

Dr. Holló Csaba, SZTE TTIK, Szoftverfejlesztés Tanszék

Programozási alapismeretek kurzus

Olvasási idő:
10 perc



Alapvető programozásnyelvi elemek C-ben és PHP-ban

V. Függvények és hatókörök

A lecke célja A változók hatóköreinek jobb megértése.

Tudás Ismeri a változók hatóköreit és a paraméterátadás lehetőségeit.

Képesség Hatékony információátadást valósít meg a függvények, illetve a globális környezet között.

Függvények definiálása

C-ben a függvényeket

- egymás után kell definiálni,

PHP-ban

- egymásba is lehetnek ágyazva
- a definíció feltételben is lehet.

Mindkét nyelvben

- a függvényhívás előtt a fordítónak / értelmezőnek találkoznia kell a függvény definíciójával (vagy legalább annak fejlécével).

❖ További leírások függvényekről C-ben és PHP-ban

Változók hatóköre

Lokális változók függvényekben

- a függvényben deklaráljuk és csak a függvényben elérhetők
- a függvény meghívásakor jönnek létre
- a függvény végén megsemmisülnek

Statikus változók függvényekben

- a függvényben deklaráljuk és csak a függvényben elérhetők
- a függvény végén **nem** semmisülnek meg

Mire jó ez?

- újabb függvényhíváskor visszkapjuk az otthagytott régi értéket; ennek érdekében nem fut le újra a deklaráció és ahhoz tartozó esetleges kezdeti értékadás sem.

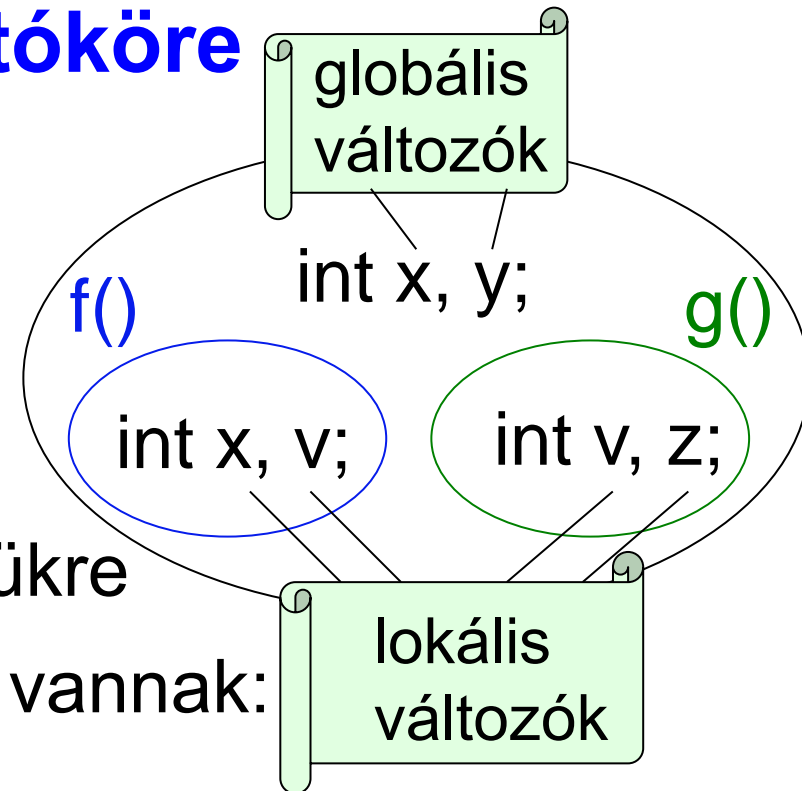
Változók hatóköre

Globális változók

- legfelsőbb szinten, minden függvényen kívül definiáljuk

Eredeti nevükkel történő elérésükre különféle alapértelmezések vannak:

