

Вывод информации в карточки и карусель Bootstrap

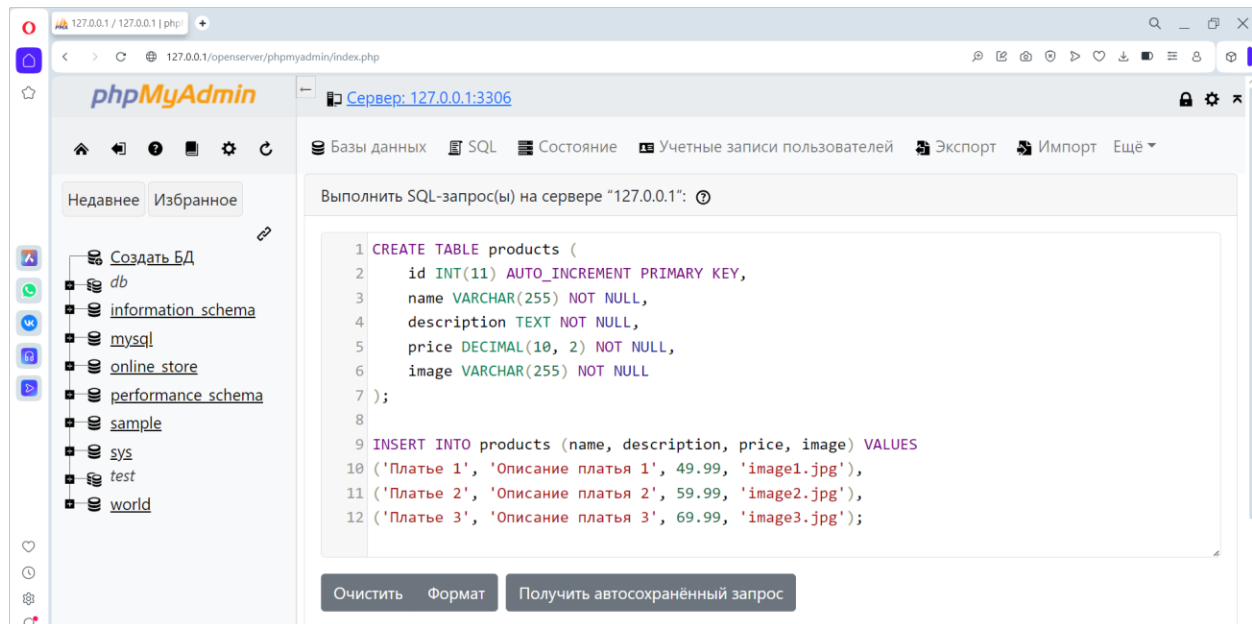
Часть 1. Простой вывод

Задание.

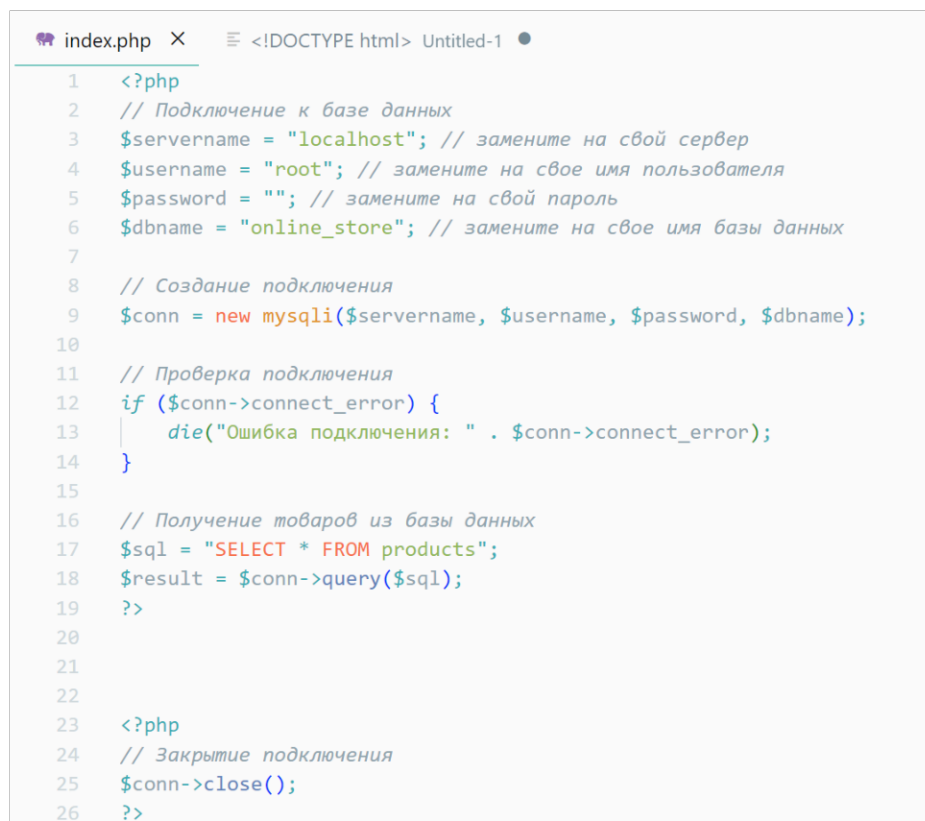
1. Изучите материал ниже и повторите код в своем проекте.
2. Реализуйте аналогичный вывод для 5 предметных областей из предыдущего задания.

