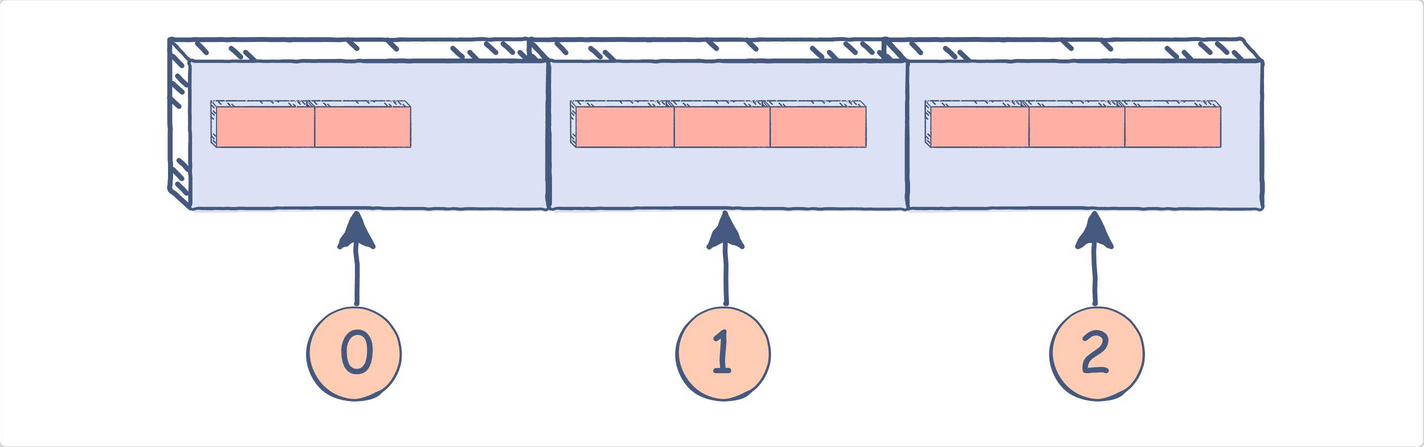
Multidimensional Array

In this lesson, we will learn how to work with multidimensional arrays.

We'll cover the following

What are Multidimensional Arrays?

PHP allows us to create multidimensional arrays. These arrays contain items that can be a single value or an array itself. These arrays can be thought of as an extension to the linear arrays. There is no limit to the dimensions of a multidimensional array. It can also be visualized as **an array of arrays**



Notice that in the figure above, the first index of the outer array is an array of two, whereas, the other two indices have arrays of three elements each. This means that PHP allows us to create multi-dimensional arrays where the elements (in this case array) at each index may vary in size.

Declaring Multidimensional Arrays

The declaration of *2-D arrays* is as follows:

```
$arrayName = array(array(), array()......array())
```

Arrays, in PHP, are resized dynamically, and attempt to write anything to non-existent element creates it and creates an entire array if needed. Before moving on to more complex multidimensional arrays, let us start with a simple example:

```
<?php
  $check=array("elephant", array("honey", "sad", 5));
  print_r($check);
?>
```

Notice, that in the above code snippet, the first index has a string “elephant”, the second index has an array with three elements (honey, sad, 5). Now, let’s declare an array that have arrays at all of its indices.

```
<?php
$comparisonAdjectives = array(
    array(
        "good",
        "better",
        "best"
    ) ,
    array(
        "bad",
        "worse",
        "worst"
    ) ,
    array(
        "tall",
        "taller",
        "tallest"
    )
);
print_r($comparisonAdjectives);
?>
```

An equivalent method of declaring the array above is the following:

```
<?php
$comparisonAdjectives[0][0] = "good";
$comparisonAdjectives[0][1] = "better";
$comparisonAdjectives[0][2] = "best";
$comparisonAdjectives[1][0] = "bad";
$comparisonAdjectives[1][1] = "worse";
$comparisonAdjectives[1][2] = "worst";
```

