

# Backend 12

Asszociatív tömbök, tömbfüggények és include

Rostagni Csaba

2024. október 9.

# Tartalom

- 1 Asszociatív tömbök
- 2 Több dimenziós tömbök
- 3 Include

# Asszociatív tömb

- Asszociatív tömb esetén nem számmal, hanem valamilyen kulcs segítségével indexeljük az elemeket.
- A kulcs lehet szöveg is.
- A **kulcs** és az **érték** között "nyíl" ( $\Rightarrow$ ) található

Asszociatív tömb felépítése:

```
$asszociativTomb = [ kulcs => érték ];
```

Doksi

Például:

```
$hetnapjai = [  
    "hétfő" => 1,  
    "kedd"  => 2,  
    "szerda" => 3,  
];
```

PHP

# Asszociatív tömb példa

PHP

```
$nagyAlex = [  
    "nev" => "Nagy Alex",  
    "kor" => 23,  
    "atlag" => 4.27,  
];
```

- A **\$nagyAlex** egy asszociatív tömb, amiben három elem van.
- A **nev** szöveges, a **kor** egész, míg az **atlag** eleme valós szám.
- Kulcsok:
  - nev
  - kor
  - atlag
- Értékek:
  - Nagy Alex
  - 23
  - 4.27

# Tartalom

- 1 Asszociatív tömbök
  - Asszociatív tömb elemeinek megjelenítése
  - Elem hozzáadása asszociatív tömbhöz
  - Kulcs ellenőrzése

# Asszociatív tömb kiírása

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

```
$at = [  
    "nev" => "Nagy Alex", "kor" => 23, "atlag" => 4.27,  
];  
echo $at;
```

Rossz példa!

Array

Eredmény

## Figyelem!

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

# Asszociatív tömb kiírása `print_r()` segítségével

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

PHP

```
$at = [  
    "nev" => "Nagy Alex",  
    "kor" => 23,  
    "atlag" => 4.27,  
];  
print_r($at);
```

Eredmény

```
Array  
(  
    [nev] => Nagy Alex  
    [kor] => 23  
    [atlag] => 4.27  
)
```

## Figyelem!

Csak és kizárólag tesztelésre alkalmazható!

# Asszociatív tömb kiírása `var_dump()` segítségével

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

PHP

```
$at = [  
    "nev" => "Nagy Alex",  
    "kor" => 23,  
    "atlag" => 4.27,  
];  
var_dump($at);
```

Eredmény

```
array(3) {  
    ["nev"]=>  
        string(9) "Nagy Alex"  
    ["kor"]=>  
        int(23)  
    ["atlag"]=>  
        float(4.27)  
}
```

**Figyelem!**

Csak és kizárólag tesztelésre alkalmazható!



# Asszociatív tömb elemeinek kiírása

PHP

```
$at = [  
    "nev" => "Nagy Alex",  
    "kor" => 23,  
    "atlag" => 4.27,  
];
```

```
echo "$at[nev]\n";  
echo "{$at["kor"]}\n";  
echo $at["atlag"];
```

Eredmény

```
Nagy Alex  
23  
4.27
```

- Idézőjelek között kiírva a kulcs idézőjeleit el kell hagyni! (nev)
- Idézőjelen belül lévő kapcsos zárójelben viszont ki kell tenni! (kor)
- Ha nem szövegbe írjuk be, hanem magában, akkor ki kell tenni a kulcs köré az idézőjeleket. (atlag)

# Tömb kiírása foreach ciklussal

"Nagy Alex"	23	4.27
"nev"	"kor"	"atlag"

PHP

```
$at = ["nev" => "Nagy Alex", "kor" => 23, "atlag" => 4.27,];  
  
foreach($at as $adat)  
{  
    echo "$adat\n";  
}
```

Eredmény

Nagy Alex  
23  
4.27

# Tömb kiírása foreach ciklussal (+kulcs)

"Nagy Alex"	23	4.27
"nev"	"kor"	"atlag"

PHP

```
$at = ["nev" => "Nagy Alex", "kor" => 23, "atlag" => 4.27,];  
  
foreach($at as $kulcs => $ertek)  
{  
    echo "$kulcs: $ertek\n";  
}
```

