



Bevezetés a PHP programozásba

Oktatási segédanyagok:

1. 5.gyakorlat – hallgatói felkészülési segédlet
2. Informatika 2 jegyzet 432-453
3. www.w3schools.com/php



Tartalom

- Mi a PHP?
- PHP alapok
- Előre definiált változók és függvények



Tartalom

- **Mi a PHP?**
- PHP alapok
- Előre definiált változók és függvények



Mi a PHP?

- Cél: dinamikus HTML oldalak generálása
- PHP == PHP: Hypertext Preprocessor

PHP motor

Bemenete: HTML+script kód

Kimenete: HTML

- A fájl kiterjesztése: .php
- Imperatív programozási nyelv
- Script nyelv ~ interpreter
- Szerver oldali programozási nyelv

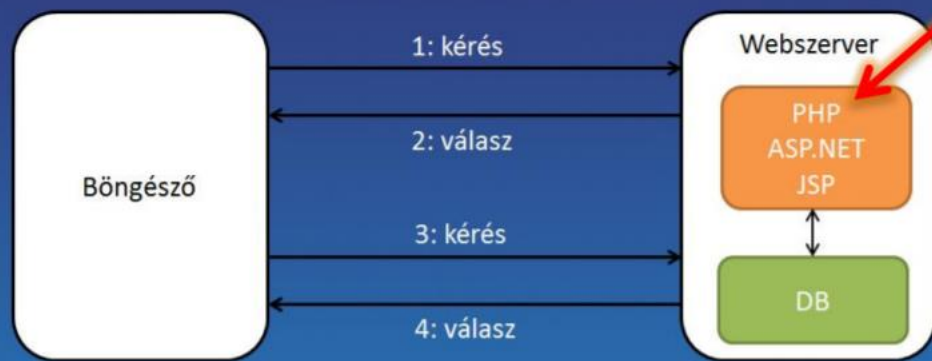
Kliens oldal: HTML, CSS, Javascript

Szerver oldal: PHP, SQL...



Mi a PHP?

A web működése



- Tipikus kérés: egy oldal címe
- Tipikus működés: sablon kitöltése DB alapján
- Tipikus válasz: HTML kód



Tartalom

- Mi a PHP?
- **PHP alapok**
- Előre definiált változók és függvények



PHP ALAPOK

- **PHP ~ “C” szövegfeldolgozást egyszerűsítő elemekkel (vezérlő szerkezetek,...)**
- **PHP = “C”+ (gyengén típusos, asszociatív tömb...)**



PHP fájl felépítése

➤ PHP fájl elemei:

■ HTML elemek

■ PHP elemek: `<?php ... ?>`

test.php fájl:

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head> <title> PHP teszt</title> </head>
```

```
  <body>
```

```
    <pre>
```

```
      <?php
```

```
        echo " Helló világ! ";
```

```
      ?>
```

```
    </pre>
```

```
  </body>
```

```
</html>
```

-> Helló világ!



Kimenet generálása, változók, szövegek

- `printf("Int: %05d Hexa: %04x \n",123,123);`
 HTML: Int: 00123 Hexa: 007b \n
 Megjelenítés: Int: 00123 Hexa: 007b
- `$v=123;` *(változónév kezdete : a-zA-Z_,[128-255 asc kar.)*
`printf("Int: %05d Hexa: %04x \n",$v, $v);`
- `$s=sprintf("Int: %05d Hexa: %04x \n",$v, $v);`
`printf("%s ",$s);`
- `print($s);` ~ `echo($s);` `print $s;` `echo $s;`
print függvény, return értéke 1, így kifejezésben használható
echo nem függvény, lehet több paramétere: `echo " STR:",$s;`
- Szövegösszefűzés: `print(" eleje " . $s . " vége\n");`



Idézőjelek használata

- szimpla idézőjel csak `\'` `\\` helyettesítés

```
print 'Ez \' \' egy $belso karaktersorozat.\n';
```

- dupla idézőjel

```
Ez ' \' egy $belso karaktersorozat.\n
```

- változó-helyettesítés
- spec. karakterkezelés `\"` `\\` `\$` `\n` ...
- kapcsos zárójel dupla idézőjelek között (kiértékelés mintha sztringen kívül lenne)

```
$belso = "próba";
```

```
print "Ez egy $belso karaktersorozat. \n";
```

```
Ez egy próba karaktersorozat.
```

```
print "xxx'yyy\\zzz\n";
```

```
xxx'yyy\zzz
```

```
print " ... {$_POST["nev"]} vagy {$_POST['nev']} ... " ;
```



Utasítások

- Minden utasítást ; zár
- Megjegyzések:
 - /* ... */
 - // ...
 - # ...



Utasítások - if

