Lecture 5

Arrays in PHP

Problem

Define more than one variable for similar purpose

Example:

Grades of a student group – define 30 variable for them

Solution:

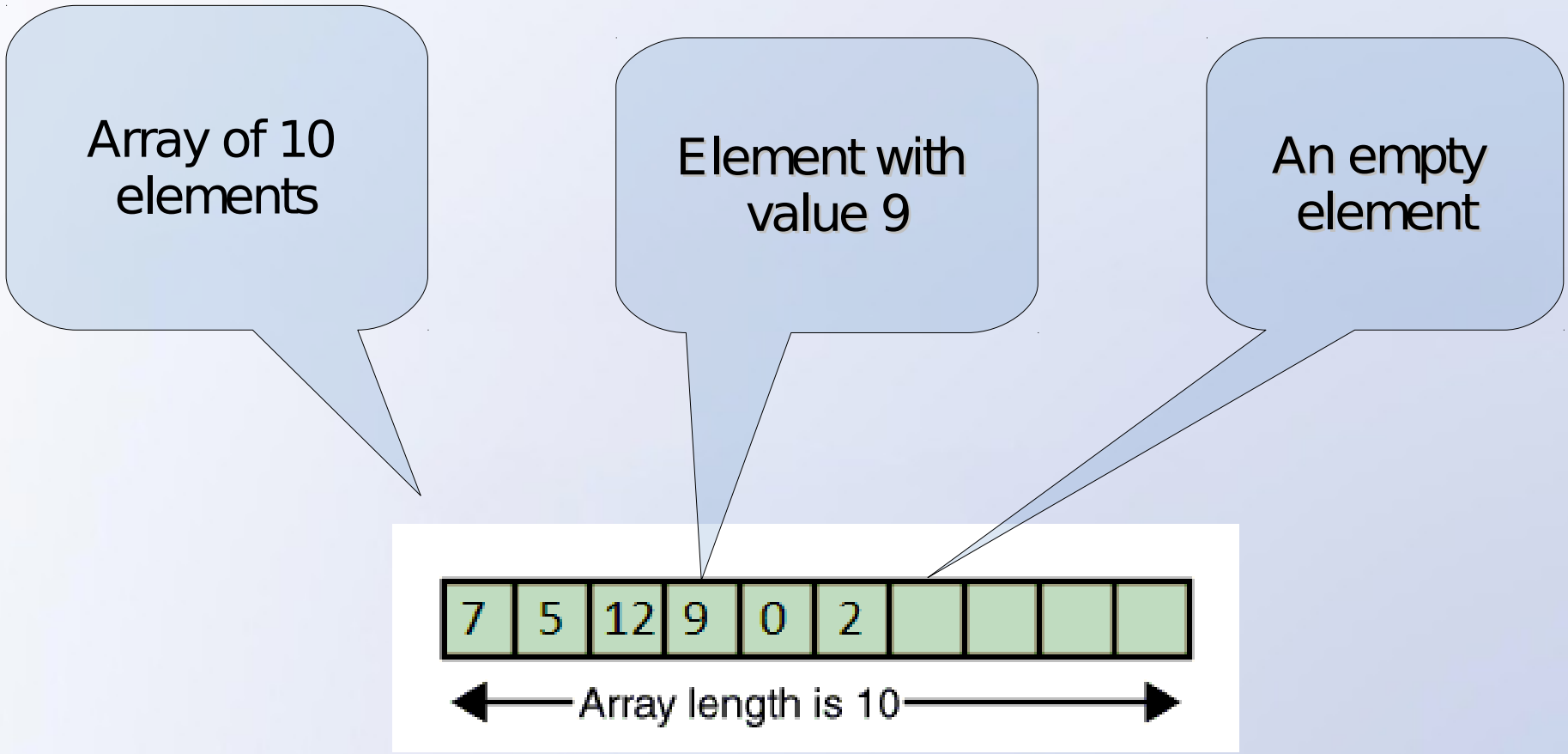
Define 30 variable of type double to hold the information

Is this OK?

What is an array?

- An array is a sequence of elements
- The order of the elements remains the same
- Arrays are used to collect together data, such as people's names and perform operations on this data as easily as possible.
- The elements are accessed through an *index* or another value

What is an array?



