

Комплект методических материалов по курсу: РНР часть 1. Основы РНР7

Автор: старший преподаватель Тарасов Алексей Владимирович

Задания на итоговую работу

Темы проектов

1. **Создание корзины книжного интернет магазина (который реализуем по умолчанию)**
2. Реализация ToDo List
3. Разработка блога на PHP
4. Разработка каталога товаров
5. Программирование календаря
6. Создание генератора HTML-форм
7. Создание сервиса для проведения тестов
8. Создание сайта по продаже курсов
9. Разработка сервиса сокращения ссылок
10. Создание сервиса генерации UTM-меток :)

Структура курса

Программа курса

<https://www.specialist.ru/course/rnr1-a>

Основные модули курса

Всего **24 ак. часа**

Модуль 1. Введение в PHP (8 ак. ч.)

Модуль 2. Управляющие конструкции (4 ак. ч.)

Модуль 3. Основы работы с функциями (4 ак. ч.)

Модуль 4. Веб-концепции PHP (4 ак. ч.)

Модуль 5. Практическая работа PHP с базами данных (4 ак. ч.)

Практические работы

Всего **15 практических** работ:

Практическая №1.1: Подготовка рабочего места.

Практическая №1.2: Константы и переменные.

Практическая №1.3: Операторы.

Практическая №1.4: Операторы подключения.

Практическая №2.1: Управляющие конструкции if и switch

Практическая №2.2: Циклические конструкции

Практическая №3.1: Создание собственных функций

Практическая №3.2: Использование встроенных функций

Практическая №4.1: Обработка GET-запросов

Практическая №4.2: Обработка POST-запроса формы

Практическая №4.3: Работа с сессией

Практическая №5.1: Создание базы и подключение

Практическая №5.2: Выборка книг

Практическая №5.3 (если будет время): Реализация входа редактора сайта

Практическая №5.4 (если будет время) Выборка книг в диапазоне цен

Ключевые фразы (хайлайты)

Установка PHP, php циклы, php if, php функции, обработка форм на PHP, PHP сессия, PHP и MySQL, личный кабинет на PHP, корзина на PHP

Модуль 1. Введение в PHP (8 ак. ч.)

Введение в курс

Введение в язык

Что такое PHP?

Зачем использовать PHP?

PHP и сервер

Синтаксис

Операторы

Переменные и константы

Типы данных

Лабораторные по теме

Практические задания

Практическая №1.1: Подготовка рабочего места.

1. Скачайте и установите OpenServer (<https://cloud.mail.ru/public/5otu/iwTH6MGUB>),
2. создайте в папке domains папку с названием вашего сайта, например **first.site**
3. и поместите туда содержимое папки раздачи.
4. Запустите или перезапустите сервер и откройте в браузере <http://first.site>
5. Измените расширение у html-файлов на php.
6. Убедитесь, что `<?php phpinfo() ?>` работает, при вставке на php-страницу и открытии этой страницы через http в браузере

Практическая №1.2: Константы и переменные.

1. В index.php создайте константы для обозначения автора (**AUTHOR**), текущего года (**YEAR**), настроек для базы данных - хоста, логина, пароля и названия базы (**DBHOST**, **DBLOGIN**, **DBPASS**, **DBNAME**), выведите две первые в подвале.
2. Создайте строковые переменные **\$firstName**, **\$lastName**, **\$email**, **\$address**.
3. Создайте переменную **\$successOrder**, со строкой, содержащей переменную **\$firstName**.
4. Создайте массив строк **\$categories** с категориями книг и массив **\$publisher** с названиями издательств, выведите содержимое массивов без использования циклов.
5. Создайте ассоциативный массив **\$book**, описывающий отдельную книгу (должен содержать ключи idbook, title, author, price, description).
6. Если будет время, создайте ассоциативный массив для верхнего меню **\$menu**. Создайте строковую переменную **\$page** со значением index

Практическая №1.3: Операторы.

1. На странице корзины, создайте переменную **\$summa** с результатом сложения нескольких стоимостей книг.
2. Создайте переменную-счётчик с начальным значением равным нулю, а затем увеличьте её значение на единицу.
3. Найдите результат сравнения этой переменной с числом 10.
4. При помощи операции конкатенации, соберите строку склеенную из переменных **\$firstName**, **\$lastName**, **\$email**, **\$address**.
5. При помощи вставки переменных в строку, создайте переменную **\$query** со строкой "INSERT INTO book VALUES (NULL, 'автор', 'название', 123, 'описание', 'категория')".
6. Создайте переменную **\$num** и поместите в неё случайное целое при помощи функции **rand()** в диапазоне от 101 до 999, затем найдите остаток деления на 100.
7. Напишите логическое выражение, которое для любого положительного целого числа не заканчивающегося на числа **от 5 до 20** будет возвращать **true** (используйте оператор получения остатка от деления).
8. Напишите логическое выражение, которое для всех чисел, завершающихся единицей даст **true**.

Практическая №1.4: Операторы подключения.

1. Вынесите константы в отдельный файл config.php и подключите его в index.php при помощи include.
2. Вынесите весь HTML-элемент footer в файл footer.inc.php в папке inc (её нужно создать) и подключите этот файл при помощи include.
3. Переименуйте расширения всех HTML-файлов с .html на .php.

Домашние задания

1. Установите WAMP сборку, если вы работаете под Windows (м.б. вам понадобится LAMP или MAMP, если вы используете другую операционную систему)
2. Узнайте каков объем памяти выделяется PHP-скрипту. Примечание: задание нужно сделать двумя способами — посмотреть в конфигурационном файле или получить параметр из php-функции.
3. Узнайте какова максимальная длительность выполнения PHP-скрипта.
Примечание: см. предыдущую задачу.

4. Установите параметры из первых двух задач в значения, отличные от начальных. Убедитесь, что изменения применены и используются.
5. Вызовите в вашем скрипте функцию `phpinfo()` и поверхностно познакомьтесь с выведенными параметрами
6. Вызовите оператор `echo` с параметром «привет»; убедитесь в использовании нужной кодировки на сервере.
7. Проверьте вашу текущую версию PHP.
8. * Пропишите путь к PHP- интерпретатору в системной переменной `PATH`.
9. * Вызовите PHP-скрипт из консоли.
10. Распечатайте результат вызова функции `date('d-m-Y H:i:s');`
11. Испытайте работу всех PHP-тегов, которые доступны в вашей версии PHP
12. Возьмите один из своих HTML-файлов. Измените его расширение на `.php`, убедитесь в доступности файла на сервере.

Модуль 2. Управляющие конструкции (4 ак. ч.)

Обзор управляющих конструкций

Операторы ветвления

Циклические операторы

Конструкция foreach

Проход по массиву

Лабораторные по теме

Практические задания

Практическая №2.1: Управляющие конструкции if и switch

1. Напишите проверку количества элементов в массивах **\$categories** и **\$publisher**, если они пусты - выведите "элементов нет".
2. При помощи switch, в центральной части index.php, сопоставьте и выведите слова (для index - Главная, basket - Корзина, contacts - Контакты, delivery - Доставка, login - Вход) в заголовках h1.
3. Если останется время, при помощи if сформируйте корректно фразу "N товаров", для любого целого положительного N.