```
$nap="kedd";  
if ($nap=="hétfő") {  
    echo "A hét első napja";  
}  
elseif ($nap=="szombat" || ($nap=="vasárnap")){  
    echo "Hétvége";  
}  
else {  
    echo "Hétköznapi";  
}
```



Utasítások - switch

```
$nap="kedd";  
switch($nap) {  
    case "hétfő":  
        echo "A hét első napja";  
        break;  
    case "szombat":  
    case "vasárnap":  
        echo "Hétvége";  
        break;  
    default:  
        echo "Hétköznapi";  
        break;  
}
```



Utasítások – while, for

```
$i=0;
```

```
while ( $i<10 ){  
    echo $i++ , ' | ' ;  
}
```

```
$i=0;
```

```
do {  
    echo $i++ , ' | ' ;  
} while ( $i<10 );
```

```
for ( $i=0 ; $i<10 ; $i++ ) {  
    echo $i , ' | ' ;  
}
```



Gyengén típusos nyelv

➤ Gyengén típusosság jelentése

- a változó bárhol létrehozható
- típusa tartalom függő: `$v= "12";` `$v= 12;`
- futásidejű típuskiértékelés

➤ Típusok

- integer (int) `$v= 12;`
- double (float) `$v= 12.;`
- boolean (bool) `$v= false;`
- string `$v= "12";`
- array `$v=array(3,2,4);`
- (object)



Gyengén típusosság – változók kezelése

- *boolean* **isset**(\$v)
- **unset**(\$v)
- *string* **gettype** (\$v) *boolean* **is_integer**(\$v) ...
- *boolean* **settype**(\$v, "string") \$v = (**string**) \$v
- **var_dump**(\$v); - típus és érték kiírása
- **print_r**(\$v); - érték kiírása
- Összehasonlítás
 - \$str='32'; \$szam=32;
 - if (\$str == \$szam) echo "érték azonos";
 - if (\$str === \$szam) echo "típus és érték is azonos";
 - if (\$str !== \$szam) echo "típus vagy érték nem azonos";



Tömbök

- Indexelt tömb
- Asszociatív tömb
- Többdimenziós tömb



Indexelt tömb (numerikus kulcs)

➤ Létrehozása

```
$a=array("szöveg",12, true);  
$a[]= "következő";  
$a[4]="utolsó";
```

➤ Hivatkozás `echo $a[3];`

➤ Bejárás

```
$n=count($a);    for ( $i=0; $i<$n; $i++ ) echo $a[i];    !!!!!  
print_r($a);
```

➤ Törlés

```
unset($a[3]);    unset($a);
```



Asszociatív tömb - kulcs-érték párok

➤ Létrehozása

```
$admin["id"]=1;
```

```
$admin["nev"]="Pál";
```

```
$admin["kor"]=21;
```

...

```
$admin=array("id"=>1, "nev"=>"Pál", "kor"=>21);
```

(A tömb eleme tömb is lehet. Lehet vegyes indexelés.)

➤ Hivatkozás

```
echo $admin["nev"] ;
```

➤ Bejárás

```
foreach ($admin as $kulcs => $ertek)
```

```
    print $kulcs. ":" . $ertek. "<br />";
```



Példa:

```
$kor = array("Lajos"=>21, "Sandor"=>37, "Eszter"=>25);
```

```
var_dump($kor);
```

```
array(3) { ["Lajos"]=> int(21) ["Sandor"]=> int(37) ["Eszter"]=> int(25) }
```

```
print_r($kor);
```

```
Array ( [Lajos] => 21 [Sandor] => 37 [Eszter] => 25 )
```

```
foreach($kor as $x_key => $x_value) {
```

```
    echo "Key=" . $x_key. ", Value=" . $x_value;
```

```
    echo "<br>";
```

```
}
```

```
Key=Lajos, Value=21
```

```
Key=Sandor, Value=37
```

```
Key=Eszter, Value=25
```



Függvények

function nev(\$a, \$b, ...) // lehet referencia szerinti is &...

{

 \$v1=...;

global \$v2=...;

static \$v3=...;

 return ...

}



Tartalom

- Mi a PHP?
- PHP alapok
- Előre definiált változók és függvények



Előre definiált változók és függvények

- Űrlapok kezelése - `$_GET`, `$_POST`, `$_REQUEST`
- MySQL kezelése – `mysqli_...`
- SESSION (állapotkezelés – munkamenet)



Űrlapok (form és input) kezelése

➤ Feladat:

Kérés: http kérés összeállítása

Form/input használata PHP-ból

Név: Kor:

Válasz:

Üdvözljük LAJOS nevű felhasználónkat, aki 23 éves.



Űrlapok (form és input) kezelése

➤ test.html

http kérés összeállítása