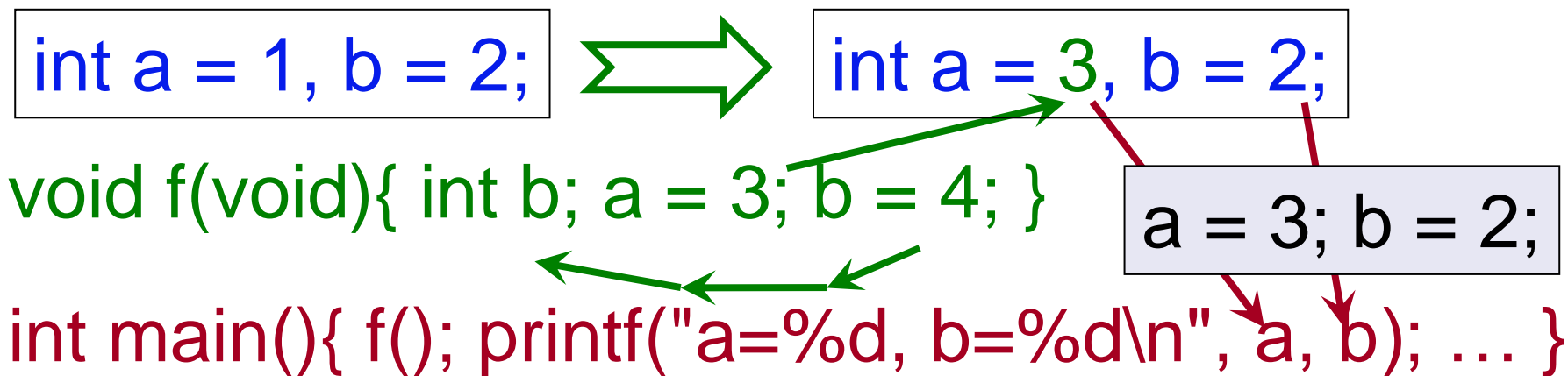
- hivatkozhatók, csak bizonyos körülmények azt akadályozzák (pl. C-ben újradeklarálás)
- alpból nem hivatkozhatók, de elérhetővé tehetők (PHP).



Globális változók C függvényekben

Mit hivatkozunk a változó nevével?

- ha nincs deklarálnva ugyanolyan nevű változó, akkor a globális változót
- ha van deklarálnva, akkor azt; ilyenkor a globális változót csak mutatóval tudjuk elérni

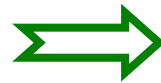


Globális változók PHP függvényekben

Mit hivatkozunk a változó nevével?

- alapesetben az első hivatkozásnál létrejön egy ugyanolyan nevű lokális változó és azt hivatkozunk;
- ha újradeklaráltuk globálisként, akkor viszont a globális változót.

```
$a = 1; $b = 2;
```



```
$a = 1; $b = 4;
```

```
function f(){ $a = 3; global $b; $b = 4; }
```

Mit teszünk, ha azt szeretnénk, hogy legyen ugyanolyan nevű globális és lokális változó is?

- nem deklaráljuk újra globálisként
- a globálisat a GLOBALS tömb elemeként tudjuk hivatkozni.

```
$a = 1; $b = 2;
```



```
$a = 5; $b = 4;
```

```
function f(){ $GLOBALS["a"] = 5; $a = 3; global $b; $b = 4; }
```

Paraméterátadás

Érték szerinti: mindkét nyelvben van

- az aktuális paraméterről egy ***másolat*** keletkezik a formális paraméterben, melynek
- módosítása ***nem módosítja*** az aktuális paraméter értékét

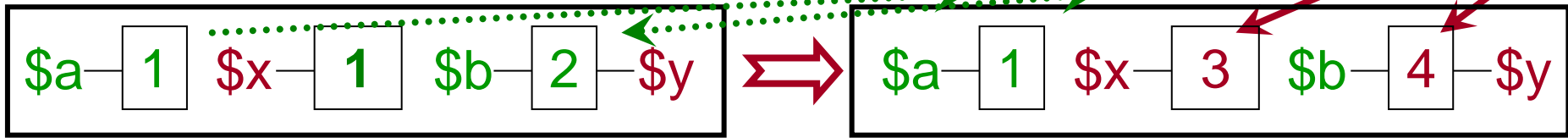
Cím szerinti: C-ben nincs, PHP-ban van

- a formális paraméter odakötődik az aktuális paraméter memóriacíméhez, így a
- ***formális paraméter*** módosítása az ***aktuális paramétert*** is ***módosítja***.

