Sohel Rana (http://sohelrana.me/)

BANGLADESHI IT SPECIALIST

Category: PHP

PHP

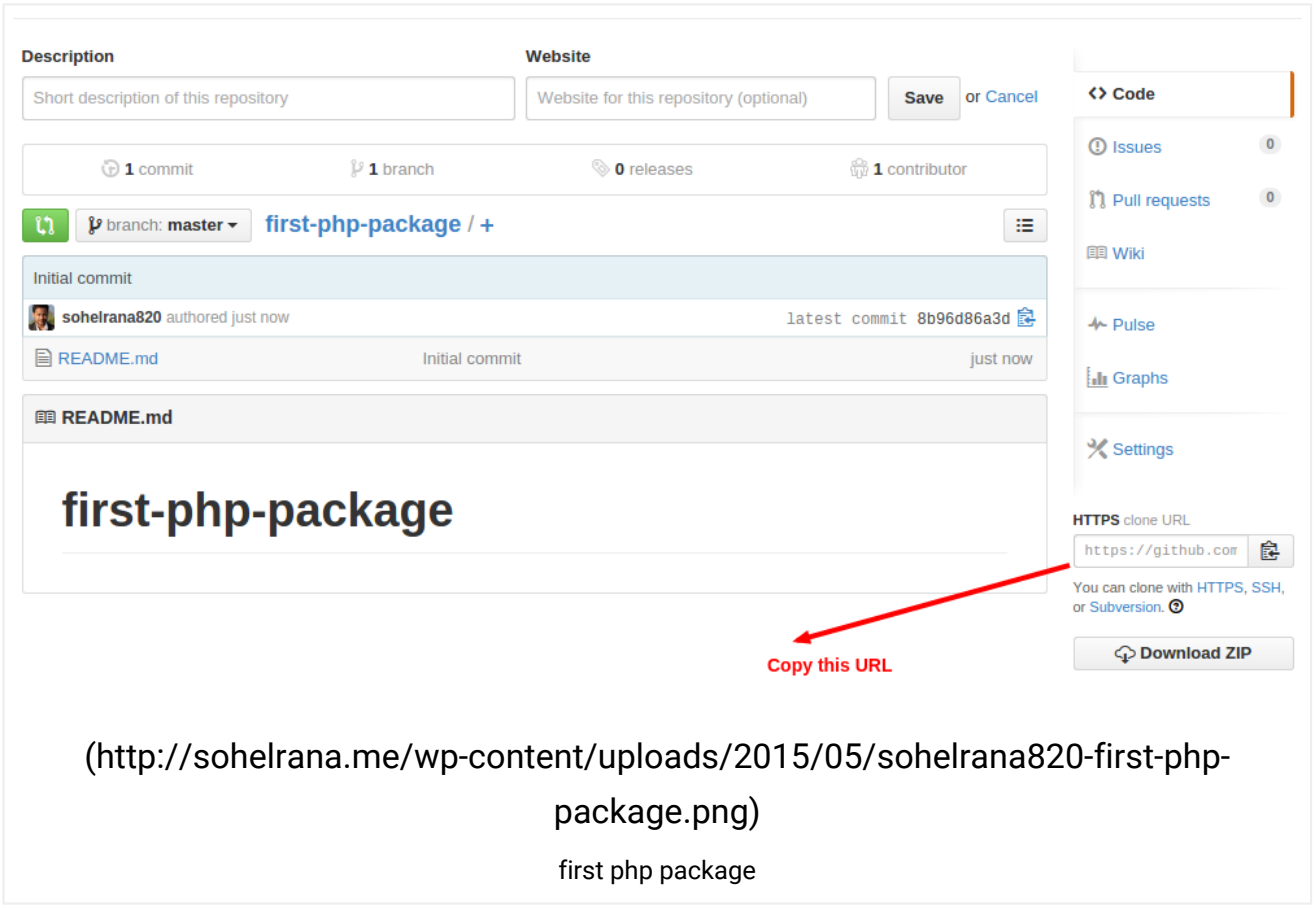
How to Create Package in PHP (http://sohelrana.me/how-to-create-package-in-php/)