Пример кода.

1. Структура базы данных online_store и контент.



2. Подключение к базе данных.



3. Стандартная разметка страницы.

```
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Карусель товаров</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-Yvpr...>
</head>
<body>

> <div class="container mt-5">...
</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-Yvp...>
</body>
</html>
```

4. Код карусели из Zeal + вывод данных из базы в цикле.

```
<div class="container mt-5">
  <h1 class="text-center mb-4">Карусель товаров</h1>
  <div id="productCarousel" class="carousel slide" data-bs-ride="carousel" data-bs-interval="3000">
    <div class="carousel-inner">
      <?php
        // Проверка на наличие товаров
        if ($result->num_rows > 0) {
          $first = true; // Для установки первого элемента в карусели активным
          while ($product = $result->fetch_assoc()) {
            echo '<div class="carousel-item ' . ($first ? 'active' : '') . '">';
            echo '<div class="card mx-auto" style="width: 18rem;">';
            echo '';
            echo '<div class="card-body">';
            echo '<h5 class="card-title">' . $product['name'] . '</h5>';
            echo '<p class="card-text">' . $product['description'] . '</p>';
            echo '<p class="card-text"><strong>Цена: ' . $product['price'] . ' </strong></p>';
            echo '<a href="#" class="btn btn-primary">Купить</a>';
            echo '</div>';
            echo '</div>';
            echo '</div>';
            $first = false; // Обновляем переменную для следующего элемента
          }
        } else {
          echo '<div class="carousel-item active"><p class="text-center">Нет товаров для отображения.</p></div>';
        }
      <?>
    </div>
    <button class="carousel-control-prev" type="button" data-bs-target="#productCarousel" data-bs-slide="prev">
      <span class="carousel-control-prev-icon" aria-hidden="true"></span>
      <span class="visually-hidden">Previous</span>
    </button>
    <button class="carousel-control-next" type="button" data-bs-target="#productCarousel" data-bs-slide="next">
      <span class="carousel-control-next-icon" aria-hidden="true"></span>
      <span class="visually-hidden">Next</span>
    </button>
  </div>
</div>
```

Подробный разбор кода карусели.

Код представляет собой динамическое формирование карусели товаров с использованием PHP и Bootstrap.

```
<div id="productCarousel" class="carousel slide" data-bs-ride="carousel" data-bs-interval="3000">
```

- **<div id="productCarousel" class="carousel slide">**: Здесь создается контейнер для карусели. Атрибут `id` назначает уникальный идентификатор для карусели, который можно использовать для управления ею с помощью JavaScript.
- **class="carousel slide"**: Эти классы связаны с Bootstrap и необходимы для стилизации карусели и добавления анимации при переходах между элементами.
- **data-bs-ride="carousel"**: Этот атрибут указывает, что карусель должна автоматически начинать прокрутку элементов при загрузке страницы.
- **data-bs-interval="3000"**: Определяет время в миллисекундах (3000 мс = 3 секунды) между автоматическими переходами на следующий элемент карусели.

```
<div class="carousel-inner">
```

- **<div class="carousel-inner">**: Контейнер для всех элементов карусели. Каждый элемент представляет собой отдельный слайд.

```
<?php
// Проверка на наличие товаров
if ($result->num_rows > 0) {
    $first = true;
```

- Этот блок PHP проверяет, есть ли товары в результате запроса к базе данных. Переменная `$result->num_rows` содержит количество записей, возвращенных запросом.
- **`$first = true;`**: Переменная инициализируется как `true`, чтобы первый элемент карусели мог быть установлен как активный.

```
while ($product = $result->fetch_assoc()) {
```

- **while (\$product = \$result->fetch_assoc())**: Вход в цикл, который будет перебирать все товары, полученные из базы данных. Функция `fetch_assoc()` возвращает следующую строку результата в виде ассоциативного массива, где ключи — это имена столбцов.

Генерация элементов карусели

```
echo '<div class="carousel-item ' . ($first ? 'active' : '') . '">';
```

- Каждый элемент карусели оборачивается в `<div class="carousel-item">`. Если `$first` равно `true`, добавляется класс `active`, что делает данный слайд видимым при загрузке.

```
echo '<div class="card mx-auto" style="width: 18rem;">';
```

- Каждый товар представлен в виде карточки Bootstrap. Параметр `style="width: 18rem;"` устанавливает фиксированную ширину карточки, а класс `mx-auto` выравнивает карточку по центру.

```
echo '';
```

