

WEBFEJLESZTÉS 2. – ADATTÁROLÁS, FÁJLOK

Horváth Győző

Egyetemi adjunktus

1117 Budapest,

Pázmány Péter sétány 1/C, 2.420

Tel: (1) 372-2500/1816

2

Ismétlés

Ismétlés

3

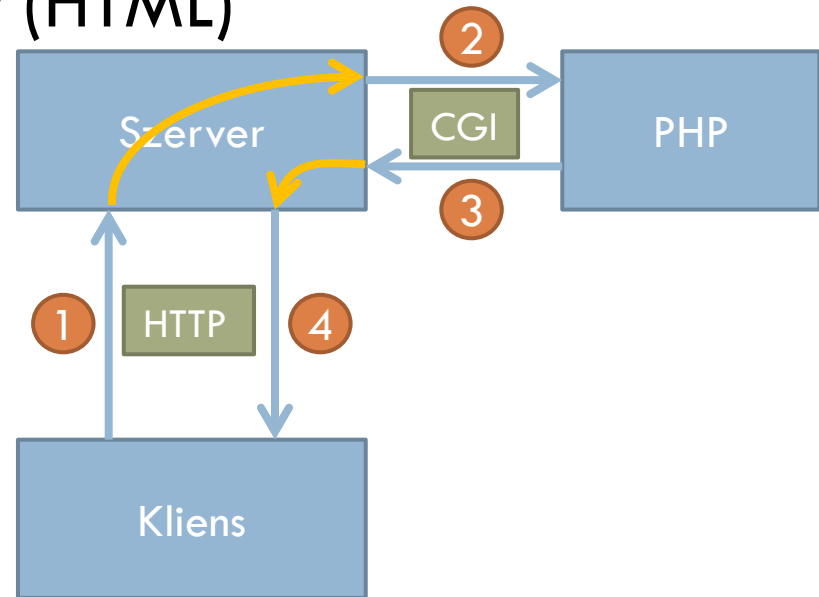
- Dinamikus szerveroldali webprogramozás: program (PHP) állítja elő a tartalmat (HTML)

- Input

- ▣ kliens: link, űrlap
- ▣ PHP: `$_GET`, `$_POST`

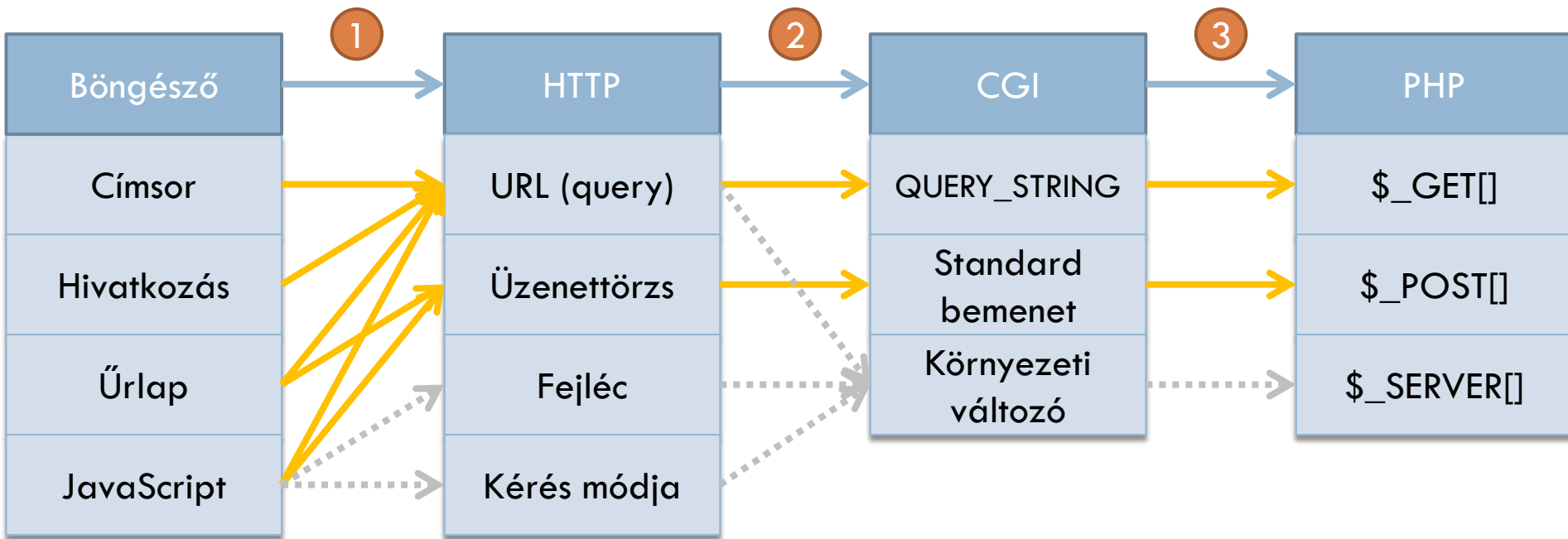
- Űrlapfeldolgozás

- ▣ Adatok (nincsen: `$_POST`, `$_GET`, HTML)
- ▣ Beolvasás, ellenőrzés (`$_POST`, `$_GET`)
- ▣ Kimenet (HTML)



