

# Honlapkészítés

## 3. Előadás – PHP+MySQL

# Van még hova fejlődni!

---

Eddig csak statikus dolgokkal foglalkoztunk. Ez mit is jelent?  
Egyszer megírtuk, úgy is marad, amíg át nem írjuk.  
Csak a kódoló beavatkozásával változik a tartalom.

Készítsünk dinamikus elemeket – például szavazás, fórum

Itt mi történik?

A felhasználók szabadon írhatnak pl egy fórumba → ezt tárolni kell!

A tárolás nem jó, ha ugyanabban a fájlban lenne → biztonság!

Tároljuk a felhasználói információkat egy másik fájlban, pl egyszerű txt-ben  
fórum → sok hozzászólás → nagy fájl → lassú!!!

szavazás → csak a válaszok darabszámait tároljuk, felülírjuk → jó lehet

Használjunk inkább adatbázist! → gyors, jól kezelhető → új nyelv: **SQL**

Az SQL az csak magának az adatbázisnak a nyelve

Kell még egy nyelv, amivel egy honlapról el tudjuk érni az adatbázist → **PHP**

PHP-ban használhatunk HTML elemeket is, sőt nem is kell, hogy feladjuk az HTML-t és a CSS-t. Elég, ha csak a szükséges részeket írjuk PHP-ban.

Mit nyújt? Változókat vezethetünk be, beépített függvények (pl szövegek kezelése – stringkezelés)

# SQL nyelv

SQL lekérdező nyelv, mely adatdefiníciós és adatmanipulációs nyelv egyben!

A MySQL egy kezelőfelület hozzá, de SQL-el működik pl az Office-ban megtalálható Microsoft Access is!

Hogyan is képzeljük el egy kezelőfelületet illetve egy adatbázist?

			id	cím	lead	szoveg	kep	datum	fooldalra	hely	tipus
<input type="checkbox"/>			15	Filmklub	Február 2-án 15:45-től vetítjük Ingmar Bergman: A ...			2007-01-24 19:58:33	nem	0	normal
<input type="checkbox"/>			16	Közszolgálati sztrájk	Tisztelt Szülők! Tájékoztatjuk Önöket, hogy...	A fenntartóval (XIII. kerületi Önkormányzat) egyez...		2007-02-09 06:26:46	nem	0	normal
<input type="checkbox"/>			17	Felvételi	Tisztelt Szülők és felvételiző Diákok! Szeretné...	<p>A Gimn&amp;acute;ziumi jelentkez&amp;acute;si lapok p...</p>		2007-02-16 06:51:49	nem	0	normal
<input type="checkbox"/>			10	Szalagavató képek	Felkerült a honlapra néhány kép a szalagavatóról.	A képeket ide kattintva lehet megtekinteni: <a hr...		2006-12-03 20:53:39	nem	0	normal
<input type="checkbox"/>			11	Szakmák napja	Szakmák napja első napi program (2006. december 6...	<p>&lt;p&gt;&lt;strong&gt;9 &amp;acute;ra&lt;/strong&gt;&lt;/p&gt; &lt;p&gt;&amp;#9679; M...</p>		2006-12-04 10:29:25	nem	0	normal
<input type="checkbox"/>			13	Felvételi	Központi írásbeli 6 osztályos: 2007. január 26. pé...	<b>Központi írásbeli eredménye megtekinthet? az a...		2007-01-11 10:25:07	nem	0	normal











































Mintha egy táblázatban tárolnánk soronként az összetartozó adatokat!






Például itt egy-egy sor egy-egy hírnek felel meg. Oszlopok (mezők):

- id – egyedi szám, azonosító (nem lehet két sornak azonos)
- cím – a hír címe
- lead – a hír bevezető szövege
- szoveg – a hír tartalma
- datum – keletkezés dátuma
- fooldalra – főoldalon kint legyen vagy csak az archívumban (igen/nem)
- hely – ha a főoldalon kint van, akkor sorrendben hanyadik legyen
- tipus – megjelenési típusa (csak címe, vagy kis bevezető is)

# SQL nyelv

A táblázat sorait rekordoknak, oszlopait mezőknek hívjuk, így  
rekord: összetartozó mezők összessége  
mező: azonos típusú adatok összessége → mezőnek van típusa

	Mező	Típus	Illesztés	Tulajdonságok	Nulla	Alapértelmezett	Extra	Művelet
<input type="checkbox"/>	id	bigint(20)			Nem	Nincs	auto_increment	     
<input type="checkbox"/>	cim	varchar(250)	latin2_hungarian_ci		Nem			     
<input type="checkbox"/>	lead	text	latin2_hungarian_ci		Nem	Nincs		     
<input type="checkbox"/>	szoveg	text	latin2_hungarian_ci		Nem	Nincs		     
<input type="checkbox"/>	kep	varchar(250)	latin2_hungarian_ci		Nem			     
<input type="checkbox"/>	datum	datetime			Nem	0000-00-00 00:00:00		     
<input type="checkbox"/>	fooldalra	enum('igen','nem','archiv')	latin2_hungarian_ci		Nem	igen		     
<input type="checkbox"/>	hely	bigint(20)			Nem	0		     
<input type="checkbox"/>	típus	enum('normal','link')	latin2_hungarian_ci		Nem	normal		     

Mind kijelölése / Kijelölés törlése A kijelöltekkel végzendő művelet:      

[Nyomtatási nézet](#) [Kapcsolat nézete](#) [Táblaszerkezet ajánlása](#)

1 mező hozzáadása A tábla végén A tábla elején után id Indítás

Egyes mezők típusa a példában:

- id – szám = bigint (20 jegy hosszú max.) – extra oszlopban auto\_increment = minden új sor egyel nagyobb azonosítót kap, mint az előzőleg felvett soré
- cím – karakterlánc (szöveg) = varchar (250 karakter hosszú max.)
- lead – szöveg = text (nincs korlátozva)
- szoveg – szöveg = text (nincs korlátozva)
- datum – dátum és idő típus = datetime (alakja: 0000-00-00 00:00:00)
- fooldalra – megadott értékekből kerülhet ki = enum (itt lehet: 'igen', 'nem', 'archiv')
- hely – szám = bigint (20 jegy hosszú max.)
- típus – megadott értékekből kerülhet ki = enum (itt lehet: 'normal', 'link')

# SQL nyelv

## Amit tennünk kell:

- Létrehozni egy új táblázatot az adatbázisban (MySQL felületen, megadva az oszlopok nevét és típusát)
- Jó, ha minden táblába elhelyezünk egy egyedi azonosítót (ID mező, extra legyen auto\_increment)
- Ha megvan a tábla szerkezete (ezt nevezzük relációs sémának), akkor kitöltjük adatokkal (reláció)

## SQL adatmanipulációs és adatdefiníciós nyelv:

Az adatokkal való feltöltést fogjuk mi végezni egy-egy oldalról (pl fórum üzenetet elküldünk) → manipulálás  
Természetesen lehet olyan eset is, hogy egy oldalról létre kell hozni egy új táblázatot és nincs lehetőség arra, hogy a MySQL felületen készítsük el, ez is megtehető az SQL-el → definiálás, erre itt nem térek ki. (lásd: SQL referencia → pl. „create table” parancs)

## SQL lekérdezés (kiolvassuk a táblázat valahány sorát) – előző példán:

```
SELECT id,cim,datum  
FROM news  
WHERE datum > '2006-12-31'
```

- ezt a három oszlopot választjuk ki
- a news táblából olvassuk ki az adatokat
- azokat a sorokat, ahol teljesül a feltétel

id	cim	datum
15	Filmklub	2007-01-24 19:58:33
16	Közzszolgálati sztrájk	2007-02-09 06:26:46
17	Felvételi	2007-02-16 06:51:49
10	Szalagavató képek	2006-12-03 20:53:39
11	Szalmák napja	2006-12-04 10:29:25
13	Felvételi	2007-01-11 10:25:07

```
SELECT id, cím, datum
```

id	cim	datum
15	Filmklub	2007-01-24 19:58:33
16	Közzszolgálati sztrájk	2007-02-09 06:26:46
17	Felvételi	2007-02-16 06:51:49
13	Felvételi	2007-01-11 10:25:07

```
WHERE datum > '2006-12-31'
```

# SQL nyelv

## Tehát a lekérdezésnél megadjuk:

mit szeretnénk látni az eredményben (SELECT)

honnan szedjük az adatot (FROM)

→ SELECT-ben csak olyan mező lehet, ami az itt megadott táblákban van  
feltétel, hogy melyik sorokat lássuk (WHERE)

## SELECT után:

**mezőnevek** lehetnek (csak a megadott oszlopokat adja vissza eredményül)

\* jel lehet = a megadott táblák összes oszlopát visszaadja

aggregátum: pl.

**min(datum)** – legkisebb dátumot adja vissza (van max() is)

**avg(mezo1)** – adott típusú mező átlagértékét adja vissza

**sum(mezo1)** – a valami oszlop összes elemének összegét adja vissza

**first(mezo1)** – első értéket adja vissza (van last() is)

**count(\*)** – sorok számát adja vissza (amire teljesül a feltétel)

egyéb: ucase(), lcase(), mid(), len(), round(), now(), format()

## FROM után:

**táblanevek**

**táblák összekapcsolása**

kod	gyereke_neve
1	ádám
1	dia
4	györgy

gyerekek tábla

id	nev	foglalkozas
1	béla	pék
2	jani	elnök
3	mari	titkár
4	zsuzsi	énekes
5	gábor	pék

emberek tábla

pl.: `f1 INNER JOIN f2 ON f1.id = f2.kod`

akkor használjuk, ha összetartozó adatokat pl két táblában tárolunk

az egyik tábla azonosítóinak (id) egy része megtalálható a másik táblázat egyik oszlopában

akár több sorban is (ez legyen most 'kod'),

akkor ezen a két mezőn össze tudjuk kapcsolni a két táblát

f1 és f2 táblákat kapcsoljuk össze, ahol id = kod

eredményben csak olyan id-jű, ami a másikban is van

id = kod	nev	foglalkozas	gyereke_neve
1	béla	pék	ádám
1	béla	pék	dia
4	zsuzsi	énekes	györgy

eredmény

# SQL nyelv

## WHERE után:

feltétel (mező >, <, <=, >=, =, <> mint egy adott érték)

mező tartalmaz-e egy szövegrészletet

– `nev LIKE 'zsuzs'` → eredmény: semmi

– `nev LIKE 'zsuzs%'` → eredmény: zsuzsi

% jel azt jelenti, hogy bármilyen karakterekkel (többel is) helyettesíthető  
az első példában viszont pontos 'zsuzs' egyezés kellett volna

több feltétel egymás után: pl.: `foglalkozas='pék' AND nev LIKE '*él'` → eredmény „béla” sor  
allekérdezés – pl.: `WHERE nev IN (SELECT...FROM...WHERE...)`

az eredeti WHERE akkor teljesül, ha a név = a zárójeles rész eredményével

használható még a BETWEEN operátor (vmilyen értékek közt – lehet szöveg is, ABC szerint)

pl.: `WHERE mezo BETWEEN érték1 AND érték2`

## Egyéb kifejezések:

GROUP BY mezo1

– azokat a mezőket adjuk meg, amiket nem aggregátumokkal (min,sum,avg...) használunk – ha van aggregátum

HAVING max(mezo1) > 3

– WHERE-hez hasonlóan feltételt tudunk vele meadni, de ezt aggregátumnál használjuk

– Aggregátum: sum(),max(),min(),avg(),count()...stb.