Eredmény

```
nev: Nagy Alex  
kor: 23  
atlag: 4.27
```

# Tartalom

## 1 Asszociatív tömbök

- Asszociatív tömb elemeinek megjelenítése
- Elem hozzáadása asszociatív tömbhöz
- Kulcs ellenőrzése

# Elem hozzáadása asszociatív tömbhöz

- A tömb elemeire szögletes zárójelekkel hivatkozhatunk.
- Előfordulhat, hogy felülírunk egy régi értéket.
- Ha kulcsot elhagyjuk sorszámozni kezdi a tömböt.

PHP

```
$t = [  
    "a" => "alma",  
    "b" => "banán",  
];  
  
$t["b"] = "barack";  
$t[] = "dinnye";
```

alma	barack	dinnye
"a"	"b"	0

# Tartalom

## 1 Asszociatív tömbök

- Asszociatív tömb elemeinek megjelenítése
- Elem hozzáadása asszociatív tömbhöz
- Kulcs ellenőrzése

# array\_key\_exists()

```
array_key_exists(string|int $key, array $array): bool
```

- Az első paraméter a keresett kulcs
- A második paraméter a tömb, amiben keresendő
- A **visszatérési értéke** (`bool`) meghatározza, az adott kulcs megtalálható-e a tömbben

```
$t = ["a", "b", "c"];  
var_dump(array_key_exists(1, $t));
```

PHP

Eredmény

`bool(true)`

Linkek:

- `array_key_exists()` - [php.net](https://www.php.net/manual/en/function-array-key-exists.php)

# Tartalom

- 1 Asszociatív tömbök
- 2 Több dimenziós tömbök
- 3 Include



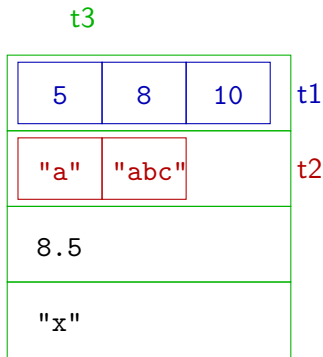
# Több dimenziós tömbök

Elkészíthetjük a tömböket külön-külön, majd egymásba pakolhatjuk...

PHP

```
$t1 = [5, 8, 10];
$t2 = ["a", "abc"];

$t3 = [
    $t1,
    $t2,
    8.5,
    "x"
];
```



**Figyelem!**

Tömbben csak akkor használunk \$ jelet, ha változót szeretnénk beleírni.

# Több dimenziós tömbök

... vagy közvetlen létrehozhatjuk a tömbben a tömböt.

```
$t3 = [  
    [5, 8, 10],  
    ["a", "abc"],  
    8.5,  
    "x"  
];
```

PHP

t3

5	8	10
"a"	"abc"	
8.5		
"x"		

# Több dimenziós tömbök

PHP

```
$t3 = [
    [5, 8, 10],
    ["a", "abc"],
    8.5,
    "x"
];
```

PHP

```
print_r($t3[0]);
```

0

5

8

10

1

"a"

"abc"

2

8.5

3

"x"

Eredmény

Array

(

[0] =&gt; 5

[1] =&gt; 8

[2] =&gt; 10

)

# Több dimenziós tömbök

PHP

```
$t3 = [
    [5, 8, 10],
    ["a", "abc"],
    8.5,
    "x"
];
```

PHP

```
var_dump($t3[0][1]);
```

0

5

8

10

1

"a"

"abc"

2

8.5

3

"x"

Eredmény

int(8)

# Több dimenziós tömbök

PHP

```
$t3 = [
    [5, 8, 10],
    ["a", "abc"],
    8.5,
    "x"
];
```

PHP

```
print_r($t3[1]);
```

0

5

8

10

1

"a"

"abc"

2

8.5

3

"x"

Eredmény

Array

(

[0] =&gt; a

[1] =&gt; abc

)

# Több dimenziós tömbök

PHP

```
$t3 = [  
    [5, 8, 10],  
    ["a", "abc"],  
    8.5,  
    "x"  
];
```