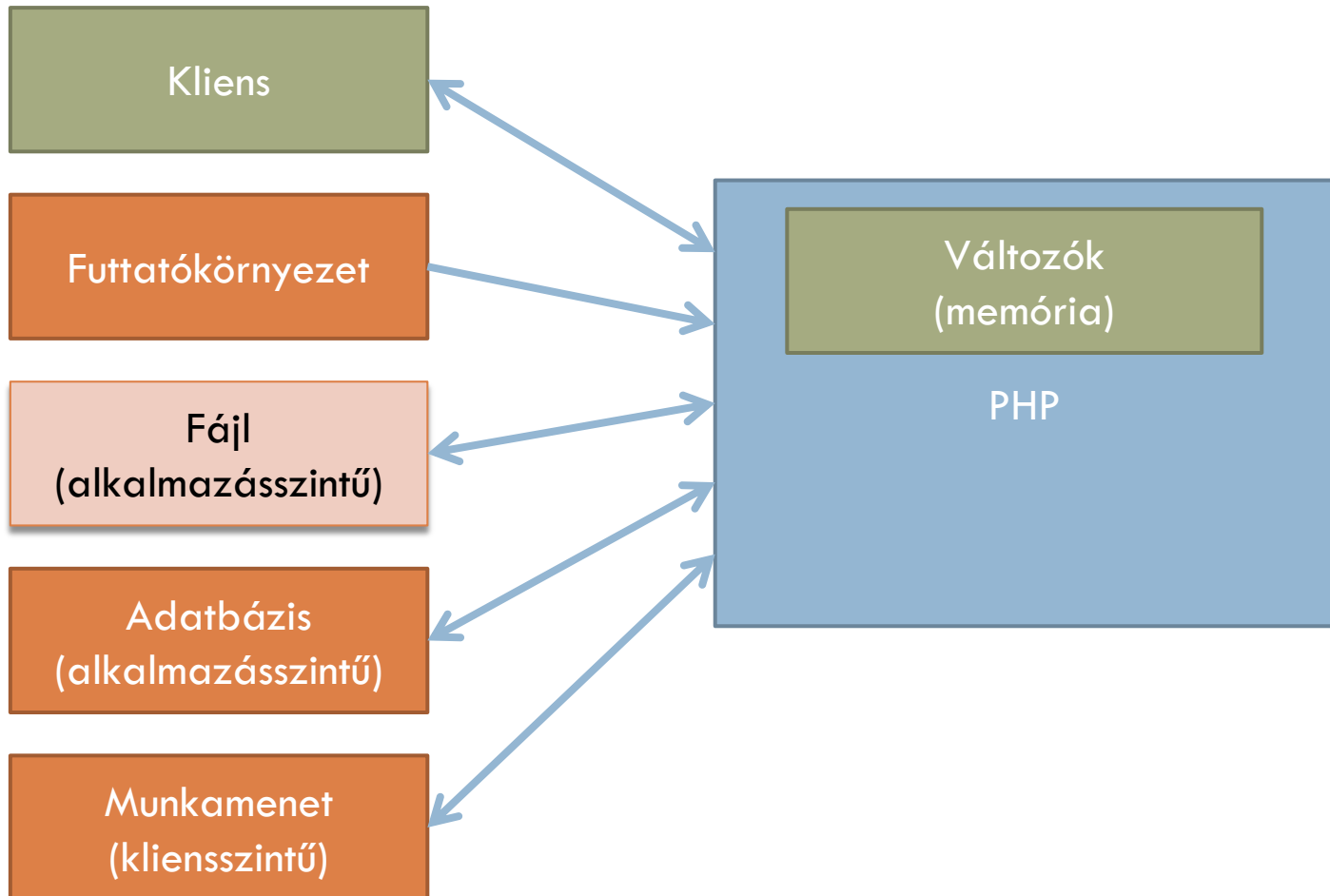
Adatok a kienstől PHP-ig

4



PHP programok adatforrása

6

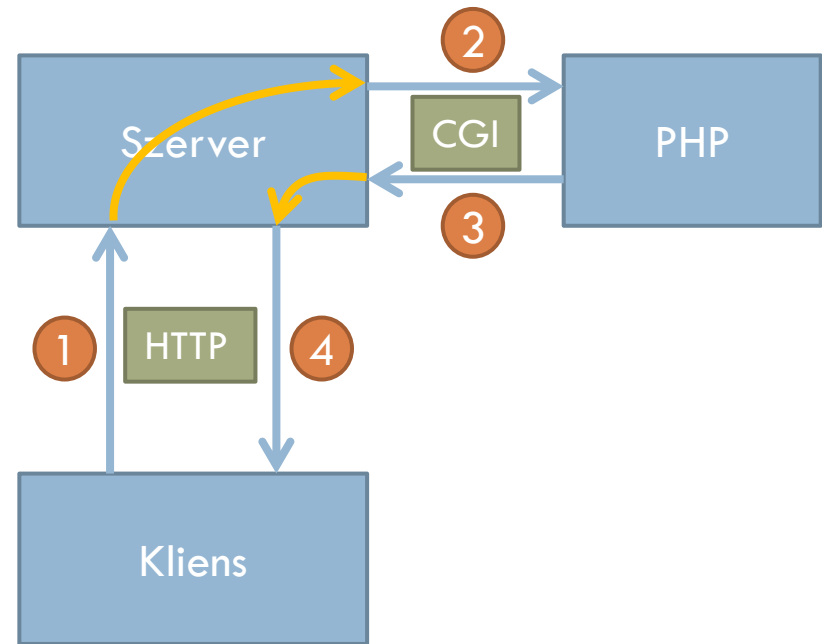


Szkriptbeli változók

7

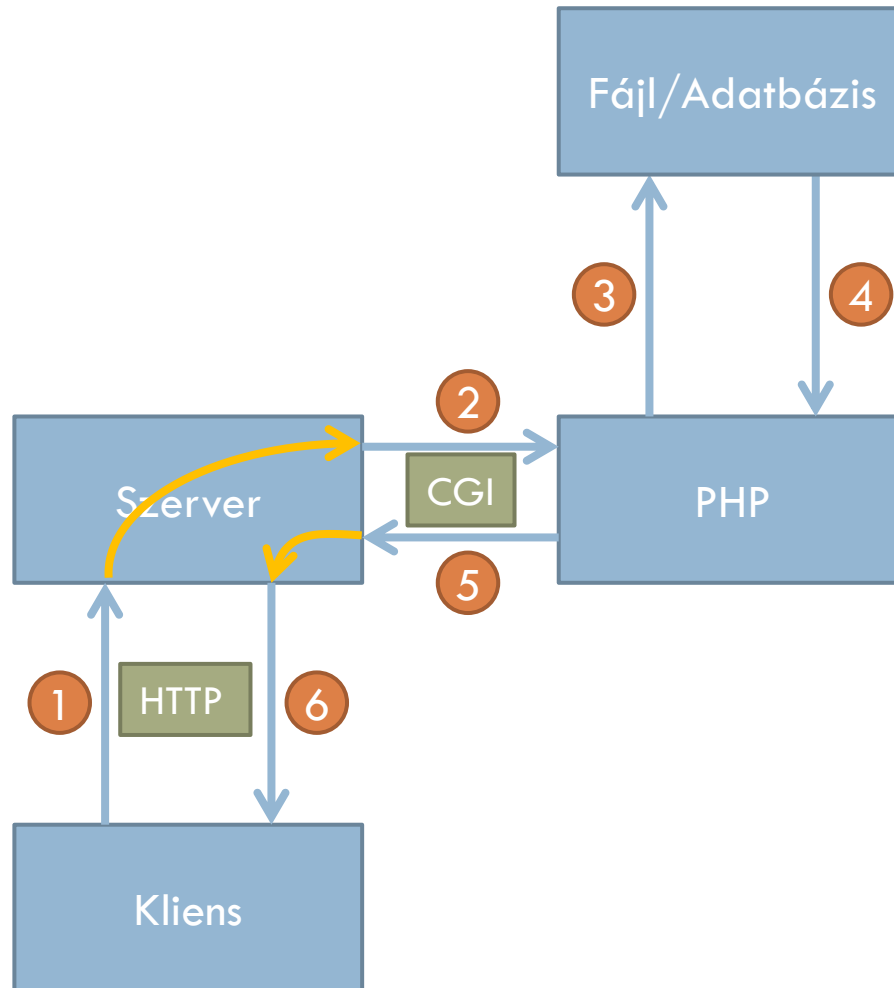
- Élettartamuk a szkript futása
- Utána megszűnnek

```
if (isset($szamlalo)) {  
    $szamlalo += 1;  
} else {  
    $szamlalo = 0;  
}  
var_dump($szamlalo);
```



Külső adattárolás

8



9

Fájlkezelés PHP-ban

Fájlműveletek PHP-ban

10

- PHP-ban van lehetőség fájlok használatára (írás, olvasás, stb)
- Mint a legtöbb programozási nyelvben
- Csak a szerver helyi fájljai érhetőek el
- Sokféle nyelvi lehetőség
- Tipikus folyamat
 - ▣ megnyitás
 - ▣ feldolgozás
 - ▣ bezárás

