



нетология  
центр онлайн-образования

# СТРОКИ, МАССИВЫ, ОБЪЕКТЫ



РОМАН ДАВЫДОВ / PHP-РАЗРАБОТЧИК



# РОМАН ДАВЫДОВ

PHP-РАЗРАБОТЧИК



[warlight@mail.ru](mailto:warlight@mail.ru)



[fb.me/rsdavydov](https://fb.me/rsdavydov)

---

# В уроке

- Строки
- Списки
- Массивы
- Ресурсы
- Объекты
- NULL

---

# СТРОКИ



---

# Объявление строк

`$a = 'Строка в одинарных кавычках';`

`$b = "В двойных кавычках появляется возможность использовать переменные вот так: $a";`

`$c = "Или так: {$b}";`

`$d = <<<PHP`

Длинную строку удобно объявить вот таким способом, особенно, если в ней используются оба типа кавычек  
`PHP;`

---

# Операции над строками

- Склеивание (конкатенация)  
`$a = "38" . 'попугаев'; // 38попугаев`

- Сравнение  
`$result = "a" > "Я"; // true`  
`$result = "a" > "я"; // false`

При нестрогом (==) сравнении интерпретатор пытается привести строки к числам!

- При произведении арифметических операций первые символы строки будут приведены, по возможности, к числу, остальные - отброшены.

```
$sum = "38попугаев" + "1 крылышко  
3333"; // 39
```



---

# Функции для работы со строками

- Вырезать подстроку из строки  
`substr ($string, $start[, $length])`
- Замена одной последовательности символов на другую  
`str_replace($search, $replace, $string)`
- Поиск вхождения подстроки в строку  
`strpos($haystack, $needle[, $offset])`
- Длина строки  
`strlen($string)`
- Очистить строку от html тегов  
`strip_tags($string, $allowable_tags)`
- MD5-хеш строки  
`md5($string)`
- Полный список: <http://php.net/manual/ru/ref.strings.php>

# Списки





---

# Список - это ловкий способ сделать несколько присваиваний за один раз

// получение строки из БД в массив

```
$row = mysqli_fetch_row($link,$result);  
list($id, $title) = $row; // $id = $row(0), $title = $row(1)
```

---

# Массивы



variable



array

---

# Объявление массива

"Старый" способ

```
$eggs = array("egg1", "egg2", "egg3"); //  
$eggs[0] == "egg1", $eggs[1] ==  
"egg2", etc...
```

"Новый" способ

```
$data = [  
10, 11, 3, 14, 8  
];  
// $data[0] == 10, $data[1] == 11,  
etc...
```

С принудительным присваиванием ключей:

```
$eggs = array ( 1 => "egg1", 2 => "egg2", 40 => "o_O");
```

```
$data = [  
100 => 2, 200 => 4  
];
```

Ассоциативный массив:

```
$array = array(  
"name" => "Виктор",  
"city" => "Москва"  
);  
echo $array['name']; // Виктор
```

---

# Работа с массивом

Добавление элементов в массив

```
$eggs[] = "новое яйцо"; // ключ  
назначается автоматически
```

```
$eggs[99] = "уейн грецки"; //  
принудительное создание элемента
```

Удаление элемента из массива:

```
unset($eggs[99]); // завершил карьеру
```

Обработка в цикле:

// важно!! работает в том случае, если все  
индексы назначены автоматически и идут от  
нуля (поведение по умолчанию)

