



# 24. ÓRA

## Teljes példa (második rész)

Az előző órában felépítettük a klubok számára a bejegyző és karbantartó felületet, amivel új klubok és rendezvények kerülhetnek a rendszerbe. Most az átlagos felhasználók számára elérhető programokat kell elkészítenünk, melyek lehetővé teszik a listák böngészését.

Ebben az órában a következőket tanuljuk meg:

- Hogyan készítsünk olyan függvényeket, amelyek egyetlen lekérdezéssel több táblából hívnak le adatokat?
- Hogyan készíthetünk a paramétereiktől függő SQL parancsokat végrehajtó függvényeket?
- Hogyan menthetjük a felhasználó régebbi beállításait munkamenet-függvényekkel?
- Hogyan kell átalakítanunk az egyszerű szöveges adatokat, hogy a HTML kódba írassuk azokat?

## Az eseménynaptár nyilvános oldalai

Miután a tagok bejegyeztethetik klubjaikat és az eseményeket, itt az ideje, hogy az átlagos felhasználó számára is elérhető oldalak készítésébe fogjunk. Ezek az oldalak a listák böngészésére és nem a bennük való keresésre szolgálnak majd, bár nem lenne túl bonyolult ez utóbbit sem megvalósítani.

Ebben az órában négy oldalt készítünk el, amelyek lehetőséget adnak a felhasználónak, hogy kiírassa egy adott terület vagy téma rendezvényeit, bármilyen időpont szerinti szűréssel párosítva.

### esemenyekinfo.php

Az `esemenyekinfo.php` oldal lehetőséget ad a rendszerben lévő rendezvények böngészésére. Bizonyos szempontból hasonlít az `esemenylista.php` oldalra, amit az előző órában tárgyaltunk, de nagyobb rugalmasságot és több információt nyújt a felhasználónak.

#### 24.1. program `esemenyekinfo.php`

```
1: <?php
2:   include("adatbazis.inc");
3:   include("datum.inc");
4:   include("kozosfv.inc");
5:
6:   if ( isset($mitkellteni) && $mitkellteni ==
       "esemenyLista" )
7:       $munkamenet["esemenyek"] = $urlap;
8:   elseif ( $munkamenet["esemenyek"] )
9:       $urlap = $munkamenet["esemenyek"];
10:  else
11:  {
12:    $datum_tomb = getDate( time() );
13:    $munkamenet["esemenyek"]["terulet"] = "BÁRMELY";
14:    $munkamenet["esemenyek"]["tipus"] = "BÁRMELY";
15:    $munkamenet["esemenyek"]["honap"] = $datum_tomb["mon"];
16:    $munkamenet["esemenyek"]["ev"] = $datum_tomb["year"];
17:  }
18:
19:  $idoszak = datumIdoszak(
       $munkamenet["esemenyek"]["honap"],
20:    $munkamenet["esemenyek"]["ev"] );
```

**24.1. program** (folytatás)

```
21: function esemenyLista( )
22:     {
23:     global $idoszak, $munkamenet;
24:
25:     $esemenyek = esemenyekLekeres( 0, $idoszak,
26:                                     $munkamenet["esemenyek"]["terulet"],
27:                                     $munkamenet["esemenyek"]["tipus"] );
28:     if ( ! $esemenyek )
29:     {
30:         print "Nincs megjeleníthető esemény<p>";
31:         return;
32:     }
33:     print "<table border=1>\n";
34:     print "<td><b>Dátum</b></td>\n";
35:     print "<td><b>Esemény</b></td>\n";
36:     print "<td><b>Klub</b></td>\n";
37:     print "<td><b>Terület</b></td>\n";
38:     print "<td><b>Típus</b></td>\n";
39:     foreach ( $esemenyek as $sor )
40:     {
41:         print "<tr>\n";
42:         print "<td>".date("j M Y H.i",
43:                             $sor["edatum"])."</td>\n";
44:         print "<td><a
45:                 href=\"esemenyinfo.php?eazonosito=\".
46:                 $sor["eazonosito"]."\"&\".SID.\"\">\".
47:                 html($sor["enev"])."</a></td>\n";
48:         print "<td><a href=\"klubinfo.php?kazonosito=\".
49:                 $sor["eklub"]."\"&\".SID.\"\">\".
50:                 html($sor["klubnev"])."</a></td>\n";
51:         print "<td>\".$sor["teruletnev"]."</td>\n";
52:         print "<td>\".$sor["tipusnev"]."</td>\n";
53:         print "</tr>\n";
54:     }
55:     print "</table>\n";
56: }
```

