

Ciekawostki do lekcji, przypomnienie lub wyjaśnienie

Cecha	Metoda GET	Metoda POST
Widoczność danych	Dane są dopisywane do adresu URL (np. <code>?id=5</code>).	Dane są ukryte w "ciele" zapytania HTTP.
Bezpieczeństwo	Małe (hasła lub dane osobowe są widoczne w pasku adresu).	Większe (dane nie są widoczne na pierwszy rzut oka).
Limit danych	Ograniczony (zazwyczaj do ok. 2000 znaków w URL).	Brak sztywnego limitu (można przysyłać duże pliki).
Historia i Zakładki	Stronę można dodać do zakładek wraz z danymi.	Strony nie da się zapisać w zakładkach z przesłanymi danymi.
Zastosowanie	Pobieranie danych, wyszukiwanie, filtrowanie.	Wysyłanie danych, logowanie, dodawanie rekordów do bazy.

Funkcja `implode()` w PHP to narzędzie służące do łączenia elementów tablicy w jeden ciąg znaków (string). Można o niej myśleć jak o „sklejaniu” rozsypanych puzzli w jedną całość za pomocą wybranego „łącznika”.

Kiedy użytkownik zaznacza kilka checkboxów w formularzu, PHP otrzymuje je w formie tablicy, np.: `['sport', 'film', 'gry']`.

Baza danych nie potrafi bezpośrednio zapisać tablicy w jednej kolumnie tekstowej. Dlatego używasz `implode()`:

PHP

```
$zainteresowania = implode(", ", $_POST['zainteresowania']);
```

- **Pierwszy argument** `(", ")`: To separator. Mówisz PHP: „rozdziel wyrazy przecinkiem i spacją”.
- **Drugi argument** `($_POST['zainteresowania'])`: To tablica z danymi, które chcesz połączyć.

Efekt: Z tablicy powstaje tekst: `"sport, film, gry"`, który możesz bezpiecznie zapisać w bazie danych.

Przeciwieństwo: `explode()`

Warto pamiętać, że `implode()` ma swojego „brata bliźniaka” o nazwie `explode()`, którego również używasz w pliku `edytuj.php`.

- `implode()`: Tablica $\rightarrow$ String (do zapisu w bazie).
- `explode()`: String $\rightarrow$ Tablica (do wyświetlenia danych z bazy w formie checkboxów).

Wyświetlenie danych:

Wyświetlanie rekordów w pętli `while`

To najważniejsza część pliku, która zamienia surowe dane z bazy na wiersze tabeli:

- `mysqli_fetch_assoc($wynik)`: Ta funkcja pobiera jeden wiersz z bazy danych i zamienia go w tablicę asocjacyjną o nazwie `$row`.
- **Pętla `while`**: Powtarza się tak długo, jak długo w zmiennej `$wynik` znajdują się jeszcze jakieś rekordy.
- `$row['nazwa_kolumny']`: Wewnątrz pętli wyciągamy konkretne wartości (np. imię, miasto) i wstawiamy je między znaczniki komórek tabeli `<td>...</td>`.

Mechanizm usuwania

Zanim strona się wyświetli, PHP sprawdza, czy użytkownik kliknął przycisk "Usuń":

- `if (isset($_POST['usun']))`: Sprawdza, czy przesłano żądanie usunięcia.
- `$id_do_usuniecie = (int)$_POST['id_klient']`: Pobiera ID rekordu i zamienia je na liczbę całkowitą (bezpieczeństwo).
- `DELETE FROM klient WHERE id_klient = ...`: Wykonuje zapytanie SQL, które trwale usuwa konkretny wiersz z bazy.

Generowanie przycisków akcji w pętli `while`

Dla każdego klienta w bazie, oprócz jego danych, skrypt generuje dwa małe formularze:

- **Przycisk "Usuń" (Metoda POST):**
 - Używa pola typu `hidden`, aby przesłać `id_klient` bez pokazywania go użytkownikowi.
 - **JavaScript**: `onclick="return confirm(...)"` to proste zabezpieczenie — przeglądarka zapyta "Czy na pewno?", zanim wyśle formularz.
- **Przycisk "Zmień" (Metoda GET):**
 - Ten przycisk nie usuwa niczego samodzielnie. Jego zadaniem jest przekierowanie użytkownika do pliku `edytuj.php`.
 - W adresie URL przesyła ID wybranego klienta (np. `edytuj.php?id_klient=5`), aby skrypt edycji wiedział, czyje dane wyświetlić.

```

// Kolumna Usuń
echo "<td>
    <form method='post' style='margin:0;'>
        <input type='hidden' name='id_klient' value='".$row['id_klient']."'>
        <button type='submit' name='usun' onclick='\"return confirm('Czy na pewno chcesz usunąć rekord?');\">Usuń</button>
    </form>
</td>";

// Kolumna Zmień
echo "<td>
    <form method='get' action='edytuj.php' style='margin:0;'>
        <input type='hidden' name='id_klient' value='".$row['id_klient']."'>
        <button type='submit'>Zmień</button>
    </form>
</td>";

```

Edytuj.php – wyjaśnienie:

1. Połączenie i identyfikacja rekordu

Skrypt łączy się z bazą i sprawdza, kogo ma edytować:

- **if (isset(\$_GET['id_klient'])):** Skrypt sprawdza adres URL (np. edytuj.php?id_klient=5). Jeśli ID tam jest, pobiera je.
- **Pobranie danych (SELECT):** PHP wysyła zapytanie do bazy, aby pobrać wszystkie aktualne informacje o tej konkretnej osobie i zapisuje je w tablicy \$rekord.

