

Web-dasturlash kirish fani bo'yicha elektron resurs uchun taqdimot materiallari

15-Ma'ruza. . PHP funksiyalari va
obyektlari.

Reja:

1. PHP funksiyalari va obyektlari;
2. PHP dasturlash tilida massivlar;
3. PHPda satrlar.



Funksiyalar

Funksiyalar nima uchun kerak? Bu savolga javob berish uchun, funksiya o'zi nima ekanligini tushunib olish lozim bo'ladi. Dasturlashda, xuddi matematikadagi kabi, unga bog'liq ko'pgina argumentlarning uning ko'pgina ma'nolarida aks etishidir. Demak, funksiya argumentning har bir ma'nolari jamlanmasi uchun uning bajargan ishi natijasi sifatida qandaydir ma'no qaytaradi. Funksiyalar nima uchun kerak, buni misollar bilan oydinlashtirishga qarakat qilamiz. Dasturlashdagi funksiya klassik misol – bu sonning faktorial ahamiyatini hisoblab beruvchi funksiya. Demak, biz unga son beramiz, u esa bizga uning faktorialini qaytaradi. Bunda biz faktorialini olishni xohlagan har bir son uchun aynan bir xil kodni qaytaravermaymiz – bu songa teng bo'lgan argumentli funksiyaning chaqirishning o'zi kifoya qiladi.

Natural son faktorialini hisoblash funksiyasi

Misol:

```
<?php
function fact($n){
    if ($n++0) return 1;
    else return $fact = $n * fact($n-1);
}
echo fact(3);
//echo (3*2) deb yozish mumkin edi; lekin son katta
bo'lsa,
echo fact(50);
//Funksiyadan foydalanish qulayroq, echo
(50*49*48*...*3*2) deb yozgandan;
?>
```

PHP-da foydalanuvchi tomonidan aniqlangan funksiya

Foydalanuvchi tomonidan belgilangan funksiya **function** so'zi bilan boshlanadi:

Sintaksis

```
function functionName() {  
    code to be executed;  
}
```

```
<?php
function writeMsg() {
    echo "Hello world!";
}
```

```
writeMsg(); // call the function
?>
```

```
<?php
function addNumbers(int $a, int $b) {
    return $a + $b;
}
echo addNumbers(5, "5 days");
```

Funksiyalarning argumentlari

Har bir funksiyada, avval aytganimizday, argumentlar ro'yxati bo'lishi mumkin. Bu argumentlar yordamida funksiyaga har xil ma'lumotlar beriladi (masalan, faktoriali hisoblanishi kerak bo'lganson ma'nosi). Har bir argument o'zgaruvchi va konstantaga ega bo'ladi.

Argumentlar yordamida ma'lumotlar funksiyaga uch xil turli usullar bilan o'tkazilishi mumkin. Bu argumentlarni ma'nosiga ko'ra (o'zgarmas holatda foydalaniladi), ilovalarga ko'ra va o'zgarmas holatda argumentlarga ma'no berishga ko'ra o'tkazish. Bu usullarni atroflicha ko'rib chiqamiz.

Argument funksiyaga ma'nosiga ko'ra o'tkazilsa, funksiya ichidagi argument ma'nosining o'zgarishi uning funksiya tashqarisidagi ma'nosiga ta'sir qilmaydi. Funksiyaga uning argumentlarini o'zgartirishga yo'l qo'yish uchun ularni havolalarga ko'ra o'tkazish kerak. Buning uchun argument nomi oldidan funksiyaning aniqlashda ampersand "&" belgisini yozish kerak.

Misol. Argumentlarni havolasiga ko'ra o'tkazish

```
<?php
//qo'shimcha qilishi mumkin bo'lgan funktsiyani yozamiz
checked so'zi qatoriga
function add_label(&$data_str){
$data_str .= "checked";
}
$str = "<input type=radio name=article ";
//bunday qator mavjud bo'lsin
echo $str."><br>";
//forma elementini keltiradi – belgilanmagan radio knopkasini
add_label($str);
//Funktsiyani chaqiramiz
echo $str."><br>";
//bu endi belgilangan radio knopkani keltiradi
?>
```


O'zgaruvchan uzunlik argumentlari ro'yxatlari

PHP4 da argumentlarning o'zgaruvchan soni bilan funksiya tuzish mumkin. Ya'ni biz uni necha argumentlar bilan chaqirilishini bilmasdan, funksiya tuzamiz. Bu kabi funksiya yozish uchun hech qanday maxsus sintaksis kerak bo'lmaydi. Hammasi uning ichiga o'rnatilgan funksiyalar `func_num_args()`, `func_get_arg()`, `func_get_args()` yordami bilan qilinadi.