Alacsony szintű általános fájlműveletek

11

- `$f = fopen($fájlnev, $mód)`
 - ▣ fájl vagy URL megnyitása
 - ▣ `$f`: logikai fájlkezelő
 - ▣ `$mód`: megnyitás módját
- `feof($f)`
 - ▣ fájl vége?
- `fclose($f)`
 - ▣ fájl bezárása

\$mód

- `r` (csak olvasás)
- `r+` (olvasás, írás)
- `w` (csak írás, üres fájl)
- `w+` (olvasás, írás, üres fájl)
- `a` (hozzáfűzés)
- `a+` (hozzáfűzés, olvasás)
- `x` (csak írás, üres fájl, ha létezik a fájl, akkor hiba)
- `x+` (olvasás, írás, üres fájl, ha létezik, akkor hiba)
- `c` (csak írás, ha létezik, akkor elejére áll)
- `c+` (írás, olvasás, ha létezik, elejére áll)

Alacsony szintű beolvasás/kiírás

12

□ Beolvasás

- fread(\$f, \$hossz)
 - \$hossz byte beolvasása
- fscanf(\$f, \$formátum)
 - formátum szerinti beolv.
- fgets(\$f[, \$hossz])
 - sor beolvasása
- fgetcsv(\$f, \$hossz[, \$elválasztó])
 - CSV sor beolvasása

□ Kiírás

- fwrite(\$f, \$s)
 - \$s kiírása
- fputs(\$f, \$s)
 - ld. fwrite()
- fprintf(\$f, \$formátum, \$változók)
 - formázott kiírás
- fputcsv(\$f, \$tömb[, \$elválasztó])
 - CSV sor kiírása

Magas szintű beolvasás/kiírás

13

□ Beolvasás

- `$tömb = file($fájlnev[, módosítók])`

- fájl soronként a tömbbe

- `$s = file_get_contents($fájlnev)`

- egész fájl szöveggént

- `readfile($fájlnev)`

- fájl kimenetre írása (pl. képek kiírása)

- `fpassthru($f)`

- megnyitott fájl maradék részének kimenetre írása (pl. képek)

□ Kiírás

- `file_put_contents($fájlnev, $s)`

- szöveg fájlba írása

További fájlműveletek

14

- `mkdir($útvonal)`
 - ▣ könyvtár létrehozása
- `rmdir($könyvtárnév)`
 - ▣ könyvtár törlése
- `copy($forrás, $cél)`
 - ▣ másolás
- `rename($mit, $mire)`
 - ▣ átnevezés, mozgatás
- `unlink($fájlnév)`
 - ▣ törlés
- `is_dir($fájlnév)`
- `is_file($fájlnév)`
- `is_readable($fájlnév)`
- `is_writable($fájlnév)`
- `is_link($fájlnév)`
- `is_executable($fájlnév)`

További fájlműveletek

15

- `basename($útvonal)`
 - ▣ fájlnevet adja vissza
- `chown($fájl, $user)`
- `chmod($fájl, $mód)`
- `chgrp($fájl, $group)`
- `stat($fájl)`
 - ▣ fájl adatai tömbben
- `fseek($f, $offset)`
 - ▣ fájlmutató mozgatása

Hibakezelés

16

- Hiba esetén → hamis visszatérési érték
- @ operátor: hibaüzenet kiírásának elnyomása

```
$f = fopen('nem_letezik.txt', 'r');  
if ($f) {  
    /* ... */  
    fclose($f);  
}
```

```
$f = @fopen('nem_letezik.txt', 'r');  
if ($f) {  
    /* ... */  
    fclose($f);  
}
```

17

Adatszerkezet-vezérelt tárolás

Adatszerkezet

18

- Adatszerkezet mentése és betöltése
- Fájl tartalma nem érdekes
- Az adat rendelkezésre állása az érdekes
- Adatszerkezet → szöveges formátum
- → Sorosítás

Sorosítás

19

- Sorosítás egy adatszerkezet visszaalakítható szöveges megfelelője.
- Tároláshoz, átküldéshez szokták használni
- Sorosító függvények
 - ▣ `serialize($érték) → szöveg`
 - ▣ `unserialize($szöveg) → érték`
 - ▣ `json_encode($érték) → szöveg`
 - ▣ `json_decode($szöveg) → érték`

Példa

20

- Rekordok tömbje
 - cím
 - rendező
 - év
 - szereplők

```
$filmek = array(
    array(
        'cim'          => 'Passió',
        'rendezo'     => 'Mel Gibson',
        'ev'          => '2004',
        'szereplok' => array(
            'Jim Caviezel',
            'Maia Morgenstern',
            'Christo Jivkov',
        ),
    ),
    array(
        'cim'          => 'Pio atya - A csodák embere',
        'rendezo'     => 'Carlo Carlei',
        'ev'          => '2000',
        'szereplok' => array(
            'Sergio Castellitto',
            'Sergio Albelli',
        ),
    ),
);
```

