

Backend 12

Tömbök

Rostagni Csaba

2024. október 1.

Tartalom

- 1 Operátorok
- 2 Változók és típusok
- 3 Vezérlési szerkezetek
- 4 Tömbök
- 5 Konzolos PHP

Tartalom

- 1 Operátorok
 - Érték növelése, csökkentése

Érték növelő/csökkentő operátorok

	pre	post
increment	<code>++\$i</code>	<code>\$i++</code>
decrement	<code>--\$i</code>	<code>\$i--</code>

Utólag növelés (post-increment) működés közben

PHP

```
$i = 1;  
$x = $i++;
```

PHP

```
$i = 1;  
$x = $i;  
$i = $i + 1;
```

Eredmény

```
i : 2  
x : 1
```

Előre növelés (pre-increment) működés közben

PHP

```
$i = 1;  
$x = ++$i;
```

PHP

```
$i = 1;  
$i = $i + 1;  
$x = $i;
```

Eredmény

```
i : 2  
x : 2
```

Tartalom

- 1 Operátorok
- 2 Változók és típusok
- 3 Vezérlési szerkezetek
- 4 Tömbök
- 5 Konzolos PHP

Tartalom

- 2 Változók és típusok
 - Változók

Változók változó nevekben

Mivel minden változó **\$** jellel kezdődik, így akár a változó nevét is megadhatjuk egy változóval.

PHP

```
$x = "nev";  
$nev = "Sámuel";  
  
echo "$x";           // nev  
echo "$nev";         // Sámuel  
echo "$$x";          // $nev  
echo "${$x}";        // Sámuel  
echo $$x;            // Sámuel
```

$$\begin{array}{c} \$ \underbrace{\$x}_{\text{nev}} == \text{"Sámuel"} \\ \underbrace{\hspace{1.5cm}}_{\$nev} \end{array}$$

Tartalom

- 1 Operátorok
- 2 Változók és típusok
- 3 Vezérlési szerkezetek
- 4 Tömbök
- 5 Konzolos PHP

Tartalom

- 3 Vezérlési szerkezetek
 - Ciklusok

Elöl tesztelős ciklus: while

PHP

```
<?php
$i = 0;
while ( $i < 11 )
{
    echo "$i ";
    $i++;
}
```

Elöl tesztelős ciklus: while alternatív szintaxis

PHP

```
<?php
$i = 0;
while ( $i < 11 ) :
    echo "$i ";
    $i++;
endwhile;
```

Számlálós ciklus: for

For ciklus:

```
<?php
for ( $i = 0 ; $i < 11 ; $i++ )
{
    echo "$i ";
}
```

PHP

Alternatív szintaxis:

```
<?php
for ( $i = 0 ; $i < 11 ; $i++ ) :
    echo "$i ";
endfor;
```

PHP

Léptető ciklus: foreach

A foreach ciklus:

```
<?php
foreach ( $tomb as $kulcs => $ertek )
{
    echo "$kulcs = $ertek, ";
}
```

PHP

Alternatív szintaxis:

```
<?php
foreach ( $tomb as $kulcs => $ertek ) :
    echo "$kulcs = $ertek, ";
endforeach;
```

PHP

Tartalom

- 1 Operátorok
- 2 Változók és típusok
- 3 Vezérlési szerkezetek
- 4 Tömbök
- 5 Konzolos PHP

Tartalom

- 4 Tömbök
 - Tömb alapismeretek
 - Tömb elemeinek megjelenítése
 - Elem hozzáadása tömbhöz

Tömbök bevezető

- A többi nyelvtől eltérően különféle típusú elemeket tárolhatunk benne.
- Valójában a PHP asszociatív tömböket (map/dictionary) használ.
- Használható klasszikus tömbként.