**24.1. program** (folytatás)

```
57: </head>
58: <body>
59: <?php
60: include("kozosnav.inc");
61: ?>
62: <h1>Esemény információk</h1>
63: <p>
64: <form action="<?php print $PHP_SELF;?>">
65: <input type="hidden" name="mitkellteni"
        value="esemenyLista">
66: <input type="hidden"
        name="<?php print session_name() ?>"
67:         value="<?php print session_id() ?>">
68: <select name=urlap[honap]>
69: <?php honapLehetosegek( $idoszak[0] ); ?>
70: </select>
71:
72: <select name=urlap[ev]>
73: <?php evLehetosegek( $idoszak[0] ); ?>
74: </select>
75:
76: <select name=urlap[terulet]>
77: <option value="BÁRMELY">Bármely terület
78: <?php optionLista( "teruletek", $urlap["terulet"]) ?>
79: </select>
80:
81: <select name=urlap[tipus]>
82: <option value="BÁRMELY">Bármely típus
83: <?php optionLista( "tipusok", $urlap["tipus"] ) ?>
84: </select>
85:
86: <input type = "submit" value="Listázás">
87: </form>
88: </p>
89:
90: <?php
91: esemenyLista( );
92: ?>
93:
94: </body>
95: </html>
```

Szokás szerint az adatbázis.inc és a kozosfv.inc külső állományok beillesztésével kezdjük. Amellett, hogy elérhetjük majd az ezekben tárolt függvényeket, biztosak lehetünk abban, hogy élő adatbáziskapcsolattal rendelkezünk és aktív munkamenettel dolgozhatunk.

Ezután ellenőrizzük, hogy érkezett-e űrlapadat. Ezt a már ismert mitkellteni űrlapelemből származó \$mitkellteni változó vizsgálatával tehetjük meg. Ha kitöltött űrlap adatai érkeztek, azok előnyt élveznek minden korábban elraktározott adattal szemben. Az előző órában láthattuk, hogyan rendeltünk egy asszociatív tömböt a munkamenethez, hogy a belépett tag adatait nyilvántartsuk. Ebben az órában új elemeket adunk ehhez, amelyek a felhasználó beállításait tartalmazzák majd.

A \$munkamenet változót a kozosfv.inc állományban rendeltük a munkamenethez (mint azt már az előző órában láttuk):

```
session_start();  
session_register( "munkamenet" );
```

Most ezt a \$munkamenet tömböt többdimenziós tömbbé alakítjuk, mivel a \$munkamenet["esemenyek"] tömbelemnek egy asszociatív tömböt adunk értékül. Ebben a tömbben fogjuk tárolni a csak ehhez az oldalhoz tartozó beállításokat. Ezek a terület, típus, hónap és év értékek. Ha űrlap adat érkezett, a korábbi beállításokat ezzel írjuk felül.

Lehetséges, hogy a program futásakor éppen nem érkezett űrlapinformáció, de korábban már tároltuk a felhasználó által előnyben részesített beállításokat. Ebben az esetben a \$mitkellteni változó nincs beállítva, viszont a \$munkamenet["esemenyek"] tömb már tartalmaz elemeket. Ebben az esetben az \$urlap tömböt írjuk felül a korábban tárolt értékekkel. Ez biztosítja, hogy az űrlapban a megfelelő beállítások látsszanak.

Ha a felhasználó nem küldött be űrlap adatokat és korábban sem tároltuk a beállításait, nekünk kell alapértékekkel feltöltenünk a \$munkamenet["esemenyek"] tömböt. A getDate() beépített PHP függvényt használjuk arra, hogy elemeire bontsuk a dátumot. A visszaadott tömb alapján állítjuk be a \$munkamenet["esemenyek"]["honap"] és \$munkamenet["esemenyek"]["ev"] elemeket az aktuális hónap és év értékekre. Már tudjuk, hogy az év és hónap értékek érvényes adatot tartalmaznak, akár űrlapértékekből, akár korábbi beállításból, akár az aktuális dátumból származnak. Ezeket az értékeket a datum.inc egy új függvényének, a datumIdoszak() függvénynek adjuk át. Ez a függvény hónap és év paramétereket vár és két időbélyeggel tér vissza, megjelölve a hónap kezdetét és végét.

## 24.2. program Részlet a datum.inc fájlból

---

```

1: function datumIdoszak( $honap, $ev )
2:     {
3:         $eleje = mktime( 0, 0, 0, $honap, 1, $ev );
4:         $vege = mktime( 0, 0, 0, $honap+1, 1, $ev );
5:         $vege--;
6:         return array( $eleje, $vege );
7:     }

```

---

A függvény által visszaadott tömböt az \$idoszak globális változónak adjuk értékül.

Létrehozunk egy új esemenyLista() függvényt, amely a megfelelő eseményekről ír majd információkat a böngészőbe. Ezt később a HTML törzsében hívjuk meg.