PHP

```
var_dump( $t3[1][1]);
```

0

5

8

10

1

"a"

"abc"

2

8.5

3

"x"

Eredmény

```
string(3) "abc"
```

# Több dimenziós tömbök

PHP

```
$t3 = [
    [5, 8, 10],
    ["a", "abc"],
    8.5,
    "x"
];
```

PHP

```
var_dump( $t3[1][1][2]);
```

0

5

8

10

1

"a"

"abc"

2

8.5

3

"x"

Eredmény

string(1) "c"

# Több dimenziós tömbök

PHP

```
$t3 = [
    [5, 8, 10],
    ["a", "abc"],
    8.5,
    "x"
];
```

PHP

```
var_dump( $t3[2]);
```

0

5

8

10

1

"a"

"abc"

2

8.5

3

"x"

Eredmény

float(8.5)



# Több dimenziós tömbök

PHP

```
$t3 = [  
    [5, 8, 10],  
    ["a", "abc"],  
    8.5,  
    "x"  
];
```

PHP

```
var_dump( $t3[3]);
```

0

5

8

10

1

"a"

"abc"

2

8.5

3

"x"

Eredmény

```
string(1) "x"
```

# Több dimenziós tömbök (tanulók példa)

PHP

```
$petra = [  
    "nev" => "Juhász Petra",  
    "kor" => 19,  
    "targyak" => [  
        "angol" => [5,2,5,4],  
        "php" => [4,4,3,4],  
    ],  
];  
$laszlo = [  
    "nev" => "Budai László",  
    "kor" => 18,  
    "targyak" => [  
        "angol" => [3,3,5,4],  
        "php" => [5,2,1,5],  
    ],  
];  
$tanulok = [$petra, $laszlo];
```

Összetettebb adatszerkezet esetén egyszerűbb lehet egy-egy blokkot külön változóként felvenni, majd összeépíteni egy nagy tömbbé.

# Több dimenziós tömbök (tanulók példa)

PHP

```
$t = [  
    "petra" => [  
        "nev" => "Juhász Petra",  
        "kor" => 19,  
        "targyak" => [  
            "angol" => [5,2,5,4],  
            "php" => [4,4,3,4],  
        ],  
    ],  
    "laszlo" => [  
        "nev" => "Budai László",  
        "kor" => 18,  
        "targyak" => [  
            "angol" => [3,3,5,4],  
            "php" => [5,2,1,5],  
        ],  
    ],  
];
```

Ha átlátjuk a tömböket, akkor  
elsőre megírhatjuk egy nagy  
tömbként az összes tanuló  
adatait.

## Figyelem!

Oda kell figyelni, hogy ne írjunk  
fölösleges \$ jeleket!

# Több dimenziós tömbök (tanulók példa)

PHP

```
$t = [  
    "petra" => [  
        "nev" => "Juhász Petra",  
        "kor" => 19,  
        "targyak" => [  
            "angol" => [5,2,5,4],  
            "php" => [4,4,3,4],  
        ],  
    ],  
    "laszlo" => [  
        "nev" => "Budai László",  
        "kor" => 18,  
        "targyak" => [  
            "angol" => [3,3,5,4],  
            "php" => [5,2,1,5],  
        ],  
    ],  
];
```

## Hány éves Petra?

PHP

```
echo $t["petra"]["kor"];
```

Eredmény

19

# Több dimenziós tömbök (tanulók példa)

PHP

```
$t = [  
    "petra" => [  
        "nev" => "Juhász Petra",  
        "kor" => 19,  
        "targyak" => [  
            "angol" => [5,2,5,4],  
            "php" => [4,4,3,4],  
        ],  
    ],  
    "laszlo" => [  
        "nev" => "Budai László",  
        "kor" => 18,  
        "targyak" => [  
            "angol" => [3,3,5,4],  
            "php" => [5,2,1,5],  
        ],  
    ],  
];
```

Hány tantárgya van Petrának?