ORDER BY mezo1,mezo2 asc

– eredmény rendezése mezo1, majd azon belül mezo2 alapján, növekvő sorrendben

– növekvő sorrend = asc (ascending) | csökkenő sorrend = desc (decreasing)

## Példa:

```
SELECT foglalkozas FROM emberek
```

Ekkor minden foglalkozást kilistázunk. A „pék” 2x szerepel, így 2x lesz az eredményben is.

Ha azt szeretnénk, hogy azonos eredmény sorok esetén csak 1x írja ki, használjunk **SELECT DISTINCT**-et:

```
SELECT DISTINCT foglalkozas FROM emberek
```

# SQL szintaxis

---

Ha egy érték szám, akkor egyszerűen megadhatjuk – pl. id=3

Ha egy érték szöveg, akkor aposztrófok között adjuk meg – pl. nev='béla'

Ha valami bonyolultabb kifejezés, azt is aposztrófok közt adjuk meg – pl. datum < '2007-01-05 17:00:00'

Kis és nagybetűérzékeny, tehát lehet SELECT, Select, select és akár SeLeCt is

Az SQL nyelv szavait írjuk nagy betűkkel – pl.: ORDER BY



# SQL nyelv

---

Táblába új sor beszúrása:

```
INSERT INTO tablanev (mezo1,mezo2,mezo3) values ('alma',5,'igen')
```

Bezúrunk egy új sort a „**tablanev**” táblába

Felsoroljuk azokat a mezőket, amikbe a values szó utáni értékeket írjuk. A két felsorolás sorrendje legyen ugyanaz! Tehát itt '**alma**' kerül a **mezo1**-be, **5** a **mezo2**-be és '**igen**' a **mezo3**-ba.

Nem kell egy tábla összes mezőjét megadni a mezőlistában, a kihagyottakba az alapértelmezett érték kerül.

Pl. az előzőhöz

```
INSERT INTO emberek (nev,foglalkozas) values ('gábor','közalkalmazott')
```

```
INSERT INTO gyerekek (kod,gyereke_neve) values (????,'adorján')
```

A példában látjuk, hogy fel szeretnénk vinni gábor gyerekeit is a másik táblába, de nem tudjuk, hiszen az emberek tábla azonosítóját (id) nem mi adtuk meg, hanem auto\_increment generálódik. Ekkor előbb le kell kérdezni, hogy mit kapott gábor id-nek, így a következő a helyes:

```
INSERT INTO emberek (nev,foglalkozas) values ('gábor','közalkalmazott')
```

```
SELECT max(id) FROM emberek
```

```
INSERT INTO gyerekek (kod,gyereke_neve) values (???, 'adorján')
```

Még mindig nem vagyunk meg, hiszen tudjuk az id-t de el kell egy változóba menteni ahhoz, hogy meg lehessen utána adni → ez majd PHP... visszatérünk rá!

# SQL nyelv

## Tábla sorainak törlése

```
DELETE * FROM tablanev WHERE ___feltétel___
```

Szinte olyan, mint a lekérdezés, csak SELECT helyett DELETE

Értelemszerűen általában egy törlésnél egy egész sort törölünk, így \*-al az összes mezőt kiválasztjuk. A fenti lekérdezéssel a táblánk minden olyan sora törlődik, amire teljesül a megadott feltétel.

```
P1: DELETE * FROM emberek WHERE foglalkozas='pék'
```

Ekkor 2 sor törlődik ki: béla-é és gáboré.

id	nev	foglalkozas	kod	gyereke_neve
1	béla	pék	1	ádám
2	jani	elnök	1	dia
3	mari	titkár	4	györgy
4	zsuzsi	énekes		
5	gábor	pék		

## Tábla sorainak módosítása

```
UPDATE tablanev SET mezo1='újérték1', mezo2='újérték2'
```

Adott táblában a megadott mezők értékeit lecseréli arra, amit új értéként megadunk.

```
P1: UPDATE emberek SET foglalkozas='rocksztár'
```

Ez a kifejezés minden ember foglalkozását megváltoztatja rocksztárra. Ha azt szeretnénk, hogy csak bizonyos sorokban változzon a mezők értéke, akkor használjuk a WHERE feltételt itt is.

```
P1: UPDATE emberek SET foglalkozas='pék' WHERE nev='mari'
```

Ekkor csak mari foglalkozása változik meg titkárról pék-re.

\* Természetesen sokmindenről nem esett szó, csak a legfontosabb dolgokat vettük át SQL-ből

# PHP

---

**Az alapjait ismerjük az SQL-nek, most következzen a PHP, amivel a honlapunkról el tudjuk érni az adatbázist és változtatásokat tudunk eszközölni.**

PHP-ból most csak az adatbázishoz szükséges alapvető dolgokat tárgyaljuk!  
Ezek nagyjából manuálisak lesznek, egyszer kell megírni, utána copy-paste.  
A lényeg mindig a lekérdezés lesz SQL nyelven.

PHP kódot bárhol elhelyezhetünk a HTML oldalon belül.

