**Практическая работа Массивы и циклы в PHP**

# 1 Вывод нескольких элементов на PHP

В проекте есть несколько написанных, но не протестированных фрагментов кода.

Например, есть код, который выводит список брендов на главную страницу. Нам нужно протестировать этот фрагмент. Для этого надо передать программе тестовые данные в новом, пока ещё не знакомом нам формате. Они уже есть в сценарии index.php. Раскомментируем эти данные и посмотрим, появятся ли марки товаров на главной странице.

1. **В файле index.php раскомментируйте код на 6 строке. Проверьте, появились ли бренды на главной странице.**
2. **Удалите фрагмент 'ehm', Да, вместе с запятой.**
3. **А теперь удалите 'tehnodom',**

# 2 Работа с массивами

Нужно вывести на страницу товара product.php список его преимуществ. это может увеличить количество покупок.

Все преимущества уже находятся в сценарии product.php. Они записаны в массив $features. Добавьте эту переменную в шаблон components/product\_info.php и посмотрите, что появится на странице товара.

**На 11 строке добавьте в элемент li массив c преимуществами товара features. Посмотрите, что вывелось в списке преимуществ в браузере.**

# 3 Индексы элементов массива

Чтобы преимущества появились на странице, нужно выводить не сам массив, а его элементы. Для этого нужно использовать *индексы*.

**В файле components/product\_info.php на 11 строке замените содержимое внутри тега li на первый элемент из массива  features[0]**

**Затем на 12 строке добавьте ещё один элемент li со вторым элементом массива features[1]**

**И на 13 строке с третьим элементом массива.**

# 4 Добавление элемента в массив по индексу

Маркетологи попросили добавить ещё несколько преимуществ в массив.  C помощью индексов запишите несколько новых преимуществ в массив features.

**На 15 строке замените значение второго элемента массива $features на 'Вызывает зависть' с помощью индекса. Посмотрите, как изменились преимущества в  браузере.**

**На 16 строке с помощью индекса добавьте в features четвёртый элемент со значением 'Мягкая'. Обратите внимание, что преимущество не вывелось на страницу в  браузере.**

# 5 Цикл while в PHP

Автоматизируйте  добавление преимуществ на страницу с помощью цикла. Напишите цикл while в сценарии product.php. В теле цикла выведем каждый элемент массива features на экран

**В файле product.php на 16 строке создайте переменную-счётчик index со значением 0**

**На 19 строке запишите пустой цикл while () {}**

**В условии цикла запишите условие $index < 6 Так как в нашем массиве 6 элементов.**

**В теле цикла запишите увеличение счётчика цикла на единицу $index = $index + 1;**

**На предыдущей строке, но тоже в теле цикла добавьте вывод на экран**

# 6 Внедрение цикла while в шаблон

Теперь внедрите его в шаблон components/product\_info.php, чтобы на странице товара появились все преимущества из массива features.

Для этого обозначим в шаблоне начало и конец цикла. Затем запишем условие и зададим увеличение индекса элемента с каждой итерацией.

*<?php while (условие): ?> // Начало цикла*

*<?php $index = $index + 1 ?> // Увеличение индекса*

*<?php endwhile; ?> // Конец цикла*

**На 12 строке запишите начало цикла**

**а на 15 строке запишите конец цикла**

**В условии запишите $index < 6**

**Затем на 14 строке добавьте увеличение счётчика**

**И на 13 строке замените код <?= $features[0] ?> на <?= $features[$index] ?>**

# 7Подсчет количества элементов в массиве

Маркетологи *неожиданно* решили увеличить количество преимуществ у товара и прислали список элементов, которые нужно добавить в массив.

На предыдущем шаге мы сами посчитали количество элементов в массиве и записали это число в условие $index < 6. После добавления новых преимуществ количество элементов изменится. Но мы же не будем каждый раз, когда массив меняется, считать самостоятельно число элементов и править условие?

