

Лекція №8 (частина 2)

"Типи даних мови PHP"

Викл. Коваль В.В.

ОСІТ

2021р.



МАСИВИ. ФУНКЦІЇ ДЛЯ РОБОТИ З МАСИВАМИ

МАСИВИ

- списки (або індексні масиви),
- асоціативні масиви

Утворення масиву

1. `$array_name = array("key1"=>"value1", "key2"=>"value2");`
2. `$array_name["key1"] = value1;`
3. `$array = ["key1" => "value1", "key2" => "value2",];`

де key – integer, string, value – будь-який тип

Приклад,

```
<?
```

```
$del_items = array("10"=>"Наука и жизнь", "12"=>"Інформатика");
```

```
$del_items["13"] = "Програмування";
```

```
?>
```

Параметр `key` необовязковий. Якщо він не зазначений, PHP буде використовувати попереднє найбільше значення ключа типу `integer`, збільшене на 1.

Приклад,

```
<?php
    $array = array(
        "a",
        "b",
        6 => "c",
        "d",
    );
    var_dump($array);/повертає тип і значення змінної
?>
```

Результат,

```
array(4) { [0]=> string(1) "a"
            [1]=> string(1) "b"
            [6]=> string(1) "c"
            [7]=> string(1) "d" }
```


Багатомірні масиви

```
<?php
$array = array(
    "foo" => "bar",
    42 => 24,
    "multi" => array(
        "dimensional" => array(
            "array" => "foo"
        )
    )
);

var_dump($array["foo"]);
var_dump($array[42]);
var_dump($array["multi"]["dimensional"]["array"]);
?>
```

```
<?php
$array = array (
    'websites' => array (
        'Search' => 'Google',
        'Social' => 'Facebook',
        'News' => 'NY Times'
    ),
    'friends' => array (
        'Chris',
        'Jim',
        'Lynn',
        'Jeff',
        'Joanna'
    )
);
```

Операції з масивами

- Додавання

<?

```
$a = array("i"=>"Інформатика", "m"=>"Математика");
```

```
$b = array("i"=>"Історія", "m"=>"Біологія", "f"=>"Фізика");
```

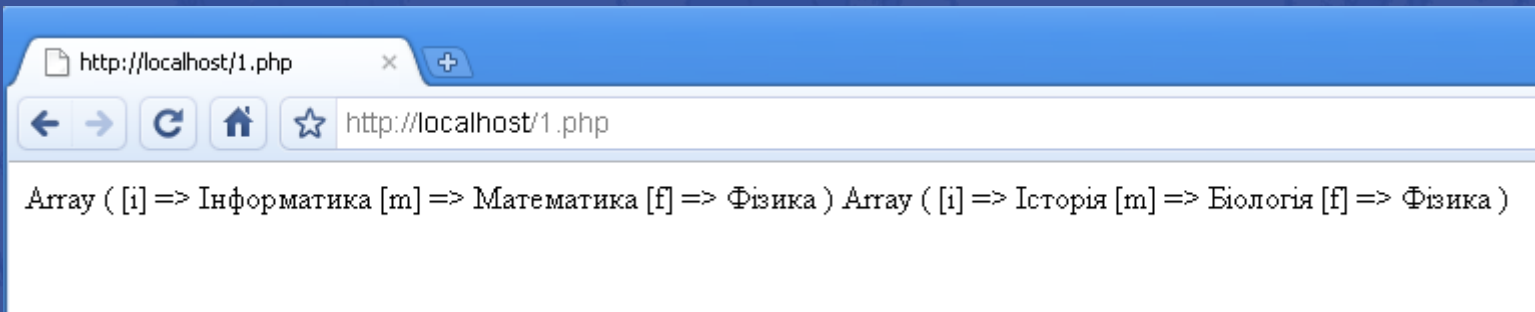
```
$c = $a + $b;
```

```
$d = $b + $a;
```

```
print_r($c);
```

```
print_r($d);
```

?>



Порівняння

Рівність – співпадіння пар ключ/значення елементів масивів.

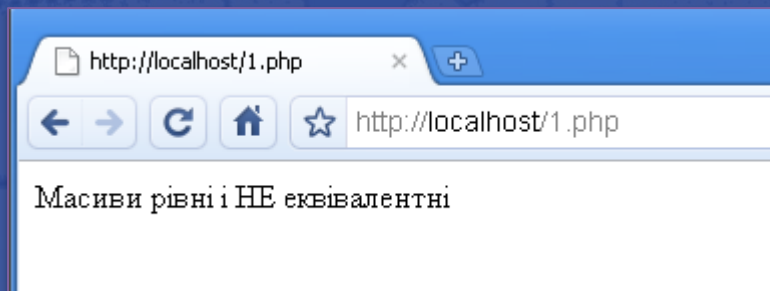
Еквівалентність – рівність значень і ключів та порядок слідування.

```
<?
```

```
$a = array("i"=>"Інформатика",    "м"=>"Математика");  
$b = array("м"=>"Математика",    "i"=>"Інформатика");
```

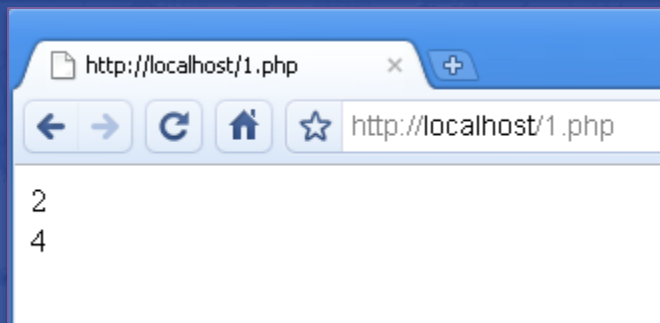
```
if ($a == $b) echo "Масиви рівні і";  
    else echo "Масиви НЕ рівні і ";  
if ($a === $b) echo " еквівалентні";  
    else echo " НЕ еквівалентні";
```

```
?>
```