Amelyik oldalban szerepel PHP kód, annak a kiterjesztését módosítsuk **.html** helyett **.php**-ra.  
Ezen kívül semmi mást nem kell tennünk, az eddigi HTML kódot is ugyanúgy fogja kezelni, csak **.html** kiterjesztéssel nem tudnánk a PHP kódot használni.

# PHP

```
<?php
$c = mysql_connect("http://adatbazishelye.hu", "felhasznamev", "jelszo");
mysql_select_db("adatbazis_neve", $c);

$sql="SELECT id,cim,lead FROM hirek ORDER BY datum desc";
$res=mysql_query($sql, $c);

while($row=mysql_fetch_assoc($res)) {
    print "<b>Hír címe: \n";
    printf ("%s" , $row["cim"]);
}
mysql_close($c);
?>
```

A PHP kód kezdetét mindig <?php, végét pedig ?> jelöli.

A lila színű részek amiket nekünk kell megadni, a többit mindig csinálhatjuk úgy, ahogy fent van.

A fenti kód első két sora teremt meg a kapcsolatot az adatbázishoz.

Első sorban megadjuk a weben lévő adatbázisunk címét, felh.nevünket és jelszavunkat.

Második sorban pedig az adatbázis nevét.

\$ jellel kezdődő szövegeket (pl. \$c, \$row, \$sql, \$akarmi) változóknak nevezzük.

\$c = valami – valami-t értékül adjuk \$c-nek, tehát \$c értéke valami lesz

Egy változó állhat az egyenlőség jobb oldalán is.

\$valtozo = "SQL lekérdezés";

Ebben a sorban adjuk meg a kívánt kifejezést (beszúrás, törlés, módosítás, lekérdezés)

Utána a \$c kapcsolaton végrehajtjuk a \$sql kifejezést, az eredményét \$res-ben elmentjük

# PHP

```
<?php
$c = mysql_connect("http://adatbazishelye.hu", "felhasznalonev", "jelszo");
mysql_select_db("adatbazis_neve", $c);

$sql="SELECT id,cim,lead FROM hirek ORDER BY datum desc";
$res=mysql_query($sql, $c);

while($row=mysql_fetch_assoc($res)) {
    print "<b>Hír címe: \n";
    printf ("%s" , $row["cim"]);
}
mysql_close($c);
?>
```

Azt szeretnénk, hogy soronként megadjuk a kiiratás módját, ehhez kell egy while ciklus a fentiek szerint, méghozzá addig, ahány sor van. A `$res` változónk tartalmazza az összes eredményt, ezt sorokra bontjuk, egy sor a `$row`-ban kap helyet. A while ciklus végigmegy az eredményünk minden során, és minden sort úgy ír ki nekünk, amilyen formában a ciklusban megadtuk.

## Kiiratás:

```
print "amit kiírunk, <span class='text'>lehet</span> itt html+css is \n";
    Ebben a sorban viszont nem használhatunk adatbázisból kiolvasott értéket.
printf („alkalmazott neve: %s <br>kora: %d", $row["mezo2"], $row["mezo1"]);
```

Itt használhatunk adatbázisból olvasott értékeket is.

`$row` az egy sor. Nekünk egyszerre csak 1-1 mezője kell: `$row["mezoamikell"]` | (~tömb és indexe)

Először " "-ok közt lehet bármi ami az előbbi print-nél is,

majd vessző után felsoroljuk, hogy melyik mezőket írjuk ki

`%s` – oda fog kerülni a kiiratáskor mezo2 értéke (s = string, szöveges mező esetén)

`%d` – oda fog kerülni a kiiratáskor mezo1 értéke (d = egész szám)

`%f` – f = float, lebegőpontos szám

Itt: `%s` után jön a sorrendben a `%d` – mezők sorrendjét eszerint rendeljük hozzá

Végén lezárjuk a kapcsolatot (`mysql_close($c)`) – ajánlott!

# Készítsünk egy vendégkönyvet!

**Először készítsük el a meglévő adatok kilistázását!**

```
<?php
$c = mysql_connect("*****", "*****", "*****");
mysql_select_db("*****", $c);

$sql="SELECT id,nev,email,szoveg,datum from $tabla order by datum desc";
$res=mysql_query($sql,$c);

print "<table width=600 cellpadding=0 border=0 align=center>\n";
while($row=mysql_fetch_assoc($res)){
    print „<tr><td width=150 class=table_fejlec_bal nowrap>\n";
    printf ("%s", $row["datum"]);
    print "</td>\n";
    print "<td width=400 class=table_fejlec_bal align=left>\n";
    $hsznev = $row["nev"];
    if($row["email"]!=""){
        printf ("<a href=mailto:%s style='color: white!important'>
                $hsznev</a>", $row["email"]);}
    else { print "$hsznev";}
    print "</td>\n";
    printf ("<td id=%s width=50 class=table_fejlec>", $row["id"]);
    printf ("#%d", $row["id"]);
    print "</td></tr>\n";
    print „<tr><td colspan=3 class=table>\n";
    $text1 = str_replace("\n", "<br>", $row["szoveg"]);
    echo "$text1\n";
    print "</td></tr> \n";
    print "<tr colspan=3><td>&nbsp;</td></tr> \n";
}
print "</table><br> \n";
mysql_close($c);
?>
```

# Készítsünk egy vendégkönyvet!

Valahogy így fog kinézni!