Практическая №2.2: Циклические конструкции

1. При помощи циклов выведите содержимое массивов **\$categories** и **\$publisher**.
2. Если создавали ранее, выведите при помощи цикла ассоциативный массив \$menu.
3. Создайте массив **\$books**, каждый элемент которого будет ассоциативным массивом типа \$book (должен содержать ключи idbook, title, author, price, description), выведите книги на главной странице так, чтобы ссылка "В корзину" содержала путь типа "?add=idbook", где idbook - номер книги.
4. На странице **basket.php** выведите произвольное количество книг в разделе "Корзина"

Домашние задания

1. Напишите скрипт, который будет показывать фон на сайте в зависимости от того, четный или нечетный час. Примечание: попробуйте применить тернарный оператор.
2. Напишите скрипт, который будет выводить предупреждение, если на скрипт выделяется объем памяти меньше 128Mb

3. Напишите скрипт, который будет, в зависимости от дня недели, выводить надпись. Например: сегодня среда. Примечание: используйте оператор switch
4. Напишите PHP цикл, который выводит числа от 1 до 100.
5. Напишите PHP цикл, который выводит числа от 23 до 78.
6. Напишите PHP цикл, который выводит нумерованный список из 10 пунктов.
7. Создайте массив из 100 случайных чисел.
8. Вывести массив из предыдущего задания, при помощи цикла while, а потом при помощи foreach.
9. Создайте массив из 10 строк и выведите их любым циклом внутри HTML-элемента div.
10. * Создайте массив, каждый элемент которого тоже массив с ключами title, description, price. Выведите все элементы этого массива, так, чтобы заголовки были в HTML-элементе h2, описания в p, а цена в гиперссылке.
11. * При выводе элементов из предыдущего задания покрасьте фон элементов ниже определенной цены в отличный от других цвет.
12. Создайте массив из 50 случайных чисел от 0 до 100. Найти все числа меньше 72 и поместить их в отдельный массив
13. Создайте цикл, который выводит числа то 0 до 100 в HTML-элементах div; окраска HTML-элементов должна чередоваться («зебра»).
14. * Создать цикл, которые выводит 20 фрагментов разметки Bootstrap — <http://getbootstrap.com/components/#thumbnails-custom-content>
15. * Создать массив с названиями и адресами ссылок. Вывести этот массив в виде выпадающего меню Bootstrap <http://getbootstrap.com/components/#dropdowns>
16. * Вывести календарь на текущий месяц. Разметку взять тут <http://htmlab.ru/calendar-html/>
17. * Взять текст с Яндекс.Рефератов. Задать ключевое слово и записать в массив все расположения этого ключевого слова.
18. * Разметить все найденные слова из предыдущего задания HTML-элементом mark

Модуль 3. Основы работы с функциями (4 ак. ч.)

Пользовательские функции

Аргументы и возврат значений

Обзор встроенных функций

Функции работы с файлами

Лабораторные по теме

Практические задания

Практическая №3.1: Создание собственных функций

1. Создайте и подключите в **index.php** файл для хранения функций **lib.inc.php**.
2. Создайте функцию `renderCategories($categories)`, которая принимает массив категорий и возвращает разметку для категорий.
3. Вызовите функцию на месте вывода категорий.
4. По аналогии создайте функции `renderPublisher($publisher)` и `renderMenu($menu)` (если будет время).
5. Создайте функцию **`saveOrder($firstName, $lastName, $email, $address)`**, которая принимает и распечатывает строковые переменные (она должна возвращать **true**).
6. Создайте функцию `getBooksByCategories($category)`, которая возвращает тестовый массив типа `$books`.
7. Если будет время, создайте функцию `calcAmount($delta = 0)`, которая прибавляет к статической переменной аргумент `$delta` и возвращает подсчёта.

Практическая №3.2: Использование встроенных функций

1. Используйте функции **trim()** и **strip_tags()** для очистки строк от пробельных символов и тегов при инициализации переменных **\$firstName**, **\$lastName**, **\$email**, **\$address**.
2. Создайте константу **ORDERS** со значением 'orders.txt' и сохраните строку из переменных **\$firstName**, **\$lastName**, **\$email**, **\$address** при помощи функции **file_put_contents()** в файл **orders.txt** (реализация должна быть внутри **saveOrder()**, не забудьте указать номер и время оформления заказа).
3. При помощи функций **file_get_contents()** или **file()** получите и выведите данные из файла (используйте **explode(\$delimiter, \$str)** для разбиения строки в массив). Для форматирования вывода даты, используйте **date()**.
4. Измените код конструкции **switch** так, что при изменении переменной **\$page**, должно меняться содержимое на странице (используйте **include**).
5. Создайте функции **getParam()** и **postParam()** с объявлением обнуляемого типа (<https://www.php.net/manual/ru/functions.returning-values.php#example-160>).

Домашние задания

1. Создать функцию в PHP — **getPlus10()**, которая будет принимать число и распечатывать сумму этого числа и 10.
2. Изменить функцию из предыдущего задания: она должна возвращать сумму, а не сразу выводить.
3. Напишите функцию **pythagoras()**, которая принимает значения двух катетов прямоугольного треугольника и возвращает размер гипотенузы этого прямоугольного треугольника. Примечание: может пригодиться встроенная PHP-функция **pow()**.
4. * Создать функцию, которая находит ипотечный платеж
5. Создайте функцию **col()**, которая распечатает количество переданных аргументов. Например: **col(12,6,123)** должна распечатать число 3.
6. Создайте функцию, которая посчитает среднее значение всех целочисленных аргументов
7. * Напишите функцию, которая принимает неограниченное количество числовых аргументов и строит столбчатую диаграмму. На каждом столбце указываются величины из аргументов. Примечание: это задание сформулировано по аналогии с заданием по созданию функций в JavaScript
<http://htmlab.ru/zadachi-po-javascript-function/>

8. Напишите функцию `op()`, которая принимает три аргумента: `$num1` и `$num2` – числовые, `$operator` – символ, обозначающий операцию. Функция должна возвращать результат выполнения оператора `$operator` над `$num1` и `$num2`
9. * Создайте две функции `add()` и `sub()`, которые принимают пару аргументов и возвращают сумму и разницу соответственно. Создайте функцию `op2()`, которая принимает два числовых аргумента `$num1` и `$num2`, и третий строковый вызываемый аргумент (callable)
10. Создайте функцию, которая при помощи статических переменных будет выполнять основную работу только один раз