MAY 3, 2015 (HTTP://SOHELRANA.ME/HOW-TO-CREATE-PACKAGE-IN-PHP/) BY ADMIN (HTTP://SOHELRANA.ME/AUTHOR/ADMIN/) · 0 COMMENT (HTTP://SOHELRANA.ME/HOW-TO-CREATE-PACKAGE-IN-PHP/#RESPOND)

One of the huge changes happened in PHP world is the Composer (http://getcomposer.org/). With the Composer (http://getcomposer.org/) you can able to use third party libraries in your project. But here I am trying to say that you are using other's libraries in your project, but why not you also create a library and use that by others as well. So let's see how to create a PHP package.

How to Create Package in PHP?

Here I am creating a sample package step by step, so you can see and do it by yourself. First of all login into your github (https://github.com) account and create a new repository called '**first-php-package**' .



(http://sohelrana.me/wp-content/uploads/2015/05/sohelrana820-first-php-package.png)

first php package

After creating a repository you will see something like that, and now copy that clone url.

Open your terminal and run this command `sudo git clone https://github.com/YOUR_GITHUB_USER_NAME/first-php-package.git` . Then open this project with your code editor. And create a **src** folder, **composer.json** file in project's root directory. Open up the composer.json file and paste this code.

```
{
    "name": "YOUR_GITHUB_USER_NAME/first-php-package",
    "description": "This is my first PHP package",
    "license": "MIT",
    "authors": [
        {
            "name": "YOUR NAME",
            "email": "YOYUR EMAIL ADDRESS"
        }
    ],
    "require": {
        "php": ">=5.3.0"

    },
    "autoload": {
        "psr-0": {
            "src\\": ""
        }
    }
}
```

Here **first-php-package'** is the name of package, Please don't change the name cause this must be the same with the github repository name. You can change the other's metadata as well in the composer.json file. Once you are done with editing the composer.json file then run `sudo composer install` command.

Go to **src** folder and create a **Hello.php** file and paste this code into **Hello.php** file.

```php
<?php

namespace src;

class Hello {

    public function sayHello()
    {
        return ' This is my first php package';
    }

}
```

Then run `sudo composer dump-autoload` command. And create a **test.php** file in your root directory and paste this code.

```php
<?php

require 'vendor/autoload.php';

$helloClass = new \src\Hello();
echo $helloClass->sayHello();
```

Finally commit and push all the code, and browse your project (e.g.http://localhost/first-php-package/test.php)

Now you have the package but you still need to submit that package. Go yo packagist (https://packagist.org) and login by your github account. Then click the 'Submit Package' button, after that put your repository url (in this case repository url is https://github.com/YOUR_GITHUB_USER_NAME/first-php-package) then check and submit. Now your package is submitted to packagist (https://packagist.org) and anyone can able to use it.

This is really basic way of creating a package and submit into packagist (https://packagist.org). Hope you can learn from this article how to create and how to submit a package.

Array Difference in nth Level Recursively in PHP (http://sohelrana.me/array-difference-in-nth-level-recursively-in-php/)

MARCH 31, 2015 (HTTP://SOHELRANA.ME/ARRAY-DIFFERENCE-IN-NTH-LEVEL-RECURSIVELY-IN-PHP/) BY ADMIN (HTTP://SOHELRANA.ME/AUTHOR/ADMIN/) · 0 COMMENT (HTTP://SOHELRANA.ME/ARRAY-DIFFERENCE-IN-NTH-LEVEL-RECURSIVELY-IN-PHP/#RESPOND)