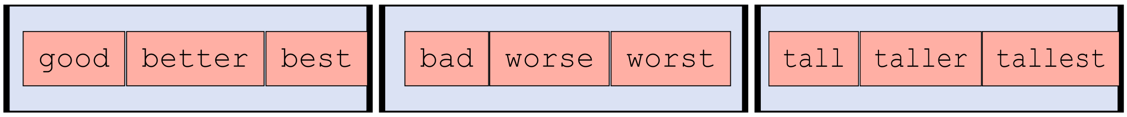
```
$comparisonAdjectives[2][0] = "tall";
$comparisonAdjectives[2][1] = "taller";
$comparisonAdjectives[2][2] = "tallest";

print_r($comparisonAdjectives);
?>
```

> **Note**: If keys are not specified by the user, PHP automatically creates an indexed array with indexes starting from zero

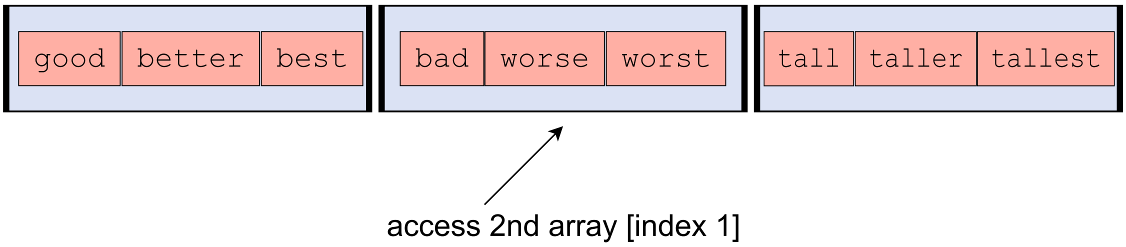
Accessing Values in Multidimensional Arrays

We can access a particular value from a multidimensional array using keys of subsequent arrays.

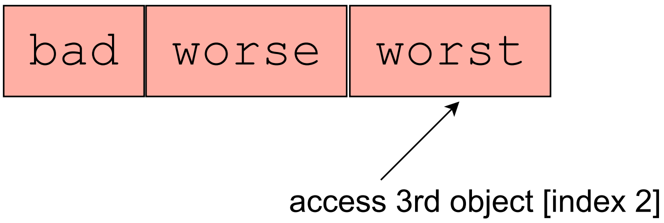
Below is a visualization for accessing **second** array's **third** value of a multidimensional array.



1 of 4



2 of 4



3 of 4

4 of 4

Run the code below to see how this is done:

```
<?php
$comparisonAdjectives = array(
    array(
        "good",
        "better",
        "best"
    ) ,
    array(
        "bad",
        "worse",
        "worst"
    ) ,
    array(
        "tall",
        "taller",
        "tallest"
    )
);
echo $comparisonAdjectives[1][2];
?>
```

In the code above, there is an indexed array `$comparisonAdjectives`. This array has three indexed arrays inside it. Each indexed array has 3 keys (indexes in this case) and they start from 0. Each key has an associated value with it. To access third value of second array, we use the `[1]` index to access the **second** array, followed by `[2]` to access the **third** value of this array. This is written as follows:

```
$comparisonAdjectives[1][2];
```

Indexed Array of Associative Arrays

Following is another example of a complex multidimensional array. It is an *indexed* array containing *associative* arrays

```
<?php
// Define a multidimensional array
$economy = array(
    array(
        "country" => "Germany",
        "currency" => "Euro",
    ),
    array(
        "country" => "Switzerland",
        "currency" => "Swiss Franc",
    ),
    array(
        "country" => "England",
        "currency" => "Pound",
    )
);

echo "Currency of Germany is: " . $economy[0]["currency"]; // Access array at [0] index

?>
```

In the code above, there is an indexed array `$economy`. This array has three associative arrays inside it. Each associative array has 2 keys: `"country"` and `"currency"`. Each key has an associated value with it. To access `"currency"` of the first array, we use the `[0]` index to access the first array, followed by `["currency"]` to access the value in this key. This is written as follows:

```
$economy[0]["currency"]
```

In the next lesson, you will be given a challenge involving multidimensional arrays.