Задачи: php строки функции

11. Создайте массив. Объедините все элементы массива в строку и распечатайте её.
12. Возьмите любой из рефератов на сервисе Яндекс.Весна. Выберите любое наиболее часто встречающееся слово. Используя функцию замены подстроки, поменяйте все вхождения отдельного слова, на это же слово, заключённое в HTML-элемент `mark`.
13. Используя функцию удаления HTML и PHP-тегов из строки, выведите на экран строку «<h1>Привет, мир!</h1>». Примечание: строка не должна выглядеть как заголовок первого уровня – все теги должны быть удалены.
14. Используя `stripos()`, найдите во фразе «Ехал Грека через реку» ближайшее вхождение «ре». Работа ведется с однобайтной кодировкой.
15. Найдите длину строки «Ехал Грека через реку» в однобайтной кодировке
16. Найдите длину строки «Ехал Грека через реку» в многобайтной кодировке. Примечание: вам пригодится `mb_strlen()`.
17. Используя встроенную функцию PHP по работе со строками, найдите количество вхождений «ре» в «Ехал Грека через реку»

Задачи: php математические функции

18. Найдите функцию, которая находит длину гипотенузы прямоугольного треугольника. Примечание: в заданиях на пользовательские функции мы писали свой вариант, но в PHP есть готовая функция
19. Используя функцию генерации случайного числа, сгенерируйте целое число в диапазоне от 45 до 234

20. Используя функцию генерации случайного числа, сгенерируйте дробное число в диапазоне от 45 до 234

Задачи: php массивы функции

21. Есть массив `$arr = array("first"=>45, "second"=>76, "third"=>12)`. Используя встроенную в PHP функцию, получить массив, элементами которого являются значениями массива `$arr`.
22. Есть массив `$arr = array("first"=>45, "second"=>76, "third"=>12)`. Используя встроенную в PHP функцию, получить массив, элементами которого являются ключами массива `$arr`
23. Используя встроенные функции, удалите первый элемент массива `$arr = [45, "тест", 100]` и добавьте в конец массива строку "тест2". Примечание: добавление элемента в конец массива также нужно выполнить функцией.
24. * Переверните массив `$arr = array("first"=>45, "second"=>76, "third"=>12)` при помощи встроенной в PHP функции по работе с массивами.
25. Есть массив `$arr = array(45, 76, 12, 12, 45, 12)`. Сколько элементов будет в массиве, который вернет функция `array_unique()`.

Модуль 4. Веб-концепции PHP (4 ак. ч.)

Стандарт HTTP/1.1

\$_GET и \$_POST

Обработка данных формы

Cookies

Сессии

Лабораторные по теме

Практические задания

Практическая №4.1: Обработка GET-запросов

1. Организуйте код так, чтобы при нажатии на ссылку в верхнем меню, в переменную \$page приходил строковый параметр соответствующий названию файла, а в основной части страницы выводилось содержимое этого файла (include). Результатом должна стать работа веб-приложения только через index.php.
2. Поправьте остальные ссылки при необходимости.

Практическая №4.2: Обработка POST-запроса формы

1. Обработайте данные, полученные методом POST из формы оформления заказа и передайте их в функцию **saveOrder()**.
2. Помните о фильтрации.

Практическая №4.3: Работа с сессией

1. Запустите работу с сессией через **session_start()** и организуйте сессионную переменную basket.
2. При нажатии на ссылку "В корзину", добавляйте книгу в сессионную корзину, а на при переходе на страницу корзины - выводите книги корзины.
3. При оформлении заказа почистите сессионную корзину.

Домашние задания

1. Создайте форму из двух полей для ввода логина и пароля на сайте. Получите данные из формы, отфильтруйте их и распечатайте на экране. Примечание: форма должна быть отправлена методом
2. * Создайте форму с одним полем ввода. В PHP создайте массив с названиями городов. Примите данные формы и пройдите по всем элементам массива: если

элементы массива содержат введенный фрагмент, они должны быть распечатаны на экране. Примечание: форма должна быть отправлена методом

3. Создайте форму для нахождения ипотечного аннуитетного платежа.
4. Создайте форму с двумя полями: логина и пароля. При введении логина «john» и пароля «qwerty» методом POST, показывать секретную часть страницы, иначе говорить, что данные введены некорректно.
5. * Создайте форму с многострочным полем ввода. Отправляя форму методом POST найдите часто встречаемости слов из форму и выведите их в порядке убывания частоты встречаемости слов. Примечание: могут пригодиться функции – разбиения строки по символам и нахождения встречаемости элементов в массиве.
6. Создайте массив имен (например, Вася, Коля, Даша и т.д.). Создайте форму с полем ввода, которая позволяет вводить текст с шаблоном @name@ и обрабатывая этот текст заменять шаблон на произвольное имя из массива.
7. * Создайте форму с многострочным полем ввода. Подключите к этому полю WYSIWYG-редактор. Принимая данные формы, очистите все теги, кроме h1-h6, p, section и распечатайте полученный фрагмент.
8. Создайте форму со всеми возможными элементами управления, присвоим им различные имена. Выведите на экран результат отправки формы методом GET. Примечание: все параметры должны быть распечатаны.
9. * Создайте форму, атрибут action которой, должен содержать строку «?param=2» и методом отправки POST. Распечатайте содержимое массивов GET и POST-данных из формы.

Модуль 5. Практическая работа PHP с базами данных (4 ак. ч.)

Введение в реляционные базы данных

Ключевые слова и операторы

PHP функции работы с MySQL

Лабораторные по теме

Практические задания

Практическая №5.1: Создание базы и подключение

1. Создайте базу данных eshop с книгами (структура таблицы book должна содержать поля idbook, author, title, price, description, category).
2. Внесите несколько книг при помощи оператора `INSERT INTO book VALUES (NULL, 'автор', 'название', 123, 'описание', 'категория')`.
3. Выполните подключение к базе в скрипте и убедитесь, что подключение произошло, иначе сообщите о проблеме.

Практическая №5.2: Выборка книг

1. Напишите запрос на выборку всех книг, выполните его и обработайте вывод книг.
2. Перепишите тело функции **getBooksByCategories(\$category)** так, чтобы она возвращала нужные книги из категории.