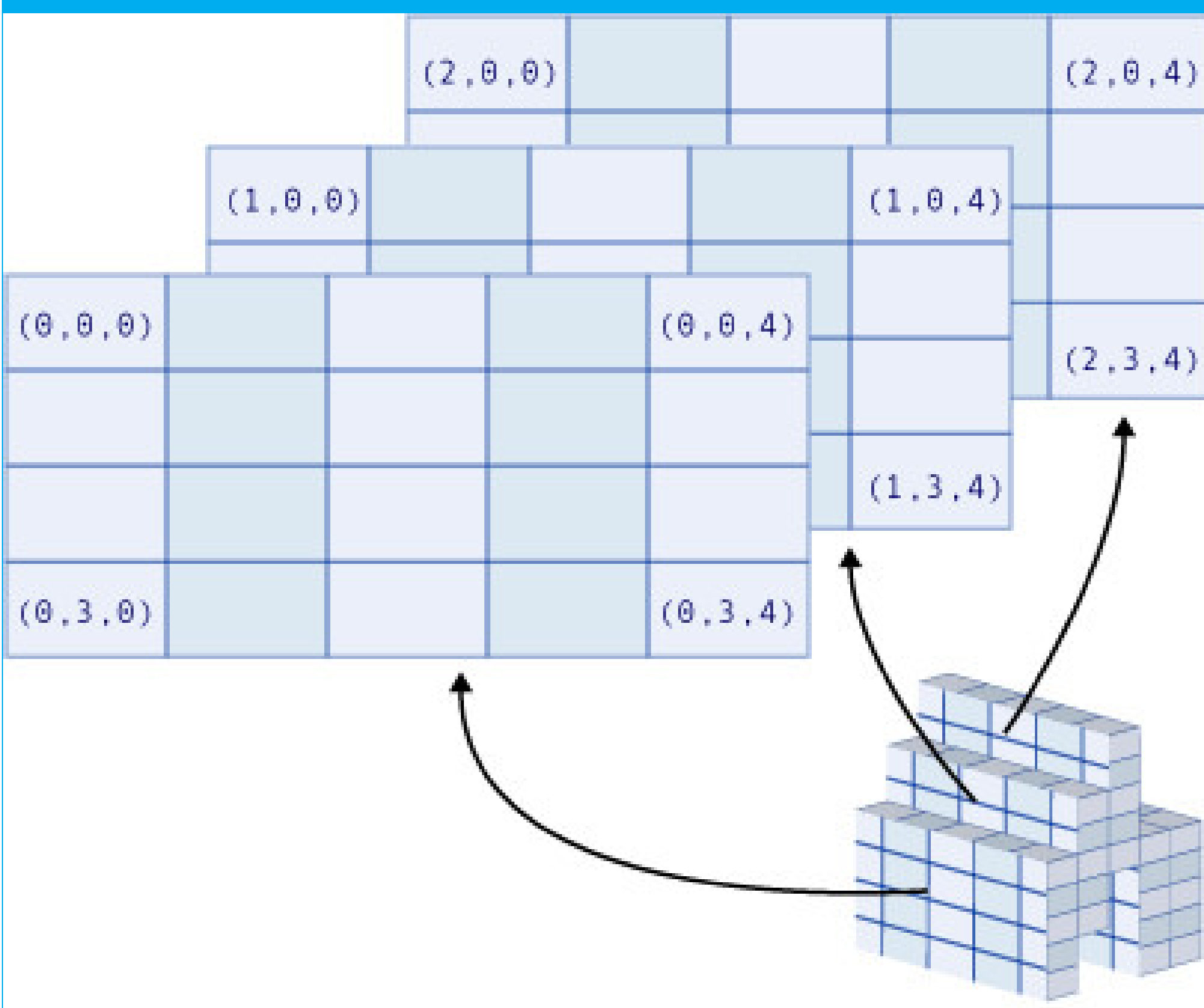
```
for ($i=0; $i<count($eggs);$i++)  
{  
    echo $eggs[$i];  
}
```

---

# Обработка ассоциативного массива и массива с индексами не по порядку

```
foreach($array as $k => $v)
{
    echo "Номер: {$k}, значение: $v";
}
$new_array = array();
foreach($array as $v)
{
    $new_array[] = $v; //"перепаковываем" ключи по порядку от 0
}
```





# Многомерный массив

---

# Многомерный массив -

- это массив, в котором значениями являются также массивы.

```
$a = array(  
    array("Москва", "Санкт-Петербург", "Нижний Новгород"),  
    array("Берлин", "Франкфурт", "Мюнхен"),  
    array("Париж", "Марсель")  
);  
echo $a[1][2]; //  
Мюнхен
```

---

# Многомерный ассоциативный массив

Ключи многомерного массива также могут быть заданы:

```
$a = array(  
    "Россия" => array("Москва", "Питер", 6 => "Н. Новгород"),  
    "Германия" => array("Берлин", 3 => "Франкфурт", 1 => "Мюнхен"),  
    "Франция" => array("Париж", "Марсель")  
);
```

```
echo $a["Россия"][6]; // Н. Новгород
```

---

# Обработка многомерного массива

Вложенные циклы:

```
foreach($a as $k => $v)
{
    $country = $k; // страна - ключ
    foreach($v as $city)
    {
        // в $city будут попадать города
    }
}
```

Если индексы многомерного массива (и всех внутренних элементов) идут по порядку, то

```
for($i = 0; $i<count($a); $i++)
{
    $row = $a[$i];
    for($j=0; $j<count($row); $j++)
    {
        $item = $row[$j];
    }
}
```

