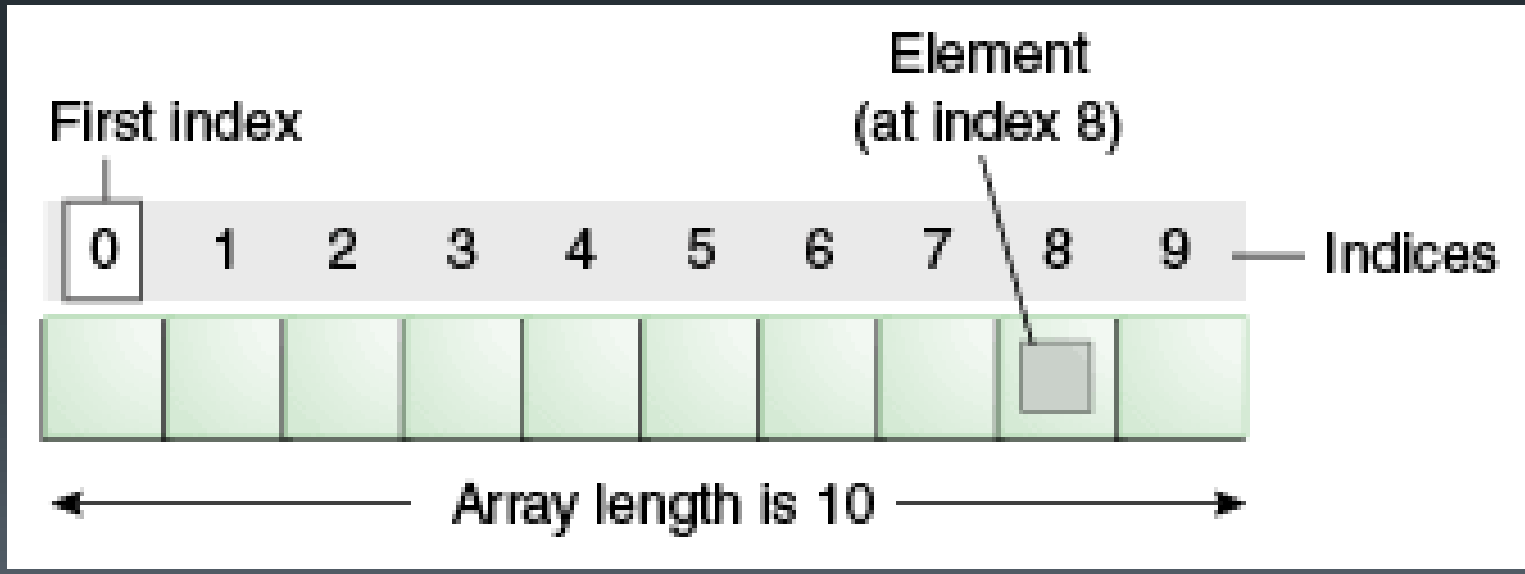
PHP Arrays

AGENDA

Index based Array
Associative Array
Accessing Array
Looping through Array
Some useful Functions

Array

- What is Array?



Indexing of Array

- Integer based Index
- Alphanumeric Index

Create a Indexed based Array

```
<?php
$fruits = array("Apple", "Mango", "Orange", "Kiwi");
print_r($fruits);
?>
```

```
Array (
[0] => Apple
[1] => Mango
[2] => Orange
[3] => Kiwi
)
```

Integer and Float Array

```
$array_int = array(1,2,3,4);
print_r($array_int);

$array_float = array(1.2, 2.2, 3.09,4.87);
print_r($array_float);
```

Array of String

```
<?php
$array_string = array("abc","xyz");
print_r($array_string);
?>
```

Mixed Array

```
<?php
$array_mixed = array("abc","xyz", 1,2 , 3.3);
print_r($array_mixed);
?>
```

Generalize a Array

- An array in PHP is actually an ordered map
- An array can be created using the array() language construct

array(key => value, key2 => value2, key3 => value3, ...)

array(1, 2) is preferred over *array(1, 2,)*

last array element is optional and can be omitted

Another way to create Indexed based Array using Implicit location

Fruits Array

```
<?php
$fruits[] = "Apple";
$fruits[] = "Mango";
$fruits[] = "Banana";
$fruits[] = "Watermelon";
print_r($fruits);
?>
```

Output

```
Array (
[0] => Apple
[1] => Mango
[2] => Banana
[3] => Watermelon
 )
```

Create Array using Explicit location

```
<?php
$fruits[0] = "Apple";
$fruits[1] = "Mango";
$fruits[2] = "Banana";
$fruits[3] = "Watermelon";
print_r($fruits);
?>
```

Accessing Array using For Loop

```
<?Php
$fruits[0] = "Apple";
$fruits[1] = "Mango";
$fruits[2] = "Banana";
$fruits[3] = "Watermelon";

for ($j = 0 ; $j < 4 ; ++$j)
 echo "$j: $fruits[$j] \n";
?>
```

```
0: Apple
1: Mango
2: Banana
3: Watermelon
```

Associative Arrays

- Keep Index based on Key
- Easy to remember
- More logical
- And work better for large code and multiple persons team

Associative Arrays for Feedback

```
<?php
$feedback['google'] = "Best Search Engine";
$feedback['yahoo'] = "Good for basic news";
$feedback['instagram'] = "Love for pics";
$feedback['facebook'] = "Stay connected with your friends";

echo $feedback['yahoo'];
?>
```

