# Лабораторная работа №2. Основы PHP, операции с массивами, строками, формы, GET- и POST-запросы.

**Цель работы:** изучение основ языка программирования PHP, работы с переменными, операторами, условными конструкциями, циклами. Изучить принципы работы с формами, GET/POST-запросами.

**Порядок выполнения работы**

1. Изучить темы 3.1–3.7 лекционного материала.

2. Выполнить задание по лабораторной работе.

3. Представить для проверки результат выполнения работы в виде файлов, распечатки и демонстрации выполнения на компьютере

**Рекомендации**

В данной лабораторной работе особое внимание следует уделить оформлению кода (отступы, комментарии, наименования переменных и функций), а также вопросам универсальности алгоритма.

Разработанный алгоритм должен быть независимым от данных, т. е. продолжать корректно функционировать, если входные данные будут заменены на иной произвольный набор значений – как корректный, так и некорректный.

Таким образом, в данной лабораторной работе мы постепенно приближаемся к вопросам обеспечения качества приложений в контексте устойчивости к входным данным.

При именовании переменных и функций, а также при оформлении кода рекомендуется придерживаться следующих правил:

* все имена пишутся в одном стиле;
* имена переменных пишутся в нижнем регистре, состоят из не более чем 2–3 слов, разделённых знаком подчёркивания и представляющих собой существительные или прилагательные;
* имена функций пишутся в нижнем регистре, состоят из не более чем 2–3 слов, разделённых знаком подчёркивания и представляющих собой глаголы или существительные;
* имена переменных и функций являются мнемоничными (отражают смысл хранимых данных или выполняемых действий);
* рекомендуемое количество комментариев – одна строка на 3–5 строк кода программы;
* отступы оформляются знаком табуляции или пятью пробелами;
* варианты поведения программы в условных конструкциях заключаются в операторные скобки даже тогда, когда состоят из одного оператора;
* в формах делать проверку на корректность введенных данных с соответствующей реакцией программы.

**Общее задание:**

На экран вывести ссылки меню с названиями (например "О компании", "Услуги", "Прайс", "Контакты"). При клике по ссылке меняется и остается измененным цвет фона вокруг активной ссылки. Весь код на одной странице. Не использовать javascript. Использовать GET-запросы.

**Задания по вариантам:**

**Вариант 1:** Пользователь вводит в поле формы названия городов через запятую. После отправки отсортируйте города по алфавиту, первые буквы должны быть в верхнем регистре, остальные в нижнем. Если в запросе будут одинаковые города, они не должны повторяться.

**Вариант 2:** сгенерировать HTML-таблицу с количеством строк, которое вводится в поле формы (вводить количество не менее 10). Каждая 5-я строка должна иметь зеленый фон, а фон остальных строк меняется в оттенках серого от #000 до #FFF (шестнадцатеричная система) с одновременным увеличением выбранного компонента цвета в каждой следующей строке.

**Вариант 3:** в произвольном тексте, введенном через форму, каждое третье слово перевести в верхний регистр, каждую третью букву всех слов сделать фиолетовой, подсчитать общее количество встречающихся в тексте букв "о" и "О".

**Вариант 4:** Создайте 2 массива с целыми числами через 2 поля формы, объедините два массива в один (не используя специальные функции PHP типа array\_merge(arr1,arr2)!), Выведите все чётные числа из получившегося массива.

**Вариант 5:** Введите через поле ввода строку состоящую из слов разделенных запятой (например, 'Один, два, три, четыре, пять.'). Первое слово начинается с большой буквы, в конце точка. Расставьте слова в обратном порядке. Выведите строку. Только первое слово новой строки должно начинаться с большой буквы, в конце предложения точка.

**Вариант 6:** Объявить пятимерный массив с целыми числами (не менее 20-ти элементов), вывести массив на экран таким образом, чтобы элементы первого уровня отображались красным цветов, второго – синим, третьего – зелёным, четвёртого – фиолетовым, пятого – жёлтым. После этого произвести сортировку элементов, где четные числа будут отображаться красным, нечетные фиолетовым, нулевые значения отображаться не будут. Данные в исходный массив вводить через поля ввода веб-формы в виде строк, где данные разделены запятыми.

**Вариант 7:** Объявить пятимерный массив с произвольными данными (не менее 20-ти элементов), в этом массиве (программно!) удалить все целые числа, дроби округлить до сотых, все текстовые элементы перевести в верхний регистр. Отсортировать все данные по возрастанию в строковом режиме. Отобразить на экран в табличном виде (или графически) исходный и полученный массивы. Данные в исходный массив вводить через поля ввода веб-формы в виде строк, где данные разделены запятыми.

**Вариант 8:** У нас есть массив часть элементов которого повторяются, необходимо вывести этот массив без дублей при помощи лишь одного цикла foreach без использования функций группировки элементов массива и не нарушая данный массив. Массив получать через веб-форму.

**Вариант 9:** Написать скрипт, который считает сумму цифр числа, введенного пользователем. Также сделать проверку на корректность введения данных пользователем. Число получать через веб-форму.

**Полезные ссылки:**

http://php.net/manual/ru/langref.php

https://metanit.com/web/php/2.1.php

Аврамец Дмитрий Владимирович (ПОИТ)

dva@bsuir.by