

1. Владимиров Иван Юрьевич
2. 31 год
3. Балашиха
4. Закончил МГУ
5. C++, Python, PHP
6. 3 из 5
7. timerke@mail.ru
8. +79067820791
9. Будние дни, 9-18
10. hh.ru

Тест

1. В) !==

В PHP есть два оператора равенства == и ===. Оператор == возвращает True, если значения сравниваемых переменных равны. Оператор === возвращает True, если значения и типы сравниваемых переменных равны. Для данных операторов есть обратные им операторы: != и !==. Оператор != возвращает True, если значения сравниваемых переменных не равны. Оператор !== возвращает True, если значения сравниваемых переменных не равны или если сравниваемые переменные разных типов.

2. Г) Переменная \$x=5

Мы используем одинарные кавычки, когда хотим вывести в неизменном виде то, что содержится внутри кавычек. Когда мы хотим вывести обработанную строку, используем двойные кавычки. Например, скрипт:

```
<?php
$x=5;
echo "Переменная $x=5";
?>
```

выведет на экран результат: Переменная 5=5

3. В) Другой ответ

Заданный скрипт:

```
<?php
$myImportantArray = [1, 2, 3, 4, 5];
foreach ($myImportantArray as $someBeautifulValue);
{
    echo $someBeautifulValue;
}
echo "end";
?>
```

можно из-за точки с запятой на строке 3 записать так:

```
<?php
$myImportantArray = [1, 2, 3, 4, 5];
foreach ($myImportantArray as $someBeautifulValue)
{
}
{
    echo $someBeautifulValue;
}
echo "end";
?>
```

В цикле `foreach` просто происходит перебор всех значений массива. В результате переменная `$someBeautifulValue` принимает после этого цикла значение последнего элемента в массиве, то есть 5. Это значение выводится в следующем блоке, выделенном фигурными скобками, а после уже выводится на экран строка “end”. В результате выполнения скрипта получим:

5end

4. В) 23default

Переменная `$s` равна 2. Следовательно, в операторе `switch` выполняются команды в блоке `case 2`. PHP продолжает исполнять операторы до конца

блока `switch` либо до тех пор, пока не встретит оператор `break`. Поскольку в скрипте нет операторов `break`, будут выполнены команды:

```
echo 2;  
echo 3;  
echo 'default';
```

5. В) true

В данном случае при сравнении оба сравниваемых значения переводятся в `boolean` значение `false`. В результате получаем:

```
if (false==false)
```

И выполняется инструкция:

```
echo "true";
```

6. Г) 13

Если значение параметра `$i` четное, то остаток от деления на 2 равен нулю. Следовательно, для четных значений параметра `$i` в цикле `for` будет выполняться инструкция `continue`, благодаря которой будет пропускаться оставшаяся часть инструкций в цикле `for`, то есть не будет выполняться `echo $i`.

7. А) key1...key2...

В операторе `if` сравнивается 0 и преобразованное в число значение параметра `$value`. Параметр `$value` принимает при выполнении цикла `foreach` строковые значения `'value1'` и `'value2'`. Данные строки не начинаются с цифр, а потому они преобразуются просто в 0.

8. Г) header("Location: http://google.com")

`header` используется для отправки HTTP-заголовка. Функцию `header` можно вызывать только если клиенту еще не передавались данные, то есть она должна идти первой в выводе, перед ее вызовом не должно быть никаких HTML-тегов, пустых строк и т.п.

9. В) 05

При объявлении функции `myfunc` мы указываем, что у ее аргумента `$x` есть значение по умолчанию, равное 0. Если при вызове функции `myfunc` ей не передается никакое значение, как на строке 6:

```
myfunc();
```

то аргумент `$x` в теле функции принимает значение по умолчанию, то есть 0. Если же при вызове функции `myfunc` ей передается какое-то значение, как на строке 7:

```
myfunc(5);
```

то аргумент `$x` в теле функции принимает переданное значение, то есть 5.

10. В) `$a='строка';`

Мы хотим в переменную `$a` записать значение типа строка. Имена переменных в PHP начинаются со знака `$`. Строковое значение можно записать либо с помощью одинарных, либо с помощью двойных кавычек. Чтобы присвоить переменной некое значение, нужно использовать оператор присваивания `=`.

11. А) `array_unshift()`

Функция `array_flip` меняет местами ключи с их значениями в массиве. Функция `array_intersect` вычисляет сходжение массивов. Функция `array_push` добавляет один или несколько элементов в конец массива.

12. Б) 7

Нам нужно посчитать значение выражения:

```
getValue1(getValue2(getValue2(getValue1(getValue1($a)))))
```

Отметим, что в функции `getValue1` используется постфиксный инкремент:

```
$a++;
```

то есть сначала возвращается значение аргумента `$a` этой функции, а потом его значение увеличивается на 1. То есть функция `getValue1` возвращает значение, переданное ей в качестве аргумента.

В функции `getValue2` используется префиксный инкремент:

```
++$a;
```

то есть сначала значение аргумента \$a этой функции увеличивается на 1, а потом возвращается. То есть функция `getValue2` возвращает увеличенное на 1 значение, переданное ей в качестве аргумента.

Сначала будет выполнена функция, покрашенная красным цветом:

```
getValue1 (getValue2 (getValue2 (getValue1 (getValue1 (5)))))
```

Здесь мы учли, что параметру \$a присвоено значение 5. Потому первая функция вернет значение 5.

Теперь посчитаем аналогично, что вернет функция, покрашенная синим цветом:

```
getValue1 (getValue2 (getValue2 (getValue1 (5))))
```

Это опять `getValue1`, а потому она также вернет значение 5.

