



# PEAR

PHP Extension and Application Repository

**Mocsnik Norbert**

PHP Roadshow 2004/1.

2004. május 29., Dunaújváros



# Mi a PEAR?

- A PHP bővítmény- és alkalmazás raktára
- Egységes gyűjtemény
- 100% PHP kód
- Sokféle csomag
- Szigorú szabályok az egység fenntartására
- Kompatibilitás és bővítési lehetőség



# Egységes gyűjtemény

- Objektorientált kód
- Egységes kódolási szabályok
- Közös hibakezelés
- Egységes verziószámozási szabályok
- Közös csomagformátum
- Közös telepítő
- Közös levelezési listák, támogatás



# Csomagok

- Egymástól független, kicsi projektek
- Egymástól függetlenül kerülnek terjesztésre, telepítésre
- Leads, developer, helpers (vezetők, fejlesztők, segítők)
- Rendszeresen kiadott új verziók
- Minden csomag regisztrálva van a <http://pear.php.net> címen



# Sokféle csomag

- Jelenleg 232 csomag 32 kategóriában
- A PEAR kategória
- A legtöbb csomagot tartalmazó kategóriák:
  - DB
  - HTTP
  - HTML
  - XML



www.phpcon.hu

# Az összes kategória

- Authentication, Benchmarking, Caching, Configuration, Console, Database, Date and Time, Encryption, File Formats, File System, Gtk Components, HTML, HTTP, Images, Internationalization, Logging, Mail, Math, Networking, Numbers, Payment, PEAR, PHP, Processing, Science, Streams, Structures, System, Text, Tools and Utilities, Web Services, XML



# PEAR webhely

- <http://pear.php.net>
- Látogatás a PEAR weboldalára



# A csomagok elnevezése

- HTML\_Template\_IT  
`require_once 'HTML/Template/IT.php';`  
`$obj = new HTML_Template_IT;`
- DB\_DataObject\_FormBuilder\_FrontEnd  
`require_once`  
`'DB/DataObject/FormBuilder/FrontEnd.php';`  
`$obj = new`  
`DB_DataObject_FormBuilder_FrontEnd`





# A PEAR telepítő

- Telepítés–frissítés–eltávolítás–letöltés
- Frissítések listázása
- Minden telepített csomag frissítése
- Tallózás a telepített csomagok között
- Tallózás a nem telepített csomagok között
- Keresés a gyűjteményben
- (Csomagok összeállítása .tar.gz formátumba)



# PEAR telepítő - Beszerzés

- A PEAR telepítő alapjai szállításra kerülnek a PHP disztribúcióval
  - Nem mindig a legfrissebb verzió
- [go-pear.org](http://go-pear.org)
  - Legújabb változat
  - Parancssori (CLI) felület
  - Webes felület
  - Ablakozós (GTK) felület



# Telepítési módok

- Cél: a PEAR fájlok a számítógépre kerüljenek, hogy include-olni lehessen őket
- Parancssori telepítő: olyan helyeken, ahol parancssori hozzáféréssel rendelkezünk
- Webes telepítő: olyan tárhelyeken, amelyekhez csak FTP-vel férünk hozzá
- GTK frontend



# Parancssori telepítés

- <http://go-pear.org> forrásának lementése
- `go-pear.php` futtatása
- Telepítési instrukciók követése
- `include_path` megfelelő beállítása
- Bemutató



# Webes telepítés

- <http://go-pear.org> forrásának lementése
- Feltöltés a tárhelyre
- Írási jogosultság biztosítása
- `go-pear.php` meghívása
- `include_path` megfelelő beállítása
- Bemutató



# A telepítő további előnyei

- Platformfüggetlen
- A PEAR csomagformátumot használja
- A központi csomagnyilvántartásból dolgozik
- Felhasználható saját csomagok terjesztésére is



# Csomagnyilvántartás

- <http://pear.php.net>
- HTML interfész
  - dokumentáció
  - release menedzselés
  - tallózás a csomagok között
- XML\_RPC interfész
  - tallózás, információ lekérés
  - frissítések listázása
  - egyéb tevékenységek



# A csomagok felépítése

- .tar.gz fájlokban kerülnek terjesztésre
- Csomagok könyvtárszerkezete a CVS-ben  
/  
docs/, docs/examples/  
data/, misc/, scripts/, test/
- package.xml csomagleíró-fájl
- Telepítés során a csomagban lévő fájlok helye változhat a könyvtárstruktúrában!