---

# Функции для работы с массивами

- `explode/implode` - пара функций для разбивания строки в массив и склеивания обратно
- `in_array($needle, $haystack)` - проверка наличия элемент в массиве
- `array_keys($array)` - получить массив, в котором элементами будут ключи исходного массива
- `array_shift($array)` - получение начального элемента массива, и удаление этого элемента из массива
- `array_search($needle, $haystack)` - поиск элемента в массиве и возврат соответствующего ключа, в случае успеха.
- Больше тут: <http://php.net/manual/en/ref.array.php>

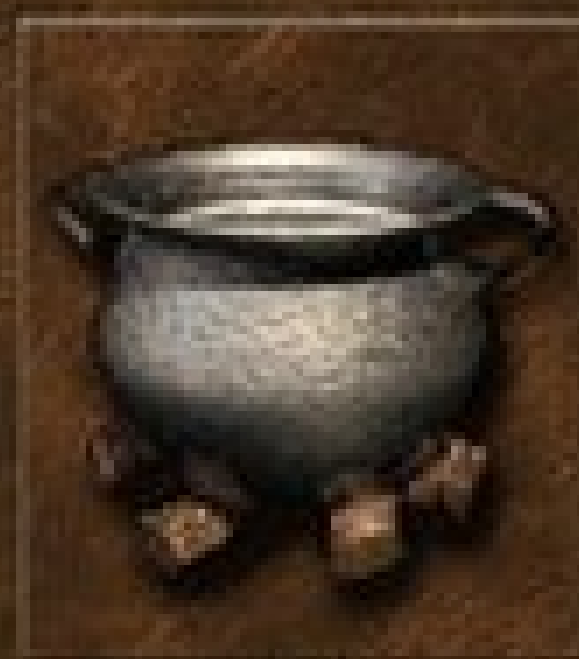


---

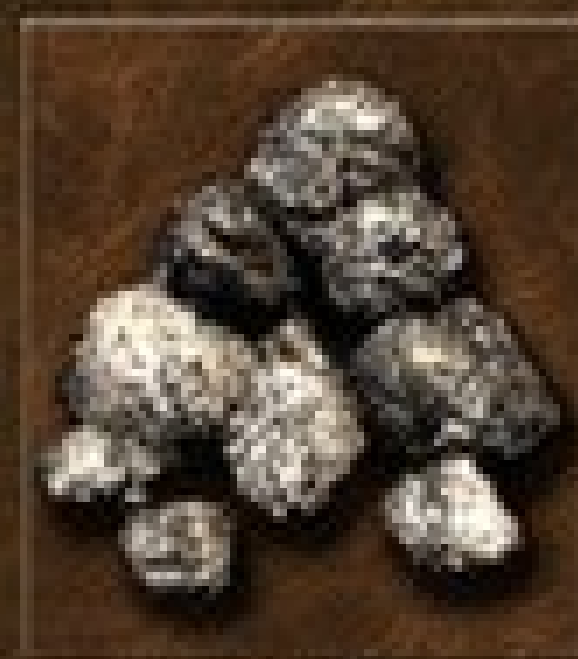
# Ресурсы



999



999



999



999



---

# Ресурсы

Ресурсы - это специальный тип данных, содержащий в себе указатель на внешний ресурс (открытый файл, соединение с БД, картинку и т.п.)

В своих приложениях, мы получаем ресурсы как результаты тех или иных функций, записываем их в переменную, и затем используем эту переменную, когда требуется обращение к внешним данным.

---

# Получаем и используем ресурсы

```
$f = fopen("file.txt", "r");
```

```
$string = fread($f, 1024); // $f - указывает  
на открытый нами файл
```

```
$link = mysqli_connect('localhost', 'my_user',  
'my_password', 'my_db');
```

```
$query = "SELECT * FROM `table` WHERE `";  
$result = mysqli_result($link, $query); // $link  
указывает на установленное с БД соединение
```

Список ресурсов (для смелых):

