



Adatbázisok gyakorlat

Mi a PHP?



- A PHP (PHP: Hypertext Preprocessor) egy nyílt forráskódú, számítógépes szkriptnyelv, legfőbb felhasználási területe a **dinamikus weboldalak készítése**.
- A PHP-kódunk általában **egy web-szerveren fut**, ahol egy PHP-elemző végrehajtja a kódban lévő PHP nyelvbeli utasításokat, és az eredményt továbbítja a kliens böngészője felé.
- Tehát ha a PHP-programunk HTML-kódokat ír ki, akkor dinamikus HTML oldalakat generálhatunk



- A XAMPP elnevezés egy betűszó:
 - X (azaz, platformfüggetlen)
 - Apache HTTP Server – webservert
 - MySQL - adatbázis
 - PHP – programozási nyelv
 - Perl – programozási nyelv
- Számunkra elegendő a XAMPP Lite csomag is.
- <http://www.apachefriends.org/en/xampp.html>
- Itt válasszuk ki az operációs rendszert és a Lite verziót!

XAMPP - folytatás



- Miután feltelepítettük a XAMPP-ot, indítsuk el a **xampp control center** alkalmazást!
- Itt kezelhetjük a XAMPP-ot alkotó alkalmazások elindítását és leállítását. Indítsuk el az **Apache** webszervert!
- A XAMPP telepítési könyvtárában lévő **htdocs** mappába kell elhelyeznünk azokat a fájlokat, amelyeket elérhetővé szeretnénk tenni a weboldalunkról.
- A böngészőbe beírva ezen a címen érhetjük el ezeket a fájlokat: http://localhost/relatív_útvonal



- **Linux** alatt elég lesz egy webservert és PHP, amit a következő csomagok telepítésével érhetünk el:
 - apache2, php5
- A telepítésükhöz használjuk a **Synaptic** csomagkezelőt, és annak a keresés funkcióját!
- A weblapokat a következő mappában kell elhelyeznünk: **/var/www**
- Változtassuk meg ennek a mappának a tulajdonosát saját magunkra: `sudo chown felhasználónevünk /var/www`



- A PHP kód egy **.php kiterjesztésű** fájlban található.
- Ebben a **<?php és ?>** tag-ek közé kell elhelyeznünk a PHP-kódunkat.
- Ami ezen kívül esik, az HTML-kódként értelmeződik.
- Tetszőlegesen váltogathatják egymást a fájlban a PHP és HTML kódrészletek.
- Hozzunk létre egy index.php nevű fájlt a példának a webszerverünk megfelelő mappájában!
(...\XAMPP\htdocs illetve /var/www)

Az első programunk



- A PHP-kódunkat legtöbbször egy HTML oldal törzsébe ágyazzuk. Erre egy példa:
- ```
<html>
 <head>
 <title>"Első programunk"</title>
 </head>
 <body>
 <?php
 echo "Hello World!\n";
 ?>
 </body>
</html>
```



- Változókat a `$változónév = kezdőérték;` utasítással deklarálhathatunk.
- A változók neve deklaráláskor és használatukkor is a `$`-jellel kezdődik.
- A típust nem kell megadnunk, mert azt a PHP automatikusan kitalálja, és mellesleg dinamikus is.
- ```
<?php  
$txt="Hello World!";  
$x=16;  
?>
```


Szöveges változók



- Egyesítés (konkatenáció): . (pont)
- `<?php`
`$txt1="Hello World!";`
`$txt2="What a nice day!";`
`echo $txt1 . " " . $txt2;`
`?>`
- Sztring hossza: `strlen("Hello world!");`
- Keresés: `strpos("Hello world!","world");`
- Az első egyezés pozícióját, vagy `FALSE`-ot ad vissza.