```
<!DOCTYPE html>
```

```
<html>
```

```
<head><title>Test php</title></head>
```

```
<body>
```

```
<h1> Form/input használata PHP-ból </h1>
```

```
<form action="test.php" method="post">
```

```
    Név: <input type="text"      name="nev" />
```

```
    Kor: <input type="text"      name="kor" />
```

```
        <input type="submit" value="Küld" />
```

```
</form>
```

```
</body>
```

```
</html>
```



Űrlapok (form és input) kezelése

- Az űrlap tartalmát a php-ban \$_GET és \$_POST, illetve \$_REQUEST asszociatív tömbökkel lehet elérni.

- test.php

```
<!DOCTYPE html>
```

```
<html>
```

```
<head><title>Test php</title></head>
```

```
<body>
```

```
<h1> Form/input használata PHP-ból </h1>
```

```
<?php
```

```
    if ( isset( $_POST['nev'] ) && isset( $_POST['kor'] ) )
```

```
        echo "Üdvözljük {$_POST['nev']} nevű felhasználónkat,  
            aki {$_POST['kor']} éves.";
```

```
?>
```

```
</body> </html>
```



SQL kezelése - mysqli_...

- Beépített eljárások:

`mysqli_connect(...)`

`mysqli_query(...)`

`mysqli_fetch_assoc(...)`

`mysqli_free_result(...)`

`mysqli_close(...)`

- Feladat: A *nevek* táblában található személyek vezetéknév- és keresztnévének kiírása táblázatos formában



SQL kezelése - 1. rész

```
<?php
```

```
$host = 'localhost'; // '127.0.0.1';
```

```
$felhasznalo = 'x';
```

```
$jelszo = 'x';
```

```
$adatbazis = 'xxxx';
```

```
$keres = 'SELECT * FROM nevek';
```

```
$con = mysqli_connect($host, $felhasznalo, $jelszo, $adatbazis )  
or exit("hiba");
```



SQL kezelése - 2. rész

```
$eredmeny = mysqli_query($con,$keres) or exit("hiba");
```

```
echo "<table border=1>";
```

```
while($sor = mysqli_fetch_assoc($eredmeny))
```

```
{
```

```
    //foreach ($sor as $kulcs=>$ertek) echo $kulcs . $ertek . "<br />";
```

```
    echo          "<tr> <td>" . $sor['Vezeteknev'].
```

```
                  "</td><td>" . $sor['Keresztnev'].
```

```
                  "</td></tr>";
```

```
}
```

```
echo "</table>";
```

```
mysqli_free_result($eredmeny);
```

```
mysqli_close($con);
```

```
?>
```



SESSION (munkamenet-kezelés)

- Fogalma:
http állapotmentes kapcsolat, de mi mégis szeretnék állapotot tárolni. (pl. bevásárló kosár)
- A felhasználó azonosításának megvalósítása: http protokoll fejrészában
szerver beteszi az azonosítót: Set-Cookie: sessionid=....
kliens visszaküldi az azonosítót: Cookie: sessionid=....
- Beépített változók és függvények:
`session_start()`; a php állomány elejére
`$_SESSION` asszociatív tömb a kosár tartalmának tárolására
(`session_destroy()`;)



SESSION - feladat

➤ Feladat:

TÉTEL:

GYŰJT

TÖRÖL

KOSÁR:

alma

banán



SESSION 1.rész

test.php

```
...
<!DOCTYPE html>
<html> <body>

    <form action="test.php" method="POST">
        TÉTEL: <input type="text" name="tetel" />
        <input type="submit" value="GYŰJT" />
    </form>
    <br />
    <form action="test.php" method="POST">
        <input type="hidden" name="torol" value="1"/>
        <input type="submit" value="TÖRÖL" />
    </form>
    <br />
    ...
</html> </body>
```




SESSION 2.rész

```
<?php
```

```
    session_start();
```

```
    if (isset($_POST['tetel']))
```

```
        $_SESSION['kosar'][]=$_POST ['tetel'];
```

```
    if (isset($_POST['torol']))
```

```
        $_SESSION ['kosar']=array();
```

```
?>
```

```
<!DOCTYPE html>
```

```
<html> <body>
```

```
...
```



SESSION 3.rész

...

```
<form action="test.php" method="POST">
```

...

```
</form>
```

```
<?php
```

```
if (isset($_SESSION['kosar'])) {
```

```
    echo "KOSÁR: ";
```

```
    foreach ($_SESSION['kosar'] as $tetel)
```

```
        echo "<br />" . $tetel;
```

```
}
```

```
?>
```

```
</html> </body>
```



Támadások elleni védekezés PHP-ben

- Adatok (inputok) szűrése
- Szimpla idézőjelek használata, ha lehet
- Escape függvények használata
 - `mysqli_escape_string()`
 - `htmlspecialchars()`
- Még több info a PHP Security Guide-ban (pl: <http://phpsec.org/>)
- ...



➤ www.w3schools.com