Havolani qaytarish

Funksiya o'z ishi natijasida shuningdek havolani biror-bir o'zgaruvchiga qaytarishi mumkin. Bu funktsiyani qanday o'zgaruvchi havolaga o'zlashtirish kerakligini aniqlash uchun foydalaniladi. Funktsiyadan havola olish uchun, e'lon oldidan ampersand (&) belgisini yozish kerak bo'ladi va har safar funksiya chaqiruvi paytida uning nomi oldidan ham ampersand (&) yozish kerak bo'ladi. Ko'pincha funksiya havolani biror-bir global o'zgaruvchiga (yoki uning qismini – havolani global massiv elementiga), havolani statistik o'zgaruvchiga (yoki uning qismini) yoki havolani argumentlardan biriga qaytaradi, agar u havola bo'yicha berilgan bo'lsa.

Misol. Havolani qaytarish

```
<?
$a = 3; $b = 2;
function & ref($par){
    global $a, $b;
    if ($par % 2 == 0) return $b;
    else return $a;
}
$var =& ref(4);
echo $var, " i ", $b, "<br>"; //2 va 2 keltiriladi
$b = 10;
echo $var, " i ", $b, "<br>"; //10 va 10 keltiriladi
?>
```

PHP dasturlash tilida Array (massiv) tipi.

PHP dasturlash tilida massiv tipi tartiblangan kartalarga o'xshaydi va qiymatini kalitga o'zlashtiradigan tipdir. Bu tip bir necha yo'nalishlarda optimallashtiriladi, shuning uchun siz uni xususiy massiv, ro'yxat (vektor), xesh-jadvali (kartani amalga oshirish uchun ishlatiladi), stek, navbat va boshhalar sifatida foydalanishingiz mumkin. Modomiki, PHP dasturlash tilida bir massivni qiymatini boshqasiga o'zlashtirish uchun daraxtlardan foydalanasiz.

Massivlarni `array()` konstruktsiyasi yordamida aniqlanadi yoki elementlariga qiymat berish bilan aniqlanadi.

`array()` konstruktsiyasi yordamida aniqlash.
`array ([key] => value,
[key1] => value1, ...)`

Misol. PHP dasturlash tilida massivlar.

```
<?php
    $books = array ("php" =>
        "PHP users guide",
        12 => true);
    echo $books["php"];
        //ekranga chiqaradi: "PHP users guide"
    echo $books[12];    //ekranga chiqaradi: 1
?>
```

PHP-da uch xil massiv mavjud:

- Indekslangan **massivlar** - sonli indeksli **massivlar**
- **Assotsiativ massivlar** - nomlangan kalitlarga ega **massivlar**
- **Ko'p o'lchovli massivlar** - Bir yoki bir nechta massivlarni o'z ichiga olgan massivlar

PHP indekslangan massivlar

Indekslangan massivlarni yaratishning ikkita usuli mavjud:

Indeks avtomatik ravishda tayinlanishi mumkin (indeks har doim 0 dan boshlanadi), shunga o'xshash:

- `$cars = array("Volvo", "BMW", "Toyota");`

yoki indeksni qo'lda tayinlash mumkin:

- `$cars[0] = "Volvo";`
`$cars[1] = "BMW";`
`$cars[2] = "Toyota";`

PHP assotsiativ massivlari

Assotsiativ massivlar - bu siz ularga tayinlangan nomlangan kalitlardan foydalanadigan massivlar. Assotsiativ massivni yaratishning ikki yo'li mavjud:

```
$age = array("Peter"=>"35", "Ben"=>"37",  
"Joe"=>"43");
```

