

Slide 3.9: PHP arrays
 Slide 3.11: PHP looping
[Home](#)



PHP Arrays (Cont.)

Associative Arrays (Cont.)

If the ID key of an associative array is a string, it could be doubly quoted, singly quoted, or not quoted.

```
<html><body>
<?php
    $hash["ID"] = 10;
    $hash['ID'] = 20;
    $hash[ID]   = 30;
    echo $hash["ID"] + $hash['ID'] + $hash[ID];
?>
</body></html>
```

output =

Check

Multidimensional Arrays

In a multidimensional array, each element in the main array can also be an array. And each element in the sub-array can be an array, and so on. The arrays could be associative arrays too. It seems variable interpolation does not work for a multidimensional array element. For example,

```
echo "$families[$arr2][$i]<br />";
```

is not working. You have to use the following command instead:

```
echo $families[$arr2][$i] . "<br />";
```

```
<html><body>
<?php
    $size = 5
    for ( $i=0; $i<$size; $i++ )
        for ( $j=0; $j<$size; $j++ )
            $arr[$i][$j] = 1;

    $total = 0;
    for ( $i=0; $i<$size; $i++ )
```

```
<html><body>
<?php
    $families = array (
        "Griffin" => array (
            "Peter",
            "Lois",
            "Megan"
        ),
        "Brown" => array (
```

```
    for ( $j=0; $j<$size; $j++ )  
        $total += $arr[$i][$j];  
    echo $total;  
?>  
</body></html>
```

Output = [Check](#)[Source](#)

```
        "Glenn",  
        "Andrew",  
        "Ella"  
    )  
);  
echo $families[][];  
?>  
</body></html>
```

Output = [Check](#)[Source](#)

[Slide 3.9: PHP arrays](#)
[Slide 3.11: PHP looping](#)
[Home](#)