Here I have written a code snippet for get array difference in nth lavel recursively in php

```
$person1 = array(
    array(
        'name' => 'A',
        'contactInfo' => array(
            'street' => 'Street',
            'phone' => 'phone',
            'email' => 'email@example.com',
        )
    ),
    array(
        'name' => 'B',
        'contactInfo' => array(
            'street' => 'Street',
            'phone' => 'phone',
            'email' => 'email@example.com',
        )
    ),
    array(
        'name' => 'C',
        'contactInfo' => array(
            'street' => 'Street',
            'phone' => 'phone',
            'email' => 'email@example.com',
        )
    ),
);

$person2 = array(
    array(
        'name' => 'A',
        'contactInfo' => array(
            'street' => '',
            'phone' => '',
            'email' => 'email@example.com',
        )
    ),
```

```
    array(
        'name' => 'D',
        'contactInfo' => array(
            'street' => 'Street',
            'phone' => 'phone',
            'email' => 'email@example.com',
        )
    ),
    array(
        'name' => 'C',
        'contactInfo' => array(
            'street' => '',
            'phone' => 'phone',
            'email' => 'email@example.com',
        )
    ),
);


/**
 * @param $array1
 * @param $array2
 * @return array
 *
 * Get different between two multidimensional array
 */
function arrayRecursiveDiff($array1, $array2)
{
    $return = array();

    foreach ($array1 as $key => $value) {
        if (array_key_exists($key, $array2)) {
            if (is_array($value)) {
                $recursiveDiff =arrayRecursiveDiff($value, $array2[$key]);
                if (count($recursiveDiff)) {
                    $return[$key] = $recursiveDiff;
                }
            } else {
                if ($value != $array2[$key]) {
                    $return[$key] = $value;
                }
            }
        } else {
            $return[$key] = $value;
        }
    }
    return $return;
}
var_dump(arrayRecursiveDiff($person1, $person2));
```

:.github.com/sohelrana820/0bece1374d24c87dd2b8/raw/array_difference_in_multidimensional_array.php) array_difference_in_multidimensional_array.php (https://gist.github.com/sohelrana820/0bece1374d24c87dd2b8#file-array_difference_in_multidimensional_array-php) hosted with ❤ by **GitHub (https://github.com)**

How to Get Data Between Two Timestamps From an Array (http://sohelrana.me/how-to-get-data-between-two-timestamps-from-an-array/)

MARCH 31, 2015 (HTTP://SOHELRANA.ME/HOW-TO-GET-DATA-BETWEEN-TWO-TIMESTAMPS-FROM-AN-ARRAY/) BY ADMIN (HTTP://SOHELRANA.ME/AUTHOR/ADMIN/) · 0 COMMENT (HTTP://SOHELRANA.ME/HOW-TO-GET-DATA-BETWEEN-TWO-TIMESTAMPS-FROM-AN-ARRAY/#RESPOND)

Sometimes programmer/developer need to get data between two timestamps from multidimensional array . Here I have written some sort of code for getting data between two timestamps from multidimensional array. I hope that will be helpful for you!

```
$start = '2015-03-31 11:00:00';
$end = '2015-03-31 20:00:00';

// Fruits array for testing
$fruits = array(
    array(
        'name' => 'Apple',
        'birthday' => '2015-03-31 13:00:00'
    ),
    array(
        'name' => 'Orange',
        'birthday' => '2015-03-31 09:00:00'
    ),
    array(
        'name' => 'Banana',
        'birthday' => '2015-03-31 17:00:00'
    ),
    array(
        'name' => 'Coconut',
        'birthday' => '2015-03-31 04:00:00'
    ),
    array(
        'name' => 'Jerry',
```

```
        'birthday' => '2015-03-31 15:00:00'
    ),
    array(
        'name' => 'Strawberry',
        'birthday' => '2015-03-31 22:00:00'
    ),
);

/**
 * @param $array
 * @param $start
 * @param $end
 * @return array
 *
 * This function will return the result between starting and endting time.
 */
function getResultBetweenTimestsmp($array, $start, $end)
{
    $result = array_map(
        function ($arrayElement) {
            return $arrayElement;
        },
        array_filter(
            $array,
            function ($arrayElement) use ($start, $end) {
                return $arrayElement['birthday'] >= $start && $arrayElement['birthday']
            }
        )
    );

    return $result;
}

$result = getResultBetweenTimestsmp($fruits, $start, $end);
var_dump($result);
```

