

## Примеры решения задач

### Задача 1

**Задача.** Дан массив с элементами 'html', 'css', 'php', 'js', 'jq'. С помощью цикла **foreach** выведите эти слова в столбик.

**Решение:** Для начала необходимо создать сам массив. В данном случае мы можем воспользоваться различными способами: объявить через `[ ]` либо просто воспользоваться присваиванием `$arr[ ] = 'html'; $arr[ ] = 'php';` и так далее. Я выбрал **первый способ**, так как он занимает меньше места при записи:

```
<?php
    $arr = ['html', 'css', 'php', 'js', 'jq'];

?>
```

Теперь необходимо воспользоваться циклом **foreach**:

```
<?php
    $arr = ['html', 'css', 'php', 'js', 'jq'];
    foreach ($arr as $elem) {
        echo $elem.'<br>';
    }

?>
```

Как это все работает: в переменной **\$elem** все элементы будут лежать по порядку: сначала 'html', потом 'css' и так далее. Я буду выводить их на экран с помощью **echo**, а после вывода элемента ставить тег **<br>** (он делает перевод строки). Содержимое **\$elem** и тег **<br>** являются строками, поэтому между ними я **ставлю точку** - соединяю тем самым две строки вместе.

### Задача 2

**Задача.** Дан массив с элементами 10, 20, 15, 17, 24, 35. Найдите сумму элементов этого массива. Запишите ее в переменную **\$result**.

**Решение:** Первое решение, которое может прийти в голову новичку - это просуммировать все элементы массива 'руками': `$arr[0]+$arr[1]+...+$arr[5]`.

Однако, этот способ не правильный, так как в случае изменения массива **код придется переписывать** (например, в него добавится еще 10 элементов, или 100, что вообще будет печально). Решение необходимо делать как можно более универсальным.

Поэтому мы должны воспользоваться циклом **foreach** - в этом случае мы не будем привязаны к количеству элементов массива.

Теперь нужно подумать о том, как просуммировать элементы массива в цикле. Для этого я введу переменную **\$result**, к которой при каждом проходе цикла буду

прибавлять ее саму и еще один элемент массива таким образом **`$result = $result + $elem`**:

```
<?php

    $arr = [10, 20, 15, 17, 24, 35];

    $result = 0;
    foreach ($arr as $elem) {
        $result = $result + $elem;
    }
    echo $result; //выведем сумму на экран

?>
```

В строке **`$result = 0`** я зануляю переменную **`$result`**, если этого не сделать - в цикле возникнет ошибка в строке **`$result = $result + $elem`**, так как при первом проходе цикла переменная **`$result`** не будет определена.

Как работает строчка **`$result = $result + $elem`**: при первом проходе в переменной **`$result`** будет 0, в **`$elem`** – 10 (первый элемент массива), тогда в **`$result`** запишется  $0+10 = 10$ .

При втором проходе цикла в **`$result`** лежит уже 10 (значение из прошлого прохода цикла), а в **`$elem`** лежит 20 (следующий элемент массива), тогда в **`$result`** запишется  $10+20=30$ .

При третьем проходе цикла в **`$result`** лежит 30, а в **`$elem`** – 15, в **`$result`** запишется  $30+15 = 45$ . И так далее пока цикл не закончится.

Строчку **`$result = $result + $elem`** можно записать короче: **`$result += $elem`**.

## Задача 3

**Задача.** Выведите столбец чисел от 1 до 100.

**Решение:** Задачу можно решить как циклом **`for`**, так и циклом **`while`**. Вначале решим через цикл **`while`**:

```
<?php

    $i = 1; //счетчик цикла устанавливаем в 1
    while ($i <= 100) {
        echo $i.'<br>'; //выведет 1,2... 100
        /*
            С помощью оператора ++ увеличиваем $i
            на единицу при каждом проходе цикла:
        */
        $i++;
    }

?>
```

Решение через цикл **`for`**:

```

<?php
    /*
        В начале цикла $i будет равно 1,
        цикл будет выполняться пока $i <= 100,
        после каждого прохода к $i прибавляется единица:
    */
    for ($i = 1; $i <= 100; $i++) {
        echo $i.'<br>'; //выведет 1,2... 100
    }
?>

```

## Задачи для решения

### Работа с foreach

1. Дан массив с элементами 'html', 'css', 'php', 'js', 'jq'. С помощью цикла **foreach** выведите эти слова в столбик.
2. Дан массив с элементами 1, 2, 3, 4, 5. С помощью цикла **foreach** найдите **сумму** элементов этого массива. Запишите ее в переменную **\$result**.
3. Дан массив с элементами 1, 2, 3, 4, 5. С помощью цикла **foreach** найдите **сумму квадратов** элементов этого массива. Результат запишите переменную **\$result**.

### Работа с ключами

4. Дан массив **\$arr**. С помощью цикла **foreach** выведите на экран столбец ключей и элементов в формате 'green - зеленый'.

```
$arr = ['green'=>'зеленый', 'red'=>'красный', 'blue'=>'голубой'];
```

5. Дан массив **\$arr** с ключами 'Коля', 'Вася', 'Петя' и с элементами '200', '300', '400'. С помощью цикла **foreach** выведите на экран столбец строк такого формата: 'Коля - зарплата 200 долларов.'.

### Циклы while и for

*Решите эти задачи сначала через цикл while, а затем через цикл for.*

6. Выведите столбец чисел от 1 до 100.
7. Выведите столбец чисел от 11 до 33.
8. Выведите столбец четных чисел в промежутке от 0 до 100.
9. С помощью цикла найдите сумму чисел от 1 до 100.

## Задачи

10. Дан массив с элементами **2, 5, 9, 15, 0, 4**. С помощью цикла **foreach** и оператора **if** выведите на экран столбец тех элементов массива, которые больше 3-х, но меньше 10.

11. Дан массив с числами. Числа могут быть положительными и отрицательными. Найдите сумму положительных элементов этого массива.

12. Дан массив с элементами **1, 2, 5, 9, 4, 13, 4, 10**. С помощью цикла **foreach** и оператора **if** проверьте есть ли в массиве элемент со значением, равным 4. Если есть - выведите на экран 'Есть!' и выйдите из цикла. Если нет - ничего делать не надо.

13. Дан массив числами, например: ['10', '20', '30', '50', '235', '3000']. Выведите на экран только те числа из массива, которые начинаются на цифру **1, 2** или **5**.

14. Дан массив с элементами **1, 2, 3, 4, 5, 6, 7, 8, 9**. С помощью цикла **foreach** создайте строку **'-1-2-3-4-5-6-7-8-9-'**.

15. Составьте массив дней недели. С помощью цикла **foreach** выведите все дни недели, а выходные дни выведите жирным.

16. Составьте массив дней недели. С помощью цикла **foreach** выведите все дни недели, а текущий день выведите *курсивом*. Текущий день должен храниться в переменной **\$day**.

## Задачи посложнее

17. С помощью цикла **for** заполните массив числами от 1 до 100. То есть у вас должен получится массив **[1, 2, 3... 100]**.

18. Дан массив **\$arr**. С помощью цикла **foreach** запишите английские названия в массив **\$en**, а русские - в массив **\$ru**.

```
$arr = [ 'green'=>'зеленый', 'red'=>'красный', 'blue'=>'голубой' ];
```

19. Дано число **\$num=1000**. Делите его на 2 столько раз, пока результат деления не станет меньше 50. Какое число получится? Посчитайте количество итераций, необходимых для этого (*итерация* - это проход цикла). Решите задачу сначала через цикл **while**, а потом через цикл **for**.