

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. Во втором случае у нас двумерный массив с ассоциативными массивами в качестве элементов, в которых хранятся пары “ключ - значение”. Когда происходит перебор в цикле, то удобнее обращаться к элементу по его ключу. В данном случае \$arr[“name”]. А в первом случае массив одномерный и к элементам нужно обращаться по индексу \$arr[0]. Если будет много элементов, то сложно вспомнить что за элемент, например, \$arr[7]. А по ключу будет сразу понятно.

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

Ответ. Мы обрабатываем все элементы массива в цикле, поэтому все элементы должны быть одинаковой структуры. В противном случае придется делать условные конструкции if else.

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. Определяем какие значения меняются и их место в верстке. Например имена, телефоны, адреса картинок и т.п. Затем составляем ассоциативный массив, у которого ключи будут этими полями (имя, телефон, адрес картинки). А значения - это уже конкретное имя, конкретный телефон и т.п.

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

Ответ. Список можно выводить с помощью цикла и структура цикла должна быть жесткой. Если структура блока не является единой, то браузер не сможет корректно обработать список и его элементы могут отображаться неправильно. Нужно будет придумывать различные if else в цикле.

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?

Ответ. Хранить значения style в таблице БД. Для первого это будет "padding: 20px;", для второго "padding: 30px;", а для третьего "". Таким образом при выводе в цикле значения подставляются в style. Будет так:

```
<div style="padding: 20px;">...</div>
<div style="padding: 30px;">...</div>
<div style="">...</div>
```

6. Два альтернативных вида записи цикла. В первом случае весь код находится внутри тега <?php ... ?>. Во втором случае теги <?php ... ?> вставляются непосредственно в нужные места в верстке.

```
<?php
foreach($items as $item) {
echo "<li>
<a href='\".$item['link'].\"'>\".$item['name'].\"</a></li>";
}
?>
```

```
<?php foreach($items as $item): ?>
<li><a href='<? echo $item["link"] ?>'>
<? echo $item["name"] ?></a></li>
<? endforeach; ?>
```