/sohelrana820/7347c08d8ee23a2c912a/raw/how_to_get_data_between_timestamps_from_an_array.php) how_to_get_data_between_timestamps_from_an_array.php (https://gist.github.com/sohelrana820/7347c08d8ee23a2c912a#file-how_to_get_data_between_timestamps_from_an_array-php) hosted with ❤ by **GitHub (https://github.com)**

How to Sort PHP Multidimensional Array by Timestamp (http://sohelrana.me/how-to-sort-php-multidimensional-array-by-timestamp/)

MARCH 31, 2015 (HTTP://SOHELRANA.ME/HOW-TO-SORT-PHP-MULTIDIMENSIONAL-ARRAY-BY-TIMESTAMP/) BY ADMIN (HTTP://SOHELRANA.ME/AUTHOR/ADMIN/) · 0 COMMENT (HTTP://SOHELRANA.ME/HOW-TO-SORT-PHP-MULTIDIMENSIONAL-ARRAY-BY-TIMESTAMP/#RESPOND)

Sometimes we need to sort out multidimensional array by timestamp. Which is really important to programmer/developer how easily they can sort out a multidimensional array by timestamp (Time). I am here to share a script which will be helpful to sort your multidimensional array by timestamp.

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
```

```php
// Feeds array with title and time
$feeds = array(
    array(
        'title'  => 'Some Feeds.',
        'time'  => '2015-03-31 13:06:00'
    ),
    array(
        'title'  => 'And another.',
        'time'  => '2015-03-23 13:06:00'
    ),
    array(
        'title'  => 'Another Feed.',
        'time'  => '2015-03-27 13:06:00'
    ),
    array(
        'title'  => 'And one more feed.',
        'time'  => '2015-03-20 13:06:00'
    ),
    array(
        'title'  => 'And one more feed.',
        'time'  => '2015-03-25 13:06:00'
    )
);

// Sorting array by time (DESC ORDER)
usort($feeds, function($firstItem, $secondItem) {
        $timeStamp1 = strtotime($firstItem['time']);
        $timeStamp2 = strtotime($secondItem['time']);
        return $timeStamp2 - $timeStamp1;
    });
var_dump($feeds);

// Sorting array by time (ASC ORDER)
usort($feeds, function($firstItem, $secondItem) {
        $timeStamp1 = strtotime($firstItem['time']);
        $timeStamp2 = strtotime($secondItem['time']);
        return $timeStamp1 - $timeStamp2;
    });
var_dump($feeds);
```

raw (https://gist.github.com/sohelrana820/861c36a511282335ce85/raw/sort_multidimensional_array_by_time.php) **sort_multidimensional_array_by_time.php (https://gist.github.com/sohelrana820/861c36a511282335ce85#file-sort_multidimensional_array_by_time-php)** hosted with ❤ by **GitHub (https://github.com)**

What is PHP Magic Method, Why and How Should Use Magic Method (http://sohelrana.me/what-is-php-magic-method-why-and-how-should-use-magic-method/)

SEPTEMBER 16, 2014 (HTTP://SOHELRANA.ME/WHAT-IS-PHP-MAGIC-METHOD-WHY-AND-HOW-SHOULD-USE-MAGIC-METHOD/) BY ADMIN (HTTP://SOHELRANA.ME/AUTHOR/ADMIN/) · 0 COMMENT (HTTP://SOHELRANA.ME/WHAT-IS-PHP-MAGIC-METHOD-WHY-AND-HOW-SHOULD-USE-MAGIC-METHOD/#RESPOND)

Magic method is called some predefined method of PHP which execute some event by the PHP compiler. The method start with the double underscore (__) known as PHP magic method (for example __construct, __destruct, __get, __set, __call etc). This is method is usually use inside any class. PHP has various magic method, I will try to discuss some most common magic method.

__construct:

The __construct method will called when someone create any instance/object of a class. This method usually declared at first of a class, but there is no any strict rules where we need to declare it. We can declare is anywhere in our class. and also we can inherit this method.

__destruct:

The __destruct method is called when object is destroyed, it's opposite of __construct method. This method usually declared at last of a class, but there is no any strict rules where we need to declare it. We can declare is anywhere in our class.

__get:

The get method is called when anyone want to read/get the value of property or variable.

__set:

The __set method is called when someone want to set value of a property or variable.

__isset:

The __isset method is called when someone applied isset() method anywhere in a class.

__unset:

__the __unset method is called when someone applied unset() method anywhere in a class, which i really opposite of isset method.

__call:

The __call method is called when someone try to call a normal method.

__sleep:

The __sleep method is called when someone trying to serialize his class object.

__wakeup:

The __sleep method is called when someone trying to deserialize his class object.

__toString:

The __toString method is called when someone try to print the object of a class.

Example:

```
<?php
class MagicClass
{
    public $classDetails = 'This is simple basic class for magic to understand magic me
    public $yourID;
    public $yourName;

    protected $isDBConnected;
    protected $isDataStored;

    private $yourAddress;

    public function __construct($yourID, $yourName)
    {
        $this->yourName = $yourName;
        $this->yourID = $yourID;
    }

    public function __set($property, $value)
    {
        if (property_exists($this, $property)) {
            $this->$property = $value;
        }
    }

    public function __get($property)
    {
        if (property_exists($this, $property)) {
            return $this->$property;
        }

```

```
37      }
38
39
40      public function  __isset($property)
41      {
42          return isset($this->$property);
43      }
44
45      public function __unset($property)
46      {
47          unset($this->$property);
48      }
49
50      public function __toString()
51      {
52          return $this->yourName;
53      }
54
55      public function __sleep()
56      {
57          return array('classDetails');
58      }
59
60      public function __wakeup()
61      {
62          $this->getConnection();
63      }
64
65      public function __call($method, $parameters)
66      {
67          if (in_array($method, array('getConnection', 'storeData')))
68          {
69              return call_user_func_array(array($this, $method), $parameters);
70          }
71      }
72
73      protected  function getConnection()
74      {
75          $this->isDBConnected;
76      }
77
78      protected function storeData()
79      {
80          $this->isDBConnected;
81      }
82
83      public function __destruct()
84      {
85          return 'Good Bye';
86      }
87  }
88
89
```

```
98  // Creating object of MagicClass and provide the argument to __construct.
99  $classObject = new MagicClass("22", 'My Name');
100
101
102 // MagicClass will react when someone try to echo it's object. (usages of __toString())
103 echo $classObject;
104 echo '<br/>';
105
106 // Set the value of yourAddress variable. (usages of __set())
    $classObject->yourAddress = 'xxx xxx xxxx';

    // Get the value of yourAddress variable. (usages of __get())
    echo $classObject->yourAddress;
    echo '<br/>';



    //Checking the yourAddress is exist or not. (usages of __isset())
    var_dump(isset($classObject->yourAddress));

    //serializing the this object. (usages of __sleep())
    $serialization = serialize($classObject);