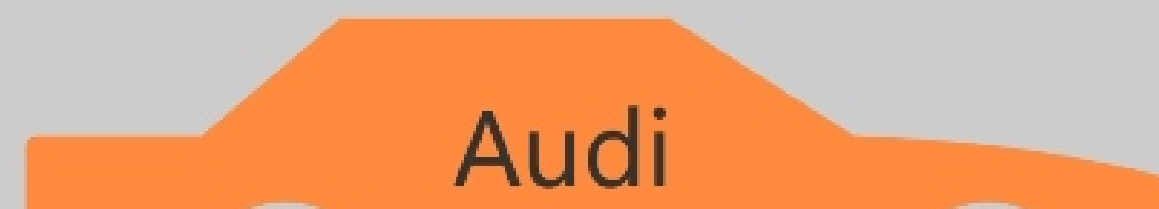
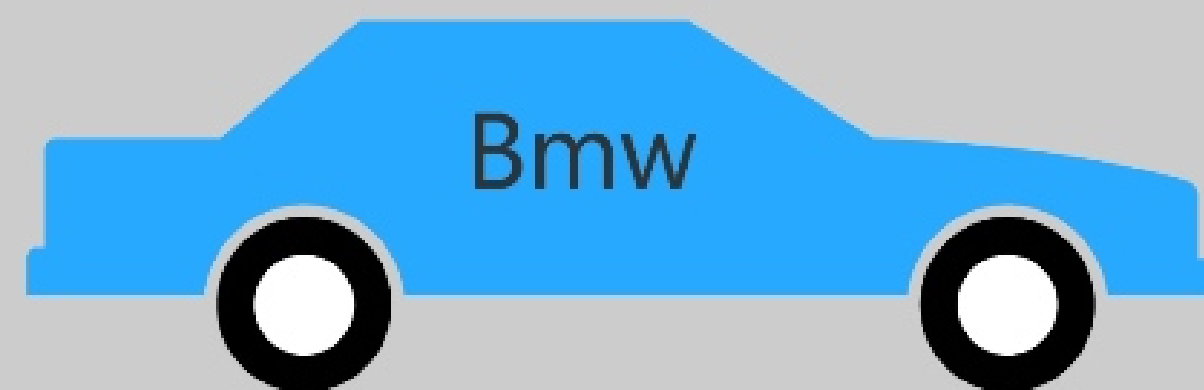
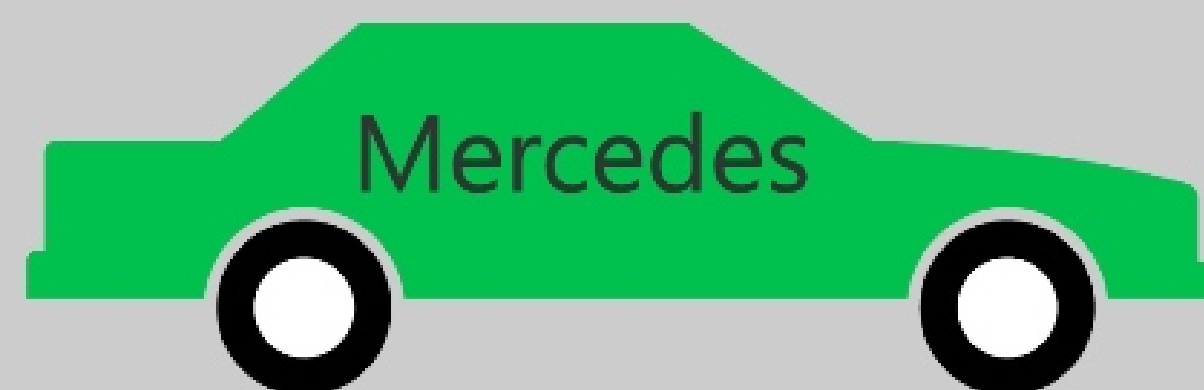
<http://php.net/manual/ru/resource.php>

# Объекты

class



objects



---

# Объекты

Объект - это структура, содержащая в себе данные и функции для работы с этими данными.

Класс - это шаблон, описывающий будущие объекты.

Созданные на основе класса объекты независимы друг от друга, живут своей жизнью, изменения в одном объекте не влияют на изменения в другом.



---

# Объявление класса

```
class Car
{
    var $brand;
    var $model;
    var $year;
    var $speed;
    function start()
    {
        if ($this->speed === 0) $this->speed = 10; // $this - этот объект
    }
}
```

---

# Использование объекта

```
$mercedes = new Car(); // создание объекта
```

```
$mercedes->brand = "Mercedes"; // установка свойств
```

```
$mercedes->start(); // обращение к методу класса
```

```
echo $mercedes->speed; // 10
```

---

NULL



N O T H I N G

---

# null и NULL

Значение NULL служит для отражения состояния переменной без какого-либо значения, не установленной, не объявленной, удаленной и т.п.

Это единственное возможное значение типа данных null

Переменная считается null, если

- ей была присвоена соответствующая константа (NULL)
- ей еще не было присвоено никакого значения
- она была удалена с помощью unset

NULL - хороший способ сообщить о том, что что-то пошло не так.



**нетология**  
центр онлайн-образования

Спасибо за внимание!

**РОМАН ДАВЫДОВ**



[warlight@mail.ru](mailto:warlight@mail.ru)



[fb.me/rsdavydov](https://fb.me/rsdavydov)