- Здесь отображается изображение товара. Путь к изображению берется из базы данных. Атрибут `alt` содержит имя товара для улучшения доступности.

```
echo '<div class="card-body">';  
echo '<h5 class="card-title">' . $product['name'] . '</h5>';  
echo '<p class="card-text">' . $product['description'] . '</p>';  
echo '<p class="card-text"><strong>Цена: ' . $product['price'] . ' </strong></p>';  
echo '<a href="#" class="btn btn-primary">Купить</a>';  
echo '</div>';
```

- Внутри карточки отображается название товара (`card-title`), описание (`card-text`), цена товара, а также кнопка "Купить". Кнопка имеет класс `btn btn-primary`, что стилизует её в основном цвете.

```
$first = false;
```

- После отображения первого элемента переменная `$first` устанавливается в `false`, чтобы последующие элементы не были активными.

```
else {  
    echo '<div class="carousel-item active"><p class="text-center">Нет товаров для  
отображения.</p></div>';  
?>
```

- Если нет товаров, отображается это сообщение об этом блоке, с классом `carousel-item active`. Надпись будет видна при отсутствии других слайдов.

Часть 2. Вывод в виде функций.

Задание.

1. Изучите материал ниже и повторите код в своем проекте.
2. Реализуйте аналогичный вывод для 5 предметных областей из предыдущего задания.

Объяснение изменений.

1. Функция `connectToDatabase`:
 - Создает подключение к базе данных и проверяет наличие ошибок при подключении. Вызывает `die()`, если возникает ошибка.
2. Функция `getProducts`:
 - Отправляет SQL-запрос для получения всех товаров и возвращает результат.
3. Функция `generateCarousel`:
 - Принимает результат выборки и генерирует HTML-код для карусели. Использует функцию `htmlspecialchars()` для предотвращения XSS-уязвимостей, экранируя специальные символы в выводимых данных.
4. Основная логика:
 - Вызов функций в основном блоке кода делает его более чистым и легким для понимания. Также это позволяет легко вносить изменения в логику, связанную с подключением к базе данных и отображением данных.

Пример кода.

1. Подключение к базе.

```
index.php X
1  <?php
2  // Конфигурация базы данных
3  $servername = "localhost"; // замените на свой сервер
4  $username = "root"; // замените на свое имя пользователя
5  $password = ""; // замените на свой пароль
6  $dbname = "online_store"; // замените на свое имя базы данных
7
8  // Функция для подключения к базе данных
9  function connectToDatabase($servername, $username, $password, $dbname) {
10     $conn = new mysqli($servername, $username, $password, $dbname);
11     if ($conn->connect_error) {
12         die("Ошибка подключения: " . $conn->connect_error);
13     }
14     return $conn;
15 }
16
```

2. Получение всех товаров.

```
// Функция для получения товаров из базы данных
function getProducts($conn) {
    $sql = "SELECT * FROM products";
    return $conn->query($sql);
}
```

3. Создание карусели.

```
// Функция для генерации HTML-кода карусели
function generateCarousel($result) {
    if ($result->num_rows > 0) {
        $first = true; // Для установки первого элемента в карусели активным
        while ($product = $result->fetch_assoc()) {
            echo '<div class="carousel-item ' . ($first ? 'active' : '') . '">';
            echo '<div class="card mx-auto" style="width: 18rem;">';
            echo '';
            echo '<div class="card-body">';
            echo '<h5 class="card-title">' . htmlspecialchars($product['name']) . '</h5>';
            echo '<p class="card-text">' . htmlspecialchars($product['description']) . '</p>';
            echo '<p class="card-text"><strong>Цена: ' . htmlspecialchars($product['price']) . ' </strong></p>';
            echo '<a href="#" class="btn btn-primary">Купить</a>';
            echo '</div>';
            echo '</div>';
            $first = false; // Обновляем переменную для следующего элемента
        }
    } else {
        echo '<div class="carousel-item active"><p class="text-center">Нет товаров для отображения.</p></div>';
    }
}
```

4. Подключение и вывод.

```
// Подключение к базе данных
$conn = connectToDatabase($servername, $username, $password, $dbname);

// Получение товаров из базы данных
$result = getProducts($conn);
?>
```

5. Далее в стандартную структуру документа добавляем разметку.

```
<div class="container mt-5">
    <h1 class="text-center mb-4">Карусель товаров</h1>
    <div id="productCarousel" class="carousel slide" data-bs-ride="carousel">
        <div class="carousel-inner">
            <?php generateCarousel($result); ?>
        </div>
        <button class="carousel-control-prev" type="button" data-bs-target="#productCarousel" data-bs-slide="prev">
            <span class="carousel-control-prev-icon" aria-hidden="true"></span>
            <span class="visually-hidden">Previous</span>
        </button>
        <button class="carousel-control-next" type="button" data-bs-target="#productCarousel" data-bs-slide="next">
            <span class="carousel-control-next-icon" aria-hidden="true"></span>
            <span class="visually-hidden">Next</span>
        </button>
    </div>
</div>
```

6. Закрываем подключение.

```
81 <script src="https://cdn.js
82 </body>
83 </html>
84
85 <?php
86 // Закрытие подключения
87 $conn->close();
88 ?>
```