    //un-serializing the this object. (usages of __wakeup())
    $deSerialization = unserialize($serialization);
    echo $deSerialization->classDetails; // Using classDetails method through un-serialize
    echo '<br/>';
```

view raw (https://gist.github.com/sohelrana820/55fec6c98a31382c6824/raw/magic_method.php) magic_method.php (https://gist.github.com/sohelrana820/55fec6c98a31382c6824#file-magic_method-php) hosted with ❤ by **GitHub (https://github.com)**

Sort Multidimensional Array by KEY (http://sohelrana.me/sort-multidimensional-array-by-key/)

SEPTEMBER 11, 2014 (HTTP://SOHELRANA.ME/SORT-MULTIDIMENSIONAL-ARRAY-BY-KEY/) BY ADMIN (HTTP://SOHELRANA.ME/AUTHOR/ADMIN/) · 0 COMMENT (HTTP://SOHELRANA.ME/SORT-MULTIDIMENSIONAL-ARRAY-BY-KEY/#RESPOND)

Array is the most important part of programming. Sometime we need to sort out our result based on key (ascending and descending order). Sometime it’s boring for us. For this reason I have created a function for sorting out the array based on my given key and of course it’s working for the multidimensional array.

```php
<?php

/**
 * @return mixed
 *
 * This is array will output array (ascending and descending order) according to KEY
 */
function sortArray()
{
    $funcArgument = func_get_args();
    $content = array_shift($funcArgument);

    foreach ($funcArgument as $n => $field) {

        if (is_string($field)) {
            $tmp = array();
            foreach ($content as $key => $row) {
                $tmp[$key] = $row[$field];
            }
            $funcArgument[$n] = $tmp;
        }

    }

    $funcArgument[] = & $content;
    call_user_func_array('array_multisort', $funcArgument);
    return array_pop($funcArgument);
}

/**
 * Create a dummy array for test.
 */
$fruitsArray = array(
    0 => array(
        'name' => 'Apple',
        'price' => '5'
    ),
    1 => array(
        'name' => 'Orange',
        'price' => '12'
    ),
    2 => array(
        'name' => 'Banana',
        'price' => '7'
    ),
    3 => array(
        'name' => 'Mango',
        'price' => '10'
    ),
```

```
    5 => array(
        'name' => 'Pineapple',
        'price' => '4'
    ),
);

$sortByPrice = sortArray($fruitsArray, 'price', SORT_DESC); // Need output based on pric
$sortByName = sortArray($fruitsArray, 'name', SORT_ACE); // Need output based on price (

var_dump($sortByPrice);
var_dump($sortByName);
?>
```

view raw (https://gist.github.com/sohelrana820/f2dad639f79768db0f35/raw/sort_multidimensional_array.php) sort_multidimensional_array.php (https://gist.github.com/sohelrana820/f2dad639f79768db0f35#file-sort_multidimensional_array-php) hosted with ❤ by **GitHub (https://github.com)**

OUTPUT

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
```

```
array (size=5)
  0 =>
    array (size=2)
      'name' => string 'Orange' (length=6)
      'price' => string '12' (length=2)
  1 =>
    array (size=2)
      'name' => string 'Mango' (length=5)
      'price' => string '10' (length=2)
  2 =>
    array (size=2)
      'name' => string 'Banana' (length=6)
      'price' => string '7' (length=1)
  3 =>
    array (size=2)
      'name' => string 'Apple' (length=5)
      'price' => string '5' (length=1)
  4 =>
    array (size=2)
      'name' => string 'Pineapple' (length=9)
      'price' => string '4' (length=1)
array (size=5)
  0 =>
    array (size=2)
      'name' => string 'Apple' (length=5)
      'price' => string '5' (length=1)
  1 =>
    array (size=2)
      'name' => string 'Banana' (length=6)
      'price' => string '7' (length=1)
  2 =>
    array (size=2)
      'name' => string 'Mango' (length=5)
      'price' => string '10' (length=2)
  3 =>
    array (size=2)
      'name' => string 'Orange' (length=6)
      'price' => string '12' (length=2)
  4 =>
    array (size=2)
      'name' => string 'Pineapple' (length=9)
      'price' => string '4' (length=1)
```

iew raw (https://gist.github.com/sohelrana820/02f11767d7c308f6e569/raw/output_of_sort_multidimensional_array) output_of_sort_multidimensional_array (https://gist.github.com/sohelrana820/02f11767d7c308f6e569#file-output_of_sort_multidimensional_array) hosted with ♥ by **GitHub (https://github.com)**