Теперь посчитаем, что вернет функция, покрашенная зеленым цветом:

```
getValue1 (getValue2 (getValue2 (5)))
```

Это функция `getValue2`, а потому она вернет значение 6.

Теперь посчитаем аналогично, что вернет функция, покрашенная оранжевым цветом:

```
getValue1 (getValue2 (6))
```

Это функция `getValue2`, а потому она вернет значение 7.

Наконец, осталась последняя функция:

```
getValue1 (7)
```

Это функция `getValue1`, а потому она вернет значение 7, которое и будет передано в качестве аргумента функции `echo`.

13. A) `bool(true)`

Функция `var_dump` выводит информацию о переменной. А функция `in_array` проверяет, присутствует ли в массиве значение. В нашем случае используется функция `in_array` без третьего параметра, то есть без строгой проверки соответствия типов элементов массива и параметра, который

нужно найти в массиве. Потому элементы массива и параметр, который нужно найти в этом массиве, переводятся к одному boolean типу:

[0, 1] -> [false, true]

'tree' -> true

То есть функция `in_array` вернет значение true. Функция `var_dump` выведет тип полученного аргумента (то есть bool) и его значение (то есть true).

14. Все

Функция `substr($string, $start, $length)` возвращает подстроку строки `string`, начинающейся с `start` символа по счету и длиной `length` символов. В первом варианте получаем:

```
echo substr($line, 3, 3);
```

То есть на экран выводим '1' под индексом 3 (начальный индекс в строке 0) и еще два символа после нее (то есть '2' и '3').

Если значение параметра `$start` в функции `substr($string, $start)` отрицательное, возвращаемая подстрока начинается с позиции, отстоящей на `$start` символов от конца строки `$string`. Во втором варианте имеем:

```
echo substr($line, -3);
```

То есть начальный символ возвращаемой подстроки опять тот же, что и в первом варианте: '1' под индексом 3. Поскольку параметр `$length` во втором варианте опущен, то будет возвращена подстрока, начинающаяся с позиции, указанной параметром `$start` и длящейся до конца строки. То есть опять подстрока '123'.

Функция `str_replace($search, $replace, $subject)` возвращает строку или массив, в котором все вхождения `$search` в `$subject` заменены на `$replace`.

То есть в третьем варианте функция

```
str_replace('123', '', $line)
```

возвращает строку '321'. Функция `strrev` переворачивает строку задом наперед. То есть из строки '321' получается опять '123'.

Наконец, в четвертом варианте написана конкатенация символов, стоящих под индексами 3, 4 и 0 в строке \$line: \$line[3].\$line[4].\$line[0]= '1'. '2'. '3'= '123'.

15. Д) Ошибка выполнения

У нас имеется массив \$items, элементами которого являются массивы ['sort'=>1], ['sort'=>3], ['sort'=>5], ['name'=>'item_1'], [['sort'=>8], ['sort'=>10]]. Все эти элементы передаются в качестве значений переменной \$item в цикле foreach. При обращении по ключу \$item['sort'] массивы ['sort'=>1], ['sort'=>3], ['sort'=>5] выдают значения 1, 3 и 5 соответственно. А вот оставшиеся два массива не имеют ключей 'sort' в результате чего появляется ошибка.

16.

Вот скрипт решения с комментариями:

```
<?php
/* Массив с данными авторов */
$authors = [
    ['name' => 'Гоголь', 'email' => 'gogol@mail.ru', 'birth_year' => 1809],
    ['name' => 'Пушкин', 'email' => 'pushkin@mail.ru', 'birth_year' => 1799],
    ['name' => 'Чехов', 'email' => 'chekhov@mail.ru', 'birth_year' => 1860]
];

/* Массив с данными о книгах */
$books = [
    ['author' => &$authors[0], 'book_name' => 'Мертвые души', 'year' =>
1842],
    ['author' => &$authors[1], 'book_name' => 'Капитанская дочка', 'year' =>
1836],
    ['author' => &$authors[0], 'book_name' => 'Ревизор', 'year' => 1836],
    ['author' => &$authors[1], 'book_name' => 'Пиковая дама', 'year' =>
1834],
    ['author' => &$authors[2], 'book_name' => 'Вишневый сад', 'year' => 1903]
];
```

```

/* Выводим информацию об авторах */
foreach ($authors as $author) {
    echo $author['name'] . ' - <a href="">' . $author['email'] . '</a> - ' .
    $author['birth_year'] . '<br/>';
}
echo '<br/>';

/* Выводим информацию о книгах */
foreach ($books as $book) {
    echo $book['book_name'] . ' - ' . $book['author']['name'] . ' - ' .
    $book['year'] . '<br/>';
}
?>

```

17.

Вот скрипт решения с комментариями:

```

<?php
$items = ['where', 'is', 'my', 'money'];
$sum = 0;

/* Функция, которая выполняется, когда индекс в цикле делится на 3 */
function divided_by_3($i, $items)
{
    $n = count($items) - 1;
    $random_index = rand(0, $n);
    echo 'i= ' . $i . ': ' . $items[$random_index] . '<br/>';
}

/* Цикл от 0 до 100 */
for ($i = 0; $i <= 100; $i++) {
    if ($i % 3 == 0 && $i % 5 == 0) {
        /* Перемешиваем массив $items */
        shuffle($items);
    } elseif ($i % 3 == 0) {
        divided_by_3($i, $items);
    } elseif ($i % 5 == 0) {
        /* Прибавляем индекс к $sum */
        $sum += $i;
    }
}

```



```
    }  
}  
  
/* Выводим массив $items */  
foreach ($items as $item) {  
    echo $item . '<br/>';  
}  
?>
```