Paraméterátadás

```
$a = 1; $b = 2; f($a, $b);
```

```
function f($x, &$y){ $x = 3; $y = 4; }
```

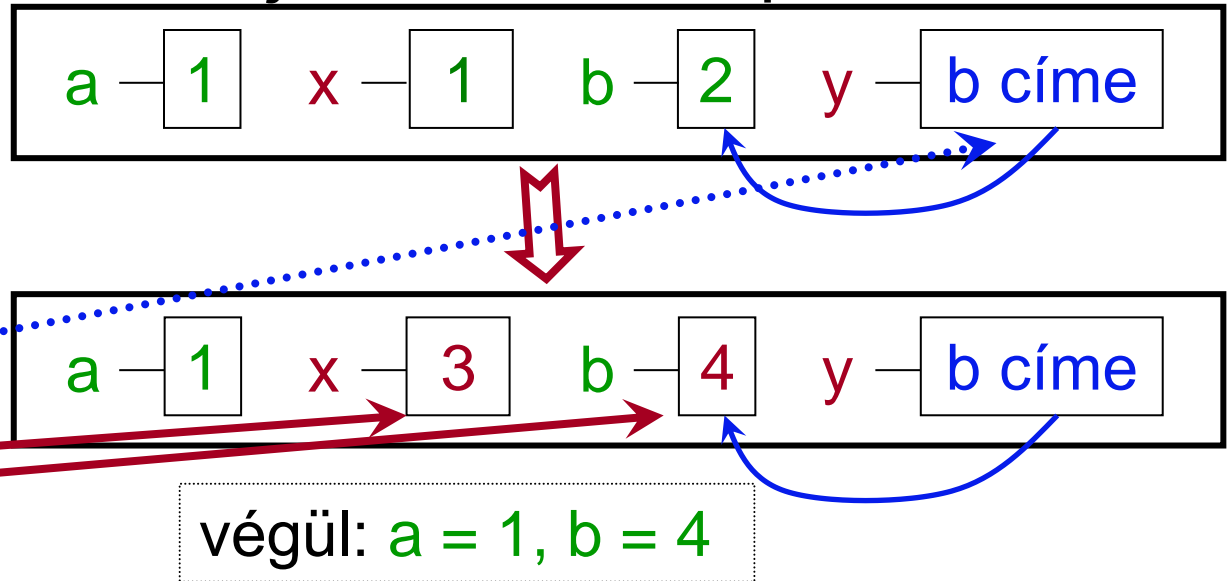


C-ben alternatív megoldásként:

végül: $a = 1, b = 4$

- értéként átadhatjuk a változó címét egy mutatóba
- mellyel módosítani tudjuk az aktuális paramétert.

```
int main(){
    int a=1, b=2;
    f(a, &b); ...
}
void f(int x, int *y){
    x = 3; *y = 4; }
```



Kérdések, feladatok

1. Írjuk meg az előbbi példákat szemléltető programokat C-ben és PHP-ban.
2. Hogyha függvényben szeretnénk kiírni egy paraméterként kapott tömb tartalmát, milyen paraméterátadási mód lesz hatékonyabb?

Megoldás. C-ben a tömb címét adjuk át (értékként), ez az alapértelmezett megoldás is. PHP-ban cím szerint adjuk át, bár az újabb verziókba be van építve a „copy on write technika”, melyről olvashatunk az [ELTE-s PHP oldalon](#).

3. Milyen esetben érdemes paraméterátadás helyett globális változót használni és miért?

Megoldás. Amikor az átadandó változót a program nagy részében használjuk és átláthatóan követni tudjuk annak módosítását. Ugyanis, a paraméterátadás több adminisztrációs tevékenységgel jár, ezért lassítja a program futását, viszont a globális változók módosítása nehezen követhető lehet, ezért rontja a program átláthatóságát.

EFOP-3.4.3-16-2016-00014

AP1 HALLGATÓI DIPLOMA-SZERZÉST SEGÍTŐ SZOLGÁLTATÁSOK

Jelen tananyag a
Szegedi Tudományegyetemen
készült az
Európai Unió támogatásával.
Projekt azonosító:
EFOP-3.4.3-16-2016-00014



SZÉCHENYI 2020

Európai Unió
Európai Szociális
Alap



BEFEKTETÉS A JÖVŐBE