IT TALENTS
Training Camp

Working With Arrays

How to make an Array

- Arrays are created using the ***array*** construct, or the **[ ]** construct
- Example

```php
<?php
// php version < 5.4
$arrayOldWay = array('Гошо', 'Пешо', 'Иван', 'Митко');

// php version >= 5.4
$arrayNewWay = ['Гошо', 'Пешо', 'Иван', 'Митко'];

```

Accessing the elements

- Every element in an array has its own index.
- The indexes starts from 0 for the first element, 1 for the second and so on, all the way to the N-1 for the Nth element.

Accessing the elements(2)

- We use [] brackets to access the individual elements of an array :

```php
<?php
$array = array('Гошо', 'Пешо', 'Иван', 'Митко');

var_dump($array[2]);
```

This will print Иван

Accessing the elements(3)

If you access non existing index in the array you will get PHP Notice.

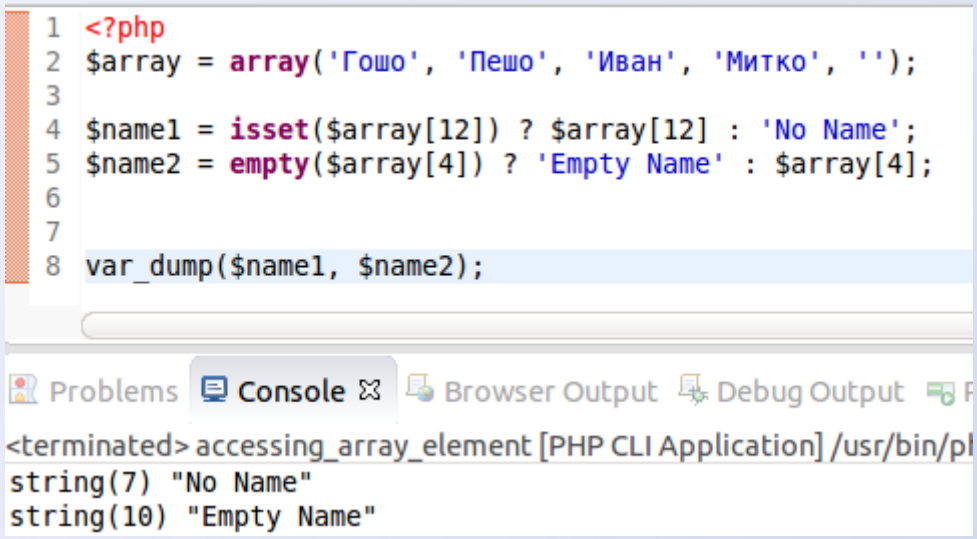
```php
<?php
$array = array('Гошо', 'Пешо', 'Иван', 'Митко');

$name = $array[12];

var_dump($name);
```

```
Problems  Console  Browser Output  Debug Output  Progress  Synchronize  Search
<terminated> accessing_array_element [PHP CLI Application] /usr/bin/php
PHP Notice:  Undefined offset: 12 in /home/vasil/server/php_course_snippets/Arrays/accessing_array_element.php on line 4
PHP Stack trace:
PHP   1. {main}() /home/vasil/server/php_course_snippets/Arrays/accessing_array_element.php:0
NULL
```

This is BAD, try to avoid this at any cost!

Safe array element access

To avoid errors when you access elements in array you should use **isset** or **empty** PHP functions to check if the index exists in the array, and then access the element on the index.



Read more about isset and empty at http://php.net

IT TALENTS
Training Camp

Number of elements

- count($arr) returns tne number of elements in array

```php
<?php

$weekdays = [
    'Monday', 'Tuesday', 'Wednesday', 'Thursday',
    'Friday', 'Saturday', 'Sunday'
];

$count = count($weekdays);

var_dump($count);
```

Modifying Array Element's Value

- It is easy, just use the [] brackets to access the element by its index and assign to it the new value.

```php
<?php
    $nums = array(
        5, 4, 1, 6, 2, 7, 4, 10
    );

    $nums[3] = 9;
    $nums[1]++;

    $nums[5] = $nums[1] + $nums[2];

    $index = 6;
    $nums[$index] = 2;

    print_r($nums);
?>
```

Array with numbers.