Практическая №5.3 (если будет время): Реализуйте вход редактора сайта

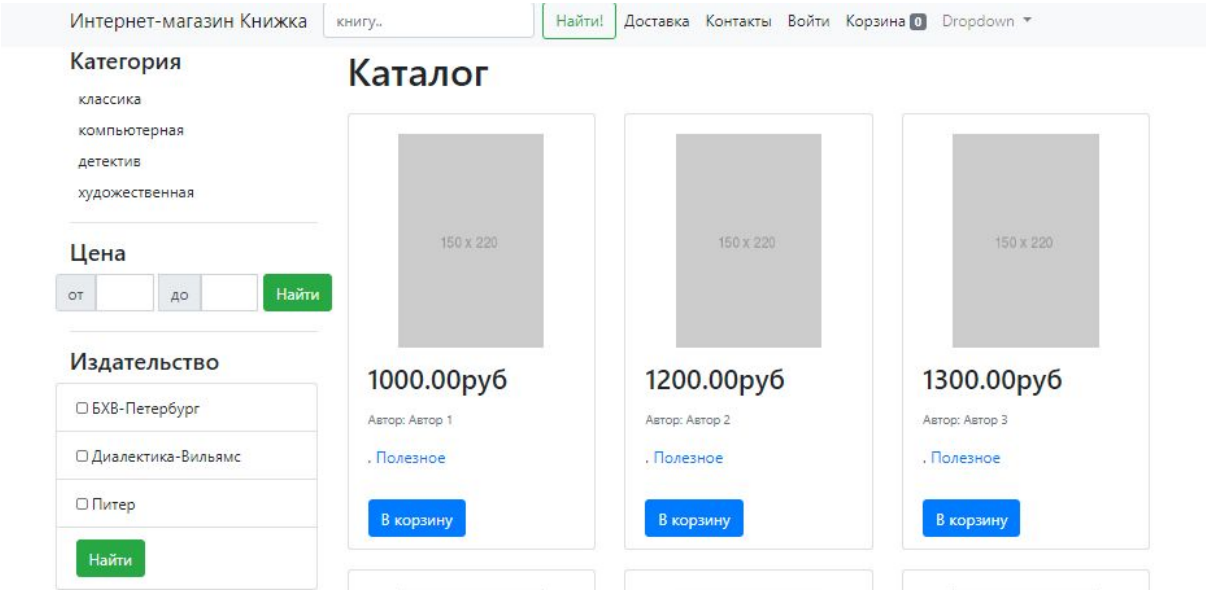
1. Обработайте форму входа на сайт, если логин и пароль введены корректно, создайте сессионную переменную с признаком входа, а в меню выведите пункт "Выйти".
2. Для редактора сайта пропишите форму добавления книг в каталог, сделайте так, чтобы книги попадали в базу.
3. Узнайте, почему здесь нужно бояться SQL-инъекции.

Практическая №5.4: (если будет время) Выборка книг в диапазоне цен

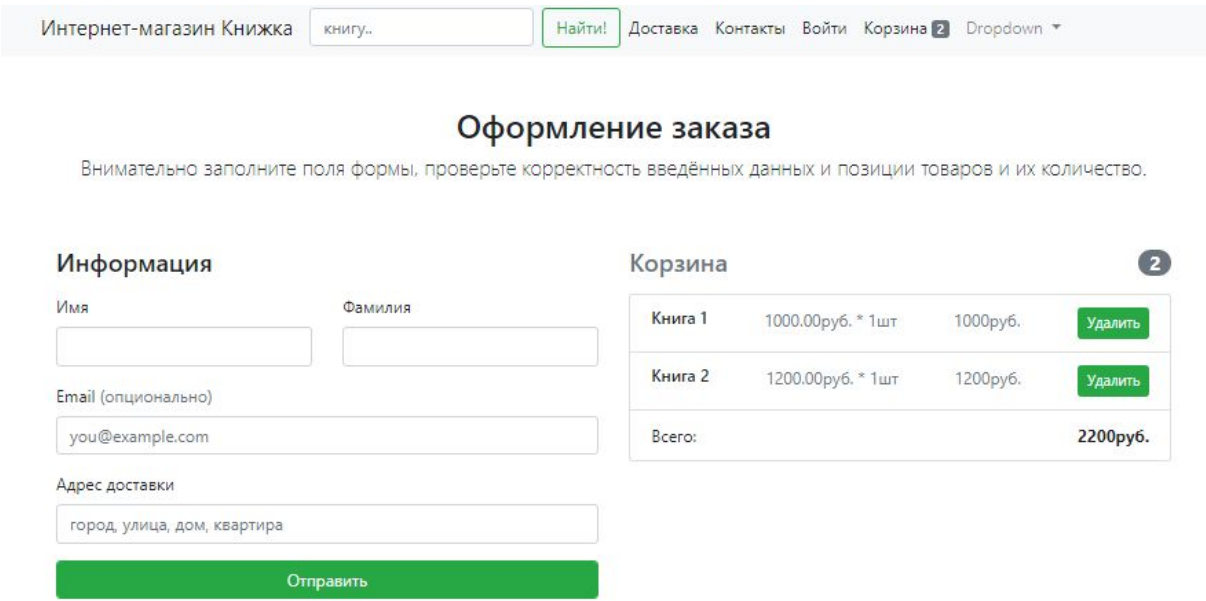
1. Обработайте запрос на получение книг в диапазоне цен, создайте SQL-запрос, выполните его и выведите все книги в нужном диапазоне (**без** учета категории)

Примеры результата итоговой работы, которая пойдёт в портфолио

Пример главной страницы каталога проекта "Создание корзины книжного интернет магазина"



Пример страницы с корзиной проекта "Создание корзины книжного интернет магазина"



Список рекомендуемой литературы/источников

1. <https://www.php.net/manual/ru/> сайт с основной документацией
2. <https://www.php-fig.org/> рекомендации для PHP-разработчиков
3. PHP 7. Котеров Дмитрий Владимирович, Симдянов Игорь Вячеславович. 2019 БХВ-Петербург
4. Разработка веб-приложений с помощью PHP и MySQL. Люк Веллинг, Лора Томсон. 2017. Вильямс, Альфа-книга
5. PHP. Объекты, шаблоны и методики программирования. Зандстра Мэтт. 2019. Диалектика-Вильямс
6. Laravel. Полное руководство. Стаффер Мэтт. 2020. Питер
7. <https://laravel.com/> сайт самого популярного PHP-фреймворка
8. Symfony 5. Быстрый старт. Fabien Potencier. 2020 (<https://leanpub.com/symfony5-bystryj-start>)
9. <https://symfony.com/> сайт одного из самых популярных PHP-фреймворков
10. <https://phpunit.de/> сайт популярного PHP-фреймворка для тестирования
11. <https://www.phpdoc.org/> сайт инструмента для создания документации в php-проекте
12. <https://getcomposer.org/> сайт менеджера зависимостей для PHP
13. <https://packagist.org/> главный репозиторий для менеджера зависимостей Composer
14. <https://www.phpclasses.org/> известный репозиторий PHP-классов

Краткий конспект

Установка/настройка веб-сервера и PHP

Как это работает?

PHP: Hypertext Preprocessor

что такое <http://php.net/manual/ru/intro-what-is.php>

взаимодействие с HTML

сервер

редактор

окружение

возможности <http://php.net/manual/ru/intro-what-cando.php>

Установка веб-сервера

Apache

nginx

сборки (ospanel - он же openserver, xampp, wampserver)

Настройка веб-сервера

Apache, httpd.conf:

LoadModule php5_module "путь/php5apache2_2.dll"

AddType application/x-httpd-php .php

Установка PHP

<http://php.net/manual/ru/install.php>

Настройка PHP

php.ini

short_open_tag = Off

max_execution_time = 30

memory_limit = 128M

error_reporting = E_ALL & ~E_DEPRECATED

display_errors = Off

error_log = ../php_errors.log

post_max_size = 8M

file_uploads = On

Первый скрипт на PHP

```
<!doctype html>
<html>
  <head>
    <title>Т е с т и р у е м  PHP</title>
  </head>
  <body>
    <?php echo '<p>П р и в е т ,  м и р!</p>'; ?>
  </body>
</html>
```

```
<!doctype html>
<html>
  <head>
    <title>PHP Test</title>
  </head>
  <body>
    <p>Hello World</p>
  </body>
</html>
```

<http://php.net/manual/ru/tutorial.firstpage.php>

Основы PHP

Синтаксис

```
<?php
echo "Hello world";
// ... еще код
echo "Последнее выражение";
// Скрипт заканчивается тут без
закрывающего тега PHP
```

Тег <?= доступен всегда, вне зависимости от настройки short_open_tag.