Serialize, unserialize

```
$s = serialize($filmek);  
echo $s;  
//a:2:{i:0;a:4:{s:3:"cim";s:7:"  
Passió";s:7:"rendezo";s:10:"Mel  
Gibson";s:2:"ev";s:4:"2004";s:9  
:"szereplok";a:3:{i:0;s:12:"Jim  
Caviezel";i:1;s:16:"Maia  
Morgenstern";i:2;s:14:"Christo  
Jivkov";}}i:1;a:4:{s:3:"cim";s:  
27:"Pio atya - A csodák  
embere";s:7:"rendezo";s:12:"Car  
lo  
Carlei";s:2:"ev";s:4:"2000";s:9  
:"szereplok";a:2:{i:0;s:18:"Ser  
gio  
Castellitto";i:1;s:14:"Sergio  
Albelli";}}}  
$filmek2 = unserialize($s);  
print_r($filmek2);
```

```
/*Array  
(  
  [0] => Array  
    (  
      [cim] => Passió  
      [rendezo] => Mel Gibson  
      [ev] => 2004  
      [szereplok] => Array  
        (  
          [0] => Jim Caviezel  
          [1] => Maia Morgenstern  
          [2] => Christo Jivkov  
        )  
      )  
    )  
  [1] => Array  
    (  
      [cim] => Pio atya - A csodák embere  
      [rendezo] => Carlo Carlei  
      [ev] => 2000  
      [szereplok] => Array  
        (  
          [0] => Sergio Castellitto  
          [1] => Sergio Albelli  
        )  
      )  
    )  
)*/
```

json_encode

23

```
$s = json_encode($filmek);  
echo $s;  
//[{"cim":"Passi\u00f3", "rendez  
o":"Mel  
Gibson", "ev":"2004", "szereplok"  
:["Jim Caviezel", "Maia  
Morgenstern", "Christo  
Jivkov"]}, {"cim":"Pio atya - A  
csod\u00e1k  
embere", "rendezo":"Carlo  
Carlei", "ev":"2000", "szereplok"  
:["Sergio Castellitto", "Sergio  
Albelli"]}]]
```

```
/*  
[  
  {  
    "cim":"Passi\u00f3",  
    "rendezo":"Mel Gibson",  
    "ev":"2004",  
    "szereplok":[  
      "Jim Caviezel",  
      "Maia Morgenstern",  
      "Christo Jivkov"  
    ]  
  },  
  {  
    "cim":"Pio atya - A csod\u00e1k  
embere",  
    "rendezo":"Carlo Carlei",  
    "ev":"2000",  
    "szereplok":[  
      "Sergio Castellitto",  
      "Sergio Albelli"  
    ]  
  }  
]  
*/
```

json_decode

24

```
$filmek3 = json_decode($s, true);  
print_r($filmek3);
```

```
/*Array  
(  
  [0] => Array  
    (  
      [cim] => Passió  
      [rendezo] => Mel Gibson  
      [ev] => 2004  
      [szereplok] => Array  
        (  
          [0] => Jim Caviezel  
          [1] => Maia Morgenstern  
          [2] => Christo Jivkov  
        )  
      )  
    )  
  [1] => Array  
    (  
      [cim] => Pio atya - A csodák embere  
      [rendezo] => Carlo Carlei  
      [ev] => 2000  
      [szereplok] => Array  
        (  
          [0] => Sergio Castellitto  
          [1] => Sergio Albelli  
        )  
      )  
    )  
)*/
```

Segédfüggvények

25

- Két segédfüggvény
 - ▣ Betöltés
 - ▣ Mentés
- Akármilyen adatszerkezetre működik

```
function fajlbol_betolt($fajlnev) {  
    $s = file_get_contents($fajlnev);  
    return json_decode($s, true);  
}  
  
function fajlba_ment($fajlnev, $adat) {  
    $s = json_encode($adat);  
    return file_put_contents($fajlnev, $s, LOCK_EX);  
}
```

Hibakezelés

26

- Hibalehetőségek beolvasásnál
 - nincs (még) meg a fájl
 - egyéb hiba történik
- → alapértelmezett szerkezet megadása

```
function fajlbol_betolt($fajlnev, $alap = array()) {  
    $s = @file_get_contents($fajlnev);  
    return (s === false  
        ? $alap  
        : json_decode($s, true));  
}
```

```
$filmek = fajlbol_betolt('filmek.txt', array());  
//vagy  
$filmek = fajlbol_betolt('filmek.txt');
```


