

قبل از توضیح به نظر بنده باید نویسنده سایت این قسمت را نیز به آموزش اصلی اضافه کند(چون خیلی مهمه)

خوب در ابتدا باید با تابع **unset()** آشنا شویم :

تابع **unset** بطور کامل **یک متغییر** و **مقدار درون** آن را نابود می کند، بصورتی که در صورتی که می خواهید دوباره از آن متغییر استفاده کنید باید آن را مجددا تعریف کنید.

مثال درس فوق رو بررسی میکنم :

اگر مثال فوق را باز نویسی کنیم:

```
$arr = array(1, 2, 3, 4);  
foreach ($arr as &$value) {  
    $value = $value * 2;  
}
```

تا اینجا مقدار آرایه **arr** برابر **array(2, 4, 6, 8)** میشود از آنجایی که ما تابع **unset(\$value)** را به کار نبرده ایم لذا این متغییر به قوت خود باقی مانده و در واقع مرجع جزء **arr[3]** به حساب می آید.(زیرا در حلقه آخر این جزء به متغییر ارجاع داده شده است) حال اگر باقی کد را به صورت زیر بنویسیم (تابع **print\_r** جهت چاپ آرایه به کار میرود):

```
foreach ($arr as $key => $value) {  
    echo "{$key} => {$value} ";  
    print_r($arr);  
    echo "<br>";  
}
```

خروجی به صورت زیر خواهد بود :

```
0 => 2 Array ( [0] => 2, [1] => 4, [2] => 6, [3] => 2 )  
1 => 4 Array ( [0] => 2, [1] => 4, [2] => 6, [3] => 4 )  
2 => 6 Array ( [0] => 2, [1] => 4, [2] => 6, [3] => 6 )  
3 => 6 Array ( [0] => 2, [1] => 4, [2] => 6, [3] => 6 )
```

همان طور که می بینید در هر مرحله مقدار **value** در همان طور که تنظیم شده است در **arr[3]** قرار میگیرد لذا به جای ۸ عدد ۶ ظاهر میشود.

- نتیجه اینکه : همیشه بعد **foreach** یا **for** هر متغییری استفاده کردیم تابع **unset** را بر روی ان اعمال کنیم .

**unset:** همان مثال همراه

```
<?php
$arr = array(1, 2, 3, 4);
foreach ($arr as &$value) {
    $value = $value * 2;
}
unset($value);

foreach ($arr as $key => $value) {
    echo "{$key} => {$value} ";
    print_r($arr);
    echo "<br>";
}
?>
```