

1. Ключевое слово `global` перед определением переменной «помещает» эту переменную в глобальную область видимости.
2. `$_SERVER`, `$GLOBALS`, `$_GET`, `$_PUT`, `$_REQUEST`, `$_FILE`, `$_COOKIES`, `$_SESSION`
3. Закрывающий дескриптор нужен в файле со смешанным содержимым, напр., когда php-код внедрён в html код. Необходим дескриптор для того, чтобы интерпретатор php понимал, что он должен обрабатывать. Всё, что вне закрывающего/открывающего тегов php интерпретатор просто выводит как есть.
4. 161 – первой выведется значение переменной, заданное при объявлении, вторым выведется результат работы функции, третьей снова выведется значение переменной. Функция не меняет значение внешней для этой функции переменной, т.к. она работает в своём контексте.
5. 012 – первый вызов функции – вывод статической переменной, после инкремент, второй вызов – вывод статической переменной, после инкремент, третий – вывод статической переменной, после инкремент. Статическая переменная сохраняется после завершения функции, и может быть обработана в дальнейшем этой же функцией.
6. Применить к массиву функцию `array_reverse()`
7. Применить к строке функцию `strrev()`
8. `Two` – оператор присвоения в данном случае возвращает 0, т.е. проверка `if` получит `false` (0 будет приведен к `false`), а значит выполнится блок `else`

```
9. <?php
function genRandNumbers() {
    $a = rand(1, 10);
    $b = rand(1, 10);
    $c = rand(1, 10);
    if (($a + $b + $c) < 15)
        genRandNumbers();
    else
        echo "$a $b $c";
}
genRandNumbers();
```

10. Данный код выведет синтаксическую ошибку)). Если её исправить, то будет 45, проверил интерпретатором. Сам я ожидал 46... Так и не понял, почему 45 =D
11. 12 – первый символ строки "1", затем символу строки с индексом 1 (т.е. второму) присваиваем "2". Итог – строка "12".