Для автоматического подсчёта элементов в массиве, или длины массива, существует команда **count.**

Посчитайте число элементов в $features и выведите результат на экран. А затем порадуйте маркетологов и добавьте преимущества в массив. Проверьте, как изменится длина массива.

**В файле product.php на 17 строке объявите переменную features\_quantity со значением count($features).**

**Строчкой ниже выведите переменную features\_quantity на экран.**

**На 15 строке вручную добавьте в конец массива новые элементы: , 'Качает не по-детски', 'Ждёт тебя дома'. Посмотрите, что вывелось на экран**

# 8 Добавление count в шаблон

Добавьте count($features) прямо в условие цикла в шаблоне. Проверьте, что при любой длине массива на странице будут отображаться все преимущества товара из этого массива.

**В шаблоне  на 12 строке в условии цикла замените число 6 на count($features)**

**В product.php на 14 строке удалите последний элемент в массиве вместе с запятой , 'Ждёт тебя дома' Убедитесь, что все элементы массива выводятся на страницу.**

**9 Вывод элементов из массива по id**

Мы научились выводить элементы из массивов разной длины на страницу. Проверим, что наш код универсален и сработает с любым товаром интернет-магазина.

В базе данных у каждого товара есть преимущества. Чтобы их получить, нужно использовать команду get\_product\_features и в скобках указать идентификатор товара.

База данных уже подключена в сценарии product.php. Замените значение $features на вызов команды get\_product\_features. Поменяйте номера товаров в адресной строке и посмотрите, как отобразятся преимущества на странице.

**На 14 строке удалите массив и запишите в переменную $features значение get\_product\_features($id).**

**В адресной строке замените 1 на 10 и посмотрите на блок преимуществ в браузере.**

**Вместо 10 напишите в адресной строке 20.**

**Поменяйте число в адресной строке на 7.**

# 10 Ассоциативный массив в PHP

В каталоге должна появиться фильтрация товаров по категориям. При клике на любую из них, на странице должны появляться товары только выбранной категории.

Сначала научимся добавлять один товар на страницу.

Разработчики подготовили для нас массив $item. В нём хранится разная информация о товаре — название, цена, фотография и категория. Только вместо индекса у каждого элемента есть своё название, или **ключ**.

В шаблоне components/product\_list.php добавим в карточку товара название, цену и фотографию товара. Для этого будем обращаться к ключам массива $item.

**В файле components/products\_list.php на 7 и 9 строке замените текст Название товара на  $item['title']**

**На 8 строке замените текст Цена товара на вывод элемента с ключом 'price' из массива $item**

**На 9 строке в атрибуте src замените адрес изображения на вывод элемента с ключом 'img\_url'**

# 11Цикл foreach в PHP

Добавьте в массив $item ещё один массив $colors, в котором хранятся цвета товара. Да, элементами массива могут быть и другие массивы. Они называются вложенными.

Наша задача — доработать отрисовку карточки товара, чтобы на странице появились все цвета из массива $colors.

Выведите цвета товара в шаблон с помощью цикла foreach. Верстальщики подготовили разметку и стили для списка цветов. Каждому цвету соответствует тег <li> с особым классом. Для каждого цвета нужно добавить отдельный <li> в разметку и задать этому элементу нужный класс. Тогда доступные цвета товара появятся на странице. Проверьте.

**В файле components/products\_list.php на 13 строке напишите начало цикла <?php foreach (): ?>**

**а на 15 строке завершение цикла <?php endforeach;?>**

**В круглые скобки запишите выражение $item['colors'] as $color**

**В элементе li в атрибут class внедрите <?= $color ?> вместо слова black**

**В файле catalog.php на 9 строке вручную добавьте в конец массива новые цвета: 'blue' и 'black' Не забудьте про запятые. Убедитесь, что все элементы массива выводятся на страницу.**

# 12Вывод элементов массива на страницу с помощью foreach

Мы научились добавлять один товар на страницу, но в каталоге их обычно больше. Потренируемся выводить несколько товаров.