Изолирование от HTML

<p>Это будет проигнорировано PHP и отображено браузером.</p>

```
<?php echo 'А это будет обработано.'; ?>
```

<p>Это тоже будет проигнорировано PHP и отображено браузером.</p>

Продвинутое изолирование

```
<?php if ($expression == true): ?>
```

Это будет отображено, если выражение истинно.

```
<?php else: ?>
```

В ином случае будет отображено это.

```
<?php endif; ?>
```

Переменные

- именованное

Типы

Скалярные типы:

- boolean - логический
- integer - целые числа
- float/double - дробные числа
- string - строчный

Смешанные типы:

- array - массивы

- object - объекты
- callable - вызываемый
- iterable - повторяемый

Специальные типы:

- resource - ресурсы
- NULL

Псевдотипы:

- mixed - смешанный (могут указываться разные типы)
- number - integer или float
- callback (он же callable)

Булев тип (логический)

Простейший тип. boolean выражает истинность значения. Он может быть либо TRUE, либо FALSE (регистронезависимы).

```
<?php
    $foo = True; // присвоить $foo значение TRUE

// == это оператор, который проверяет
// эквивалентность и возвращает boolean
if ($action == "show_version") {
    echo "The version is 1.23";
}

?>
```

Преобразование в булев тип

При преобразовании в boolean, следующие значения рассматриваются как FALSE:

- само значение boolean FALSE
 - integer 0 (ноль)
 - float 0.0 (ноль)
 - пустая строка, и строка "0"
 - массив без элементов
 - особый тип NULL (включая неустановленные переменные)
 - Объекты SimpleXML, созданные из пустых тегов
1. все остальные значения рассматриваются как TRUE
 2. 1 рассматривается как TRUE, как и любое другое ненулевое (отрицательное или положительное) число!

Целые числа

```
<?php
$a = 1234; // десятичное число
$a = -123; // отрицательное число
$a = 0123; // восьмеричное число (эквивалентно 83
в десятичной системе)
$a = 0x1A; // шестнадцатеричное число
(эквивалентно 26 в десятичной системе)
$a = 0b1111111; // двоичное число (эквивалентно 255
в десятичной системе)
?>
```

Дробные числа

```
<?php
$a = 1.234;
$b = 1.2e3;
$c = 7E-10;
?>
```

Математические операторы

```
$a = 10;
$b = 3;
echo "<br>";
echo '$a + $b = ', $a + $b, "<br>";
echo '$a - $b = ', $a - $b, "<br>";
echo '$a * $b = ', $a * $b, "<br>";
echo '$a / $b = ', $a / $b, "<br>";
echo '$a % $b = ', $a % $b, "<br>";
```

```
10 + 3 = 13
10 - 3 = 7
10 * 3 = 30
10 / 3 = 3.3333333333333
10 % 3 = 1
```

```
$a = 10;
```

```

$b = 3;
echo "<br>";
echo "$a + $b = ", $a + $b, "<br>";
echo "$a - $b = ", $a - $b, "<br>";
echo "$a * $b = ", $a * $b, "<br>";
echo "$a / $b = ", $a / $b, "<br>";
echo "$a % $b = ", $a % $b, "<br>";

```

Задача на ипотеку

```

//Найти ипотечный платёж
$S = 5e6;
$p = 12;
$n = 20;

$p = $p / 1200; //переводим доли за месяц
$n = $n * 12;  //находим кол-во месяцев

echo $S * $p / (1 - pow(1+$p, -$n));

```

Строки

Строка (тип string) - это набор символов, где символ - это то же самое, что и байт. Это значит, что PHP поддерживает ровно 256 различных символов, а также то, что в PHP нет встроенной поддержки Unicode.

```

<?php
$name = 'В а с и л и й';
echo "П р и в е т , $name<br>";
echo 'П р и в е т , $name<br>';
echo <<<LABEL
    П р и в е т , м и р ! П р и в е т , $name. <br>
LABEL;
echo <<<'LABEL'
    П р и в е т , м и р ! П р и в е т , $name. <br>
LABEL;
?>

```

Экранирование

```
<?php
    echo "Этот квадрат шириной {$square}м.";
?>
```

Обращение к символу

```
<?php
// Получение первого символа строки
$str = 'This is a test.';
$first = $str[0];

// Получение третьего символа строки
$third = $str[2];

// Получение последнего символа строки
$str = 'This is still a test.';
$last = $str[strlen($str)-1];

// Изменение последнего символа строки
$str = 'Look at the sea';
$str[strlen($str)-1] = 'e';

?>
```

Практическая работа

Нужно в самом начале скрипта index.php создать переменные year и author. Вывести переменные в подвале

Строковый оператор

```
$firstName = "В а с я";
$firstName .= "И в а н о в";
$firstName = $firstName . "И в а н о в";
echo $firstName;
```


Комбинированные операторы

```
$a = 45;  
$a += 10; //$a = $a + 10;  
echo $a;
```

Копирование переменных

```
$a = 5;  
$b = 2;  
$b = $a;  
$a = 3;  
echo "$a $b";//3 5
```

```
$a = 5;  
$b = 2;  
$b = &$a;  
$a = 3;  
echo "$a $b";//3 3
```

Переменные переменной

```
$name = "В а с я";  
$a = "name";  
echo $a,$$a;  
$name
```

Побитовые сдвиги

```
64 32 16 8 4 2 1  
0 1 1 0 0 1 0 -> 32 + 16 + 2 -> 50  
$a << $b
```

```
50 * 2  
50 << 1
```

```
64 32 16 8 4 2 1  
1 1 0 0 1 0 0 -> 64+32+4 -> 100
```

```
50 << 2
```

Выполнение команды

```
$a = `dir`;  
echo $a;
```

Приведение типа

```
$a = (int) "10 ш о к о л а д о к";  
echo $a;
```

Константы

```
define("COMPANY",'С п е ц и а л и с т . р у');  
const PI = 3.1415;  
const MY = "С л о в о л ю б о е";  
echo COMPANY, PI, MY;
```

- **const** должны быть объявлены в верхней области
- **const** всегда регистрозависимая

Тест

<https://goo.gl/forms/PUGjEM1riHcLOrAa2>

Массивы

```
$a = array(23,67,12,9);  
$b = [100,300,200];  
  
$a[10] = 43;  
$a[] = 1;  
$a[] = null;  
  
//echo $a[0]; //и н д е к с 0 - 23  
echo "<pre>", print_r($a), "</pre>";  
//echo "<pre>", var_dump($a), "</pre>";  
echo "<pre>", count($a), "</pre>";
```