# PEAR kódolási szabályok

- A PEAR-ben lévő csomagokra vonatkoznak
- Alkalmazhatjuk saját kódunkban is
- Fontos az átláthatóság
- `<?php ?>` forma használata a rövid forma helyett
- Sorhossz max. 75-85 karakter
- TAB helyett 4 db szóköz
- Kommentek kizárólag `//` és `/* */`



# Példák a kódolási szabályokból

- If utasítás

```
<?php
if ((feltetel1) || (feltetel2)) {
    muvelet1;
} elseif ((feltetel3) && (feltetel4)) {
    muvelet2;
} else {
    defaultmuvelet;
}
?>
```



# Példák a kódolási szabályokból

- Switch utasítás

```
<?php
switch (feltétel) {
    case 1:
        muvelet1;
        break;
    default:
        defaultmuvelet;
        break;
}
?>
```

# Példák a kódolási szabályokból

- Függvényhívás

```
<?php
    $var = foo($a, $b, $c);
?>
```

- Olvashatóság

```
<?php
    $rovid           = foo($bar);
    $hosszu_valtozo = foo($bar);
?>
```



# Példák a kódolási szabályokból

- Függvénydefiníciók

```
<?php
```

```
function tesztFuggveny($p1, $p2 = '')
```

```
{
```

```
    if (feltétel) {
```

```
        utasítás;
```

```
    }
```

```
    return $ertekek;
```

```
}
```

```
?>
```



# Verziószámozás

- major [.minor [.patch]] [pre]  
minor: új funkciók OK, BC rendben  
patch: bugfixek, apró változtatások  
pre: {dev, a, b, RC, pl}
- 0.x verzió "kísérleti" csomag
- $0.x < 1.x < 2.x$
- $0.1 < 0.9 < 0.10 < 1.0$
- $0.1dev1 < 1.0a1 < 1.0b1 < 1.0RC1 < 1.0 < 1.0pl1$

- Az alkalmazott licenszek csomagonként eltérőek
- Engedélyezett licenszek:
  - PHP Licenz
  - Apache Licenz
  - LGPL
  - BSD-stílusú licenszek
- Egy-egy csomag fejlesztőjének kérésére egyedi elbírálás alapján el lehet térni ezektől



# Hibakezelés

- PEAR\_Error
- Alosztályok származtathatók belőle (DB\_Error)
- Hibakezelés

```
<?php
$res = open_file($filename);
if (PEAR::isError($res)) {
    die($res->getMessage());
}
// sikeres, $res érvényes
?>
```



# Alkalmazás saját kódban

- Hibamentes lefutás esetén a várt értékkel térjünk vissza
- Hiba esetén generáljunk egy PEAR\_Error objektumot:

```
<?php
function teszt()
{
    return PEAR::raiseError(
        'A fájl nem található');
}
?>
```



# HTML\_Template\_IT

- ```
<html>
<table border>
  <!-- BEGIN row -->
  <tr>
    <!-- BEGIN cell -->
    <td>
      {DATA}
    </td>
    <!-- END cell -->
  </tr>
  <!-- END row -->
</table>
</html>
```



# HTML\_Template\_IT

```
• $data = array (  
    '0' => array('Stig', 'Bakken'),  
    '1' => array('Martin', 'Jansen'),  
    '2' => array('Alexander', 'Merz')  
);  
  
require_once 'HTML/Template/IT.php';  
  
$tpl = new HTML_Template_IT  
('./templates');  
  
$tpl->loadTemplatefile('main.tpl.htm',  
true, true);
```



# HTML\_Template\_IT

- ```
foreach($data as $name) {  
    foreach($name as $cell) {  
        $tpl->setCurrentBlock('cell') ;  
        $tpl->setVariable('DATA', $cell)  
  
        ;  
  
        $tpl->parse('cell') ;  
    }  
}  
$tpl->show() ;  
// vagy: $st = $tpl->get() ;
```