2009-08-15 20:05:57	bbb	#8
bbbbbbbbbbbbbbbbbbbb		
bbb		
2009-08-14 23:34:51	aaa	#7
aaaaaa		
2009-08-13 22:17:45	aaa	#6
aaabbb		
bac		
2009-08-13 21:12:48	aaa	#5
aaa		
2009-08-10 18:02:38	próba3	#4
van dátum is?		
0000-00-00 00:00:00	próba2	#3
ezaz		

A kódban a „\n” jel a sortörést jelenti a php kódban, tehát ettől még az oldalon nem lesz sortörtés (az <br>) de a böngészőben a forráskód megtekintésekor ott lesznek a php kód sortörései. Ha nem használjuk akkor olyan lesz, mint egy folytonos szöveg.

```
str_replace( "\n" , "<br>" , $row[ "szoveg" ] );
```

Ez a kifejezés kicseréli az üzenet készítése során leütött entereket sortörésekre, tehát ha az üzenet írója a beíráskor nyomott valahol entert akkor a megjelenítésnél is ott lesz az a sortörés.

# Készítsünk egy vendégkönyvet!

---

**Készítsük el az űrlapot, amin felvihetünk egy új üzenetet!**

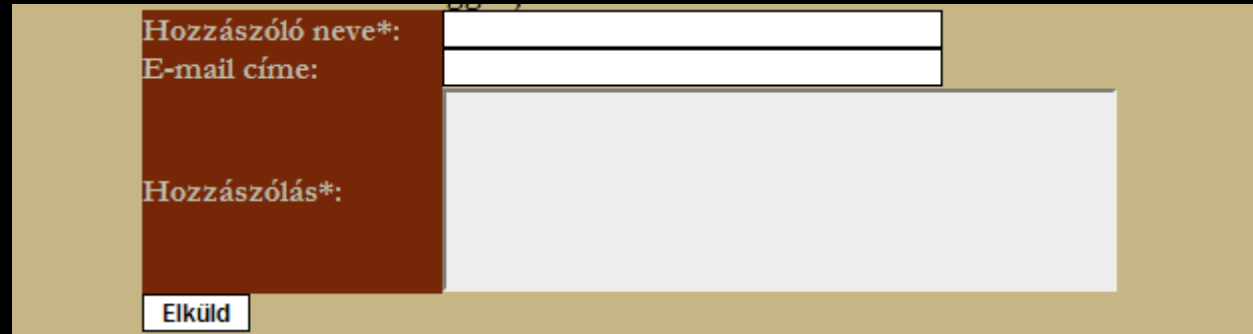
```
<form name="form1" method="post" action="">
  <table width="500" border="0" cellpadding="0" cellspacing="0">
    <tr>
      <td width="150" class="table_fejlec_bal">Hozzászóló neve*:</td>
      <td width="350"><input type="text" name="nev" id="textfield" class="hosszu"></td>
    </tr>
    <tr>
      <td class="table_fejlec_bal">E-mail címe:</td>
      <td><input type="text" name="email" id="textfield2" class="hosszu"></td>
    </tr>
    <tr>
      <td class="table_fejlec_bal">Hozzászólás*:</td>
      <td>
        <textarea name="szoveg" cols="45" rows="5" class="textmezo"> </textarea>
      </td>
    </tr>
    <tr>
      <td colspan="2"><input type="submit" name="vksubmit" id="button" value="Elküld"></td>
    </tr>
  </table>
</form>
```



# Készítsünk egy vendégkönyvet!

---

Valahogy így fog kinézni!



The image shows a web form for a guest book. It has a light beige background. On the left, there is a dark brown vertical bar containing the labels for the form fields. The fields are: 'Hozzászóló neve\*' (Guest name), 'E-mail címe:' (Email address), and 'Hozzászólás\*' (Message). The first two fields are short text boxes, and the third is a larger text area. At the bottom left of the form is a button labeled 'Elküld' (Send).

Figyeljünk, hogy jól nevezzük el az egyes űrlapelemeket (mind különböző és ha lehet beszédes nevű).

# Készítsünk egy vendégkönyvet!

---

És végül jöjjön az a PHP kód, amivel beszúrjuk az új adatokat az adatbázisba!

A kommentek segítik a megértést – php kommentek:

**// ez komment, mert két perjel után áll (de csak egysoros)**

**/\* ez egy többsoros komment kezdete (per-csillag), lezárása pedig csillag-per, tehát \*/**

```
<?php
```

```
/*
```

```
Itt először létre kell hozni a kapcsolatot... lásd előző diák
```

```
*/
```

```
/*$_POST egy globális változó, ez itt mindent jelent ami el lett küldve
```

```
az űrlap összes mezője küldött valamit, így ki kell választani egyenként (~printf)
```

```
tehát az egyes űrlapelemek értékeit adjuk át változóknak (a küldés gombot is!!!)*/
```

```
$nev = $_POST["nev"];
```

```
$email = $_POST["email"];
```

```
$szoveg = $_POST["szoveg"];
```

```
$vksubmit = $_POST["vksubmit"];
```

```
//folytatás a következő dián!
```

