PHP | array\_filter() Function

* Last Updated : 08 Mar, 2018

This built-in function in PHP is used to filter the elements of an array using a user-defined function which is also called a callback function. The array\_filter() function iterates over each value in the array, passing them to the user-defined function or the callback function. If the callback function returns true then the current value of the array is returned into the result array otherwise not. This way the keys of the array gets preserved, i.e. the key of element in the original array and output array are same.

**Syntax:**

*array* array\_filter($array, $callback\_function, $flag)

**Parameters**: The function takes three parameters, out of which one is mandatory and the other two are optional.

1. **$array** (mandatory): This refers to the input array on which the filter operation is to be performed.
2. **$callback\_function** (optional): Refers to the user-defined function. If the function is not supplied then all entries of the array equal to FALSE , will be removed.
3. **$flag** (optional): Refers to the arguments passed to the callback function.
   * ARRAY\_FILTER\_USE\_KEY – passes key as the only argument to a callback function, instead of the value of the array.
   * ARRAY\_FILTER\_USE\_BOTH – passes both value and key as arguments to callback instead of the value.

**Return Value**: The function returns a filtered array.

Below is a program showing how to return or filter out even elements from an array using array\_filter() function.

filter\_none

edit

play\_arrow

brightness\_4

|  |
| --- |
| <?php    // PHP function to check for even elements in an array  function Even($array)  {      // returns if the input integer is even      if($array%2==0)         return TRUE;      else         return FALSE;  }    $array = array(12, 0, 0, 18, 27, 0, 46);  print\_r(array\_filter($array, "Even"));    ?> |

Output:

Array

(

[0] => 12

[1] => 0

[2] => 0

[3] => 18

[5] => 0

[6] => 46

)

CREATE INDEX Example

CREATE INDEX index\_name  
ON table\_name (column1, column2, ...);

The SQL statement below creates an index named "idx\_lastname" on the "LastName" column in the "Persons" table, now when we search something with select query like => “SELECT \* FROM table WHERE dob=”12-1-1997”,

Then if we write this below “index” query, the search will first see the indexes, and when it founds the lastname index, it will enter that column only, instead of checking all the columns one by one.

CREATE INDEX idx\_lastname  
ON Persons (LastName);

Link for better understanding:=> <https://www.youtube.com/watch?v=vRJM1sDrocM&t=364s>

## SQL CREATE VIEW Statement

In SQL, a view is a virtual table based on the result-set of an SQL statement.

A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database.

You can add SQL functions, WHERE, and JOIN statements to a view and present the data as if the data were coming from one single table.

### **CREATE VIEW Syntax**

CREATE VIEW view\_name AS  
SELECT column1, column2, ...  
FROM table\_name  
WHERE condition;

LINK : <https://youtu.be/iLHolv9Oeyg>

1)CREATE VIEW studentdata

AS

SELECT id, name, course\_name FROM table INNER JOIN course c ON s.courses =c.course\_id;

2)SELECT \* FROM studentdata;

Explanation:at first we are creating a view named “studentdata” and then adding the complex sql query which is to be repetedly used throught the file and saved it in database by creating the view. Then later whenever we select the studentdata in “point 2“ then the “table” of point 1 INNER JOIN query gets selected and we don’t have to write the big complex query of “point 1” every time.

1)routes, sql queries, log for error messages, result\_array objects which is array in array,

**//CODEIGNITER//**

A CodeIgniter helper is a set of related functions (Common functions) which you could use them within *Models*, *Views*, *Controllers*,.. everywhere.

Once you load (include) that file, you can get access to the functions.

But a Library is a class, which you need to make an instance of the class (by $this->load->library()). And you'll need to use the object $this->... to call the methods.

**As a thumb rule:** A library is used in object oriented context (Controller, ...), while a helper is more suitable to be used within the *Views* (non object oriented).

Q.1)When to use array object and normal array and when we use row vs row\_array nd vice versa for result

Q.2) How to fetch the returned value of an object and array?

$q->row()->id returns the id, $q->row()->email returns the email…

**//TERMS TO KNOW//**

1)setting database unique email colummn : Unique function is used to set all values in a databse column to unique

Ans. CREATE TABLE table\_name(

   ...

   column\_name1 column\_definition,

   column\_name2 column\_definition,

   ...,

   UNIQUE(column\_name1,column\_name2)

);

2).htaccess use3

Ans. .htaccess is a configuration file for use on web servers running the Apache Web Server software.

    When a .htaccess file is placed in a directory which is in turn 'loaded via the Apache Web Server', then the .htaccess file is detected and executed by the Apache Web Server software.

    These .htaccess files can be used to alter the configuration of the Apache Web Server software to enable/disable additional functionality and features that the Apache Web Server software has to offer.

    These facilities include basic redirect functionality, for instance if a 404 file not found error occurs, or for more advanced functions such as content password protection or image hot link prevention.

    Whenever any request is sent to the server it always passes through .htaccess file. There are some rules are defined to instruct the working.