Ahhoz, hogy az események listáját megkapjuk, az esemenyekLekeres() függvényt használjuk, amely az adatbazis.inc állományban található. Ezzel a függvénnyel érintőlegesen már az előző órában is találkoztunk, de kevés jó tulajdonságát ismerhettük meg.

## 24.3 program Részlet az adatbazis.inc fájlból

---

```

1: function esemenyekLekeres( $kazonosito=0, $idoszak=0,
                             $terulet=0, $tipus=0 )
2:     {
3:         global $kapcsolat;
4:         $lekeres = "SELECT klubok.klubnev, esemenyek.*,
                       teruletek.terulet as teruletnev,
                       tipusok.tipus as tipusnev ";
5:         $lekeres .= "FROM klubok, esemenyek, teruletek,
                       tipusok WHERE ";
6:         $lekeres .= "klubok.azonosito=esemenyek.eklub
7:                     AND esemenyek.eterulet=teruletek.azonosito
8:                     AND esemenyek.etipus=tipusok.azonosito ";
9:         if ( ! empty( $kazonosito ) && $kazonosito
              !="BÁRMELY" )
10:            $lekeres .= "AND esemenyek.eklub=
                          '$kazonosito' ";
11:         if ( ! empty($idoszak) )

```

**24.3 program** (folytatás)

---

```

12:         $lekeres .= "AND esemenyek.edatum
                >= '$idoszak[0]' AND
                esemenyek.edatum
                <='$idoszak[1]' ";
13:     if ( ! empty($terulet) && $terulet != "BÁRMELY" )
14:         $lekeres .= "AND
                esemenyek.eterulet='$terulet' ";
15:     if ( ! empty($tipus) && $tipus != "BÁRMELY" )
16:         $lekeres .= "AND esemenyek.etipus='$tipus' ";
17:     $lekeres .= "ORDER BY esemenyek.edatum";
18:     $eredmeny = mysql_query( $lekeres, $kapcsolat );
19:     if ( ! $eredmeny )
20:         die ( "esemenyLekeres hiba: ".mysql_error() );
21:     $vissza = array();
22:     while ( $sor = mysql_fetch_array( $eredmeny ) )
23:         array_push( $vissza, $sor );
24:     return $vissza;
25:     }

```

---

A függvény neve nem igazán fejezi ki sokoldalúságát. Négy paramétert vár: egy klubazonosítót, egy két időbélyeget tartalmazó tömbből álló időszakot, egy terület- és egy típuskódot. Minden paramétere elhagyható vagy hamis, illetve üres értékkel helyettesíthető.

A függvény lényegében egy SQL kérést állít össze a paraméterek alapján. A lekérés alapvetően összekapcsolja az adatbázisban lévő táblákat, ezzel biztosítva, hogy a klubnév, területnév és típusnév mezők is elérhetőek legyenek az eredménytáblában.

```

$lekeres = "SELECT klubok.klubnev, esemenyek.*,
                teruletek.terulet as területnev,
                tipusok.tipus as tipusnev ";
$lekeres .= "FROM klubok, esemenyek, teruletek,
                tipusok WHERE ";
$lekeres .= "klubok.azonosito=esemenyek.eklub
                AND esemenyek.eterulet=teruletek.azonosito
                AND esemenyek.etipus=tipusok.azonosito ";

```

Ezután a függvény további feltételeket ad ehhez, attól függően, hogy a megfelelő függvényparaméter be van-e állítva. A \$tipus és \$terulet paraméterek akkor is figyelmen kívül maradnak, ha a "BÁRMELY" karakterláncot tartalmazzák.

A gyakorlatban ez azt jelenti, hogy minél több paramétert kap a függvény, annál szűkebb eredménytáblát fogunk kapni. Ha nem adunk át semmilyen paramétert, az összes esemény információit megkapjuk. Ha csupán egy `$kazonosito` paramétert adunk meg, csak az adott klubhoz tartozó eseményeket kapjuk vissza. Ha csak a második `$idoszak` paramétert adjuk meg, csak az adott időszakba eső rendezvényeket láthatjuk és így tovább.

Végül a függvény az SQL parancsot végrehajtja és egy kétdimenziós tömbbel tér vissza, amely a kiválasztott események részleteit tartalmazza.

Miután a függvény által visszaadott kétdimenziós tömböt az `$esemenyek` változóba helyeztük, végig kell lépkednünk annak elemein. Az `edatum` elemeket felhasználva kiírjuk a formázott dátumokat a `date()` beépített függvény segítségével. Ezután egy hivatkozást készítünk az `esemenyinfo.php` oldalra, ahol több információ érhető el egy rendezvényről. Ehhez az esemény azonosítóját és a `SID` állandót kell átadnunk. Ebben a ciklusban egy újabb HTML hivatkozást is készítünk a `klubinfo.php` oldalra, a klub azonosítójával és a `SID` állandóval, végül kiírjuk a terület- és típusneveket a böngészőbe.