Управляющие конструкции

Управляющие конструкции

Операторы инкремента/декремента

Цикл for

Цикл while

Цикл do-while

Итерирование массива

Цикл foreach

Лабораторные работы по теме

IF

```
if ( ВЫРАЖЕНИЕ )  
    ОПЕРАТОР;
```

```
if ( ВЫРАЖЕНИЕ ){  
    ОПЕРАТОР1;  
    ОПЕРАТОР2;  
}
```

IF ELSE

```
if ( ВЫРАЖЕНИЕ ){  
    ОПЕРАТОР1;  
    ОПЕРАТОР2;  
} ELSE {  
    ОПЕРАТОРЫ;  
}
```

Пример работы IF

```
$salary = 80000;  
define("MIDDLE",107000);  
  
if( $salary < MIDDLE ) {  
    echo "У ч и т ь с я...";  
} else {  
    echo "Р а б о т а т ь";  
}  
  
// т е р н а р н ы й   о п е р а т о р  
echo $salary < MIDDLE ? "У ч и т ь с я...":"Р а б о т а т ь";
```

```
<?php if( $salary <= MIDDLE ): ?>
    У ч и т ь с я ...
<?php endif ?>
```

```
<?php if( $salary <= MIDDLE ): ?>
    У ч и т ь с я ...
<?php elseif( true ): ?>
    Р а б о т а т ь
<?php endif ?>
```

SWITCH

```
switch( ВЫРАЖЕНИЕ ){
    case РЕЗУЛЬТАТ1: код: break;
    case РЕЗУЛЬТАТ2: код: break;
    ...
}
```

Пример SWITCH

```
$day = strftime("%u");
switch($day){
    case 1: $menu = "п р о т е и н";           break;
    case 2: $menu = "к у р и н а я   г р у д к а"; break;
    case 3: $menu = "г р е ч к а";           break;
    case 4: $menu = "м о р к о в к а";       break;
    case 5: $menu = "о в с я н к а 1";       break;
    case 6: $menu = "о в с я н к а 2";       break;
    case 7: $menu = "о в с я н к а 3";       break;
    //default: $menu = "с г у щ е н к а";
}
echo "$day $menu";
```

Ещё один пример работы if

```
$h = strftime("%H");
if($h >= 8 && $h <= 19){
    $navbarClass = "navbar-default";
} else {
    $navbarClass = "navbar-inverse";
}
```

```
}
```

```
<nav class="navbar <?= $navbarClass ?> navbar-fixed-top">
```

Формирование строки для корзины

```
// < > == <= >= ===  
/*  
Описать корректно кол-во товаров  
в корзине покупателя  
*/  
$goods = 23;  
$word = "т о в а р о в";  
if( $goods % 100 < 5 or $goods % 100 > 20 ){  
    if( $goods % 10 == 1 ) $word = "т о в а р";  
    if( $goods % 10 > 1 and $goods % 10 < 5 )  
        $word = "т о в а р а";  
}  
echo "В корзине $goods $word";
```

Тест

<https://goo.gl/forms/DRo2oVGNRMjOqUVd2>

FOR

```
for(ИНИЦИАЛИЗАЦИЯ; УСЛОВИЕ;  
ИЗМЕНЕНИЕ_СЧЕТЧИКА){  
    ТЕЛО_ЦИКЛА;  
}
```

1. ИНИЦИАЛИЗАЦИЯ - один раз
2. УСЛОВИЕ
3. ТЕЛО_ЦИКЛА
4. ИЗМЕНЕНИЕ_СЧЕТЧИКА
5. переход на шаг 2.

Пример for

```
for($i = 0; $i < 10; $i++){  
    echo "$i<hr>";  
}
```

Пример формирования HTML-элемента select

```
echo "<select name=w>";  
for($i = 0; $i < 10; $i++){  
    echo "<option>П у н к т $i";  
}  
echo "</select>";  
  
$select = "<select name=w>";  
for($i = 0; $i < 10; $i++){  
    $select .= "<option>П у н к т $i";  
}  
$select .= "</select>";  
  
echo $select;
```

НЕ НАДО ТАК ДЕЛАТЬ :)

```
$i = 0;  
for( ; ; ){  
    echo "$i<hr>";  
    if($i == 10) break;  
    $i++;  
}
```

Массив \$goods

```
$goods = [  
[  
    "title" => "Разработка веб-приложений с помощью  
PHP и MySQL",  
    "description" => "Люк Веллинг, Лаура Томсон",  
    "price" => 1280  
],  
[
```

```

"title" => "PHP. Объекты, шаблоны и методики
программирования",
"description" => "Мэтт Зандстра",
"price" => 2124
],
[
"title" => "PHP5",
"description" => "Дмитрий Котеров, Алексей
Костарев ",
"price" => 543
],
[
"title" => "PHP и MySQL Исчерпывающее руководство",
"description" => "Бретт Маклафлин",
"price" => 638
],
[
"title" => "Head First PHP & MySQL",
"description" => "Lynn Beighley, Michael Morrison",
"price" => 2310
]
]

```

While

```

while(УСЛОВИЕ){
    ТЕЛО_ЦИКЛА;
}

```

Пример работы while

```

$i = 0;
while($i < 7){
    echo $i++, "- ";
}

```

Пример формирования таблицы

```

define("WEEKS",6);

```

```

define("DAY_WEEKS",7);

<?php

echo '<table class="table">';
$i = 0;
$k = 1;
while($i < WEEKS){
    echo "<tr>";
    $j = 0;
    while($j < DAY_WEEKS ){
        echo "<td>";
        if ($k < 32) echo $k++;
        $j++;
    }
    $i++;
}
echo '</table>';
?>

```

Вывести текущий месяц

```

$months = ["1"=>"Я н в а р ь", "2"=>"Ф е в р а л ь", "3"=>"М а р т",
"4"=>"А п р е л ь", "5"=>"М а й", "6"=>"И ю н ь",
"7"=>"И ю л ь", "8"=>"А в г у с т", "9"=>"С е н т я б р ь",
"10"=>"О к т я б р ь", "11"=>"Н о я б р ь", "12"=>"Д е к а б р ь"];

echo '<table class="table">';
echo "<caption>",$months[strftime("%m")], "</caption>";

```

Подсветить текущий день

```

    echo $k == strftime("%e") ? "<td class='bg-warning text-white'>" :
    "<td>";

```

do/while

```

$i = 10;
do{
    echo $i++, "<br>";
}while($i < 7);

```


foreach

```
/*
foreach(МАССИВ as ПЕРЕМЕННАЯ){
    //на каждой итерации ПЕРЕМЕННАЯ
    //принимает значение очередного
    //элемента массива
}
*/
$arr = ["id12"=>"П е т я","id22"=>"К а т я"];
foreach($arr as $value){
    echo $value,"<hr>";
}
```

foreach key value

```
$arr = ["id12"=>"П е т я","id22"=>"К а т я"];
foreach($arr as $key => $value){
    echo "$key = $value<hr>";
}
```

foreach изменение цикла

```
$arr = ["id12"=>"П е т я","id22"=>"К а т я"];
foreach($arr as $key => &$value){
    $value .= "*";
    echo "$key = $value<hr>";
}
print_r($arr);
```

