# Backend 12 Tömbök

Rostagni Csaba

2024. október 1.

- Operátorok
- Változók és típusok
- Vezérlési szerkezetek
- Tömbök
- 6 Konzolos PHP

### <u>Tartalom</u>

- Operátorok
  - Érték növelése, csökkentése

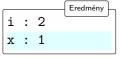
# Érték növelő/csökkentő operátorok

	pre	post
increment	++\$i	\$i++
decrement	\$i	\$i

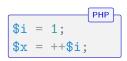
Utólag növelés (post-increment) működés közben

```
x = i++:
```

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



Előre növelés (pre-increment) működés közben





- Operátorok
- Változók és típusok
- Vezérlési szerkezetek
- Tömbök
- 6 Konzolos PHP

- Változók és típusok
  - Változók

#### Változók változó nevekben

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

```
$ $x =="Sámuel"
*nev **snev**
```

Rostagni Csaba Backend 12 2024. október 1. 7/39

- Operátorok
- Változók és típusok
- Vezérlési szerkezetek
- Tömbök
- 6 Konzolos PHP

- Vezérlési szerkezetek
  - Ciklusok

#### Elöl tesztelős ciklus: while

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

Rostagni Csaba Backend 12 2024. október 1. 10/39

### Elöl tesztelős ciklus: while alternatív szintaxis

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

2024. október 1. Rostagni Csaba Backend 12 11/39

### Számlálós ciklus: for

#### For ciklus:

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

#### Alternatív szintaxis:

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

Rostagni Csaba Backend 12 2024. október 1. 12 / 39

## Léptetős ciklus: foreach

#### A foreach ciklus:

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

#### Alternatív szintaxis:

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

Rostagni Csaba Backend 12 2024. október 1. 13 / 39

- Operátorok
- Változók és típusok
- Vezérlési szerkezetek
- Tömbök
- 6 Konzolos PHP

- 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

Rostagni Csaba Backend 12 2024. október 1. 16/39

# Tömb létrehozása (régi forma)

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

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

$tomb = array("alma", "Banán");

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

Rostagni Csaba Backend 12 2024. október 1. 17/39

#### Tömb létrehozása

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

```
PHP
tomb = [3, 5, 8];
                                                           PHP
$tomb = ["alma", "Banán"];
                                                           PHP
$tomb = [1, "Alma", false];
```

Rostagni Csaba Backend 12 2024. október 1. 18 / 39

#### 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)

Rostagni Csaba Backend 12 2024. október 1. 19/39

#### 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]
];
```

Amint megadtuk az első elemet onnantól magától folytatja.

Rostagni Csaba Backend 12 2024. október 1. 20/39

#### A tömb mérete

• A tömb méretét a count() függvény adja meg.

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

```
Bredmény _____
```

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

```
Eredmény 4
```

21/39

Rostagni Csaba Backend 12 2024. október 1.

- 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.

```
Rossz példa!
t = [3, 5, 8]:
echo $t;
                                                                      Eredmény
Array
```

#### Figyelem!

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

Rostagni Csaba Backend 12 2024. október 1. 23 / 39

# 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);
```

```
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);
```

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

25/39

#### Figyelem!

Csak és kizárólag tesztelésre alkalmazható!

Rostagni Csaba Backend 12 <u>2024. október 1.</u>

#### Tömb elemeinek kiírása

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

```
3
5
8
8
```

#### Figyelem!

A szöveggel ellentétben az index nem lehet negatív!

#### Tömb kiírása for ciklussal

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

for( $i = 0 ; $i < count( $t ) ; ++$i )
{
    echo "{$t[$i]}, ";
}</pre>
```

```
3, 5, 8, Eredmény
```

Rostagni Csaba Backend 12 2024. október 1. 27 / 39

Eredmény

28 / 39

# Tömb kiírása for ciklussal (átirat)

```
PHP
$nevek = ["András", "Emese", "Peti"];
for( $i = 0 ; $i < count( $t ) ; ++$i )</pre>
    $nev = $t[$i];
    echo "$nev\n";
```

András

Emese

Peti

Rostagni Csaba Backend 12 2024. október 1.

Eredmény

#### Tömb kiírása foreach ciklussal

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

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

András Emese

Peti

Rostagni Csaba Backend 12 2024. október 1. 29/39

# 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
```

- 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.

```
PHP
t = [3, 5, 8]:
t[0] = 0;
t[3] = 9;
t[10] = 10
$t[] = 11;
```

5 8 10 11 0 2 3 10 11

Rostagni Csaba Backend 12 2024. október 1. 32 / 39

- Operátorok
- Változók és típusok
- Vezérlési szerkezetek
- Tömbök
- 6 Konzolos PHP

- 6 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.

Súgó megjelenítése

```
php -h
```

Rostagni Csaba Backend 12 2024. október 1. 35/39

# PHP konzolos verzió: php-cli (windows)

A konzolos php futtatásának lehetőségei:

• cmd.exe: Megadhatjuk a php.exe teljes útvonalát!

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

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

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

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

```
php elso.php
```

Rostagni Csaba Backend 12 2024. október 1. 36/39

- Konzolos PHP
  - Konzolos szkript használata
  - A szkript argumentumai

## Konzolos szkript paraméterezése

- A legtöbb program indításakor paramétereket adhatunk meg, hogy speciálisan indíthassuk el, példáult 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 hágy argumentumot kapott.
- A tömb minden eleme string típusú lesz, még a számok is!

Rostagni Csaba Backend 12 2024. október 1. 38 / 39

## Konzolos szkript paraméterezése példa

Futtassuk a hello.php-t a megadott módon.

```
CMD
php hello.php a 10 "x y"
                                                          hello.php
<?php
var dump($argv);
var dump($argc);
                                                         Eredmény
array(4) {
    <= [0]
             string(8) "hello.php"
    [1]=>
             string(1) "a"
    [2]=>
             string(2) "10"
                                   // szövegként tárolt szám!
    [3]=>
             string(3) "x y"
int(4)
```

Rostagni Csaba Backend 12 2024. október 1. 39/39