Talán feltűnt az előző órában, hogy egy `html()` nevű függvényt használtunk arra, hogy szöveges adatokat írjunk ki a böngészőbe. Az `esemenyLista()` függvényben, ahogy az események információin végiglépkedünk, ismét a `html()` függvényt hívjuk meg. Ez az általunk írt függvény a `kozosfv.inc` állományban található. Egy karakterláncot vár és annak böngészőbe íráshoz alkalmassá tett, átalakított változatával tér vissza. A különleges karakterek HTML elemekké alakulnak, az újsor karakterek például `<br>`-é.

#### 24.4. program Részlet a `kozosfv.inc` fájlból

---

```
1: function html( $szoveg )
2:     {
3:         if ( is_array( $szoveg ) )
4:             {
5:                 foreach ( $szoveg as $kulcs=>$ertek )
6:                     $szoveg[$kulcs] = htmlszoveg( $ertek );
7:                 return $szoveg;
8:             }
9:         return htmlszoveg( $szoveg );
10:    }
11:
```



**24.4. program** (folytatás)

```

12: function htmlspecialchars( $szoveg )
13:     {
14:     $szoveg = htmlspecialchars( $szoveg );
15:     $szoveg = nl2br( $szoveg );
16:     return $szoveg;
17:     }

```

Látható, hogy ez valójában nem egy, hanem két függvény. A `html()` egy karakterláncot vagy egy tömböt vár paraméterül. Ha tömböt kap, végiglépked rajta, átalakítva minden elemet, egyéb esetben magát a kapott karakterláncot alakítja át. A tényleges átalakítás egy másik függvényben, a `htmlspecialchars()`-ben valósul meg, amely két beépített függvényt alkalmaz a kapott karakterláncra. A `htmlspecialchars()` minden karaktert, ami nem jelenne meg helyesen a HTML kódban, a hozzá tartozó HTML elemre cserél. Az `nl2br()` minden újsor karaktert `<br>` sortöréssel helyettesít.

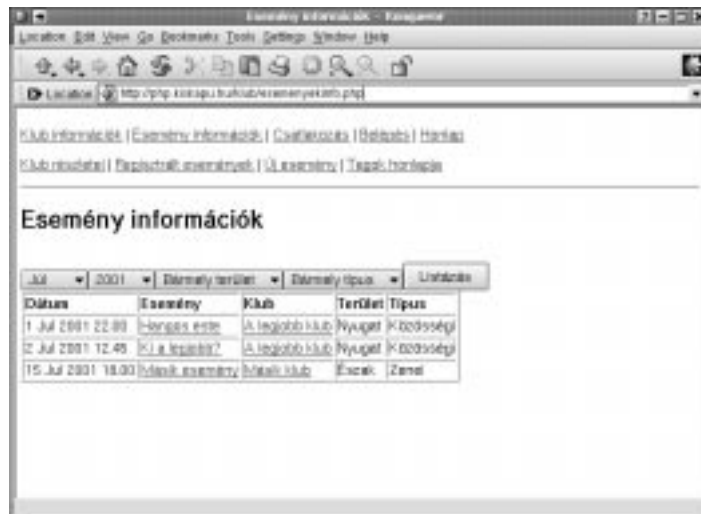
Visszatérve az `esemenyekinfo.php` oldalra, miután beállítottuk az alapadatokat és létrehoztuk az események kiírására szolgáló függvényt, már csak annyi van hátra, hogy egy űrlapot biztosítsunk a felhasználónak, amely lehetőséget ad a lista szűkítésére, illetve bővítésére.

A felhasználónak választási lehetőséget adó űrlap egyszerűen elkészíthető az előző órában megismert függvényekkel, melyek közvetlenül OPTION HTML elemeket írnak ki a böngészőbe.

Az `esemenyekinfo.php` egy lehetséges kimenetét a 24.1. ábra mutatja.

**24.1. ábra**

*Az `esemenyekinfo.php` kimenete*



## klubokinfo.php

A felhasználó jogos igénnyel nem csak rendezvények, hanem klubok szerint is ki szeretné íratni az adatbázis adatait, típus vagy terület szerint szűkítve azokat. A klubokinfo.php oldal ezt az igényt elégíti ki.