Change 4-th element.

Increment second element.

Using a variable to point the index.

Deleting elements of array

- To delete element or multiple elements

```php
<?php

$array = [1, 2, 3, 4, 5, 6];
// single element deletion
unset($array[0]);

// multiple elements deletion
unset($array[0], $array[1], $array[2]);

var_dump($array);
```

Exercises

1. Create an array with some integer elements

2. Modify values of the first 3 elements

3. Print the second element

4. Multiply the 3rd element with 2nd,set the result to 5th, and then print it

5- Bonus Task : Create console script which creates array with 5 elements. While the array is not empty ask user to enter a index of the array and delete the element at that index from the array. If the index does not exist in the array print a message to inform the user. Otherwise print a message for successful operation

An Array with Keys

There is a powerful magic here…

- We can create an array and specify our own index key at the same time
- But we can put the index items in a different order and even missed out an index number

```php
$users = array(
        2 => 'Пешо',
        3 => 'Гошо',
        4 => 'Иван',
        5 => 'Митко'
);
```

```php
$users = array(
        5 => 'Пешо',
        4 => 'Гошо',
        7 => 'Иван',
        1 => 'Митко'
);
```

Non-Numeric Keys

PHP has one more trick for you..

- Arrays don't have to have a numerical key

They are known as **associative** arrays. They are very important to understand and are esential in PHP development

```php
<?php

$users = [
        'pesho' => 'Пешо',
        'gosho' => 'Гошо',
        'ivan' => 'Иван',
        'mitko' => 'Митко',
];
```

Example

Map username to password for this user.

```php
<?php

$passForUser = [
        'pesho' => '1234',
        'kiro' => '1234',
        'miro' => '1234',
];

echo $passForUser['pesho'], PHP_EOL;
```

Get the stored password for user *pesho*.

Iterating the array

•The *foreach* loops are specifically designed for iterating through arrays.

```
foreach (array as value) {
    expression;
    expression;
    ...
}
```

Example-Iterating with foreach cycle

```php
<?php

$users = [
    2 => 'Pesho',
    3 => 'Gosho',
    4 => 'Ivan',
    5 => 'Mitko',
];

foreach ($users as $user) {

    echo $user, PHP_EOL;

}
```

Example(2) – Iterating with for cycle

```php
<?php

$users = [
      'Pesho',
      'Gosho',
      'Ivan',
      'Mitko',
];

$len = count($users);

for ($i = 0; $i < $len; $i++) {

    echo $users[$i], PHP_EOL;

}
```

Retrieve all elements by index.

The *foreach* loop second form

```
foreach (array as key => value) {
    expression;
    expression;
    ...
}
```

Get all indexes(or keys) along with values.

How does the foreach loop work in PHP

- **foreach** works only on arrays and objects.
- It issues an error when you use it on anything else.
- The arrays in PHP have internal pointer which points to the first element.
- On each iteration the internal pointer is moved to the next element of the array.
- When the loop is over the internal pointer is reset to the first position in the array.

Example

```php
<?php

$nums = [
    5, 4, 1, 6, 2, 7, 4, 10
];

foreach ($nums as $index => $value) {
    echo $index, ' => ', $value, PHP_EOL;
}
```

Advanced Example

Represents each key

Represents each value

```php
<?php

$klass = [
    'Номер 1' => 'Ани',
    'Номер 2' => 'Боби',
    'Номер 3' => 'Вики',
    'Номер 4' => 'Георги',
    'Номер 5' => 'Димитър',
    'Номер 6' => 'Евгени',
    'Номер 7' => 'Живка',
];

foreach ($klass as $number => $name) {
    echo $number, ' в клас е ', $name, PHP_EOL;
}
```

Most common usages

```php
<?php
$array = [
    'one' => 1,
    'two' => 2,
    'three' => 3,
    'four' => 4,
];

foreach ($array as $key => $value) {
    echo $key, ' => ', $value, PHP_EOL;
}

foreach ($array as $k => $v) {
    echo $k, ' => ', $v, PHP_EOL;
}

foreach ($array as $key => $val) {
    echo $key, ' => ', $val, PHP_EOL;
}

var_dump($val);
```

Be aware that **$key** a **$value** are accessibe
After the loop ends

Exploding a String Into An Array

We can split a string to an array containing chunks which are defined by a specific separator.

```
$result = explode($delimiter, $string);
```

```php
<?php

$planets= "Mercury, Venus, Earth, Mars, Jupiter, Saturn, Uranus,
Neptune";

$planets = explode(", ", $planets);

print_r($planets);

?>
```

Now **$planets** is an array with all planets as values.