yoki:

```
$age['Peter'] = "35";  
$age['Ben'] = "37";  
$age['Joe'] = "43";
```


Misol. Massivni qayta indekslaymiz.

```
<?php
$arr = array ("a","b","c");
/* "a", "b" va "c" qiymatli massivni yaratamiz. Bu erda kalit ko'rsatilmagan
   Biroq mos ravishda ular 0,1,2 bo'ladi. */
print_r($arr); //massivni ekranga chiqaramiz (kaliti va qiymatini)
unset($arr[0]);
unset($arr[1]);
unset($arr[2]);
//massivdan hamma elementini o'chiramiz
print_r($arr); //massivni ekranga chiqaramiz (kaliti va qiymatini)
$arr[] = "aa"; //massivga yangi element qo'shamiz. Uni indeksi(kaliti) 3
               bo'ladi, 0 emas.
print_r($arr);
$arr =
array_values($arr); //massivni qayta indekslaymiz.
$arr[] = "bb"; //bu elementni kaliti 1 bo'ladi.
print_r($arr);
?>
```

Bu skriptning natijasi quyidagicha bo'ladi:

- `Array ([0] => a [1] => b [2] => c)`
- `Array ()`
- `Array ([3] => aa)`
- `Array ([0] => aa [1] => bb)`

Massivlar initsializatsiyasi

PHP da massivlarni initsializatsiya qilishning 2 usuli mavjud. Birinchisi massiv elementlariga qiymat berishdan iborat:

```
<?
$car[] = "passenger car";
$car[] = "land-rover";
echo($car[1]); // chiqaradi "land-rover"
?>
```

Massiv indeksini ochiq ko'rsatish mumkin:

```
<?
$car[0] = "passenger car";
$car[1] = "land-rover";
echo($car[1]); // chiqaradi "land-rover"
?>
```

Yangi element yaratsak, uning indeksi avtomatik ravishda 31 bo'ladi:

```
<?
$car[10] = "passenger car";
$car[20] = "land-rover";
$car[30] = "station-wagon";
$car[] = "victoria";
echo($car[31]);
?>
```

Alternativ usul **array()** konstruktsiyasidan foydalanishdan iborat:

```
<?
$car = array("passenger car","land-rover");
echo($car[1]); // chiqaradi "land-rover"
?>
```

Massivlarni ko'rib chiqish uchun foreach sikli

PHP4 da massiv elementlarini ko'rib chiqish uchun **foreach** operatoridan foydalanish mumkin. Bu operator sintaksisi:

```
foreach (array as [$key =>] $value)  
{  
    statements;  
}
```

Busikl ma'nosi sodda: har bir element ko'rilganda uning indeksi **\$key** o'zgaruvchiga, qiymati bo'lsa **\$value** o'zgaruvchiga joylashtiriladi. Bu ikki o'zgaruvchilarning nomlari ixtiyoriydir.

Misol:

```
<?
$car = array("passenger car", "land-rover",
    "station-wagon","victoria");
foreach($car as $index => $val)
{
    echo("$index -> $val <br>");
}
?>
```

6. Ko'p o'lchovli massivlar

Ko'p o'lchovli massivlarni ko'rib chiqish uchun ichki joylashgan **array()** konstruktsiyasidan foydalaniladi. Ko'p o'lchovli massivlarni o'qib chiqish joylangan sikllar yordamida amalgam oshiriladi. Quyidagi scriptda ko'p o'lchovli massiv yaratish va ko'rib chiqish ko'rsatilgan.

Misol:

```
<?    $ship = array(
    "Passenger ship" => array("Yacht","Liner","Ferry"),
    "War ship" => array("Battle-wagon","Submarine","Cruiser"),
    "Freight ship" => array("Tank vessel","Dry-cargo ship","Container
cargo ship")    );    foreach($ship as $key => $type)
{    echo(
    "<h2>$key</h2>\n"."<ul>\n");
    foreach($type as $ship)
    {        echo("\t<li>$ship</li>\n");    }    }    echo("</ul>\n");    ?>
```

Massivlarni tartiblash funksiyalari

Quyidagi tartiblash funksiyalari mavjud:

- **sort()**

Massivni o'sish bo'yicha tartiblash funksiyasi.

Sintaksis:

```
void sort(array array [, int sort_flags])
```

- **asort()**

Assotsiativ massivni o'sish bo'yicha tartiblash.

Sintaksis:

```
void asort(array arr [, int sort_flags])
```

- **arsort()**

Assotsiativ massivlarni kamayish bo'yicha tartiblash.

Sintaksis:

```
void arsort(array arr [, int sort_flags])
```


- **ksort()**

Massivlarni kalit o'sishi bo'yicha tartiblash.

