```
if (          ) {
      ;  // Odczyt wartości zmiennej
      if (          ) {
            echo "<p>Nie podano nazwy klasy!</p>";
      }           {
            echo "<p>Wpisana nazwa klasy: $klasa; </p>";
      }
}
```

`empty($klasa)` `isset($_POST["klasa"])`

`$_SERVER["REQUEST_METHOD"] == "GET"` `$klasa = $_POST["klasa"]` `else`

Ćwiczenie 4

Dokończ zdanie.

Wieloliniowe echo, zakończone następującą etykietą: STOP ; powinno zostać rozpoczęte z użyciem instrukcji...

- ◯ `echo<<<STOP.`
- ◯ `echo<<STOP.`
- ◯ `echo[STOP].`
- ◯ `echo(STOP).`

Ćwiczenie 5

Uzupełnij poprawne zdefiniowanie etykiety dla pola tekstowego.

```
<label [      ]="klasa">Podaj nazwę klasy:</label>
<input [      ]="text" name="klasa" [      ]="klasa">
```

`method` `name` `type` `for` `id`

Ćwiczenie 6

Wskaż, co nie jest operatorem komentarza w języku SQL.

- [ ] // (dwa znaki forward slash)
- [ ] \\ (dwa znaki backslash)
- [ ] -- (dwa myślniki i spacja)
- [ ] <!-- (operator mniejszy niż, wykrzyknik i dwa myślniki)

Ćwiczenie 7

Z użyciem metody GET przesłano na serwer formularz zawierający dwa pola tekstowe, których atrybuty `name` mają wartości kolejno: `imie` oraz `nazwisko`. Atrybut `action` formularza wskazywał na skrypt o nazwie `index.php`. Wskaż, jaki adres pojawi się w pasku adresu przeglądarki po podsumowaniu formularza, jeśli osobą przesyłającą dane był Jan Nowak.

- ◯ `index.php&imie=Jan?nazwisko=Nowak`
- ◯ `index.php%imie=Jan%%nazwisko=Nowak`
- ◯ `index.php?imie=Jan&nazwisko=Nowak`
- ◯ `index.php%imie=Jan&nazwisko=Nowak`

Ćwiczenie 8

Uzupełnij odpowiednio elementy formularza HTML umożliwiającego przesłanie na serwer swojego imienia i nazwiska.

```
<form
        ="index.php"         ="post">
    <label>Imię: <input type="text" name="imie"></label>
    <label>Nazwisko: <input type="text" name="nazwisko"></label>
    <input type="        "         ="Wyślij dane">
</form>
```

`method` `caption` `button` `submit` `action` `value`

Dla nauczyciela

Autor: Mirosław Zelent
Przedmiot: Informatyka

Temat: Współpraca bazy danych ze stroną internetową, etap III

Grupa docelowa:

Liceum ogólnokształcące i technikum, 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:
>
> e) programuje elementy strony internetowej współpracujące z sieciową bazą danych;

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):

- Korzystając z języka HTML, skonstruujesz formularz ułatwiający korzystanie z bazy danych.

- Poznasz różnice między dwiema metodami przesyłania informacji z formularza na serwer.
- Unikniesz błędów związanych z brakiem sprawdzenia faktu wysłania informacji z formularza na serwer.
- Rozpoznasz możliwość zaatakowania formularzy przetwarzanych przez skrypt PHP z wykorzystaniem tzw. wstrzykiwania SQL.

Strategie nauczania:

- konstruktywizm;
- konektywizm.

Metody i techniki nauczania:

- dyskusja;
- rozmowa nauczająca z wykorzystaniem multimedium i ćwiczeń interaktywnych;
- ćwiczenia praktyczne.

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 III". Nauczyciel prosi uczniów o zapoznanie się z treściami w sekcji „Przeczytaj".

Faza wstępna:

1. Nauczyciel wyświetla temat i cele zajęć. Prosi uczniów, by na podstawie wiadomości zdobytych przed lekcją zaproponowali kryteria sukcesu.
2. **Rozpoznanie wiedzy uczniów.** Uczniowie tworzą pytania dotyczące tematu zajęć, na które odpowiedzą w trakcie lekcji.

Faza realizacyjna:

1. **Praca z tekstem.** Jeżeli przygotowanie uczniów do lekcji jest niewystarczające, nauczyciel prosi o indywidualne zapoznanie się z treścią zawartą w sekcji „Przeczytaj". Każdy uczestnik zajęć podczas cichego czytania wynotowuje najważniejsze kwestie poruszane w tekście.
2. Nauczyciel pozostawia wyświetloną zawartość sekcji „Przeczytaj", czyta treść polecenia nr 1: „Tuż pod tabelą zawierającą wypisanie imion i nazwisk uczniów należących do wybranej klasy powinna się pojawić także informacja (odczytana z bazy danych) o tym, kto jest wychowawcą tej grupy uczniów.". Prosi uczniów, aby w parach przeanalizowali rozwiązanie problemu. Wybrana para prezentuje wynik swojej pracy na forum klasy.
3. **Ćwiczenie umiejętności.** Prowadzący zapowiada uczniom, że w kolejnym kroku będą rozwiązywać ćwiczenia nr 1-5 z sekcji „Sprawdź się". Każdy z uczniów robi to samodzielnie. Po ustalonym czasie wybrani uczniowie przedstawiają rozwiązania. Nauczyciel w razie potrzeby koryguje odpowiedzi, dopowiada istotne informacje, udziela uczniom informacji zwrotnej.

Faza podsumowująca:

1. Nauczyciel ponownie wyświetla na tablicy temat i cele lekcji zawarte w sekcji „Wprowadzenie". W kontekście ich realizacji następuje omówienie ewentualnych problemów z rozwiązaniem ćwiczeń z sekcji „Sprawdź się".

Praca domowa:

1. Uczniowie opracowują FAQ (minimum 3 pytania i odpowiedzi) do tematu lekcji („Współpraca bazy danych ze stroną internetową, etap III").
2. Uczniowie wykonują ćwiczenia 6-8 z sekcji „Sprawdź się".

Wskazówki metodyczne:

- Treści w sekcji „Film samouczek" można wykorzystać na lekcji jako podsumowanie i utrwalenie wiedzy uczniów.