Exploding a String Into An Array(3)

The explode() function
- Accepts two parameters
  - delimiter - The boundary string.
  - string - The input string

- Returns an array of strings created by splitting the string parameter on boundaries
formed by the delimiter.

IT TALENTS
Training Camp

Imploding an Array

Of course we could do the opposite action and we can implode an array to a string :

```
$result = implode($glue, $pieces);
```

```php
<?php

   $planets= array(
        "Mercury",
        "Venus",
        "Earth",
        "Mars",
        ...
   );

   $planets = implode(", ", $planets);

   echo $planets;

?>
```

Now **$planets** is a string with all planets in the array separated with ,

Exercise

1. Enter an array elements from the console
2. Print its max number
3. Print the array
4. Print the max number

Exercise (Advanced)

1. Invert given array using two approaches:

   - Using second array

   - Without second array

2. Write a program, which reads an array and then creates a new array, which is with 1 element bigger than the original one. All elements, except the first one, should be the previous element, multiplied with the current index.

For example:

[9 2 4 -3 7 5]

[6 9 4 12 -12 35 30]

Array Comparison And Union

Example	Name	Result
$a + $b	Union	Union of *$a* and *$b*.
$a == $b	Equality	**TRUE** if *$a* and *$b* have the same key/value pairs.
$a === $b	Identity	**TRUE** if *$a* and *$b* have the same key/value pairs in the same order and of the same types.
$a != $b	Inequality	**TRUE** if *$a* is not equal to *$b*.
$a <> $b	Inequality	**TRUE** if *$a* is not equal to *$b*.
$a !== $b	Non-identity	**TRUE** if *$a* is not identical to *$b*.

Array Union with +

```php
<?php

$a = [1, 2, 3, 4, 5];
$b = [6, 7, 8, 9, 10];

$c  = $a + $b;
var_dump($c);

```

```
array(5) {
  [0] =>
  int(1)
  [1] =>
  int(2)
  [2] =>
  int(3)
  [3] =>
  int(4)
  [4] =>
  int(5)
}
```

```php
<?php

$a = ['one' => 1, 'two' => 2,
        'three' => 3, 'four' => 4, 'five' => 5];
$b = ['six' => 6, 'seven' => 7, 'eight' => 8,
        'nine' => 9, 'ten' => 10];

$c  = $a + $b;
var_dump($c);
```

```
array(10) {
  'one' =>
  int(1)
  'two' =>
  int(2)
  'three' =>
  int(3)
  'four' =>
  int(4)
  'five' =>
  int(5)
  'six' =>
  int(6)
  'seven' =>
  int(7)
  'eight' =>
  int(8)
  'nine' =>
  int(9)
  'ten' =>
  int(10)
}
```

IT TALENTS
Training Camp

Array Union with **array_merge**

```php
<?php

$a = [1, 2, 3];
$b = [4, 5, 6];

$c = array_merge($a, $b);
var_dump($c);
```

```
array(6) {
  [0] =>
  int(1)
  [1] =>
  int(2)
  [2] =>
  int(3)
  [3] =>
  int(4)
  [4] =>
  int(5)
  [5] =>
  int(6)
}
```

```php
<?php
$a = ['first' => 1, 'second' => 2, 'third' => 3];
$b = ['first' => 4, 'second' => 5, 'third' => 6];

$c = array_merge($a, $b);
var_dump($c);
```

```
array(3) {
  'first' =>
  int(4)
  'second' =>
  int(5)
  'third' =>
  int(6)
}
```

IT TALENTS
Training Camp

Exercises

1. Enter an array from console and multiply even elements by 2 and divide odd by 2.

2. Enter two arrays and compare if the sums of their elements are equal.

3. Write a program to randomly generate an array of a given size.

4. Enter two arrays A and B. Then construct new array C where C consists of the highest 5 numbers from A and B.

Adding an element to an array

- Method one: just use empty [] brackets and assign new value.
- Method two: just use [] brackets with new key and assign new value to it.

```php
<?php

    $users = array(
        'lyubo' => 'galabov',
        'daniel' => 'radev',
        'emil' => 'dimitrov'
    );

    $users[] = 'Unknown';
    $users['viktor'] = 'hristov';

    print_r($users);

?>
```

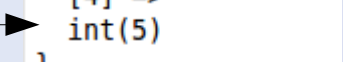
What will be the index of this element ?