Добавьте в массив $items еще элементы:

$items = [

0 => [

'img\_url' => 'img/item-mseyulida.jpg',

'title' => 'Лампа Мсеюлида',

'price' => 4000,

'type' => 'lighting',

'colors' => ['orange', 'green', 'yellow', 'blue', 'black']

],

1 => [

'img\_url' => 'img/item-nnulm.jpg',

'title' => 'Рабочий стол Ннулм',

'price' => 5000,

'type' => 'furniture',

'colors' => ['green']

],

2 => [

'img\_url' => 'img/item-tre.jpg',

'title' => 'Набор мебели Тре',

'price' => 20000,

'type' => 'furniture',

'colors' => ['yellow', 'black']

],

3=> [

'img\_url' => 'img/item-badeta.jpg',

'title' => 'Кресло Бадета',

'price' => 500,

'type' => 'furniture',

'colors' => ['blue', 'black']

]

];

Добавьте в шаблон components/products\_list.php цикл foreach и с его помощью выведите на страницу все товары из массива $items.

**В файле components/products\_list.php на 5 строке напишите начало цикла  foreach.  Информация на странице временно исчезнет.**

**На 20 строке напишите конец цикла**

**Внутри условия цикла напишите $items as $item. Товары появятся на странице.**

# 13 Подключение базы данных

Мы научились отображать карточки товара на странице. До этого мы работали с временными данными из массива $items, но ведь у нас есть база c настоящими данными.

Подключите базу данных products\_db.php в сценарий catalog.php. Вместо массива запишите в переменную $items команду get\_products — так получите массив со всеми товарами из базы. Убедитесь, что код по-прежнему работает и товары добавляются на страницу.

**В файле catalog.php на 2 строке подключите базу данных products\_db.php**

**На 6 строке удалите массив со списком товаров из переменной $items.**

**Запишите в переменную $items команду get\_products()**

# 14 Фильтрация товаров

Мы научились добавлять товары из массива на страницу. Теперь мы готовы приступить к фильтрации. Разработчики уже добавили фильтры в сценарий catalog.php и в шаблон components/products\_list.php.

Наша задача — дополнить цикл, который отображает карточки на странице, так, чтобы на страницу добавлялись не все товары из массива, а только товары выбранной категории. Например, пользователь выбирает фильтр «Мебель» и на странице появляются товары только такого типа.

Для начала надо выяснить, какой фильтр сейчас выбран. У каждого фильтра есть свой адрес с параметром запроса product\_type:

catalog.php?product\_type=furniture

Чтобы определить, какой фильтр выбрал пользователь, достаточно получить значение параметра запроса с помощью [уже знакомой](https://htmlacademy.ru/courses/337/run/14) нам команды $\_GET.

У каждого товара в массиве $items есть ключ 'type'. Это и есть категория товара.

Достаточно внутри цикла foreach сравнить активный фильтр из $type со значением 'type' каждого товара в массиве. Если они совпадают, проверка пройдёт и товар появится на странице.

1. **В шаблоне product\_list на 19 строке напишите начало условия <?php if ($item['type'] === $type): ?>**
2. **На 34 строке напишите конец условия <?php endif; ?>**
3. **В адресной строке вместо категории other напишите lighting Убедитесь, что на странице отобразились товары этой категории.**
4. **Теперь на странице каталога нажмите на категорию «Мебель». На странице должны появиться товары только этой категории.**

# 15 Вывод всех товаров на страницу

Доработаем условие так, чтобы при нажатии на «Все товары» на странице появлялись все товары из массива $items, независимо от их типа.

Воспользуемся логическим оператором ИЛИ

1. **На 19 строке добавьте в условие логическое ИЛИ || и сравнение $type === 'all'**
2. **Выберите категорию «Все товары» и убедитесь, что все товары появились на странице.**