3)site\_url vs base\_url

Ans. echo base\_url(); // http://example.com/website

echo site\_url(); // http://example.com/website/index.php

if you want a URL access to a resource (such as css, js, image), use base\_url(), otherwise, site\_url() is better.

4)implode

Ans.

// $arr = array('Hello','World!','Beautiful','Day!');

// echo implode("/",$arr);

OUTPUT: Hello/World!/Beautiful/Day!

5)SELECT \* FROM `login` WHERE subject="eng" AND subject="math"

6)unit of varchar(255);

Ans. 32,672 bytes

7)select \* from subject WHERE subject LIKE '%eng%' or '%math%'

(i.e.,) Finds any values that have "or" in any position

id1 = 12 , subject1:beng,eng,math

id2 = 24 , subject1:eng,math,his

id3 = 43 , subject1:geo,bio,phy

Ans. here we search the value in a given set(eng,beng,hindi) with LIKE %eng%, and wil search the value in that set of a particular row and return that id.

8)find\_in\_set('value\_name','table\_name')

Ans. here set refers to the list of subjects given for a particular user id in a row, in that set we will search the value name and if present then return the id

9)SYNTAX : select \* from table\_name where field\_name IN(value1, value2);

Ans. select \* from customer\_tbl where country IN('Germany','uk','france') => this sql will return

all the rows having those country name , and it is a multiple or function because after locating any one of the above names of the country, it will return the whole id

**1)setting database unique email colummn : Unique function is used to set all values in a databse column to unique**

**Ans. CREATE TABLE table\_name(**

**...**

**column\_name1 column\_definition,**

**column\_name2 column\_definition,**

**...,**

**UNIQUE(column\_name1,column\_name2)**

**);**

**2).htaccess use3**

**Ans. .htaccess is a configuration file for use on web servers running the Apache Web Server software.**

**When a .htaccess file is placed in a directory which is in turn 'loaded via the Apache Web Server', then the .htaccess file is detected and executed by the Apache Web Server software.**

**These .htaccess files can be used to alter the configuration of the Apache Web Server software to enable/disable additional functionality and features that the Apache Web Server software has to offer.**

**These facilities include basic redirect functionality, for instance if a 404 file not found error occurs, or for more advanced functions such as content password protection or image hot link prevention.**

**Whenever any request is sent to the server it always passes through .htaccess file. There are some rules are defined to instruct the working.**

**3)site\_url vs base\_url**

**Ans. echo base\_url(); // http://example.com/website**

**echo site\_url(); // http://example.com/website/index.php**

**if you want a URL access to a resource (such as css, js, image), use base\_url(), otherwise, site\_url() is better.**

**4)implode**

**Ans.**

**// $arr = array('Hello','World!','Beautiful','Day!');**

**// echo implode("/",$arr);**

**OUTPUT: Hello/World!/Beautiful/Day!**

**5)SELECT \* FROM `login` WHERE subject="eng" AND subject="math"**

**6)unit of varchar(255);**

**Ans. 32,672 bytes**

**7)select \* from subject WHERE subject LIKE '%eng%' or '%math%'**

**(i.e.,) Finds any values that have "or" in any position**

**id1 = 12 , subject1:beng,eng,math**

**id2 = 24 , subject1:eng,math,his**

**id3 = 43 , subject1:geo,bio,phy**

**Ans. here we search the value in a given set(eng,beng,hindi) with LIKE %eng%, and wil search the value in that set of a particular row and return that id.**

**8)find\_in\_set('value\_name','table\_name')**

**Ans. here set refers to the list of subjects given for a particular user id in a row, in that set we will search the value name and if present then return the id**

**9)SYNTAX : select \* from table\_name where field\_name IN(value1, value2);**

**Ans. select \* from customer\_tbl where country IN('Germany','uk','france') => this sql will return**

**all the rows having those country name , and it is a multiple or function because after locating any one of the above names of the country, it will return the whole id**

cal\_days\_in\_month — Return the number of days in a month for a given year and calendar

### **Description**[**¶**](https://www.php.net/manual/en/function.cal-days-in-month.php#refsect1-function.cal-days-in-month-description)

**cal\_days\_in\_month** ( int $calendar , int $month , int $year ) : int

This function will return the number of days in the month of year for the specified calendar.

|  |
| --- |
| <?php  // PHP program to demonstrate the **strtotime**()  // function when the english text is "**now**"    // prints **current time** in second  // since now means current  echo **strtotime**("now"), "\n";    // prints the current time in date format  echo date("Y-m-d", strtotime("now"))."\n";  ?> |

Output:

1525378260 //this is a format obtained from strtotime("now") where the //date month and year can be fetched using date("Y-m-d")

2018-05-03

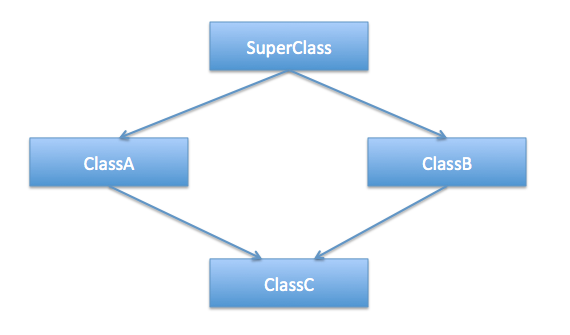
**Terms to know =>**

The difference between attributes and properties can be important in specific situations. Before jQuery 1.6, the .attr() method sometimes took property values into account when retrieving some attributes, which could cause inconsistent behavior. As of jQuery 1.6, the .prop() method provides a way to explicitly retrieve property values, while .attr() retrieves attributes.