Sintaksis:

`int ksort(array arr [, int sort_flags])`

- **krsort()**

Indekslar kamayishi bo'yicha massivlarni tartiblash.

Sintaksis:

`int krsort(array arr [, int sort_flags])`

- **array_reverse()**

Massiv elementlarini teskari joylashtirish.

Sintaksis:

`array array_reverse(array arr [, bool preserve_keys])`

- **shuffle()**

Massiv elementlarini tasodifiy joylashtirish.

Sintaksis:

`void shuffle(array arr)`

- **natsort()**

Tabiiy tartiblashni bajaradi.

Sintaksis:

`void natsort(array arr)`

Misol:

`<?`

```
$array1 = $array2 = array("pict10.gif", "pict2.gif", "pict20.gif", "pict1.gif");
```

```
echo ("oddiy tartiblash:"); echo ("<br>");
```

```
sort($array1);
```

```
print_r($array1);
```

```
echo ("<br>"); echo ("tabiiy tartiblash:"); echo ("<br>");
```

```
natsort($array2);
```

```
print_r($array2);
```

`?>`

Natija:

oddiy tartiblash:

Array ([0] => pict1.gif [1] => pict10.gif [2] => pict2.gif [3] => pict20.gif)

estestvennaya sortirovka:

Array ([3] => pict1.gif [1] => pict2.gif [0] => pict10.gif [2] => pict20.gif)

PHP dasturlash tilida String (satr) tipi

Satr – bu belgilar to'plamidir. PHP dasturlash tilida belgi bu bir bayt va 256 ta turli belgilar mavjud. PHP dasturlash tili Unicode tipidagi belgilarni qabul qilmaydi. PHP dasturlash tilida amalda satrlarga chegirma mavjud emas, shuning uchun satrlarni ishlatganda uning aniq uzunligi haqida o'ylash shart emas.

PHP dasturlash tilida satrlar uchta turli xil usullarda aniqlanadi:

- bittalik qo'shtirnoqlar yordamida ("");
- qo'shtirnoqlar yordamida ("");
- heredoc-sintaksisi yordamida.

Bittali tirnoqlar

Misol:

```
<?php
echo 'Satrlar majmui';
//Ekranga chiqaradi: ' belgini chiqarish uchun undan oldin \ belgi qo'yiladi.
echo ' Belgini \' chiqarish uchun undan oldin'
    ' //belgini qo'yish kerak';
//Ekranga chiqaradi: Siz shuni o'chirmoqchimisiz C:\*.*?
echo ' Siz shuni o'chirmoqchimisiz C:\\*.*?';
//Ekranga chiqaradi: Buni qo'ymang: \n
//yangi qatorga
echo ' Buni qo'ymang: \n yangi qatorga ';
//Ekranga chiqaradi: o'zgaruvchi $expand ham
//$eithar qo'yilmaydi
echo 'o'zgaruvchi $expand ham $eithar' .
    'qo'yilmaydi';
?>
```

Satrlar bilan ishlaydigan funksiyalar

- `strlen ()` - Satr uzunligini qaytaradi
- `str_word_count ()` - Satrdagi so'zlarni hisoblash
- `strrev ()` - Satrni teskari yo'naltirish
- `strpos ()` - satr ichidagi matnni qidirish
- `str_replace ()` - Matnni satr ichida almashtirish
- PHP `trim ()` funktsiyasi bush joylarni olib tashlaydi
- `Strtok ()` funktsiyasi satrni kichikroq satrlarga (jetonlarga) ajratadi.

Satrlar bilan ishlaydigan funksiyalar

- `strlen ()` - Satr uzunligini qaytaradi
- `str_word_count ()` - Satrdagi so'zlarni hisoblash
- `strrev ()` - Satrni teskari yo'naltirish
- `strpos ()` - satr ichidagi matnni qidirish
- `str_replace ()` - Matnni satr ichida almashtirish
- PHP `trim ()` funktsiyasi bush joylarni olib tashlaydi
- `Strtok ()` funktsiyasi satrni kichikroq satrlarga (jetonlarga) ajratadi.

- ```
<?php
echo strlen("Hello world!"); // outputs 12
?>
```
- ```
<?php  
echo str_word_count("Hello world!"); //  
outputs 2  
?>
```
- ```
<?php
echo strrev("Hello world!"); // outputs !dlrow
olleH
?>
```