






How to add elements to an empty array in PHP?

Asked 14 years, 1 month ago Modified 1 month ago Viewed 1.4m times

  If I define an array in PHP such as (I don't define its size):

642 `$cart = array();`

 Do I simply add elements to it using the following?




`$cart[] = 13;
$cart[] = "foo";
$cart[] = obj;`

Don't arrays in PHP have an add method, for example, `cart.add(13)` ?


[php](#) [arrays](#) [variables](#)

Share Edit Follow Flag

edited Apr 14, 2014 at 8:20

 [Peter Mortensen](#)
Morte 31k 21 106 130


asked Mar 24, 2009 at 9:35

 [AquinasTub](#)
8,315 6 22 15




Sorted by:

Highest score (default) 

9 Answers

 Both [array_push](#) and the method you described will work.

1019 `$cart = array();
$cart[] = 13;
$cart[] = 14;
// etc`

`//Above is correct. but below one is for further understanding
$cart = array();
for($i=0;$i<=5;$i++){
 $cart[] = $i;
}
echo "<pre>";
print_r($cart);
echo "</pre>";`

Is the same as:

```
<?php  
$cart = array();  
array_push($cart, 13);  
array_push($cart, 14);
```

```
// Or
$cart = array();
array_push($cart, 13, 14);
?>
```

Share Edit Follow Flag

edited Nov 22, 2018 at 8:58

answered Mar 24, 2009 at 9:42



Yoram de Langen

5,311 3 24 31



Bart S.

11.1k 1 19 26

- 223 ▲ As stated in the PHP documentation, if you're only pushing a single element every time (like in a loop) or a single element once, it's best to use the `$cart[] = 13` method not only because it's less characters to do the same operation, but it also doesn't impose the performance overhead of a function call, which `array_push()` would. Edit: But, great answer. Effectively the same, and majority of uses won't even notice a performance difference, but helps to know those nuances. – [Mattygabe](#) Jan 15, 2011 at 5:10 ✎
- 87 ▲ Is it just me or does the `$cart[] = ...` syntax, at first glance, look like a variable assignment and not an implicit `array_push`? – [Brad Hein](#) Feb 5, 2014 at 16:36 ✎
- 11 ▲ It definitely does to me. I wouldn't mind an explanation of why its **not** an assignment. – [limeandcoconut](#) May 20, 2014 at 4:05
- 4 ▲ `$cart[] = 13;` is faster. has less characters and looks better. – [Gal Bracha](#) Jul 28, 2014 at 6:54
- 33 ▲ I'll just offer my alternative viewpoint that it's VERY confusing for other language programmers to read the syntax of `cart[] = ...`, I've got experience with a lot of languages and I'd never guess that's what it does. – [Erti-Chris Eelmaa](#) Oct 13, 2016 at 18:02

It's better to not use [array_push](#) and just use what you suggested. The functions just add overhead.

105

```
//We don't need to define the array, but in many cases it's the best solution.
$cart = array();
```

```
//Automatic new integer key higher than the highest
//existing integer key in the array, starts at 0.
```

```
$cart[] = 13;
$cart[] = 'text';
```

```
//Numeric key
$cart[4] = $object;
```

```
//Text key (assoc)
$cart['key'] = 'test';
```

Share Edit Follow Flag

edited Jul 31, 2015 at 7:16

answered Mar 24, 2009 at 9:47



kamal pal


4,157 5 25 40






OIS

9,773 2 32 41

- 13 ▲ "If you're adding multiple values to an array in a loop, it's faster to use `array_push` than repeated `[] = statements`" php.net/manual/en/function.array-push.php#84959 – [Ollie Glass](#) Dec 18, 2010 at 17:15

3  Absolutely correct if your use-case is adding a single item or items one at a time. If all values are known at the same time, it's probably best just to use the `array_push` notation depending on how many items must be added the extra characters from re-typing the array name each time may be more of a performance hindrance than the function call over-head. As always, judgment should be exercised when choosing. Good answers! – [Mattygabe](#) Jan 15, 2011 at 5:13