Példa – új szereplő hozzáadása

27

```
$filmek = fajlbol_betolt('filmek.txt');  
  
$filmek[0]['szereplok'][] = 'Monica Bellucci';  
  
fajlba_ment('filmek.txt', $filmek);
```

```
[{"cim":"Passi\u00f3", "rendezo":"Mel  
Gibson", "ev":"2004", "szereplok":["Jim  
Caviezel", "Maia Morgenstern", "Christo  
Jivkov", "Monica Bellucci"]}, {"cim":"Pio atya - A  
csod\u00e1k embere", "rendezo":"Carlo  
Carlei", "ev":"2000", "szereplok":["Sergio  
Castellitto", "Sergio Albelli"]}]]
```

28

Fájlszerkezet-vezérelt tárolás

Használati esetek

29

- Fájl szerkezete kötött
 - mikroformátum
 - áttekinthetőség
 - interoperabilitás
- → Magas és alacsony szintű fájlműveletek

1. példa

30

- Adott filmcímek listája egy fájlban, soronként egy filmmel. Olvassuk ezt be egy tömbbe!

```
$filmek = file('lista.txt',  
              FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);  
/*  
Array  
(  
    [0] => A hobbit  
    [1] => A Gyűrűk Ura  
    [2] => Út a vadonba  
    [3] => Passió  
)  
*/
```

```
A hobbit  
A Gyűrűk Ura  
Út a vadonba  
Passió
```

1. példa – hibellenőrzés

31

```
$filmek = @file('lista.txt',  
                FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES)  
or die('Nincs meg a fájl');
```

□ vagy

```
$filmek = @file('lista.txt',  
                FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);  
if (!$filmek) {  
    $filmek = array();  
}
```

2. példa

32

- Egy filmcímeket tartalmazó tömb fájlba mentése (egy cím egy sor)

```
//Kiírás alacsony szintű
műveletekkel
define('SORVEG', "\n");
$f = @fopen('mesek.txt', 'w')
    or die('Hiba!');
if ($f) {
    foreach ($filmek as $film) {
        fputs($f, $film . SORVEG);
    }
    fclose($f);
}
```

```
//A tömb filmcímekkel
$filmek = array(
    'Vuk',
    'Lolka és Bolka',
    'Macskafogó',
    'Kisvakond és barátai',
);
```

```
define('SORVEG', "\n");
//Elemek összefűzése
$s = implode(SORVEG, $filmek) . SORVEG;
//Kiírás magas szintű művelettel
$siker = @file_put_contents('mesek.txt', $s);
```

3. példa

33

- Adott egy rekordokból álló tömb. Végezzük el a kiírását úgy, hogy egy sorban egy rekordnyi információ legyen, az egyes értékeket soron belül tabulátorral válasszuk el!

```
$filmek = array(
    array(
        'cim'      => 'Passió',
        'rendezo' => 'Mel Gibson',
        'ev'       => '2004',
    ),
    array(
        'cim'      => 'Pio atya - A csodák embere',
        'rendezo' => 'Carlo Carlei',
        'ev'       => '2000',
    ),
);
```

3. példa

34

```
$filmek = array(
    array(
        'cim'      => 'Passió',
        'rendezo' => 'Mel Gibson',
        'ev'       => '2004',
    ),
    array(
        'cim'      => 'Pio atya - A csodák embere',
        'rendezo' => 'Carlo Carlei',
        'ev'       => '2000',
    ),
);
```

```
define('SORVEG', "\n");
$f = @fopen('filmek.txt', 'w')
    or die('Hiba!');
if ($f) {
    foreach ($filmek as $film) {
        fputcsv($f, $film, "\t");
    }
    fclose($f);
}
```

Passió	Mel Gibson	2004	
Pio atya - A csodák embere	Carlo Carlei	2000	

4. példa

Passió Mel Gibson	2004
Pio atya - A csodák embere	Carlo Carlei 2000

35

```
//Beolvasás a file paranccsal, és az utólagos soronkénti bontás
$filmek = file('filmek.txt',
              FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
foreach($filmek as &$film) {
    list($c, $r, $e) = explode("\t", $film);
    $film = array(
        'cim'      => $c,
        'rendezo' => $r,
        'ev'      => $e,
    );
}
```

- Az előző példában kapott fájlt olvassuk be rekordok tömbjeként!

```
//Beolvasás soronként az fgetcsv-vel
$filmek = array();
$f = @fopen('filmek.txt', 'r');
if ($f) {
    while ($sor = fgetcsv($f, 0, "\t")) {
        $filmek[] = array(
            'cim'      => $sor[0],
            'rendezo' => $sor[1],
            'ev'      => $sor[2],
        );
    }
    fclose($f);
}
```

5. példa

36

- Az előző feladatbeli rekordok tömbjét tároljuk úgy a fájlban, hogy a rekord minden egyes mezeje külön sorba kerüljön, és az egyes rekordokat üres sor válassza el egymástól.
- Oldjuk meg a tömb beolvasását is!
- Alacsony szintű fájlműveletek