### 24.5. program klubokinfo.php

---

```
1: <?php
2:   include("adatbazis.inc");
3:   include("datum.inc");
4:   include("kozofv.inc");
5:   if ( isset($mitkellteni) &&
        $mitkellteni == "klubLista" )
6:       $munkamenet["klubok"] = $urlap;
7:   elseif ( $munkamenet["klubok"] )
8:       $urlap = $munkamenet["klubok"];
9:   else
10:      {
11:          $munkamenet["klubok"]["terulet"] = "BÁRMELY";
12:          $munkamenet["klubok"]["tipus"] = "BÁRMELY";
13:      }
14:   function klubLista( )
15:      {
16:          global $munkamenet;
17:          $klubok = klubokLekeres
                ( $munkamenet["klubok"]["terulet"],
18:                $munkamenet["klubok"]["tipus"] );
19:          if ( ! $klubok )
20:              {
21:                  print "Nincs megjeleníthető klub<p>\n";
22:                  return;
23:              }
24:          print "<table border=1>\n";
25:          print "<td><b>Klub</b></td>\n";
26:          print "<td><b>Terület</b></td>\n";
27:          print "<td><b>Típus</b></td>\n";
28:          foreach ( $klubok as $sor )
29:              {
30:                  print "<tr>\n";
```

**24.5. program** (folytatás)

```
31:         print "<td><a
           href=\"klubinfo.php?[kazonosito]=".
           $sor["azonosito"]."\"&\".SID.\"\">\".
32:         html($sor["klubnev"])."</a></td>\n";
33:         print "<td>$sor["teruletnev"]</td>\n";
34:         print "<td>$sor["tipusnev"]</td>\n";
35:         print "</tr>\n";
36:     }
37:     print "</table>\n";
38: }
39: ?>
40: <html>
41: <head>
42: <title>Klub információk</title>
43: </head>
44: <body>
45: <?php
46: include("kozosnav.inc");
47: ?>
48: <P>
49: <h1>Klub információk</h1>
50: <p>
51: <form action="<?php print $PHP_SELF;?>">
52: <input type="hidden" name="mitkelltenni"
   value="klubLista">
53: <input type="hidden" name="<?php print
   session_name() ?>"
54:         value="<?php print session_id() ?>">
55: <select name=urlap["terulet"]>
56: <option value="BÁRMELY">Bármely terület
57: <?php optionLista( "teruletek",
   $urlap[terulet] ) ?>
58: </select>
59: <select name=urlap[tipus]>
60: <option value="BÁRMELY">Bármely típus
61: <?php optionLista( "tipusok", $urlap["tipus"] ) ?>
62: </select>
63: <input type = "submit" value="Rendben">
64: </form>
```

### 24.5. program (folytatás)

---

```

65:  </p>
66:  <?php
67:  klubLista( );
68:  ?>
69:  </body>
70:  </html>

```

---

Láthatjuk, hogy ez a program szerkezetében és logikájában igencsak hasonlít az előző példában szereplőre. Az ehhez az oldalhoz tartozó „memória” azonban a \$munkamenet [ "klubok" ] tömbbe kerül. Ha űrlapadatok érkeznek, azokkal írjuk felül a korábbi tartalmát, ha nem érkezett űrlap és már be van állítva, akkor az \$urlap tömböt írjuk felül a \$munkamenet [ "klubok" ] változóval. Ha egyik feltétel sem teljesül, alapértékekkel töltjük fel.

Egy klubLista() függvényt hozunk létre, amelyben az adatbazis.inc egy új függvényét, a klubokLekeres()-t hívjuk meg. Ez a függvény a klub területére, illetve típusára vonatkozó elhagyható paramétereket vár.

### 24.6. program Részlet az adatbazis.inc fájlból

---

```

1:  function klubokLekeres( $terulet="", $tipus="" )
2:  {
3:      global $kapcsolat;
4:      $lekeres = "SELECT klubok.*, teruletek.terulet
                as területnev, tipusok.tipus
                as tipusnev ";
5:      $lekeres .= "FROM klubok, teruletek,
                tipusok WHERE ";
6:      $lekeres .= "klubok.terulet=teruletek.azonosito
                AND klubok.tipus=tipusok.azonosito ";
7:      if ( $terulet != "BÁRMELY" &&
            ! empty( $terulet ) )
8:          $lekeres .= "AND klubok.terulet='&#36;terulet' ";
9:      if ( $tipus != "BÁRMELY" && ! empty( $tipus ) )
10:         $lekeres .= "AND klubok.tipus='&#36;tipus' ";
11:     $lekeres .= "ORDER BY klubok.terulet,
                klubok.tipus, klubok.klubnev";
12:     $eredmeny = mysql_query( $lekeres, $kapcsolat );
13:     if ( ! $eredmeny )
14:         die ( "klubokLekeres hiba: ".mysql_error() );

```

## 24.6. program (folytatás)

---

```
15:     $vissza = array();
16:     while ( $sor = mysql_fetch_array( $eredmeny ) )
17:         array_push( $vissza, $sor );
18:     return $vissza;
19: }
```

---

A `klubokLekeres()` függvény paramétereit függvényében dinamikusan állít elő egy SQL utasítást. Alap esetben csupán összekapcsolja a klubok, területek és típusok táblákat. Ha üres karakterláncra és a "BÁRMELY" karaktersorozatra kívül bármi másra kap a `$terulet` vagy `$típus` paraméterekben, tovább szűkíti az eredménytáblát a `WHERE` feltételhez adott újabb elemekkel. Végül egy kétdimenziós tömbbel tér vissza.

