



5. A PHP alapjai

BODÓ ZALÁN

Matematika és Informatika Kar,
BBTE, Kolozsvár

PHP

www.php.net



- ❑ PHP: Hypertext Processor (eredetileg: Personal Home Page)
- ❑ Rasmus Lerdorf, Andi Gutmans, Zeev Suraski
- ❑ 1994: eleinte: CGI programokból állt
- ❑ 1995: Form Interpreter + PHP: PHP/FI (PHP version 2)
- ❑ 1997: PHP 3 (PHP/FI 2)
- ❑ 2000: PHP 4 – Zend Engine
- ❑ 2004: **PHP 5**

☐ `<?php ... ?>`

☒ között szerepel a PHP-kód, amit a webserver értelmez:

☐ CGI-szkriptként való futtatás – minden egyes kéréskor új processz elindítása, majd befejezése (NE HASZNÁLJUK!)

☐ a PHP értelmező modulként beépül a webserverbe – sokkal hatékonyabb (!)

☐ `include fajlnev;` - beszúrja és kiértékeli a fájlt (általában: **.php** vagy **.inc**)

☐ `require fajlnev;` - ugyanaz mint az `include`, de "fatal error"-t ad nemlétező fájl esetén

☐ megjegyzések (kommentek): `//... és /* ... */`

A PHP nyelv elemei

regex:
[a-zA-Z_][a-zA-Z0-9_]*

- változók:
 - logikai, egész szám, lebegőpontos szám, sztring, tömb, objektum, erőforrás, NULL
 - nevük: \$-ral kezdődik

Pl.: `$a = "Hello, világ!";`
`echo "$a\n";`
- konstansok: `define(KNEV, ertekek);`
Pl.: `define(KONS1, 42); echo KONS1;`
- sztringek: `"..."` és `'...'`
 - adott elem: `$sztring{2}`
- aritmetikai, értékadó, összehasonlító operátorok - C és Perlhez hasonlóak
 - `===` - típus és érték egyenlőségvizsgálata

□ tömbök:

- numerikus - automatikus numerikus kulcsok
`$names = array("Peter","Quagmire","Joe");`
`echo $names[0];`
- asszociatív – tetszőleges kulcsok
`$ages = array("Jack"=>32, "Kate"=>30, "Sawyer"=>34);`
`echo $ages['Kate'];`
- többdimenziós (egymásba ágyazott)
`$families = array (
"Griffin"=>array ("Peter", "Lois", "Megan"),
"Quagmire"=>array ("Glenn"),
"Brown"=>array ("Cleveland","Loretta","Junior")
);`
`echo $families['Griffin'][2]; //ki: Megan`

Nyelvi szerkezetek

- if - hasonló a C és Perlhez
 - új: elseif
- switch – C
- while, do...while, for – C
- **foreach** – Perlhez hasonló:
foreach (\$array as \$value) {
 ...
}
és
foreach (tömb as kulcs => tömbelem) {
 ciklusmag
}

Függvények

```
function fnev(parameterok) {  
    ...  
}
```

- ❑ fnev – hasonlóan a változónevekhez
- ❑ paraméterek: \$p1, ..., \$pn
- ❑ visszatérési érték: nem adható meg;
return használata (C/C++)

Referenciák

□ referencia: `$a = &$b;`

□ paraméterátadás:

```
<?php
function foo(&$var) {
    $var++;
}
$a = 5;
foo($a); //ki: 6
?>
```

□ az objektumok referenciaként adódnak át

□ referencia visszatérítése:

```
function &valami(&$a) {  
    $a++;  
    return $a;  
}
```

```
$a = 1;  
$b = &valami($a);  
$a++;  
echo $a; //ki: 3  
echo $b; //ki: 3
```

□ referencia megszüntetése: unset(\$a);

Néhány fontosabb függvény

- ❑ **echo**
- ❑ **print_r** és **var_dump**
- ❑ **eval** – kiértékeli a paraméterként megadott PHP kifejezést tartalmazó sztringet
- ❑ **die** – hibaüzenet + szkript leállítása
- ❑ **unset** – a változó null-ra állítása
- ❑ *matematikai függvények*: sqrt, ceil, floor, rand, ...
- ❑ *fájlkezelés*: fopen, fclose, fwrite, fread, fgets, filesize, feof, file_exists, delete, rename, ...

Beépített globális változók

- ❑ `$_GLOBALS` – összes globális változó
- ❑ `$_SERVER` – a webszerver által beállított vagy az aktuális szkripttel kapcsolatos értékek: pl. `SERVER_ADDR`
- ❑ `$_GET` – a GET paraméterei
- ❑ `$_POST` – a POST paraméterei
- ❑ `$_COOKIE` – HTTP-n küldött cookiek
- ❑ `$_FILES` – feltöltött fájlok:
 - `$_FILES["file"]["name"]` - a feltöltött fájl neve
 - `$_FILES["file"]["type"]` - a feltöltött fájl típusa
 - `$_FILES["file"]["size"]` - a feltöltött fájl mérete byte-okban
 - `$_FILES["file"]["tmp_name"]` - a szerveren tárolt fájl másolatának átmeneti neve
 - `$_FILES["file"]["error"]` - a fájl feltöltése során kapott hibakód
- ❑ `$_ENV` – környezeti változók
- ❑ `$_REQUEST` – GET, POST és COOKIE paraméterei
- ❑ `$_SESSION` – a session-höz tartozó változók