# Készítsünk egy vendégkönyvet!

```
// leellenőrizzük, hogy minden szükséges űrlapmező ki lett-e töltve, ha nem jelezzük (else)
// isset = létezik-e valamilyen változó
// pl. $nev<>" = a nev változó nem üres (tehát valami meg lett-e adva az űrlapmezőben)
if (isset($_POST["vksubmit"]) && $nev<>" && $szoveg<>" ){
    /*ellenőrizzük, hogy az email címnek megadott szöveg lehet-e email cím
    ezt az úgynevezett reguláris kifejezéssel tesszük, csak annyi érdekel minket, hogy
    ilyen alakú-e: valami@ezmegaz.rovid*/
    if($_POST["email"]<>" && $email ==
        (!ereg("^[_\.0-9a-z-]+@[0-9a-z][0-9a-z-]+\.[a-z]{2,6}$", $email))){
        //ha nem lehet ez fut le
        print "<span class=text_nagy_piros>Hibás e-mail cím formátum!</span>";
    }
    else {
        //ha lehet, akkor a beszúrás SQL kifejezése jön
        //NOW() az éppen aktuális időt jelenti
        $sql = "INSERT INTO vendegkonyv (nev,email,szoveg,datum) values
            ('$nev', '$email', '$szoveg', NOW())";
        //ha sikertelen a lekérdezés (FELKIÁLTÓJEL tagadást jelent)
        if(!mysql_query($sql,$c))
            //akkor ezt írja ki
            printf("<span class=text_nagy_kiemelt>HIBA!</span>");
        //ha pedig sikeres akkor ezt:
        else printf ("<span class=text_nagy_kiemelt>SIKER!<br></span>");
    }
}
/*ha nincs kitöltve minden szükséges, de el lett küldve az űrlap, vagyis a globális
változónak van Submit gomb értéke, ha ez nincs itt, akkor még űrlapküldés előtt is kiírja */
else if(isset($_POST["vksubmit"])){
    print "<span class=text_nagy_kiemelt>Nincs minden mező kitöltve!</span>";
}
//ide még kell mysql_close(**);
?>
```

# PHP – egy fontos funkció

Egy jó honlapon minden oldalnak ugyanaz a szerkezete, csak a tartalom változik.

Ezt eddig hogyan oldottuk meg?

**HTML:** minden egyes oldalra külön meg kell írni a szerkezetet, formázást

→ *változtatáskor*: mindenhol módosítani kell!

**CSS:** a formázás itt már egy fájlban külön szerepel (\*.css), de mondjuk a táblázatos módszert használjuk a szerkezethez, azt pedig minden oldalba be kell tenni

→ formázást csak 1 helyen kell módosítani, a css fájlba

→ ha a szerkezetet módosítjuk, akkor minden fájlban!

→ MEGOLDÁS CSS-ben: a már többször emlegetett <div>-ek használata HTML-ben a táblázat helyett

→ MEGOLDÁS máshogyan? Van!

**PHP:** php-ban egyszerűen megoldható, hogy csak egyszer kelljen módosítani a szerkezetet is!

Lépések:

1. Készítsük el egy oldalon (továbbiakban eredeti fájl) a szerkezetet, helyezzük el bele a grafikai-elemeket (kép) is.
2. Ahova a tartalmat szeretnénk elhelyezni, oda helyezzük el a következő php kódot:

```
<?php include "fajlnev.php"; ?>
```

Ekkor a fajlnev.php kódja egy az egyben be fog másolódni az eredeti fájlunk kódjába oda, ahova elhelyeztük.

Érezhetjük, hogy ez így még nem teljes! Ekkor mindig csak ugyanaz az egy fájl fog oda bemásolódni.

**Segítenek a változók!**

Az include-oláskor mindig egy változót hívunk meg, és mindig azt a fájlt „includoljuk be”, ami a változó tartalma.

Legegyszerűbb, ha a változó tartalma az elérési út az eredeti fájlunktól (ami az oldal keretét tartalmazza)

```
<?php  
    include "$fajlutvonal";  
?>
```

Hogyan fog ez a változó mindig különböző értéket kapni?

Űrlapoknál láthattuk, hogy a <form> tagnak van egy **method** attribútuma, mely lehet **post** és **get**.

Mi a **post** metódust használtuk, hiszen elküldtük az űrlapelemek tartalmát.

PHP globális változók közül megismertük a **\$\_POST** változót, mely pont az űrlapelemek elküldött értékeit tartalmazta.

Van olyan is, hogy **\$\_GET**! Ez kell nekünk! A get valahonnan kapja az értékét! Még hozzá az oldal címéből!

# PHP – egy fontos funkció

A POST-nál egy adott űrlapelemet így adtuk meg: `$_POST["neve"]`  
GET-nél sem lesz ez másként: `$_GET["neve"]`

De hol található ezt a „neve” attribútumot? Az oldal címében!

Pl: `http://www.honlapcime.hu/index.php?neve=ezaazerteke&xxx=yyyyy`

Ebből a címből két változót is kapunk:

```
$_GET["neve"]
```

```
$_GET["xxx"]
```

**Szintaxis:** a fájl neve után egy KÉRDŐJEL, majd az attribútum=érték párok, egymástól & jellel elválasztva  
Ékezetes betűket és szóközöket ne nagyon használjunk a cím utáni értékeknél  
Ilyenkor ki kell írni a fájlnevet is (pl index.php)

Ekkor így módosul a beírandó php kódunk:

```
<?php
    $fajlutvonal = $_GET["path"];
    include "$fajlutvonal";
?>
```

Készen vagyunk? NEM! Hogyan érjük el, hogy valamilyen változó bekerüljön az oldal címébe?

Az oldalon a likeket úgy adjuk meg:

Pl.: `<a href="index.php?path=./mappa/fajlnev.html">Link</a>`

**FONTOS!** Itt ha az index.php az a fájl ahol az includolás van, tehát az tartalmazza a keretét az oldalnak,  
akkor mindig az index.php-val kell kezdődnie a link útvonalának,

hiszen abban van a kódrészlet, ami az utána megadott „path” nevű attribútumot fel tudja dolgozni

Ha pl. erre a linkre kattintunk, akkor az index.php fog bejönni. A `$_GET["path"]` az a `./mappa/filenev.html` lesz,  
ezt az értéket fogja megkapni a `$fajlutvonal` változó is, így ezt a fájlt fogjuk beincludolni.