A `klubokinfo.php` oldalon végiglépünk ezen a tömbön, kiírva a területnevet és típusnevet értékeket is a böngésző számára. Mint eddig, a klub neve most is egy hivatkozásban szerepel, amely a `klubinfo.php` oldalra mutat.

## klubinfo.php

A `klubinfo.php` oldal lehetőséget ad egyetlen klub minden adatának megtekintésére. Hivatkozások segítségével juthat ide a felhasználó, akár az `esemenyekinfo.php`, az `esemenyinfo.php` vagy a `klubokinfo.php` oldalakról is. Az oldalt az eddig megismert megoldások és függvények segítségével építhetjük fel, amint az alábbi kód mutatja.

## 24.7. program klubinfo.php

---

```
1: <?php
2: include("adatbazis.inc");
3: include("kozosfv.inc");
4: if ( ! isset($kazonosito) )
5:     header( "Location: klubokinfo.php?".SID );
6: $klub = klubLekeres( $kazonosito );
7:
8: $klub = html( $klub );
9: if ( $klub["email"] != "" )
10:    $klub["email"] = "<A HREF=\"mailto:$klub[email]\"
                    >".$klub["email"]."</A>";
```

**24.7. program** (folytatás)

```
11: function klubEsemenyei ()
12:     {
13:     global $kazonosito;
14:     $esemenyek = esemenyekLekeres( $kazonosito );
15:     if ( ! $esemenyek )
16:     {
17:         print "Nincs megjeleníthető esemény<P>";
18:         return;
19:     }
20:     print "<table border=1>\n";
21:     print "<td><b>Dátum</b></td>\n";
22:     print "<td><b>Esemény</b></td>\n";
23:     print "<td><b>Terület</b></td>\n";
24:     print "<td><b>Típus</b></td>\n";
25:     foreach ( $esemenyek as $sor )
26:     {
27:         print "<tr>\n";
28:         print "<td>".date("j M Y H.i",
29:             $sor["edatum"])."</td>\n";
30:         print "<td><a
31:             href=\"ese menyinfo.php?eazonosito= ".
32:             $sor["eazonosito"]." & ".SID." \">>.htm
33:             l($sor["enev"])."</a></td>\n";
34:         print "<td>".$sor["teruletnev"]."</td>\n";
35:         print "<td>".$sor["tipusnev"]."</td>\n";
36:         print "</tr>\n";
37:     }
38:     print "</table>\n";
39: }
40: ?>
41: <html>
42: <head>
43: <title>Klub részletek</title>
44: </head>
45: <body>
46: <?php
47: include("kozonav.inc");
48: ?>
49: <p>
50: <h1>Klub részletek</h1>
51: <h4><?php print $klub["klubnev"] ?></h4>
```

## 24.7. program (folytatás)

---

```
48: <p>
49: Terület: <b><?php print $klub["teruletnev"] ?></b>
50: <br>
51: Típus: <b><?php print $klub["tipusnev"] ?></b>
52: <br>
53: Email: <b><?php print $klub["email"] ?></b>
54: </p>
55: Ismertető:<br>
56: <b><?php print $klub["ismerteto"] ?></b>
57: <hr>
58: <?php
59: klubEsemenyei();
60: ?>
61: </body>
62: </html>
```

---

A program egy \$azonosito paramétert vár, amely egy klub azonosítószámát kell, hogy tartalmazza. Ha az oldal nem kapott ilyen paramétert, a felhasználót a klubokinfo.php oldalra küldjük. A klub adatainak kiderítésére egy új adatbázis.inc függvényt, a klubLekeres() -t alkalmazzuk. Ez a függvény egy klubazonosítót vár és egy tömböt ad vissza:

## 24.8. program Részlet az adatbázis.inc fájlból

---

```
1: function klubLekeres( $azonosito )
2:     {
3:         global $kapcsolat;
4:         $lekeres = "SELECT klubok.*, teruletek.terulet
                    as teruletnev, tipusok.tipus
                    as tipusnev ";
5:         $lekeres .= "FROM klubok, esemenyek, teruletek,
                    tipusok WHERE ";
6:         $lekeres .= "klubok.terulet=teruletek.azonosito
7:                     AND klubok.tipus=tipusok.azonosito
8:                     AND klubok.azonosito='$azonosito'";
9:         $eredmeny = mysql_query( $lekeres, $kapcsolat );
10:        if ( ! $eredmeny )
11:            die ( "klubLekeres hiba: ".mysql_error() );
12:        return mysql_fetch_array( $eredmeny );
13:    }
```

