**Вопросы по пройденным заданиям**

**1. Почему мы не использовали такую структуру массива**

["john","jane","alex"];

а использовали такую

[

["name"⇒"John"],

["name"⇒"Jane"]

];

Ведь первый вариант тоже будет технически работать одинаково. Тогда зачем мы использовали второй вариант?

**Ответ:** второй вариант является ассоциативным массивом,использующих пару ключ - значение. Этот вариант позволяет нам работая с вёрсткой настроить логику таким путём, что-бы мы видели с какими элементами вёрстки работаем, так сказать установили визуальную связь, строка используемая вместо индекса даёт не только нам, но и другим участникам разработки понять какой элемент относится к другому, мы создаём автоматизацию для простоты, быстроты, удобства пользования. Так же мы легко можем добавить новые элементы в массив, не нарушая структуру.

**2. Мы хотим добавить новый элемент массива. Почему это так важно не менять его структуру при добавлении нового элемента?**

**Ответ:** если поменяем структуру массива, то можем сломать код, который расчитан на определенный порядок, так же значит поменяем логику - нами выстроенная свять от массива к вёрстке и к базе данных будет нарушена, значит придётся тратить время на дополнения кода, что приведёт к его усложнению, к трудности понимания другими лицами. При добавлении нового элемента в массив, лучше использовать ассоциативный массив в виду его расширяемости.

**3. Каким образом формируется массив элементов? Нужно описать процесс формирования массива. Например, смотря на верстку, как вы понимаете какая у массива должна быть структура? Как вы определяете какие будут поля в массиве?**

Ответ: сначала меняем расширение с HTML на PHP, затем смотря на вёрстку мы видем какие элементы вёрстки нам необходимо разбить на блоки, а именно что мы знали какие элементы кода наполняются контентом или могут быть изменены, а какие являются не изменными. Далее создаём массив, вырезаем или копируем контент и вставляем его в массив. Обязатально создаём массив с понятной не только для себя структурой,но и что бы при добавлении нового элемента не нарушалась структура массива. Выделяем блок который будет повторяться и оборачиваем его в цикл

foreach: <?php foreach() : ?> <?php endforeach; ?> - альтернативный синаксис в данном случае является понятнее и убоднее в запоминании. Затем передаём массив циклу и выводим данные, обновляем страницу в браузере и проверяем, если визуально ни чего не поменялось, значит мы успешно провели автоматизацию. Для проверки работы кода можно добавить элемент в массив и обновить страницу, если элемент появился на странице, значит всё правильно.

**4. Почему для вывода списка важно, чтобы структура html блока была единой?**

**Ответ:** это необходимо потому, что структуру html блока можно нарушить, нарушив структуру ломается разметка страницы.

**5. Дан список элементов. Структура html блока у них разная. У одного элемента div style=”padding: 20px;”, у следующего style=”padding: 30px;”, а у третьего вообще нет стилей. Ваши действия?**

foreach($items as $item) {

}

а когдa

foreach($items as $item):

endforeach;

Приведите примеры.

**Ответ:** мне нравится вариант с альтернативным синтаксисом, так как в массив вносятся данные какие нам нужны, затем оборачиваем нужный блок, добавляем скрипты в html разметку там где нам необходимо.

$items = [

[

............

"value" => 20px,

],

[

.............

"value" => 30px,

],

[

..........

],

];