1. **What are the problems of arises for multiple inheritance and How to make multiple inheritance possible in PHP?**

The “diamond problem” (sometimes referred to as the “deadly diamond of death”) is an ambiguity that arises when two classes B and C inherit from A, and class D inherits from both B and C. If there is a method in A that B and C have overridden, and D does not override it, then which version of the method does D inherit: that of B, or that of C?

Take this for example by assuming PHP is supporting multiple inheritance.



Let’s say **SuperClass** is an abstract class declaring some method and **ClassA**, **ClassB** are concrete classes.

**<?php**

***class* SuperClass**

**{**

**protected function greet()**

**{**

**echo "Grandparent";**

**}**

**}**

**// First Parent class**

***class* ClassA *extends* SuperClass**

**{**

**protected function greet()**

**{**

**echo "Parent1";**

**}**

**}**

**// Second Parent class**

***class* ClassB *extends* SuperClass**

**{**

**protected function greet()**

**{**

**echo "Parent2";**

**}**

**}**

***class* ClassC *extends* ClassA, ClassB**

**{**

**public function test()**

**{**

***$c* = new self();**

***$c*->greet();**

**}**

**}**

As you can see from the code, on calling the method **greet()** using object **ClassC**, it’s impossible for the compiler to decide whether it has to call ClassA’s **greet()** or ClassB’s **greet()** method. So, this is to avoid such complications, PHP does not support multiple inheritance.

You may also like: [A closer look at Invokable classes in PHP](https://www.amitmerchant.com/invokable-classes-php/)

## Mitigation of Diamond problem in PHP

One solution to mitigate not having multiple inheritance in PHP is to use traits. Traits are a mechanism for code reuse in single inheritance languages such as PHP which you’d use multiple inheritance for. Basically, traits are like classes except for one fact that you can’t initantiate an instance of a trait. That is like utilising class members directly into the class without needing to instantiate or inherit them.

Below is an example of how you can define a trait and utilize the same in the class.

***trait* myTrait**

**{**

**public function whereAmI()**

**{**

**echo \_\_CLASS\_\_;**

**}**

**}**

***class* Hello**

**{**

**use myTrait;**

**}**

***$a* = new Hello;**

***$a*->whereAmI(); //Hello**

Similarly, you can use multiple traits in a single clss comma-separated like this.

**use Hello, World;**

You can read further about Traits [here](https://www.php.net/manual/en/language.oop5.traits.php).

Another solution here would be to [use composition](https://www.amitmerchant.com/reasons-use-composition-over-inheritance-php/) while designing your software. Basically, Composition is the mechanism to reuse code across classes by containing instances of other classes that implement the desired functionality. Check below example.

**<?php**

***class* Vehicle**

**{**

**public function move()**

**{**

**echo "Move the car";**

**}**

**}**

***class* Car**

**{**

**private *$vehicle*;**

**public function \_\_construct(Vehicle *$vehicle*)**

**{**

***$this*->vehicle = *$vehicle*;**

**}**

**public function accelarate()**

**{**

***$this*->vehicle->move();**

**}**

**}**

As you can see, we’ve [injected](https://www.amitmerchant.com/dependency-injection-container-php/) **Vehicle** class to the **Car** class through constructor and this way we can access the class members of **Vehilcle** class into the **Car** class. Now, if you want to use to use an another class called **Tyre** in class **Car**, all you have to do is to inject it’s instance in the constructor like so.

**<?php**

***class* Vehicle**

**{**

**public function move()**

**{**

**echo "Move the car";**

**}**

**}**

***class* Tire**

**{**

**public function addAlloys()**

**{**

**echo "Adding alloy wheels...";**

**}**

**}**

***class* Car**

**{**

**private *$vehicle*;**

**private *$tire*;**

**public function \_\_construct(Vehicle *$vehicle*, Tire *$tire*)**

**{**

***$this*->vehicle = *$vehicle*;**

***$this*->tire = *$tire*;**

**}**

**public function accelarate()**

**{**

***$this*->vehicle->move();**

***$this*->tire->addAlloys();**

**}**

**}**

This approach is called as “Composition over Inheritance” in object oriented programming and I’ve written [a whole article](https://www.amitmerchant.com/reasons-use-composition-over-inheritance-php/) around it.