PHP

```
echo count($t["petra"]["targyak"]);
```

Eredmény

2

# Több dimenziós tömbök (tanulók példa)

## Mennyi László átlaga angolból?

PHP

```
$osszeg=0;
$db=count($t["laszlo"]["targyak"]["angol"]);
for($i = 0; $i < $db; $i++ )
{
    $osszeg += $t["laszlo"]["targyak"]["angol"][$i];
}
echo $osszeg/$db;
```

Eredmény

3.75

# Több dimenziós tömbök (tanulók példa)

Jelenítse meg a tanulók nevét és tárgyaként a jegyeiket.

PHP

```
foreach($t as $tanulo){
    echo $tanulo['nev'] . "\n";

    foreach($tanulo['targyak'] as $targy => $jegyek ) {
        echo "\t{$targy}:";

        foreach($jegyek as $jegy)
        {
            echo "$jegy,";
        }
        echo "\n";
    }
}
```

# Több dimenziós tömbök (tanulók példa)

Eredmény

Juhász Petra

angol:5,2,5,4,

php:4,4,3,4,

Budai László

angol:3,3,5,4,

php:5,2,1,5,



# Több dimenziós tömbök (tanulók példa)

Egy kis trükkel megoldható, hogy ne legyen a sorok végén fölösleges vessző.

PHP

```
foreach($t as $tanulo){
    echo $tanulo['nev'] . "\n";

    foreach($tanulo['targyak'] as $targy => $jegyek ) {
        echo "\t{$targy}:";

        for($i = 0; $i < count($jegyek)-1; ++$i)
        {
            echo "$jegyek[$i],";
        }
        echo $jegyek[count($jegyek)-1] . "\n";
    }
}
```

# Több dimenziós tömbök (tanulók példa)

Eredmény

Juhász Petra

angol:5,2,5,4

php:4,4,3,4

Budai László

angol:3,3,5,4

php:5,2,1,5

# Tartalom

- 1 Asszociatív tömbök
- 2 Több dimenziós tömbök
- 3 Include

# Az include() függvény

Az include függvény a hívás helyére "másolja" a megadott fájlban lévő kódot és értelmezi is.

- Elsőként a megadott útvonalon keresi a fájlt, amennyiben létezik.
- Utána a konfigurban beállított **include\_path** mappában keresi.
- Végezetül az aktuális mappában keresi a fájlt.

A második lépést kihagyja, ha gyökérkönyvtárral (c:\, vagy /home), egy vagy két ponttal kezdjük a hivatkozást.

```
<?php include('menu.php') ?>
```

PHP

# Az include() és a hatókör (scope)

- A külső fájlban található **változók** hatóköre ugyanaz lesz, mintha az adott sorba írtuk volna őket
- A függvények hatóköre globális lesz.

Linkek:

- PHP dokumentáció: `include()`
- PHP dokumentáció: `include_path`

# include() vs. require()

PHP

```
<?php include("test.php"); ?>  
<?php require("test.php"); ?>
```

- Míg az `include()` hibás fájl esetén figyelmeztetést ad, addig a `require()` esetén megáll a szkript futása végzetes hibával.
- Az oldal működéséhez feltétlen szükséges részeket érdemes a `require()` segítségével hozzá adni a kódunkhoz.

# include() vs. include\_once()

PHP

```
<?php include("reklam.php"); ?>  
<?php include_once("fuggvenyek.php"); ?>
```

- Míg az `include()` annyiszor "bemásolja" és végrehajtja a fájlt ahányszor szerepel a kódban, addig az `include_once()` csak az első előfordulásakor.
- Ahol gondot okoz az, hogy már szerepelt korábban (például függvényeknél, amiket nem lehet újra definiálni) ott az `include_once`-t célszerű alkalmazni.

# include, require, once összefoglaló

**Közös:** A függvény hívás helyére bemásolja a fájl tartalmát, értelmezi a benne található kódot.

- `include()`
  - Többszöri híváskor újra végrehajtódik.
  - Hibás fájl esetén figyelmeztet.
- `include_once()`
  - Többszöri híváskor NEM hajtódik újra végre.
  - Hibás fájl esetén figyelmeztet.
- `require()`
  - Többszöri híváskor újra végrehajtódik.
  - Hibás fájl esetén elszáll hibával.
- `require_once()`
  - Többszöri híváskor NEM hajtódik újra végre.
  - Hibás fájl esetén elszáll hibával.