Сферический пример в вакууме

```
$arr = [
    "php"=>"с е р в е р н ы й",
    "js"=>"к л и е н т с к и й",
    "mysql"=>"с у б д"
];
foreach($arr as $key => $value){
```

```
    echo "<p title='$value'>$key";  
}
```

Тест

<https://goo.gl/forms/Czq2xnH9y7UQ2P873>

Пользовательские функции

Описание функции
Вызов функции
Аргументы функции
Аргументы функции по умолчанию
Область видимости переменных
Статические переменные
Возврат значений
Уточнение типа
Лабораторные работы по теме

Функции

```
//описание функции  
function НАЗВАНИЕ_ФУНКЦИИ(АРГУМЕНТ1, АРГУМЕНТ2,...,АРГУМЕНТ3){  
    ТЕЛО ФУНКЦИИ  
}
```

```
//вызов функции  
НАЗВАНИЕ_ФУНКЦИИ();
```

Пример функции

```
function ipoteka($S,$p,$n){  
    $p = $p / 1200;  
    $n = $n * 12;  
    return $S * $p / (1 - pow(1+$p, -$n));  
}
```

```
echo ipoteka(5e6,11,20),"<hr>";  
echo ipoteka(5e6,11,30),"<hr>";  
echo $t = ipoteka(5e6,11,40),"<hr>";
```

Платеж и переплата по ипотеке

```
$S = 5e6;
$n = 20;
$p1= 10;
$p2= 20;
$delta = 2;

for($p = $p1; $p <= $p2; $p += $delta){
    echo "$p% = ";
    echo round(ipoteka($S,$p,$n));
    echo " П е р е п л а т а ",ipoteka($S,$p,$n)*$n*12-$S,"<hr>";
}
```

Вызов строчной переменной как функции

```
$foo = "ipoteka";
echo $foo(5e6,11,20),"<br>";
echo $foo(),"<br>";

$q = 3e6;
echo $foo($q),"<br>";
```

Условное определение функции

```
if( false ){
    function ipoteka($S = 5e6, $p = 11, $n = 20){
        $p = $p / 1200;
        $n = $n * 12;
        return $S * $p / (1 - pow(1+$p, -$n));
    }
}

echo ipoteka(5e6,11,20),"<br>";
```

Функции и пространства имён

```
<?php
namespace Specialist;
```

```
function strftime(){
    echo "п р и в е т>";
}

\Specialist\strftime();
echo \strftime("%e");
?>
```

Статические переменные

```
function test(){
    $a = 0;
    echo ++$a;
}
test();
test();
test();
```

Функция на один раз

```
function test(){
    static $a = 0;
    if($a == 3) return;
    echo ++$a;
}
test();
test();
test();
test();
```

Демонстрация работы static

```
function card($check){
    static $balance = 1e5;
    if($check <= $balance){
        $balance -= $check;
        echo "С п и с а н о $check. ";
        echo "Н а с ч е т у $balance<hr>";
    }
}
```

```

    } else {
        echo "Недостаточно средств. ";
        echo "На счету $balance<hr>";
    }
}

card(30000);
card(69000);
card(100000);

```

Нескалярные аргументы

```

function test($a = [4,6,9]){
    print_r($a);
}
test();

```

Типизация

```

function test(array $a = [4,6,9]){
    print_r($a);
}
test("п р и в е т");

```

Callable

```

function test( callable $foo ){
    echo $foo();
}
test("rand");

```

Тип возврата

```

function test($a, $b): string {
    return $a . $b;
}
echo test("Е х а л", "Г р е к а");

```

Переменное число аргументов

```
function avg() {
    /* echo func_num_args();//ко л-во а р г у м е н т о в
    echo func_get_arg(2);//т р е т и й э л е м е н т
    print_r(func_get_args());//в с е а р г.*/

    $t = 0;
    for($i = 0; $i < func_num_args(); $i++)
        $t += func_get_arg($i);
    return $t / func_num_args();
}
echo avg(3,6,10,7);
```

Построение столбчатой диаграммы

```
/**
 * О т р и с о в к а г р а ф и к а
 * @params integer значение первого столбца
 * @params integer значение второго столбца
 * @return string строка с графиком из div
 */
function graph() {
    $length = func_num_args();
    $str = "";
    for($i = 0; $i < $length; $i++){
        $j = func_get_arg($i);
        $str .= "<div style='background:#ff9;margin:2px;width:".($j*2)."%>";
        $str .= $j."</div>";
    }
    return $str;
}
echo graph(3,6,10,7,5);
```

Аргументы в массив

работает с PHP5.6!

```
function some(...$num){
    print_r($num);
}
some(3,6,10,3);
```

```

/**
 * О т р и с о в к а   г р а ф и к а
 * @params integer значение первого столбца
 * @params integer значение второго столбца
 * @return string строка с графиком из div
 */
function graph(...$num) {
    $length = count($num);
    $str = "";
    for($i = 0; $i < $length; $i++){
        $j = $num[$i];
        $str .= "<div style='background:#ff9;margin:2px;width:".($j*2)."%'>";
        $str .= $j."</div>";
    }
    return $str;
}
echo graph(3,6,10,7,5);

```

а можно и так:

```

$a = [3,6,10,7,5];
echo graph(...$a);

```

Разыменование массива

```

function some(){
    return [34,"п р и в е т"];
}
/*$a = some()[0];
$b = some()[1];
echo $a,$b;*/

list($a,$b) = some();
echo $a,$b;

```

Глобальные переменные

```

$t = 100;
function some(){
    /*global $t;
    echo $t;*/
    echo $GLOBALS["t"];
}

```

```
some();//100
```

Передача параметра по ссылке

```
$a = 45;  
function foo(&$p){  
    $p += 10;  
}  
foo($a);  
echo $a;
```

Тест

<https://goo.gl/forms/jZWJy2VSBspNenQj1>

Задания для практики

1. Написать функцию нахождения ипотечного аннуитетного платежа `ipoteka($S,$p,$n)` и возвращать массив с ключами: `payment` - размер ежемесячного платежа, `overpay` - размер переплаты по кредиту

```
function ipoteka(float $S, float $p, float $n):array{  
    $p = $p / 1200;  
    $n = $n * 12;  
  
    $A = $S * $p / (1 - pow(1 + $p, -$n));  
    return ["payment"=>$A, "overpay" => $A *$n - $S];  
}
```

2. Создать функцию `menuHTML()`, которая на вход получается ассоциативный массив, а возвращает HTML разметку меню на сайте
3. Дан древовидный массив строк. Написать рекурсивную функцию `getListByTree($tree)`, которая вернет HTML разметку в виде многоуровневого HTML списка.