Formok (űrlapok) kezelése

□ **\$_GET, \$_POST** (és \$_REQUEST)

Pl.:

```
print_r($_GET);  
echo $_GET["param"];
```

Fájlkezelés

- ❑ megnyitás:
 `$file=fopen("welcome.txt","r");`
 - módok: `r`, `r+`, `w`, `w+`, `a`, `a+`, `x`, `x+`
- ❑ fájl bezárása: `fclose($file)`
- ❑ fájl vége: `feof($file)`
- ❑ sor kiolvasása: `fgets($file)`
- ❑ karakter kiolvasása: `fgetc($file)`
- ❑ olvasás: `fread($file, $meret)`
- ❑ írás: `fwrite($file, $string)`

...

Cookie-k kezelése



- ☐ magyarul: **süti** (nem igazán használjuk)
- ☐ azonosításra használjuk
- ☐ **cookie** = a kliens gépen tárolt szöveges fájl
- ☐ műveletek:
 - cookie készítése
 - cookie ellenőrzése, kiolvasása

□ készítés:

`setcookie(name, value, expire, path, domain, secure, httponly);`

`name` – név (sztring)

`value` – érték (sztring)

`expire` – a cookie érvényességének ideje
(érdemes a `time()`, ill. `mktime()` függvényeket használni)

`path` – milyen könyvtárakban lesz érvényes a szerveren

`domain` – milyen doménen érvényes

`secure` – HTTP vagy HTTPS

`httponly` – csak HTTP protokollon elérhető

□ kiolvasás:

`$_COOKIE["cnev"]`

`isset($_COOKIE["cnev"])`

□ törlés: süti lejárt sütivé történő beállítása

- gyakran használt utasítás: az érvényességi időt 1 órával ezelőttre állítjuk:

`setcookie("cnev", "", time()-3600);`

Session-ök

- ❑ magyarul: **munkamenet** vagy **szesszió** (inkább az első)
- ❑ információ tárolása a felhasználó beállításairól a szerveren
 - az UID vagy sütiben van tárolva vagy az URL-en keresztül adódik át
- ❑ session indítása:
`session_start();`
- ❑ session-változók beállítása és kiolvasása:
`$_SESSION['vnev']`
- ❑ törlés:
 - `unset($_SESSION['vnev']);`
 - teljes session törlése: `session_destroy();`

MySQL adatbázisok



□ kapcsolódás:

`mysql_connect(server, user, passwd);`

server – szerver címe: port;

alapértelmezetten: localhost:3306

user – felhasználói azonosító

passwd – jelszó

□ kapcsolat bontása:

`mysql_close($con);` //a \$con a mysql_connect által
visszaadott erőforrás

□ AB létrehozása, kiválasztása, törlése:

`mysql_create_db($adatb, $con);` //vissza: bool

`mysql_select_db($adatb, $con);`

`mysql_drop_db($adatb, $con);`

□ mysql parancs végrehajtása:

`$res = mysql_query($sql, $con);`

□ adatok feldolgozása:

`$row = mysql_fetch_assoc($res);`

az aktuális rekordot a \$row asszociatív tömbbe tölti
adat elérése: pl. `$row['ID']`

OOP PHP-ban

- ❑ nagyon hasonló a C++-hoz
- ❑ egyszerű példa:

```
<?php
    class SimpleClass {
        // adattag
        public $var = 'a default value';

        // metódus
        public function displayVar() {
            echo $this->var;
        }
    }
?>
```

-
- hozzáférés-módosítók: public, protected, private (C++)
 - konstansok és statikus változók:
 - const, static
 - konstruktor/destruktor:
 - __construct(...)
 - __destruct()
 - öröklődés: extends kulcsszó (Java)
Pl.: class B extends A {

 ...

}

-
- ❑ `::` hatáskör(feloldó) operátor vagy PHP-ban *Paamayim Nekudotayim* (héberül: *dupla kettőspont*)
 - ❑ szülő osztály: `parent::$adat`
 - ❑ absztrakt osztályok, metódusok:
 - absztrakt metódus: `abstract protected function valami(...)`
 - absztrakt osztály: ha legalább 1 metódusa absztrakt: `abstract class oszt ...`

-
- ❑ interfészek: mint Java-ban (implements)
 - ❑ túlterhelés: mást jelent, mint más programozási nyelvekben
 - itt: amikor elérhetetlen adatot vagy metódust próbálunk elérni
 - ❑ `__set($name, $value)`
 - ❑ `__get($name)`
 - ❑ `__isset($name)`
 - ❑ `__unset($name)`
 - ❑ `__call($name, $args)`
 - ❑ `__callStatic($name, $args)`

-
- ❑ iterátorok: végigmenni (iterálni) az objektum adattagjain (pl. `foreach`)
 - ❑ *final* kulcsszó: (mint Java-ban) lehet adattagra, metódusra vagy osztályra alkalmazni – nem lehet felüldefiniálni; ha osztály, akkor nem lehet származtatni belőle
 - ❑ objektummásoló: `clone`
 - ❑ típusjavaslat (*type hint*): objektum és array típusú paraméterek esetén