Difference between arrays

- You can check this by using two functions in PHP – array_diff and array_diff_assoc
- **array_diff($array1, $array2, …)** - returns the values of **$array1**, that are not present in other arrays

```php
<?php

$a = [1, 2, 3, 4, 5];
$b = [1, 2, 3];

$c = array_diff($a, $b);

var_dump($c);
```

```
array(2) {
  [3] =>
  int(4)
  [4] =>
  int(5)
}
```



Difference between arrays(2)

- **array_diff_assoc($array1, $array2, ...)** - same as **array_diff** but also checks if the keys are the same

```php
<?php

$a = ['first' => 1, 'second' => 2, 'third' => 3];
$b = ['first_entry' => 1, 5, 6];

$c = array_diff_assoc($a, $b);

var_dump($c);
```

```
array(3) {
  'first' =>
  int(1)
  'second' =>
  int(2)
  'third' =>
  int(3)
}
```

Note that the values are equal,
but the keys are different and the entry
is in the returned array

Similarity between arrays

You can easily get the common elements between two and more arrays using the functions **array_intersect** and **array_intersect_key**

- **array_intersect($array1, $array2, …)** - returns array of all elements from the first array contained in all other arguments

```php
<?php

$a = [1, 2, 3, 4, 5];
$b = [1, 2, 3];

$c = array_intersect($a, $b);

var_dump($c);
```

```
array(3) {
  [0] =>
  int(1)
  [1] =>
  int(2)
  [2] =>
  int(3)
}
```

All the elements are present int both arrays

IT TALENTS Training Camp

Similarity between arrays(2)

- **array_intersect_key($array1, $array2, ...)** - returns array of all elements on keys from the first array contained in all other arguments

```php
<?php

$a = ['first' => 1, 'second' => 2, 'third' => 3];
$b = ['first' => 'one',  2, 3];

$c = array_intersect_key($a, $b);

var_dump($c);
```

```
array(1) {
  'first' =>
  int(1)
}
```

Only the 'first' key is present in both arrays, so the result array contains the key and value from the first array

How to get part of the array

- **array_slice($array, $offset, [$length])** - returns part of the $array, $length long, from the $offset. If $length is ommited it returns the rest of the array from the $length

```php
<?php

$array = [1, 2, 3, 4, 5];

var_dump(array_slice($array, 2));
```

```
array(3) {
  [0] =>
  int(3)
  [1] =>
  int(4)
  [2] =>
  int(5)
}
```

All the elements of the array from the second (exclusive) to the last

IT TALENTS Training Camp

Randomizing array elements

- You can easy sort the PHP array in random order using **shuffle** function

```php
<?php

$array = [1, 2, 3, 4, 5];

shuffle($array);

var_dump($array);
```

```
array(5) {
  [0] =>
  int(3)
  [1] =>
  int(4)
  [2] =>
  int(5)
  [3] =>
  int(2)
  [4] =>
  int(1)
}
```

Resources

http://php.net/manual/en/function.isset.php

http://php.net/manual/en/function.empty.php

http://php.net/manual/en/function.explode.php

http://php.net/manual/en/function.implode.php

http://php.net/manual/en/function.array-merge.php

http://php.net/manual/en/function.array-diff.php

http://php.net/manual/en/function.array-diff-assoc.php

http://php.net/manual/en/function.array-intersect.php

http://php.net/manual/en/function.array-intersect-key.php

http://php.net/manual/en/function.array-slice.php

http://php.net/manual/en/function.shuffle.php

Summary

- What is array
- How to declare and initialize array
- How to access and change elements
- How to get the array length
- How to read an array from the console