

## PHP Array

An array is a special variable, which can hold more than one value at a time. An array is a data structure that stores one or more similar type of values in a single value. For example, if you want to store 100 numbers then instead of defining 100 variables it's easy to define an array of 100 length.

An array index always starts at 0.

There are three different kind of arrays and each array value is accessed using an ID c which is called array index.

- **Numeric array** – An array with a numeric index. Values are stored and accessed in linear fashion.
- **Associative array** – An array with strings as index. This store element values in association with key values rather than in a strict linear index order.
- **Multidimensional array** – An array containing one or more arrays and values are accessed using multiple indices

In PHP, the `array()` function is used to create an array

## PHP Arrays Rock!

- Better than Python Dictionaries
- Better than Java Hash Maps
- PHP Arrays have all the benefits of Python Dictionaries but they can also maintain the order of the items in the array

# Key / Value

```
<?php
    $stuff = array("name" => "Chuck",
                   "course" => "WA4E");
    echo $stuff["course"], "\n";
?>
```

WA4E

# Integer Indices

```
<?php
    $stuff = array("Hi", "There");
    echo $stuff[1], "\n";
?>
```

There

# Dumping an Array

The function `print_r()` shows PHP data - it is good for debugging.

```
<?php
    $stuff = array("name" => "Chuck",
                   "course" => "WA4E");

    echo("<pre>\n");
    print_r($stuff);
    echo("\n</pre>\n");
?>
```

```
Array(
    [name] => Chuck
    [course] => WA4E
)
```

# var\_dump vs. print\_r

```
<?php
    $stuff = array("name" => "Chuck",
                   "course" => "SI664");

    var_dump($stuff);
?>
```

```
array(2) {
    ["name"]=>
    string(5) "Chuck"
    ["course"]=>
    string(5) "SI664"
}
```

# var\_dump() and false

```
<?php
    $thing = FALSE;
    echo("One\n");
    print_r($thing);
    echo("Two\n");
    var_dump($thing);
?>
```

```
One
Two
bool(false)
```

# Building Up an Array

You can allocate a new item in the array and append a value at the same time using empty square braces [ ] on the right hand side of an assignment statement.

```
$va = array();  
$va[] = "Hello";  
$va[] = "World";  
print_r($va);
```

```
Array(  
    [0] => Hello  
    [1] => World  
)
```

# Building Up an Array

You can also add new items in an array using a key.

```
$za = array();  
$za["name"] = "Chuck";  
$za["course"] = "WA4E";  
print_r($za);
```

```
Array(  
    [name] => Chuck  
    [course] => WA4E  
)
```

# Looping Through an Array

```
<?php
    $stuff = array("name" => "Chuck",
                  "course" => "SI664");
    foreach($stuff as $k => $v ) {
        echo "Key=", $k, " Val=", $v, "\n";
    }
?>
```

Key=name Val=Chuck  
Key=course Val=SI664

# Looping Through an Array

```
<?php
    $stuff = array("Chuck", "SI664");
    foreach($stuff as $k => $v ) {
        echo "Key=", $k, " Val=", $v, "\n";
    }
?>
```

Key=0 Val=Chuck  
Key=1 Val=SI664

# Counted Loop Through an Array

```
<?php
    $stuff = array("Chuck", "SI664");
    for($i=0; $i < count($stuff); $i++) {
        echo "I=", $i, " Val=", $stuff[$i], "\n";
    }
?>
```

I=0 Val=Chuck

I=1 Val=SI664

## Multidimensional array

A multi-dimensional 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. Values in the multi-dimensional array. are accessed using multiple index

## Arrays of Arrays

The elements of an array can be many things other than a string or integer. You can even have objects or other arrays.

```
$products = array(
    'paper' => array(
        'copier' => "Copier & Multipurpose",
        'inkjet' => "Inkjet Printer",
        'laser' => "Laser Printer",
        'photo' => "Photographic Paper"),
    'pens' => array(
        'ball' => "Ball Point",
        'hilite' => "Highlighters",
        'marker' => "Markers"),
    'misc' => array(
        'tape' => "Sticky Tape",
        'glue' => "Adhesives",
        'clips' => "Paperclips")
);
```

```
echo $products["pens"]["marker"];
```

Markers



# Array Functions

`array_key_exists($key, $ar)` - Returns TRUE if key is set in the array

`isset($ar['key'])` - Returns TRUE if key is set in the array

`count($ar)` - How many elements in an array

`is_array($ar)` - Returns TRUE if a variable is an array

`sort($ar)` - Sorts the array values (loses key)

`ksort($ar)` - Sorts the array by key

`asort($ar)` - Sorts array by value, keeping key association

`shuffle($ar)` - Shuffles the array into random order

```
$za = array();
$za["name"] = "Chuck";
$za["course"] = "WA4E";

if (array_key_exists('course',$za) ) {
    echo("Course exists\n");
} else {
    echo("Course does not exist\n");
}

echo isset($za['name']) ? "name is set\n" : "name is not set\n";
echo isset($za['addr']) ? "addr is set\n" : "addr is not set\n";
```

Course exists  
name is set  
addr is not set

# Exploding Arrays

```
$inp = "This is a sentence with seven words";
$temp = explode(' ', $inp);
print_r($temp);
```

```
Array(
    [0] => This
    [1] => is
    [2] => a
    [3] => sentence
    [4] => with
    [5] => seven
    [6] => words
)
```



## HTTP & PHP Arrays