5. példa – kiírás

37

```
$filmek = array(
    array(
        'cim'      => 'Passió',
        'rendezo' => 'Mel Gibson',
        'ev'       => '2004',
    ),
    array(
        'cim'      => 'Pio atya - A csodák embere',
        'rendezo' => 'Carlo Carlei',
        'ev'       => '2000',
    ),
);
```

```
//Mentés fájlba
define('SORVEG', "\n");
$f = @fopen('filmek_tobbsor.txt', 'w')
    or die('Hiba!');
if ($f) {
    foreach ($filmek as $film) {
        fputs($f, $film['cim'] . SORVEG);
        fputs($f, $film['rendezo'] . SORVEG);
        fputs($f, $film['ev'] . SORVEG);
        fputs($f, SORVEG);
    }
    fclose($f);
}
```

```
Passió
Mel Gibson
2004
```

```
Pio atya - A csodák embere
Carlo Carlei
2000
```

5. példa – beolvasás

38

Passió
Mel Gibson
2004

Pio atya - A csodák embere
Carlo Carlei
2000

```
//Beolvasás fájlból
$filmek = array();
$f = @fopen('filmek_tobbsor.txt', 'r');
if ($f) {
    while (!feof($f)) {
        $cim = trim(fgets($f));
        $rendezo = trim(fgets($f));
        $ev = trim(fgets($f));
        $ures = fgets($f);
        if ($cim != '') {
            $filmek[] = array(
                'cim'           => $cim,
                'rendezo'      => $rendezo,
                'ev'           => $ev,
            );
        }
    }
    fclose($f);
}
```

Konkurens fájlhasználat

39

- Ha egyszerre többen hívják meg a szkriptet
→ konkurens használat
- flock(\$f, \$op)
 - \$op
 - LOCK_SH (olvasáshoz)
 - LOCK_EX (íráshoz)
 - LOCK_UN (kioldáshoz)
- fflush(\$f)
 - fájlpuffer ürítése

```
define('SORVEG', "\n");
$f = @fopen('mesek.txt', 'w')
    or die('Hiba!');
if ($f) {
    if (flock($f, LOCK_EX)) {
        foreach ($filmek as $film) {
            fputs($f, $film . SORVEG);
        }
        flock($f, LOCK_UN);
    }
    fclose($f);
}
```

Konkurens fájlhasználat

40

- Olvasáshoz is érdemes lockolni a fájlt, hogy közben ne kezdődhessen el egy írási folyamat.

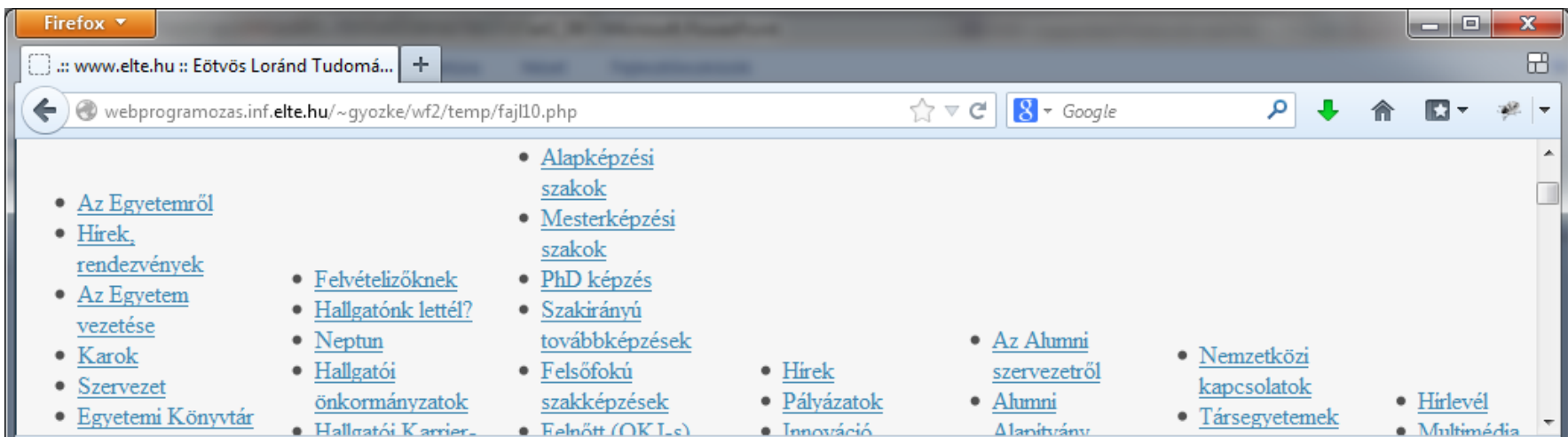
```
$filmek = array();  
$f = @fopen('lista.txt', 'r');  
if ($f) {  
    if (flock($f, LOCK_SH)) {  
        while (!feof($f)) {  
            $sor = trim(fgets($f));  
            if ($sor != '') {  
                $filmek[] = $sor;  
            }  
        }  
        flock($f, LOCK_UN);  
    }  
    fclose($f);  
}
```