Функція count

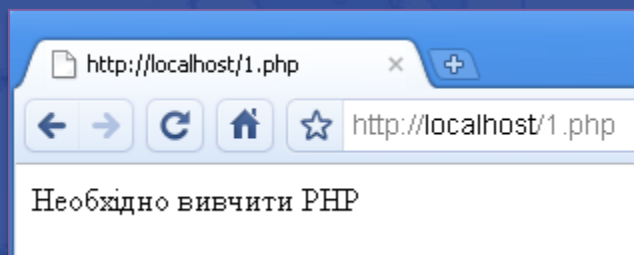
```
<?
$del_items = array(
    "langs" => array("10"=>"Python", "12"=>"Lisp"),
    "other"=>"Інформатика");
echo count($del_items) . "<br>";
echo count($del_items,COUNT_RECURSIVE);
?>
```



Функція in_array

`in_array(" значення", "масив",
["обмеження на тип"]);`

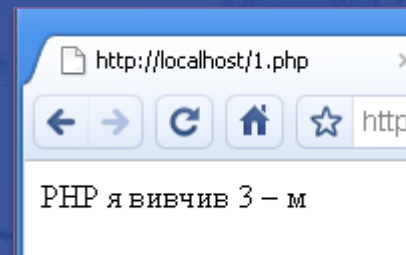
```
<?php  
$langs = array("Lisp","Python","Java", "PHP","Perl");  
  
if (in_array("PHP",$langs))  
    echo "Необхідно вивчити PHP<br>";  
if (in_array("php",$langs))  
    echo "Необхідно вивчити php<br>";  
?>
```



Функция array_search

array_search(" значення", "масив",
["обмеження на тип"]);

```
<?php
$langs = array("JavaScript","Python","Java", "PHP","Perl");
if (!array_search("PHP",$langs))
    echo "Необхідно вивчити PHP<br>";
else {
    $k = array_search("PHP",$langs);
    echo "PHP я вивчив $k - м";
}
?>
```



Функція array_keys

array_keys ("масив" [, "значення для пошуку"])

```
<?php
$langs = array("Lisp", "Python", "Java", "PHP", "Perl", "Lisp");

$lisp_keys = array_keys($langs, "Lisp");

echo "Lisp входить у масив ". count($lisp_keys) ." рази:<br>";
foreach ($lisp_keys as $val) {
    echo "під номером $val <br>";
}
?>
```

Lisp входить у масив 2 рази:
під номером 0
під номером 5

Функція array_values()

```
<?php
$langs =array("Lisp","Python","Java","PHP", "Perl","Lisp");
$lisp_keys = array_values($langs);
echo "Елементів масиву: ".
    count($lisp_keys) . "<br>";
foreach ($lisp_keys as $val) {
    echo " $val <br>";
}
?>
```

```
Елементів масиву: 6
Lisp
Pithon
Java
PHP
Perl
Lisp
```


Сортування масивів

Функція sort

<?

```
$items = array(10 => "хліб", 20 => "молоко", 30 => "бутерброд");  
sort($items);
```

```
print_r($items);
```

```
$rev_items = array("хліб" =>10, "бутерброд"=>30, "молоко" =>20);  
sort($rev_items);
```

```
print_r($rev_items);
```

?>

```
Array ( [0] => бутерброд [1] => молоко [2] => хліб ) Array ( [0] => 10 [1] => 20 [2] => 30 )
```

Функції `asort`, `rsort`, `arsort`

Функція `sort()` сортує список (по значенням) в порядку зростання, а `rsort()` — в порядку спадання

```
<?php
$books = array("Пушкин"=>"Руслан и Людмила",
    "Толстой"=>"Война и мир",
    "Лермонтов"=>"Герой нашего времени");
asort($books);
    // сортуємо масив,
    // зберігаючи значення ключів
print_r($books);
echo "<br>";
rsort($books);
    // сортуємо масив у зворотному порядку,
    // ключі будуть замінені
print_r($books);
?>
```

```
Array ( [Толстой] => Война и мир [Лермонтов] => Герой нашего времени [Пушкин] => Руслан и Людмила )
Array ( [0] => Руслан и Людмила [1] => Герой нашего времени [2] => Война и мир )
```

Сортування масива по ключам

```
<?php
$books = array("Пушкин"=>"Руслан и Людмила",
               "Толстой"=>"Война и мир",
               "Лермонтов"=>"Герой нашего времени");
ksort($books);
// сортування масиву
// із збереженням значення ключів
print_r($books);
?>
```

```
Array ( [Лермонтов] => Герой нашего времени
        [Пушкин] => Руслан и Людмила
        [Толстой] => Война и мир )
```



Виділення підмасива

Функція `array_slice`

`array_slice` (масив, номер_елемента [, довжина])

```
<?php
$arr = array(1,2,3,4,5);
$sub_arr = array_slice($arr, 2, 2);
    // масив з елементів 3, 4
$sub = array_slice($arr,-3, 2);
    // масив з елементів 3, 4
$sub1 = array_slice($arr,0, -1);
    // масив з елементів 1, 2, 3, 4
$sub2 = array_slice($arr,-4, -2);
    // масив з елементів 2, 3
?>
```


Функція `array_chunk`- розбиває масив на декілька.

`array_chunk` (масив, розмір [, зберігати_ключі])

Обернення масивів

`array_reverse` (array array [, bool reserve_keys])



Дякую за увагу!

Подивитись