---

A klub adatainak megszerzésére a `sorLekeres()` függvényt is használhattuk volna. Azért készítettünk mégis egy célfüggvényt erre a feladatra, mivel így az eredménytábla a területnev és típusnev elemeket is tartalmazza majd. Ezt a klubok, teruletok és típusok táblák összekapcsolásával érhetjük el.

A `klubLekeres()` által visszaadott tömböt a `$klub` változóban tároljuk, tartalmát pedig a HTML törzsrészában írjuk ki a böngésző számára. Az oldalon létrehozunk még egy `klubEseményei()` nevű függvényt is, amely a klubhoz tartozó események kiírását végzi, felhasználva az `esemenyekLekeres()` néven korábban létrehozott függvényünket. A `klubinfo.php` egy lehetséges kimenetét a 24.2. ábra mutatja.

## 24.2. ábra

*A `klubinfo.php` kimenete*



## esemenyinfo.php

Az `esemenyinfo.php` az utolsó oldal, amit az alkalmazáshoz el kell készítenünk. Erre az oldalra bármely eseményinformációt adó lapról el lehet jutni egy hivatkozás segítségével. Az oldal minden információt megad az `$kazonosito` segítségével kiválasztott eseményről.



## 24.9. program eseményinfo.php

---

```
1: <?php
2:   include("adatbazis.inc");
3:   include("kozosfv.inc");
4:   if ( ! isset($eazonosito) )
5:       header( "Location: esemenyeklista.php?".SID );
6:   $esemeny = esemenyLekeres( $eazonosito );
7:   html( $esemeny );
8:   ?>
9:   <html>
10:  <head>
11:  <title>Esemény részletei</title>
12:  </head>
13:  <body>
14:  <?php
15:  include("kozosnav.inc");
16:  ?>
17:  <P>
18:  <h1>Esemény részletei</h1>
19:  <h4><?php print $esemeny["enev"] ?></h4>
20:  <p>
21:  Klub:
22:  <b>
23:  <?php print "<a href=\"klubinfo.php?kazonosito=\".
24:           $esemeny["eklub"]."\"&\".SID.\"\">
25:  $esemeny["klubnev"]</a>"
26:  ?>
27:  </b>
28:  <br>
29:  Terület: <b><?php print $esemeny["teruletnev"]?></b>
30:  <br>
31:  Típus: <b><?php print $esemeny["tipusnev"] ?></b>
32:  </p>
33:  Ismertető:<br>
34:  <?php print $esemeny["eismerteto"] ?>
35:  </body>
36:  </html>
```

---

Az eddigiek után ez az oldal már elég egyszerűnek látszik. Egyetlen új függvénnyel találkozhatunk, az `adatbazis.inc` állomány `esemenyLekeres()` függvényével. Ennek segítségével kapjuk meg az adott `$eazonosito`-hoz tartozó esemény részleteit. Mivel ezek egy tömbben rendelkezésre állnak, már csak ki kell írunk azokat a böngészőbe.

Az `esemenyLekeres()` függvényt a 24.10. példában láthatjuk. Tulajdonképpen csupán egy egyszerű SQL lekérésből áll, amely a klubok és az esemenyek táblákat kapcsolja össze.

### 24.10. program Részlet az `adatbazis.inc` fájlból

---

```
1: function esemenyLekeres( $azonosito )
2:     {
3:         global $kapcsolat;
4:         $lekeres = "SELECT klubok.klubnev as klubnev,
                    esemenyek.*, teruletek.terulet
                    as teruletnev, tipusok.tipus
                    as tipusnev ";
5:         $lekeres .= "FROM klubok, esemenyek, teruletek,
                    tipusok WHERE ";
6:         $lekeres .= "klubok.azonosito=esemenyek.eklub
7:                     AND esemenyek.eterulet=teruletek.azonosito
8:                     AND esemenyek.etipus=tipusok.azonosito
9:                     AND esemenyek.eazonosito='$azonosito'";
10:        $eredmeny = mysql_query( $lekeres, $kapcsolat );
11:        if ( ! $eredmeny )
12:            die ( "esemenyLekeres hiba: ".mysql_error() );
13:        return mysql_fetch_array( $eredmeny );
14:    }
```

---

## A jövő

Mostanra elkészültünk a teljes rendezvéynaptárral. Remélem sikerült érzékeltetni ezzel a valós életbeli kisalkalmazások dinamikus kialakításának lehetőségeit és a PHP szerteágazó képességeit.