Észrevehetjük, hogy fölöslegesen használjuk a `$fajlutvonal` változót, hiszen a `$_GET[]` is ugyanazt tartalmazza

→ Egyszerűbben: `<?php include " '$_GET[„path”]." ' "; ?>`

→ Szintaxis bonyolultabb lett!!!

# PHP – egy fontos funkció

A beincludolt fájlokban csak és kizárólag a tartalom legyen, megfelelő formázással!

**FONTOS!** Azoknál a fájlknál, amiket beincudolunk figyelni kell a linkekre!

Ha abszolút útvonalat használunk pl egy képhez, akkor nem lesz gondunk, hiszen az mindenhol ugyanúgy éri el.

Relatív útvonalnál ha pl. van az index.php, amibe includolunk, és van egy „mappa” könyvtárunk, amiben van egy kép és mondjuk a szoveg.html is, amiben megjelenítjük a képet, akkor a szoveg.html-ből így hivatkoznánk rá: ./kep.jpg.

Ez addig jó, amíg nem includoljuk be a szoveg.html-t az index.php-ba!

Beincludoláskor olyan, mintha az index.php-ba kerülne a szoveg.html (a megfelelő helyre, ahol a PHP kód van).

De emiatt a relatív útvonal nem a szoveg.html helyétől számít, hanem az index.php helyétől!!!

Tehát includoláskor így jelenik meg jól a kép: ./mappa/kep.jpg

**Figyeljünk erre**, hogy

vagy mindig az index.php (keretet tartalmazó fájl) helyétől nézzük a relatív hivatkozásokat

vagy csak abszolút hivatkozásokat használjunk!

Persze ha egy mappában van minden akkor ez nem fordulhat elő, de ez nem célszerű! Sok fájlnál kell a mappa-struktúra!

A beincludolt fájloknak szedjük le a <html>...<body> illetve a </body></html> részeit, hiszen ezek a keretfájlban is megtalálhatóak és includoláskor nem szép, ha még egyszer bekerülnek:

```
<html>
<head>
<meta http-equiv="Content-Type"
content="text/html; charset=utf-8" />
<title>Untitled Document</title>
</head>

<body>
<strong><font face="Courier New, Courier,
monospace">Szövegelek</font></strong>
</body>
</html>
```

```
<strong><font face="Courier New, Courier,
monospace">Szövegelek</font></strong>
```

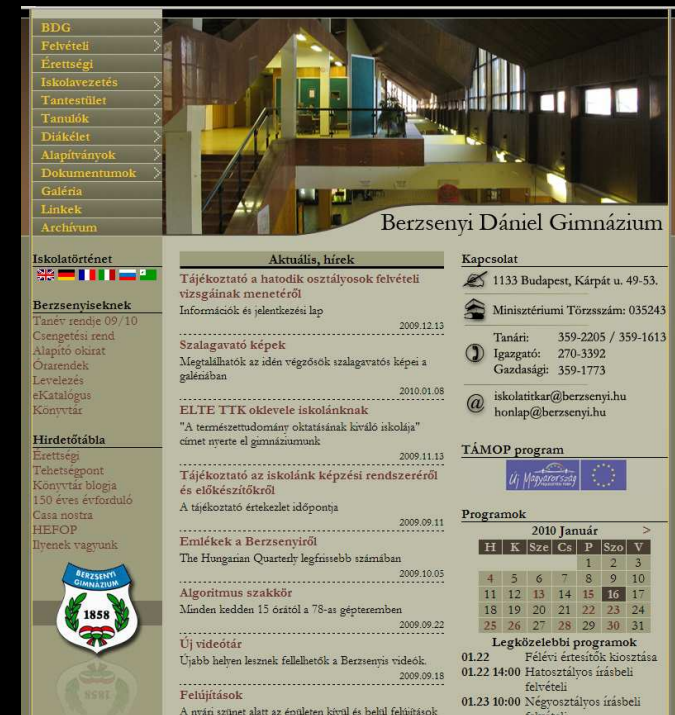
**ELÉG CSAK ENNYI!**

# PHP – egy fontos funkció (példa)



Ide kell képzelni az includolás PHP kódját!

Berzsenyi.hu keretfájl (fent) és két példa, hogyha van is valami beincludolva






# PHP – egy fontos funkció (példa)



Berzsényi.hu keretfájl (fent), egy példa, hogyha van is valami beincludolva (jobb) és **alul pedig a beincludolt fájl**. Látható, hogy az oldal kerete nincs benne! A táblázat az az incudolt rész formázása csak. Megfigyelhető, hogy sok helyen használunk PHP-t (kis sárga ikon), ezeken a helyeket adatbázisból nyerünk ki információt!

**Iskolatörténet**

**Hirdetőtábla**



**Aktuális, hírek**

**Kapcsolat**

✉ 1133 Budapest, Kárpát u. 49-53.

🏠 Minisztériumi Törzsszám: 035243


👤 Tanári: 359-2205 / 359-1613

📞 Igazgató: 270-3392

💰 Gazdasági: 359-1773

@ iskolatitkar@berzsényi.hu  
honlap@berzsényi.hu


**TÁMOP program**



**Programok**

**BDG**

- Felvételi
- Érettségi
- Iskolavezetés
- Tanterület
- Tanulók
- Diákélet
- Alapítványok
- Dokumentumok
- Galéria
- Linkek
- Archívum



**Berzsényi Dániel Gimnázium**

**Kapcsolat**

✉ 1133 Budapest, Kárpát u. 49-53.