2  This answer is the most complete. – [Lokiare](#) May 24, 2018 at 16:19

 1) `array_push()` has a *return value*, whereas the others do not. Perhaps this is the/one reason for its overhead? It seems to be a consensus to use the other methods, unless you need that return value. 2) *If you need elements to be added to the **end** of the array*, use either `array_push()` or `+=` method of concatenation (not shown in this answer), or `$cart[] = 13` methods. Using the named/numeric key method (`$cart[4] = $object` and `$cart['key'] = 'test'` methods do not *guarantee* the element will be added to the *end* of the array, only that it will be *in* the array. – [SherylHohman](#) Jun 30, 2020 at 23:40 

 @SherylHohman: This `$cart[]` will add values to the end of the array. – [OIS](#) Sep 11, 2020 at 5:48

 Based on my experience, solution which is fine(the best) when keys are not important:

19

```
$cart = [];
$cart[] = 13;
$cart[] = "foo";
$cart[] = obj;
```



Share Edit Follow Flag

edited Nov 5, 2020 at 14:09

answered Oct 14, 2015 at 13:15

[fico74](#)

7,815 7 54 89

 You can use [array_push](#). It adds the elements to the end of the array, like in a stack.

11

You could have also done it like this:

```
$cart = array(13, "foo", $obj);
```



Share Edit Follow Flag

edited Mar 24, 2009 at 10:03

answered Mar 24, 2009 at 9:37

[andi](#)

14.3k 9 47 46

```
$cart = array();
$cart[] = 11;
$cart[] = 15;
```

7

// etc

//Above is correct. but below one is for further understanding

```
$cart = array();
```



```
for($i = 0; $i <= 5; $i++){
    $cart[] = $i;
```

//if you write \$cart = [\$i]; you will only take last \$i value as first element in array.

```
}
echo "<pre>";
print_r($cart);
echo "</pre>";
```

Share Edit Follow Flag

answered Oct 23, 2019 at 1:59



unpluggeDloop

109 1 5



\$cart[] = \$i; - that part of code add elements to array ----> \$cart = [\$i]; - this will pass compiler but you will not get what you want – unpluggedloop Oct 28, 2019 at 12:08



REMEMBER, this method overwrites first array, so use only when you are sure!

4

```
$arr1 = $arr1 + $arr2;
```

[\(see source\)](#)

Share Edit Follow Flag

edited May 23, 2017 at 11:47



Community Bot

1 1

answered May 3, 2015 at 17:05



T.Todua

T.Tod 52.1k 19 228 233

1 Why the downvote, can someone explain why this is bad? is it insecure? – Sandy Jan 18, 2017 at 14:14

6 @SandyBeach it's not an answer – mateos Mar 25, 2017 at 6:39



2

```
$products_arr["passenger_details"]=array();
array_push($products_arr["passenger_details"],array("Name"=>"Isuru Eshan", "E-
Mail"=>"isuru.eshan@gmail.com"));
echo "<pre>";
echo json_encode($products_arr, JSON_PRETTY_PRINT);
echo "</pre>";
```



//OR

```
$countries = array();
$countries["DK"] = array("code"=>"DK", "name"=>"Denmark", "d_code"=>"45");
$countries["DJ"] = array("code"=>"DJ", "name"=>"Djibouti", "d_code"=>"253");
$countries["DM"] = array("code"=>"DM", "name"=>"Dominica", "d_code"=>"1");
foreach ($countries as $country){
    echo "<pre>";
    echo print_r($country);
    echo "</pre>";
}
```

Share Edit Follow Flag

edited Nov 22, 2018 at 6:42

answered Nov 22, 2018 at 6:37



Isuru Eshan

23 4



If you are trying to append to an associative array

0



```
//append to array
$countries["continent"] = "Europe";
```



Share Edit Follow Flag



answered Mar 29 at 10:07



Sammy

39 6



When one wants elements to be added with zero-based element indexing, I guess this will work as well:

-1



```
// adding elements to an array with zero-based index
$matrix= array();
$matrix[count($matrix)]= 'element 1';
$matrix[count($matrix)]= 'element 2';
...
$matrix[count($matrix)]= 'element N';
```



Share Edit Follow Flag

answered Jun 23, 2014 at 9:19



Gestix Team

17 1