2. Przygotowanie formularza (Wypełnienie danymi)

To jest część wizualna w HTML. Skrypt musi wpisać stare dane w odpowiednie pola, abyś nie musiała wypełniać wszystkiego od nowa:

- **Pola tekstowe:** Używasz value="<?php echo \$rekord['imie']; ?>", co "wpisuje" imię do ramki.
- **Radio i Select:** PHP sprawdza warunkiem if, czy dana opcja była wybrana (np. płeć 'k'). Jeśli tak, dopisuje słowo checked lub selected.
- **Zainteresowania (Checkboxy):**
 1. **explode:** Rozbija tekst z bazy (np. "sport, film") na tablicę.
 2. **foreach:** Przechodzi przez wszystkie możliwe zainteresowania.
 3. **in_array:** Sprawdza, czy dane hobby jest na liście użytkownika – jeśli tak, zaznacza "ptaszka".

3. Obsługa zapisu (if (isset(\$_POST['zapisz'])))

Gdy zmienisz dane i klikniesz przycisk "Zapisz zmiany", uruchamia się druga część skryptu:

- **Odebranie danych:** PHP zbiera nowe wartości z pól formularza przez \$_POST.
- **implode:** Ponownie skleja zaznaczone checkboxy w jeden tekst, aby zmieściły się w jednej komórce bazy danych.

4. Zapytanie UPDATE SQL

```
UPDATE klient SET imie='NoweImie', nazwisko='NoweNazwisko', ... WHERE id_klient=5
```

- **WAŻNE:** Klauzula `WHERE id_klient=$id` jest niezbędna. Bez niej skrypt zmieniłby imiona i nazwiska **wszystkich** osób w bazie danych na te same!
-

5. Finalizacja

- **mysqli_query:** Wysyła gotowe zapytanie do bazy.
 - **exit:** Po wyświetleniu komunikatu o sukcesie, skrypt kończy działanie, aby nie przetwarzać reszty strony niepotrzebnie.
-

Podsumowanie przepływu w tym pliku:

1. **START:** Skrypt dostaje ID przez `GET`.
2. **ODCZYT:** Pobiera dane z bazy i wkłada je do pól formularza.
3. **EDYCJA:** Użytkownik zmienia co chce w formularzu.
4. **WYŚLIJ:** Użytkownik klika "Zapisz" (dane idą przez `POST`).
5. **ZAPIS:** Skrypt wykonuje `UPDATE` w bazie danych.
6. **KONIEC:** Wyświetla link powrotny do listy wszystkich klientów.

Wyszukiwanie danych po fragmencie np. imienia i nazwiska

Jakie dane pobrać z bazy na podstawie tego, czy coś wpisano w wyszukiwarkę:

PHP

```
if (isset($_GET['szukaj'])) {  
    $imie_nazwisko = mysqli_real_escape_string($connect,  
$_GET['imie_nazwisko']);  
    $zapytanie = "SELECT * FROM klient WHERE imie LIKE '%$imie_nazwisko%'  
OR nazwisko LIKE '%$imie_nazwisko%'";  
} else {  
    $zapytanie = "SELECT * FROM klient";  
}
```

- **mysqli_real_escape_string:** To funkcja bezpieczeństwa. Czyści tekst wpisany przez użytkownika, aby nikt nie próbował wpisać komend SQL (ochrona przed atakami).
- **Operator LIKE '%...%':**
 - `LIKE '%Jan%'` oznacza: znajdź tekst, który ma w środku "Jan" (znajdzie: **Jan**, **Janusz**, **Damjan**).
- **OR:** Szukamy dopasowania albo w kolumnie `imie`, albo w kolumnie `nazwisko`.

3. Formularz wyszukiwania (HTML)

- **method="get"**: Używamy GET, aby fraza wyszukiwania była widoczna w adresie URL. Dzięki temu możesz np. zapisać wynik wyszukiwania w zakładkach.
- **value="<?php echo htmlspecialchars(\$imie_nazwisko); ?>"**: Dzięki temu po kliknięciu "Wyszukaj", wpisane słowo nie znika z pola tekstowego – użytkownik widzi, co wpisał.

4. Generowanie tabeli (Pętla `while`)

Skrypt wykonuje zapytanie `$zapytanie` (albo "pobierz wszystko", albo "pobierz przefiltrowane") i w pętli generuje wiersze tabeli.