- A C nyelvben megszokott módon. Kivétel az osztás, ami mindig valós osztást jelent, a % pedig egész értéket ad vissza.
- Az aritmetikai operátoroknál itt is használhatjuk az összevont értékadási formákat: `$a += $b;`
- Logikai operátorok: `==`, `!=`, `<`, `<=`, `>`, `>=`

If, elseif, else



- ```
<?php
$d=date("D");
if ($d=="Fri")
 echo "Kellemes hétvégét!";
elseif ($d=="Mon")
 echo "Hétfőn még a fű se nő!";
else
 echo "Szép napot!";
?>
```
- `date("D")` – a hét napjának lekérdezése



- Kétféleképpen hozhatunk létre tömböket:
- A tömbelemek felsorolásával az indexek automatikusan hozzárendelődnek 0-tól kezdődően:  
`$autok=array("Saab","Volvo","BMW","Toyota");`
- A tömbelemeket egyenként megadva az indexeket mi magunk határozzuk meg:  
`$autok[0]="Saab"; $autok[1]="Volvo";  
$autok[2]="BMW"; $autok[3]="Toyota";`
- `echo $autok[0] . " és " . $autok[1] .  
" svéd autók.";`

# Asszociatív tömbök



- Kulcs-érték párokat tartalmazó adatszerkezet.
- Az elemek felsorolásával:  

```
$eletkorok = array("Péter"=>32,
"András"=>30, "Tibor"=>34);
```
- Az elemek egyenkénti megadásával:  

```
$eletkorok['Péter'] = "32";
$eletkorok['András'] = "30";
$eletkorok['Tibor'] = "34";
```
- ```
echo "Péter " . $eletkorok['Péter'] .  
" éves.";
```

Ciklusok



- `$i=1;`
`while ($i<=5) { $i++; }`
- `$i=1;`
`do { $i++; }`
`while ($i<=5);`
- `for ($i=1; $i<=5; $i++) {`
 `echo "i = " . $i . "
";`
`}`

A foreach ciklus



- Tömböket járhatunk be a segítségével.
- ```
<html>
<body>
<?php
$szamok=array("egy","keet","haar");
foreach ($szamok as $ertek) {
 echo $ertek . "
";
}
?>
</body>
</html>
```

# A foreach ciklus - folytatás



- Asszociatív tömbök esetén a kulcs-érték párokat is bejárhatjuk vele:
- ```
<?php  
$szamok=array("egy"=>1,"keet"=>2,  
"haar"=>3);  
foreach ($szamok as $kulcs => $ertek)  
{  
    echo $kulcs."=".$ertek."<br>";  
}  
?>
```

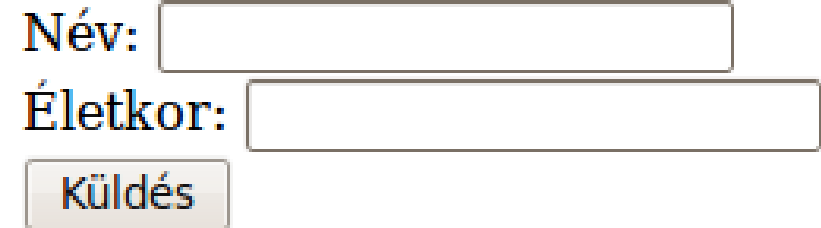



- ```
<?php
function összead ($a,$b) {
 $osszeg=$a+$b;
 return $osszeg;
}
echo "2 + 2 az néha " . összead(2,3);
?>
```
- A visszatérési érték típusát nem kell megadni, mert a típusok dinamikusak a PHP-ben.
- Nem kötelező visszaadni egy értéket, lehet egyszerű eljárást is írni (a `return` elhagyásával).

# Űrlapok kezelése



- Vegyünk egy egyszerű HTML-űrlapot:
- ```
<html>
<body>
<form action="udvozol.php" method="post">
Név: <input type="text" name="nev" /> <br>
Életkor: <input type="text"
name="eletkor" /> <br>
<input type="submit" />
</form>
</body>
</html>
```

A screenshot of the HTML form rendered in a browser. It shows two text input fields labeled 'Név:' and 'Életkor:'. Below the 'Életkor:' field is a button labeled 'Küldés'.

- Az űrlapot kitöltve, és a „Küldés” gombra kattintva a böngészőnk az „udvozol.php” oldalra ugrik.