Встроенные функции

Документация PHP

Обзор встроенных функций

Встроенные константы и псевдоконстанты

Суперглобальные переменные

Функции подключения файлов

Лабораторная работа по теме

Документация PHP

- <http://php.net/>
- описание функций
- псевдотипы

Обзор встроенных функций

- математические
- строковые
- для работы с массивами
- дата и время
- другие

Математические функции

```
echo base_convert("ffffff",16,10);  
$t = 36.6;  
echo ceil($t), "<br>";  
echo floor($t), "<br>";  
echo round($t), "<br>";  
echo round($t,1), "<br>";  
  
echo max(34,56,100), "<br>";  
echo max([34,56,100]), "<br>";  
echo min([34,56,100]), "<br>";
```

Строковые функции

```
echo str_repeat(" т ы н ц - ",4), "<br>";
```

```

$str = "  john  ";
echo "<pre>", chop($str), "'<br>";
echo "'", ltrim($str), "'<br>";
echo "'", rtrim($str), "'<br>";
echo "'", trim($str), "'</pre><br>";

echo "<pre>", print_r(explode(" ", "Е х а л  Г р е к а  ч е р е з")),
"</pre>";
explode("/", __DIR__)

echo "<ul><li>", implode("<li>", [45, 67, 100]), "</ul>";
echo md5("qwerty"."#kjh&#");
print "п р и в е т";

$goods = "apples";
$num    = 5;
printf("%s %d", $goods, $num);

$csv = "'col,1','col2','col3'";
$csv = str_getcsv($csv, ',', '"');
echo "<pre>", print_r($csv), "</pre>";

$str = "Е х а л  Г р е к а  ч е р е з  р е к у";
$str = str_replace(' е ', '<mark>е</mark>', $str);
echo "<pre>", $str, "</pre>";

$str = "Е х а л  <b>Г р е к а</b> <mark>ч е р е з</mark> р е к у";
$str = strip_tags($str, '<mark>');
echo "<pre>", $str, "</pre>";

$str = "Е х а л  Г р е к а  ч е р е з  р е к у";
echo strpos($str, "р е", 2), "<br>";
echo strpos($str, "р е", 12), "<br>";

$str = "Е х а л  Г р е к а  ч е р е з  р е к у";
echo substr_count($str, "р е", 0, strlen($str)), "<br>";

$str = "Hello world!";
echo substr($str, 2, 7), "<br>";

```

Функции для работы с массивами

```

$arr = [12, 45, 12, 67, 12, 67, 100];

```

```

echo "<pre>";
print_r(array_count_values($arr));
echo "</pre>";

$arr = [12,45,12,40=>67,12,67,100];
echo "<pre>";
print_r(array_values($arr));
print_r(array_keys($arr));
echo "</pre>";

$arr = [12,45];
echo "<pre>";
echo array_pop($arr), "<br>";
echo array_push($arr,100), "<br>";

echo array_shift($arr), "<br>";
echo array_unshift($arr,500,400), "<br>";
print_r($arr);
echo "</pre>";

$arr = [12,45];
echo "<pre>";
echo in_array(12,$arr), "<br>";
print_r($arr);
echo "</pre>";

function mysort($a,$b){
    return $a < $b ? 1 : -1;
}

$arr = [12,45,23,100,1];
echo "<pre>";
usort($arr, 'mysort');
print_r($arr);
echo "</pre>";

```

Функции даты времени

```

echo time(), "<hr>";
echo date("d-m-Y H:i:s"), "<hr>";
echo date("d-m-Y H:i:s",1612580958), "<hr>";
echo mktime(20,23,12,12,6,17), "<hr>";

```

```

echo "<pre>";
print_r(getdate());
echo "</pre>";

$start = mktime(0,0,0,$month,1,$year);
$wday = getdate($start)["wday"];
$wday = ($wday + 6) % 7;

/*
П В С Ч П С В
1 2 3 4 5 6 0
0 1 2 3 4 5 6

(5 + 6) % 7 -> 4
(1 + 6) % 7 -> 0
*/

$k = 1 - $wday;

function calendar($year, $month){
    $months = [1=>"Я н в а р ь","Ф е в р а л ь","М а р т",
    "А п р е л ь","М а й","И ю н ь","И ю л ь","А в г у с т","С е н т я б р ь",
    "О к т я б р ь","Н о я б р ь","Д е к а б р ь"];

    $days = cal_days_in_month(CAL_GREGORIAN, $month, $year);

    $table = "<table class=\"table\">";
    $table .= "<caption>".$months[$month]. " $year</caption>";
    $k = 1;
    for($i = 0; $i < 6; $i++){
        $table .= "<tr>";
        for($j = 0; $j < 7; $j++){
            $table .= "<td>";
            if($k <= $days)
                $table .= $k;
            $k++;
        }
    }
    return $table .= "</table>";
}

```

Встроенные константы и псевдоконстанты

```
__LINE__  
__FUNCTION__  
__DIR__  
__FILE__  
__NAMESPACE__
```

Суперглобальные переменные

```
$GLOBALS
```

Функции подключения файлов

```
include  
require  
include_once  
require_once
```

include_once

```
<?php  
  
include "my.php";  
include_once "my.php";  
  
foo();  
  
<?php  
  
    function foo(){  
        echo "123";  
    }  
  
?>
```

Тест

<https://goo.gl/forms/x8uyHsYt0jEMUCyw2>

Изучаем HTTP: формы

Стандарт HTTP/1.1

- rfc2616

Заголовки запроса и ответа

GET / HTTP/1.1

Host: www.specialist.ru

Connection: keep-alive

User-Agent: Mozilla/5.0 (Win...

Accept: text/html

Accept-Encoding: gzip, deflate

HTTP/1.1 200 OK

Cache-Control: private

Content-Type: text/html; charset=utf-8

Content-Encoding: gzip

...

Content-Length: 33540

GET /info.php?login=vasya&pass=qwerty HTTP/1.1

Host: phpbase-htmlab.c9users.io

...

POST /info.php HTTP/1.1

Host: phpbase-htmlab.c9users.io

...

login=john&pass=12345

Статус сервера

- 200 Ok

Доступ к заголовкам запроса

- \$_GET
- \$_POST
- \$_REQUEST

Работа с веб-формами

```
<form action="info.php" method="POST">
  <input type="text" name="login"> <br>
  <input type="password" name="pass"> <br>
  <input type="submit">
</form>
```

HTTP-методы запросов

- GET
- POST

Проверка передаваемых значений

```
if( !empty($_GET["login"]) ){
    $login = trim(strip_tags($_GET["login"]));
    echo $login;
}

- строки, числа, флажки
- filter_input
- http://php.net/manual/ru/function.filter-input-array.php
```

Пример обработки формы и сохранения результата в файл

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

    $args = [
        "from" => FILTER_SANITIZE_EMAIL,
        "msg" => FILTER_SANITIZE_STRING,
    ];

    $params = filter_input_array(INPUT_POST,$args);
    $from = $params["from"];
    $msg = $params["msg"];
```

```
$str = "$from|$msg|\r\n";  
    file_put_contents("content.txt",$str,FILE_APPEND);  
  
}
```

Тест

<https://goo.gl/forms/LbO9UrKu2AnwEby82>