Részletek: <http://php.net/manual/en/language.types.array.php>

Tömb létrehozása (régí forma)

- Az 5.4-es verzió előtt az alábbi módon lehetett tömböket létrehozni.

```
$tomb = array(3, 5, 8);
```

Régi

```
$tomb = array("alma", "Banán");
```

Régi

```
$tomb = array(1, "Alma", false);
```

Régi

Tömb létrehozása

- Az 5.4-es verziótól kezdve létezik a rövid (JSON) formátum.

```
$tomb = [3, 5, 8];
```

PHP

```
$tomb = ["alma", "Banán"];
```

PHP

```
$tomb = [1, "Alma", false];
```

PHP

Tömb létrehozása

- A tömb megadásakor az elemek elrendezése tetszés szerint tördelhető.
- Az újabb PHP verziókban az utolsó elem után is kirakhatjuk a vesszőt. (korábban végzetes hibának számított)

```
$nevsor = [  
    "Anna",      // $nevsor[0]  
    "Peti",      // $nevsor[1]  
    "Mónika",    // $nevsor[2]  
    "Zoltán",    // $nevsor[3]  
];
```

PHP

Tömb létrehozása

- A tömbök alapértelmezetten 0-tól indexelődnek.
- Mi is megadhatjuk az elemek indexét.

```
$napok = [  
    1 => "Hétfő",    // $napok[1]  
    2 => "Kedd",     // $napok[2]  
    3 => "Szerda",   // $napok[3]  
];
```

PHP

- Amint megadtuk az első elemet onnantól magától folytatja.

```
$napok = [  
    4 => "Csütörtök", // $napok[4]  
    "Péntek",         // $napok[5]  
    "Szombat"];       // $napok[6]
```

PHP

A tömb mérete

- A tömb méretét a `count()` függvény adja meg.

PHP

```
$tomb = [3, 5, 8];  
$hossz = count($tomb);  
echo $hossz;
```

Eredmény

3

PHP

```
$nevsor = [  
    "Anna",  
    "Peti",  
    "Mónika",  
    "Zoltán",  
];  
echo count($nevsor);
```

Eredmény

4

Tartalom

- 4 Tömbök
 - Tömb alapismeretek
 - Tömb elemeinek megjelenítése
 - Elem hozzáadása tömbhöz

Tömb kiírása

- Előfordulhat, hogy szeretnénk látni egy tömb tartalmát és a felépítését.
- Tesztelési célzattal felfedhetjük a tömbünk adatait.

```
$t = [3, 5, 8];  
echo $t;
```

Rossz példa!

Array

Eredmény

Figyelem!

A tömb egy összetett adatszerkezet, így egy egyszerű `echo` paranccsal nem tudjuk kiírni a tartalmát.

Tömb kiírása `print_r()` segítségével

- A `print_r()` függvény rekurzívan írja ki a tömb tartalmát.
- Megjeleníti a tömb elemeihez tartozó indexeket is.

```
$t = [3, 5, 8];  
print_r($t);
```

PHP

Eredmény

```
Array  
(  
    [0] => 3  
    [1] => 5  
    [2] => 8  
)
```

Figyelem!

Csak és kizárólag tesztelésre alkalmazható!

Tömb kiírása `var_dump()` segítségével

- A `var_dump()` függvény rekurzívan írja ki a tömb tartalmát.
- Megjeleníti a tömb elemeihez tartozó indexeket és a típusokat is.

```
$t = [3, 5, 8];  
var_dump($t);
```

PHP

Eredmény

```
array(3) {  
    [0]=>  
        int(3)  
    [1]=>  
        int(5)  
    [2]=>  
        int(8)  
}
```

Figyelem!

Csak és kizárólag tesztelésre alkalmazható!

Tömb elemeinek kiírása

PHP

```
$t = [3, 5, 8];  
echo $t[0] . "\n";  
echo $t[1] . "\n";  
echo $t[2] . "\n";  
echo $t[count($t)-1]
```

Eredmény

3
5
8
8

Figyelem!

A szöveggel ellentétben az index nem lehet negatív!