Output : Good for basic news

Assignment Using the array Keyword (format *index* => *value*)

```
<?php
$feedback = array(
'google' =>"Best Search Engine",
'yahoo' => "Good for basic news",
'instagram' => "Love for pics",
'facebook' => "Stay connected with your friends“
);
echo $feedback['yahoo'];
?>
```

foreach loop

```
$j = 0;
foreach($feedback as $item)
{
echo "$j: $item \n";++$j;
}
```

0: Best Search Engine

1: Good for basic news

2: Love for pics

3: Stay connected with your friends

Accessing Key and Value

- <?php
- $feedback = array('google' =>"Best Search Engine",
- 'yahoo' => "Good for basic news",
- 'instagram' => "Love for pics",
- 'facebook' => "Stay connected with your friends");
-
- while (list($item, $description) = each($feedback))
- echo "$item: $description \n";
-
- ?>

Output

- google: Best Search Engine
- yahoo: Good for basic news
- instagram: Love for pics
- facebook: Stay connected with your friends

Multi-D Array

- <?php
- $feedback = array(1,2,array(3,4));
- print_r($feedback);
- ?>

- Array
- (
- [0] => 1
- [1] => 2
- [2] => Array
- (
- [0] => 3
- [1] => 4
-)

-)

M-D Associative Array

- $feedback = array(
- 'user1' => array('google' =>"Best Search Engine",
- 'yahoo' => "Good for basic news",
- 'instagram' => "Love for pics",
- 'facebook' => "Stay connected with your friends"),
-
- 'user2' => array('google' =>"Search Engine",
- 'yahoo' => "I am not using",
- 'instagram' => "For young",
- 'facebook' => "For friends"),
-
- 'user3' => array('google' =>"I love it",
- 'yahoo' => "use only yahoo mail",
- 'instagram' => "I follow celebraity",
- 'facebook' => "I like it")
-
-);

Accessing M-D Associative Array

```
foreach($feedback as $user => $users)
        foreach($users as $key => $value)
                echo "$user:\t$key\t($value) \n";
```

Output

- user1: google (Best Search Engine)
- user1: yahoo (Good for basic news)
- user1: instagram (Love for pics)
- user1: facebook (Stay connected with your friends)
- user2: google (Search Engine)
- user2: yahoo (I am not using)
- user2: instagram (For young)
- user2: facebook (For friends)
- user3: google (I love it)
- user3: yahoo (use only yahoo mail)
- user3: instagram (I follow celebraity)
- user3: facebook (I like it)

Array Functions

- is_array
- Count
- Sort
- Shuffle
- Explode
- Extract
- Compact
- Reset
- End

is_array()

- To check variable is array or not
- echo (is_array($fruits)) ? "Is an array" : "Is not an array";

count($array)

- Count all the top elements in a array

count($feedback) ; # 3

- $p1 = array("Copier", "Inkjet", "Laser", "Photo");
- echo count($p1);#4

count(array,top_level)

- Top Level – 0 /1
- 0 : counting only top level array
- 1 : force recursive counting of sub arrays

count($feedback); # 3

count($feedback); # 15 (12+3)

- $p1 = array("Copier", "Inkjet", "Laser", "Photo");
- echo count($p1,1);#4

sort(array)

- Return true on success
- Return false on error

```
<?php
$array = array(2,4,1,5,3);
echo "\n";
echo sort($array);
print_r($array);

$fruits = array("Mango",
"Apple", "Fig", "Grapes");
echo sort($fruits);
print_r($fruits);
?>
```

- ?>
- 1
- Array
- (
- [0] => 1
- [1] => 2
- [2] => 3
- [3] => 4
- [4] => 5
-)
- 1
- Array
- (
- [0] => Apple
- [1] => Fig
- [2] => Grapes
- [3] => Mango
-)

Force sorting to be made either numerically or as strings

- sort($array, SORT_NUMERIC);
- sort($array, SORT_STRING);

- Reverse Order

- rsort($fred, SORT_NUMERIC);
- rsort($fred, SORT_STRING);

shuffle(array)

- Return true on success
- Return false on error

- <?php
- $nos = array(1,2,3,4,5,6,7,8,9);
- shuffle($nos);
- print_r($nos)
- ?>

- Array
- (
- [0] => 7
- [1] => 2
- [2] => 6
- [3] => 3
- [4] => 4
- [5] => 8
- [6] => 1
- [7] => 9
- [8] => 5
-)

explode()

- Convert string into array using some separator
- Separator may be any char
- <?php
- $temp = explode(' ', "I am a coder");
- print_r($temp);
- ?>
- Array ([0] => I [1] => am [2] => a [3] => coder)

explode()

- <?php
- $temp = explode('#', "BCCD#ADDF#EEFG#AAAA");
- print_r($temp);
- ?>

- Array ([0] => BCCD [1] => ADDF [2] => EEFG [3] => AAAA)

extract()

- Convert key/value pairs from an array into PHP variables
- Example : $_GET or $_POST variables as sent to a PHP script by a form

- extract($_GET);

compact()

- Inverse of extract()

- <?php
- $fname = "Krish";
- $sname = "Richard";
- $address = "A-3,NY, USA";
- $contact = "854716632";
- $contact = compact('fname', 'sname', 'address', 'contact');
- print_r($contact);
- ?>

output

- Array
- (
- [fname] => Krish
- [sname] => Richard
- [address] => A-3,NY, USA
- [contact] => 854716632
-)

Thank You

Q& A