Űrlapok kezelése - folytatás



- Az „udvozol.php” tartalma a következő:
- ```
<html>
<body>
Szia <?php echo $_POST["nev"]; ?>!

Te <?php echo $_POST["eletkor"]; ?> éves
vagy.
</body>
</html>
```
- A `$_GET` és a `$_POST` változókkal kérhetjük le az űrlapokban megadott információkat.

# A `$_GET` és `$_POST` asszociatív tömbök



- Ha HTML űrlapunkban, azaz a form elemünkben a `method` attribútumnak a `"get"` és a `"post"` értékeket adhatjuk meg.
- Ettől függően kell a `$_GET` vagy a `$_POST` tömböt használnunk a PHP kódunkban az űrlapban megadott adatok elérésére.
- A `$_REQUEST` asszociatív tömbben viszont elérhető a `$_GET`, a `$_POST`, és a `$_COOKIE` tömbök tartalma.

# Get vagy post?



- Ha a get metódussal küldjük az adatokat tovább, akkor azok a URL-ben meg fognak jelenni, és azok nem lehetnek bármilyen hosszúak, míg a POST metódusra ezek nem igazak.
- `http://localhost/proba/udvozol.php?nev=Endre&eletkor=22` – ez get esetén lesz így.
- A get metódussal jelszavakat ne küldjünk!
- Ha könyvjelzőt szeretnénk létrehozni egy oldalhoz, akkor viszont szükséges a get metódus használata.



- A `$_SESSION` szintén egy asszociatív tömb. Egy weboldal minden egyes látogatójához külön session változó tartozik, így minden látogatóról külön-külön információkat tudunk eltárolni.
- A `$_SESSION`-ben lévő információk a weboldal minden lapjáról elérhetőek, tehát egy látogató globális változóiként foghatók fel.
- A PHP-ban a session-t a `<html>` elem előtt kell elindítani, mivel az nem jön létre automatikusan:  

```
<?php session_start(); ?>
```

# Session-ök - folytatás



- Mivel a `$_SESSION` egy asszociatív tömb, így változók tárolására használhatjuk:
- `$_SESSION["nev"]="Péter";`  
`echo "A nevem: " . $_SESSION["nev"];`
- Az `isset($_SESSION["valtozo"])` utasítással tudjuk lekérdezni, hogy egy változót tároltunk-e a session-ben.
- Az `unset($_SESSION["valtozo"]);` utasítással kitöröljük a változót a session-ből, a `session_destroy();` utasítással pedig az egész session-t megszüntethetjük.

# A \$\_SERVER asszociatív tömb



- Hasznos információkat tartalmaz:
- DOCUMENT\_ROOT: a webről elérhető dokumentumokat tartalmazó mappa fájlneve
- PHP\_SELF: az aktuálisan futtatott PHP-szkript neve a DOCUMENT\_ROOT-hoz képest
- SERVER\_ADDR: a szerver IP-címe
- SERVER\_NAME: a szerver URL-je
- REMOTE\_ADDR: a kliens IP-címe



# Gyakorló feladatok



- Írjunk egy olyan rekurzív függvényt, amely visszaadja az első  $n$  szám összegét!
- Hozzunk létre egy asszociatív tömböt, amely az általunk felvett kurzusok név-kreditérték párosait tartalmazza, majd számítsuk ki az általunk felvett kreditek számát egy foreach ciklussal!
- Kérd be egy személy adatait egy űrlapon, majd írassuk azokat ki egy másik PHP-oldalon!
- Tárold el a neved egy session-változóban, majd egy másik PHP-oldalon írasd azt ki!



- Ugyanez megtalálható bővebben angolul itt:
- [http://www.w3schools.com/PHP/php\\_intro.asp](http://www.w3schools.com/PHP/php_intro.asp)
- A HTML-ről is hasznos bevezető érhető el itt:
- <http://www.w3schools.com/html/default.asp>
- További PHP referencia:
- <http://hu2.php.net/manual/en/index.php>
- Haladóbb linuxosok az általam készített LAMP-demót is böngészhetik:
- <http://www.stud.u-szeged.hu/Palatinus.Endre/LAMP%20demo.zip>