Különösen érdemes megfigyelni, mennyire megkönnyítik munkánkat a PHP 4 munkameneteket kezelő függvényei, mivel ezek használatával igen egyszerű az adatok megőrzése kérésről kérésre. Ha a felhasználónk a munkamenet során később visszatér mondjuk az `esemenyekinfo.php` oldalra, pontosan azokat a beállításokat fogja találni, mint amelyeket legutóbb otthagytott. A munkamenet-függvények nélkül feltehetően sokkal több információt kellett volna kérésről-kérésre átadnunk oldalainknak az URL-ekben.

Bár az eseménynaptár bizonyos megközelítésből teljesen tekinthető, mindazonáltal csak egy jó prototípus, amit meg tudunk mutatni a megrendelőnek, ahogy haladunk a munkával. Az alkalmazás hasznára válna néhány további szolgáltatás, különösképpen egy kulcsszavas keresés. Kellemes lenne lehetőséget adni a látogatóknak, hogy megjegyzéseket fűzhessenek a rendezvényekhez. Ez új dimenziót nyitna az alkalmazás számára, igazán érdekes környezetté alakítva azt.

Lehetőséget adhatnánk a tagok számára, hogy képekre mutató hivatkozásokat helyezzenek el a klubok ismertetőiben. Akár azt is megoldhatnánk, hogy képeket tölthessenek fel böngészőjük segítségével.

A tagok feltehetően örülnének, ha az eseményeket lemásolhatnák vagy a rendezvények ismétlődő jellegét is kezelhetnék.

Mielőtt az alkalmazást átnyújtanánk a megbízónak, szükségünk lesz egy karbantartó felületre, amely alkalmas arra, hogy egy technikailag képzetlen adminisztrátor is módosíthasson vagy törölhessen tagokat, illetve az eseményeket, valamint a terület- és típuskategóriákat megváltoztathassa.

Valószínűleg feltűnt már, hogy programjaink kimenete eléggé spártai. Végül át kell adnunk a munkát egy grafikus tervezőnek, aki ügyes vezérlősvot, komoly grafikát és más elemeket ad az oldalhoz. Szerencsére a legtöbb PHP kód külső állományokban található, a HTML-től függetlenül, de elképzelhető, hogy szükség lesz ránk is a ciklusok újraírásánál.

## Összefoglalás

Ebben és az előző órában elkészítettünk egy teljesen működő többoldalas PHP alkalmazást. Gyakoroltuk az állapot mentésének, a felhasználói azonosításnak, az adatbázis-adatok módosításának és megjelenítésének módjait és sok más kérdést is érintettünk.

Egy többoldalas alkalmazás végigvitele egy könyvben nem könnyű dolog, de megéri a kitartást. Olyan kérdésekre adtunk válaszokat, olyan problémákat oldottunk meg, amelyekkel munkánk során időről időre találkozni fogunk. Majdnem minden programnak, amit írunk, egynél több felhasználó kéréseit kell majd egyidőben kiszolgálnia, tehát egyes megoldásokra van szükség az állapotok mentésére.

## Kérdések és válaszok

### **Ennyi volt. Hogyan tovább?**

Ez már nem a könyvön múlik. Elég információt találhatunk ebben a könyvben ahhoz, hogy saját kifinomult alkalmazásainkat el tudjuk készíteni. Ezzel és a Világhálón elérhető rengeteg információval nincs, ami megállíthatna bárkit is a fejlődésben!

## Műhely

A műhelyben kvízkérdések találhatók, melyek segítenek megszilárdítani az órában szerzett tudást. A válaszokat az A függelékben helyeztük el.

## Kvíz

1. Melyik függvény használható új elem tömb végére való beszúrásához?
2. Lehetséges egy új elem felvétele a tömb végére függvény használata nélkül is?
3. Melyik függvény alkalmazható a különleges karakterek átalakítására HTML-kiírás céljából?
4. Melyik függvény alakítja át az újsor karaktereket `<br>` elemekké?
5. A `SID` állandó a munkamenet-azonosító oldalról-oldalra küldéséhez hivatkozások készítésénél használható. Hogyan érhető el ugyanez a hatás űrlap készítésekor?

## Feladatok

1. Tekintsünk végig az órában tárgyalt forráskódokon. Találunk olyan megoldásokat, amiket fel tudunk használni saját programjainkban?
2. Lapozzuk át a könyvet és jegyzeteinket, ha írtunk ilyeneket. Jusson később eszünkbe, hogy érdemes kis idő elteltével újra átnézni a jegyzeteket, hogy minél több hasznunk legyen a befektetett munkából.