# ЛАБОРАТОРНАЯ РАБОТА №9

# Дисциплина: Программирование на PHP.

# Преподаватель: Рашидинов Дамир Рашидинович.

# Студент: Жандарова Саида.

# Группа: ИС-23-22.

# Дата: 30.10.2025.

# Тема: Клиент–серверные технологии. Обработка HTTP-запросов в PHP.

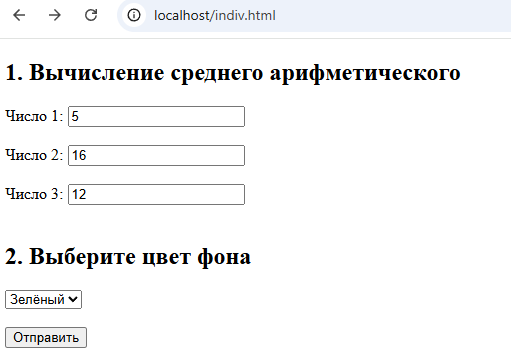
# Цель работы: Научиться принимать и обрабатывать данные, передаваемые с клиента на сервер по протоколу HTTP с использованием методов GET и POST, а также освоить базовые принципы работы с суперглобальными массивами PHP.

# Задание

* Создайте форму для ввода трёх чисел и реализуйте PHP-страницу, которая:
  + вычисляет среднее арифметическое,
  + выводит результат пользователю.
* Создайте форму с выбором цвета (select) и отобразите выбранный цвет на следующей странице (измените цвет фона страницы).
* Используйте $\_SERVER['HTTP\_USER\_AGENT'], чтобы определить, с какого браузера пользователь открыл сайт.

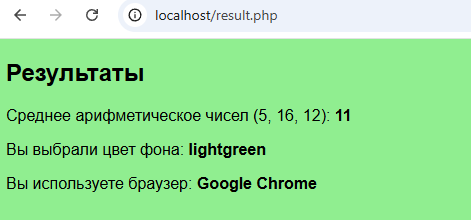
### **Создание формы для ввода данных — indiv.html**

Форма содержит три поля для ввода чисел и выпадающий список для выбора цвета.  
После нажатия на кнопку данные передаются на страницу result.php.



### **Обработка данных — result.php**

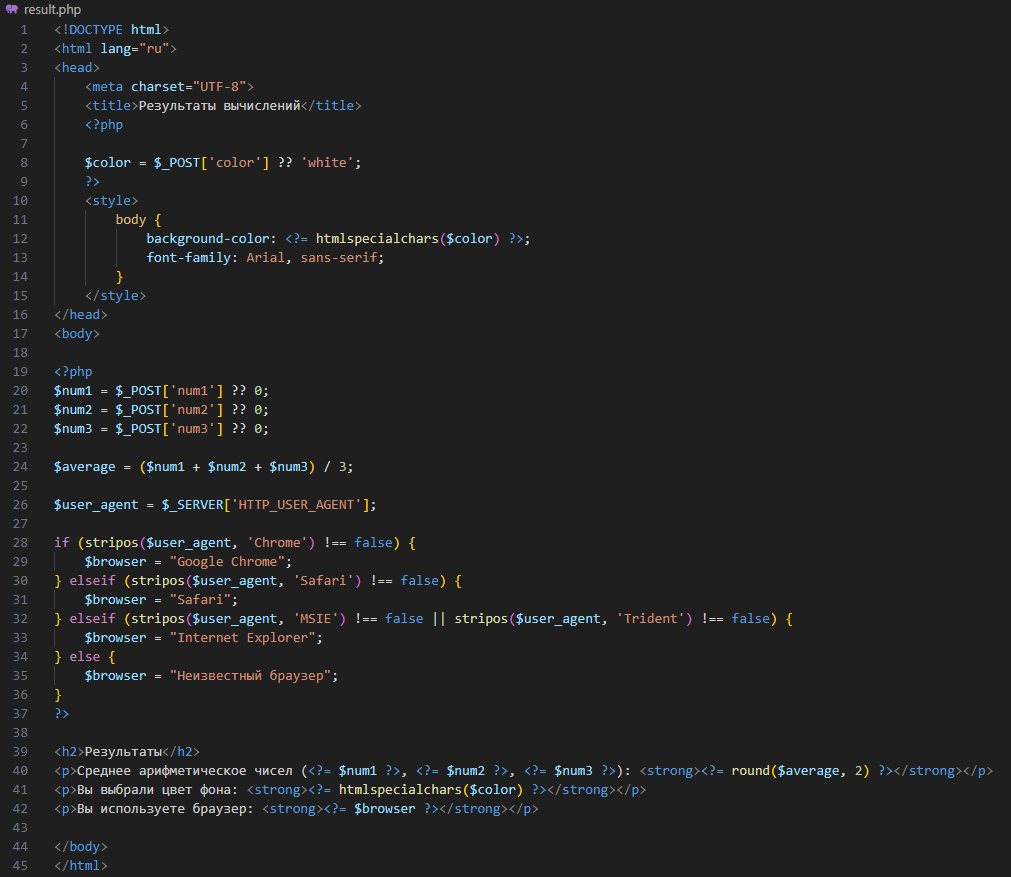
На этой странице вычисляется среднее арифметическое, изменяется цвет фона и выводится информация о браузере пользователя.



**HTML код – indiv.html:**



**PHP код – result.php:**



**Контрольные вопросы**

1. **Чем отличаются методы передачи данных GET и POST**

* Метод GET передаёт данные через адресную строку браузера, добавляя их к URL после знака вопроса. Эти данные видны пользователю и ограничены по длине, поэтому GET не подходит для передачи конфиденциальной информации.  
  Метод POST отправляет данные скрыто — в теле HTTP-запроса. Они не отображаются в адресной строке и могут быть любого объёма. POST чаще используется при отправке форм, регистрации, авторизации и работе с базами данных.

1. **Что содержат массивы $\_GET, $\_POST и $\_SERVER**

* $\_GET — хранит данные, переданные методом GET (из адресной строки).  
  Пример: при переходе по адресу page.php?name=Saida значение $\_GET['name'] будет Saida.  
  $\_POST — содержит данные, отправленные методом POST (например, из формы).  
  Пример: $\_POST['email'] вернёт значение, введённое пользователем в поле формы.  
  $\_SERVER — содержит информацию о сервере и запросе: метод (например, $\_SERVER['REQUEST\_METHOD']), адрес страницы, IP пользователя, данные о браузере ($\_SERVER['HTTP\_USER\_AGENT']) и другие технические сведения.

1. **Почему важно фильтровать данные, полученные от пользователя**

* Потому что пользовательские данные могут содержать вредоносный код. Без фильтрации возможны XSS-атаки или SQL-инъекции, которые могут повредить сайт или украсть данные.  
  Фильтрация защищает систему, удаляя опасные символы и теги, а также предотвращает выполнение непредусмотренного кода.

1. **Для чего используется функция htmlspecialchars()**

* Функция htmlspecialchars() преобразует специальные HTML-символы (<, >, ", ', &) в безопасные текстовые аналоги.  
  Например, <script> превращается в <script>.  
  Это предотвращает выполнение HTML и JavaScript-кода, введённого пользователем, и защищает сайт от XSS-атак.

1. **Что произойдёт, если пользователь отправит запрос на process\_post.php методом GET**

* Если скрипт ожидает данные методом POST, а запрос придёт через GET, массив $\_POST будет пустым.  
  Скрипт не получит нужные значения и не сможет корректно обработать запрос.  
  Если не предусмотрена проверка метода, возможны ошибки или некорректная работа страницы.