URL-ek olvasása

41

- A fájlműveletek többsége nemcsak fájlt, hanem URL-t is képes olvasni (általában folyamokat)
- → PHP-ban fel tudunk dolgozni más oldalakat
- Egy egyszerű oldal megjelenítése:

```
$s = file_get_contents('http://www.elte.hu/');  
echo $s;
```



Hármas struktúra

43

- Adatok és függvények
 - adatszerkezetek megadása
 - adatok betöltése
 - betöltő és mentő függvény
 - adatok feldolgozásához kapcsolódó függvények
- Kérés feldolgozása
 - beolvasás (`$_POST`, `$_GET`, stb.) és ellenőrzés
 - feldolgozás meghívása
 - kimeneti adatok előkészítése
- Kimenet
 - HTML sablon

Vázlat

44

```
<?php
```

```
//Adatok
```

```
//IO függvények
```

```
function fajlbol_betolt($fajlnev, $alap = array()) {  
    $s = @file_get_contents($fajlnev);  
    return (s === false ? $alap : json_decode($s, true));  
}
```

```
function fajlba_ment($fajlnev, $adat) {  
    $s = json_encode($adat);  
    return file_put_contents($fajlnev, $s, LOCK_EX);  
}
```

```
//Adatokkal kapcsolatos függvények
```

```
function feldolgozas{$adat} {  
    //...  
}
```

```
//-----  
//Kérés feldolgozása  
//...
```

```
//-----  
//Kimenet  
?>
```

```
HTML sablon
```

Külön fájlba helyezés

45

- Funkcionálisan elváló részeket külön fájlokban tárolni
 - karbantartás
- Fájlok beemelése:
 - `include($fájlnév)` – hiba esetén warninggal továbbmegy
 - `include_once($fájlnév)`
 - `require($fájlnév)` – hiba esetén hibával megáll
 - `require_once($fájlnév)`
- Mintha odamásolnánk a fájl tartalmát

Vázlat

46

- fileio.php
 - ▣ betöltés és mentés
 - ▣ fix
- adat.php
 - ▣ feladatfüggő név és tartalom
- kimenet.php
 - ▣ feladatfüggő név és tartalom

```
<?php
include('fileio.php');
include('adat.php');

//Kérés feldolgozása
//...

include('kimenet.php');
```

Szétszedés hátránya

47

- Minden funkcióhoz három fájl tartozik
 - adat, kimenet, kérést fogadó
 - ez növeli a bonyolultságot
 - rontja az áttekinthetőséget
 - (javítja a karbantarthatóságot)
- Csak a kérés fogadót szabad meghívni!
 - ő húzza be a többi
 - erre figyelni kell

Tokenes védelem

48

- Kérést kiszolgáló szkriptben token definiálása

```
<?php
define('TOKEN', 'Védelem');

include('fileio.php');
include('adat.php');

//Kérés feldolgozása
//...

include('kimenet.php');
```

- Többi szkript első sora ezt vizsgálja

```
<?php if ( ! defined('TOKEN') ) exit('Közvetlenül nem elérhető!');
```

.htaccess védelem

49

- Könyvtár alapú konfigurációs fájl
- Apache képes kezelni
- Ehhez külön könyvtárba kell

Könyvtárba szervezés

50

- Az azonos funkcionalitású fájlok azonos könyvtárba kerüljenek
- Könyvtárak
 - ▣ közös
 - ▣ adat
 - ▣ kimenet
- Könyvtáranként .htaccess állomány

Könyvtárba szervezés

51

□ Könyvtárszerkezet

□ (D) kozos

- fileio.php
- .htaccess

□ (D) adat

- adat.php
- .htaccess

□ (D) kimenet

- kimenet.php
- .htaccess

□ fajl.php

□ fajl.php

```
<?php
define('TOKEN', 'Védelem');

include('kozos/fileio.php');
include('adat/adat.php');

//Kérés feldolgozása
//...

include('kimenet/kimenet.php');
```

□ .htaccess

```
deny from all
```

Összefoglalás

52

- Adattárolás a kérés-válaszon kívül
- Fájelkezelés
 - ▣ Adatszerkezet-vezérelt
 - sorosítás + magas szintű fájl műveletek
 - ▣ Fájl szerkezet-vezérelt
 - alacsony vagy magas szintű fájl műveletek
- Kódszervezés
 - ▣ hármas egység
 - ▣ fájlba és könyvtárba szervezés