🏠 Minisztériumi Törzsszám: 035243


👤 Tanári: 359-2205 / 359-1613

📞 Igazgató: 270-3392

💰 Gazdasági: 359-1773

@ iskolatitkar@berzsényi.hu  
honlap@berzsényi.hu

**TÁMOP program**



**Programok**

2010 Január

H	K	Sze	Cs	P	Szo	V
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

**Legközelebbi programok**

01.22 Félévi értesítők kiosztása

01.22 14:00 Határozatnyos írásbeli felvételi

01.23 10:00 Négyosztályos írásbeli felvételi

**Iskolatörténet**

**Aktuális, hírek**

Tájékoztató a hatodik osztályosok felvételi vizsgáinak menetéről

Információk és jelentkezési lap

2009.12.15

Szalagavató képek

Megtekinthető az idén végzősök szalagavató képei a galériában

2010.01.08

ELTE TTK oklevele iskolánkknak

"A természettudomány oktatásának kiváló iskolája" címet nyerte el gimnáziumunk

2009.11.13

Tájékoztató az iskolánk képzési rendszeréről és előkészítőkről

A tájékoztató értekezlet időpontja

2009.09.11

Emlékek a Berzsényiről

The Hungaria Quartet legfinest számában

2009.10.05

Algoritmus szakkönyv

Minden kedden 15 órától a 78-as géptermében

2009.09.22

Új videótár

Ujabb helyen lesznek fellelhetők a Berzsényis videók

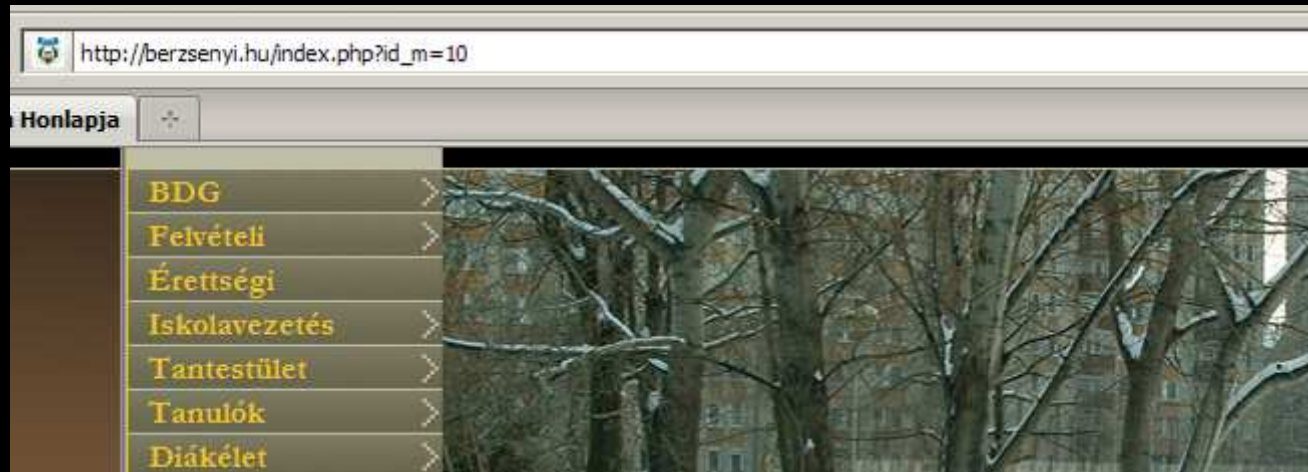
2009.09.18

Felújítások

A nyári szünet alatt az épületen kívül és belül felújítások



# PHP – egy fontos funkció (példa)



**Böngésző címsora:** berzsényi.hu/index.php?id\_m=10

Persze ilyen fájl nincs, itt egy bonyolultabb technikával van megvalósítva, de láthatjuk, hogy van valami nyoma. Egy baj még mindig van!

Mi történik, ha csak valaki beírja, hogy „berzsényi.hu”. Az index.php fog megnyílni, de mit fog beincludolni?

→ Semmit, hiszen nem létezik a keresett attribútum!

**Megoldás:**

```
<?php
    if(isset($_GET[„path"])){
        $fajlutvonal = $_GET["path"];
        include "$fajlutvonal";
    }
    else include „eztnyitomhanincs.php”;
?>
```

- ha van path  
- ha nincs

Tehát megadunk egy „alapértelmezett oldalt”, ami mindig betöltődik, ha nincs „path”-unk a címsorban. Ez lesz a főoldalunk. A többi oldalt úgymint navigálással érjük el.

# PHP – egy fontos funkció (példa)

## ÖSSZEFOGLALVA:

Készítsünk egy keretfájlt, amiben az oldal szerkezetét megadjuk (tipikusan ez az index.php lesz).  
Ebbe a fájlba helyezzük el a következő kódot oda, ahova majd mindig kerül a tartalom:

```
<?php
    if(isset($_GET[„path“])){
        $fajlutvonal = $_GET["path"];
        include "$fajlutvonal";
    }
    else include „eztnyitomhanincs.php“;
?>
```

- ha van path  
- ha nincs

Az összes többi fájlban csak a tartalmat kell tárolni

- csak a <body> tagon belül lévő kód maradjon meg ezekben a fájlokban
- figyeljünk, hogyha relatív hivatkozásokat használunk itt

Linkeket pedig mindig a következő alakban adjuk meg (ha azt szeretnénk, hogy a keretben nyíljon meg):

keretfajl.php?attrib1=ertek1

Itt egyelőre mindig elég lesz 1 db attribútum-érték pár!

Vége