Tömb kiírása for ciklussal

PHP

```
$tomb = [3, 5, 8];  
  
for( $i = 0 ; $i < count( $t ) ; ++$i )  
{  
    echo "{$t[$i]}, ";  
}
```

Eredmény

3, 5, 8,

Tömb kiírása for ciklussal (átirat)

PHP

```
$nevek = ["András", "Emese", "Peti"];

for( $i = 0 ; $i < count( $t ) ; ++$i )
{
    $nev = $t[$i];
    echo "$nev\n";
}
```

Eredmény

András
Emese
Peti

Tömb kiírása foreach ciklussal

PHP

```
$nevek = ["András", "Emese", "Peti"];

foreach($nevek as $nev)
{
    echo "$nev\n";
}
```

Eredmény

```
András
Emese
Peti
```

Tömb kiírása foreach ciklussal (indexelve)

PHP

```
$nevek = ["András", "Emese", "Peti"];

foreach($nevek as $index => $nev)
{
    echo "$index: $nev\n";
}
```

Eredmény

```
0: András
1: Emese
2: Peti
```

Tartalom

- 4 Tömbök
 - Tömb alapismeretek
 - Tömb elemeinek megjelenítése
 - Elem hozzáadása tömbhöz

Elem hozzáadása tömbhöz

- A tömb elemeire szögletes zárójelekkel hivatkozhatunk, így felülírhatunk egy régi értéket.
- Amennyiben "túlindexelnénk", akkor új elemként jön létre.
- Jóval nagyobb index esetén, sem fog létrehozni üres elemeket.
- Az index elhagyható, ilyenkor a soron következő indexet veszi.

```
$t = [3, 5, 8];
```

```
$t[0] = 0;
```

```
$t[3] = 9;
```

```
$t[10] = 10
```

```
$t[] = 11;
```

PHP

0	5	8	9	10	11
0	1	2	3	10	11

Tartalom

- 1 Operátorok
- 2 Változók és típusok
- 3 Vezérlési szerkezetek
- 4 Tömbök
- 5 Konzolos PHP

Tartalom

- 5 Konzolos PHP
 - Konzolos szkript használata
 - A szkript argumentumai

PHP konzolos verzió: php-cli

- PHP-ban lehetséges konzolos alkalmazások készítése is. Ezeket a **php-cli**, azaz a **PHP Command Line Interface** segítségével futtathatjuk.

Teszteljük a konzolos PHP verzióját.

```
php -v
```

CMD

Súgó megjelenítése

```
php -h
```

CMD

PHP konzolos verzió: php-cli (windows)

A konzolos php futtatásának lehetőségei:

- **cmd.exe**: Megadhatjuk a **php.exe** teljes útvonalát!

CMD

```
C:\xampp\php\php.exe elso.php
```

- **cmd.exe**: Beállíthatjuk a %PATH% környezeti változóban a **php.exe** elérési útvonalát!

CMD

```
SET PATH=C:\xampp\php\;%PATH%  
php elso.php
```

- **xampp**: A XAMPP Control Panelen kattintsunk a **Shell** gombra és ebben dolgozzunk!

CMD

```
php elso.php
```

Tartalom

- 5 Konzolos PHP
 - Konzolos szkript használata
 - A szkript argumentumai

Konzolos szkript paraméterezése

- A legtöbb program indításakor paramétereket adhatunk meg, hogy speciálisan indíthassuk el, például a böngészőt rögtön privát módban.
- PHP-ban, ha nincs letiltva erre jó az `$argv` és az `$argc`
- Az `$argv` (argumentum values) a paraméterül kapott értékek (azaz az argumentumok) tömbjét jelenti
- Az `$argc` (argument count) egy olyan szám, ami megmondja, hogy hány argumentumot kapott.
- A tömb minden eleme string típusú lesz, még a számok is!

Konzolos szkript paraméterezése példa

Futtassuk a hello.php-t a megadott módon.

```
php hello.php a 10 "x y"
```

CMD

```
<?php  
var_dump($argv);  
var_dump($argc);
```

hello.php

```
array(4) {  
    [0]=>    string(8) "hello.php"  
    [1]=>    string(1) "a"  
    [2]=>    string(2) "10"        // szöveggént tárolt szám!  
    [3]=>    string(3) "x y"  
}  
int(4)
```

Eredmény