Setter and Getter through PHP magic methods (http://sohelrana.me/setter-and-getter-through-php-magic-methods/)

SEPTEMBER 10, 2014 (HTTP://SOHELRANA.ME/SETTER-AND-GETTER-THROUGH-PHP-MAGIC-METHODS/) BY ADMIN (HTTP://SOHELRANA.ME/AUTHOR/ADMIN/) · 0 COMMENT (HTTP://SOHELRANA.ME/SETTER-AND-GETTER-THROUGH-PHP-MAGIC-METHODS/#RESPOND)

Setter and Getter (SET/GET) is the most important and useful technique of programming. There are several way to do that, but the most effective and efficiency way is using PHP magic functions. Here I am gonna show you how to set and get through PHP magic method.

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
```

```php
<?php
class MagicClass
{

    /**
     * @var
     *
     * Declaring two private variable.
     */
    private $myName;
    private $myAddress;

    /**
     * @param $property
     * @param $propertyValue
     *
     * Setting the value of property
     */
    public function __set($property, $propertyValue)
    {
        if(property_exists($this, $property)){
            $this->$property = $propertyValue;
        }
    }

    /**
     * @param $property
     * @return mixed
     *
     * Getting the property value.
     */
    public function __get($property)
    {
        if(property_exists($this, $property)){
            return $this->$property;
        }
    }
}

$myObject = new MagicClass(); // Create instance or object of Magic Class

$myObject->myName = 'Write your name'; // Set the value of myName variable.
$myObject->myAddress = 'Your address'; // Set the value of myAddress variable

echo $myObject->myName; // Displaying the name
echo $myObject->myAddress; // Displaying the address

?>
```

view raw (https://gist.github.com/sohelrana820/97609e6c5ac9e15076cb/raw/setter_and_getter_in_php.php) setter_and_getter_in_php.php (https://gist.github.com/sohelrana820/97609e6c5ac9e15076cb#file-setter_and_getter_in_php-php) hosted with ❤ by **GitHub (https://github.com)**

How to Install xDebug in Ubuntu (http://sohelrana.me/how-to-install-xdebug-in-ubuntu/)

SEPTEMBER 1, 2014 (HTTP://SOHELRANA.ME/HOW-TO-INSTALL-XDEBUG-IN-UBUNTU/) BY ADMIN (HTTP://SOHELRANA.ME/AUTHOR/ADMIN/) · 0 COMMENT (HTTP://SOHELRANA.ME/HOW-TO-INSTALL-XDEBUG-IN-UBUNTU/#RESPOND)

xDebug is most famous and popular PHP extension which provides the debugging of your code very smartly. It is most important to a programmer to debugging their code/data, xDebug gives us more flexibility to debug our code. Here I am gonna show you how to install xDebug in your ubuntu machine.

Run this code through your terminal.

```
~$ sudo apt-get install php5-xdebug
```

Then open your php.ini file and add these code in your ini file.

```
# Added for xdebug
zend_extension="/usr/lib/php5/20100525/xdebug.so"
xdebug.remote_enable=1
xdebug.remote_handler=dbgp xdebug.remote_mode=req
xdebug.remote_host=127.0.0.1 xdebug.remote_port=9000
```

Finally restart the server.

```
~$ sudo service SERVER_NAME restart
```

All done! now you can enjoy your debugging.

Search Value from Multi Dimension Array (http://sohelrana.me/search-value-from-multi-dimension-array/)

AUGUST 31, 2014 (HTTP://SOHELRANA.ME/SEARCH-VALUE-FROM-MULTI-DIMENSION-ARRAY/) BY ADMIN (HTTP://SOHELRANA.ME/AUTHOR/ADMIN/) · 0 COMMENT (HTTP://SOHELRANA.ME/SEARCH-VALUE-FROM-MULTI-DIMENSION-ARRAY/#RESPOND)

Here is simple example of searching specific value from a multi dimension array.

```
<?php
class ArraySearch{

    /**
     * @param $array
     * @param $key
     * @param $value
     * @return array
     *
     * Searching elements from the array.
     * This function will search among the whole array by specific key value.
     * And the return result the possible element.
     */
    public  function searchProfession($array, $key, $value)
    {
        $results = array();
        if (is_array($array)) {
            if (isset($array[$key]) && $array[$key] == $value) {
                $results[] = $array;
            }

            foreach ($array as $subArray) {
                $results = array_merge($results, $this->searchProfession($subArray, $key
            }
        }
        return $results;
    }
}

$data = array(
    0 => array(
        'contact' => array(
            'name' => 'Your Name',
            'Phone' => 'Your Phone Number',
            'profession' => 'Teacher',
        )
    ),
```

```
    1 => array(
        'contact' => array(
            'name' => 'Your Name',
            'Phone' => 'Your Phone Number',
            'profession' => 'Doctor',
        )
    ),
    3 => array(
        'contact' => array(
            'name' => 'Your Name',
            'Phone' => 'Your Phone Number',
            'profession' => 'Teacher',
        )
    ),
    4 => array(
        'contact' => array(
            'name' => 'Your Name',
            'Phone' => 'Your Phone Number',
            'profession' => 'Engineer',
        )
    ),
);


$arrayObj = new ArraySearch();
var_dump($arrayObj->searchProfession($data, 'profession', 'Engineer'));
var_dump($arrayObj->searchProfession($data, 'profession', 'Teacher'));
```

ew raw (https://gist.github.com/sohelrana820/abf192e5eef85dd128fc/raw/search_value_multi_dimantion_array.php)
search_value_multi_dimantion_array.php (https://gist.github.com/sohelrana820/abf192e5eef85dd128fc#file-search_value_multi_dimantion_array-php) hosted with ❤ by **GitHub (https://github.com)**

Singleton Design Pattern (http://sohelrana.me/singleton-design-pattern/)

AUGUST 28, 2014 (HTTP://SOHELRANA.ME/SINGLETON-DESIGN-PATTERN/) BY ADMIN (HTTP://SOHELRANA.ME/AUTHOR/ADMIN/) · 0 COMMENT (HTTP://SOHELRANA.ME/SINGLETON-DESIGN-PATTERN/#RESPOND)

In software engineering, a singleton the design pattern which allowed only object of a class. That means singleton pattern restricts to create multi instance of a class. This is useful, when we need to create only one instance or object of a class. To do this we

must need to ensure a mechanism where we can access that class without creating object for them.

```php
<?php
class Singleton
{
    public static function classInstance()
    {
        static $instance = null;
        if (null === $instance) {
            $instance = new static();
        }

        return $instance;
    }

    protected function __construct()
    {

    }
}

class SingletonChild extends Singleton
{

}
```

view raw (https://gist.github.com/sohelrana820/930bb683270ab1326dee/raw/Singleton.php) Singleton.php (https://gist.github.com/sohelrana820/930bb683270ab1326dee#file-singleton-php) hosted with ❤ by **GitHub (https://github.com)**

```php
/** Here is the OUTPUT **/

$obj = Singleton::classInstance();
var_dump($obj === Singleton::classInstance());             // bool(true)

$anotherObj = SingletonChild::classInstance();
var_dump($anotherObj === Singleton::classInstance());      // bool(false)

var_dump($anotherObj === SingletonChild::classInstance()); // bool(true)
```

output.php view raw (https://gist.github.com/sohelrana820/930bb683270ab1326dee/raw/output.php) (https://gist.github.com/sohelrana820/930bb683270ab1326dee#file-output-php) hosted with ❤ by **GitHub (https://github.com)**

This is very basic concept of singleton design pattern.