# HTML\_Template\_Flexy

- `<html>`
  - `<head>`
    - `<title>{title}</title>`
  - `</head>`
  - `<body>`
    - `<H1>{title}</H1>`
    - `<form name="form">`
      - `<table>`
        - `<tr>`
          - `<td>`
            - Input Box:
            - `<input name="input">`
    - `</td>`
  - `</form>`



# HTML\_Template\_Flexy

- ```
<tr flexy:foreach="numbers,number,string">
  <td>
    <a href="mypage.html?id={number}">
      {string}
    </a>
  </td>
</tr>
</table>
  this is number 2 : {numbers[2]}
  This is a {anObject.member}
  {someMethod()}
  {someMethod():h} <!-- prevent -->
</body>
</html>
```



# DB\_DataObject

- ```
$options = &PEAR::getStaticProperty  
( 'DB_DataObject', 'options' );  
$options = array(  
  
    'database' =>  
    'mysql://user:password@localhost/database',  
  
    'schema_location' =>  
    '/home/peartest/DataObjects',  
  
    'class_location' =>  
    '/home/peartest/DataObjects',  
  
    'require_prefix' => 'DataObjects/',  
    'class_prefix'   => 'DataObjects_',  
);
```



# DB\_DataObject

- `class DataObjects_Cat extends DB_DataObject`

```
{  
    ###START_AUTOCODE  
    /* the code below is auto generated  
do not remove the above tag */  
    var $__table = 'cat';  
    var $id;  
    var $nev;  
[...]  
    ###END_AUTOCODE  
}  
?>
```





# DB\_DataObject + Validate

- ```
function validateEmail() {  
    return Validate::email($this->email,  
true);  
}  
  
function validateHomepage() {  
    return Validate::url($this->homepage,  
true);  
}
```



# DB\_DataObject

- ```
<?php
$person = new DataObjects_Person;
$person->get(12);
echo $person->name;
print_r($person);
?>
```



# DB\_DataObject

- ```
<?php
$person = DB_DataObject::factory
('person');

$person->hair = 'red';
$person->has_glasses = 1;
$number_of_rows = $person->find();
while ($person->fetch()) {
    echo $person->name . '<BR>';
}
?>
```



# DB\_DataObject

- ```
<?php
$person = DB_DataObject::factory
('person');

$person->get(12);
$person->name = 'fred';
$person->update();
?>
```



# DB\_DataObject

- ```
<?php
$person = new DataObjects_Person;
$person->get(12);
$original = $person; // clone or copy
$person->name='fred';
$person->update($original);
?>
```



# DB\_DataObject

- ```
<?php
$person = new DataObjects_Person;
$person->removed = 1;
$person->whereAdd('age > 21');
$person->update();
?>
```



# DB\_DataObject

- `dbname.links.ini`

```
[person]
eyecolor = colors:name
owner = grp:id
picture = attachments:id
```



# DB\_DataObject

- ```
<?php
$person = new DataObjects_Person;
$person->eyeColour = 'red';
$person->find();
while ($person->fetch()) {
    $person->getLinks();

    echo "{ $person->name } has red eyes
and owns a { $person->_car->type } \n";
}
?>
```





# HTTP\_Upload

- ```
<html><body>
<form action="send.php" method="post"
enctype="multipart/form-data">
  Send these files:<br>
  <INPUT TYPE="hidden"
name="MAX_FILE_SIZE" value="100000">
  <input name="userfile"
type="file"><br>
  <input type="submit" value="Send
file">
</form>
</body></html>
```



# HTTP\_Upload

- ```
<?php
require 'HTTP/Upload.php';
$upload = new http_upload('hu');
$file = $upload->getFiles('userfile');
if (PEAR::isError($file)) {
    die ($file->getMessage());
}
```



# HTTP\_Upload

- ```
if ($file->isValid()) {  
    $file->setName('uniq');  
    $dest_dir = './uploads/';  
    $dest_name = $file->moveTo($dest_dir);  
    if (PEAR::isError($dest_name)) {  
        die ($dest_name->getMessage());  
    }  
    $real = $file->getProp('real');  
    echo "$real->$dest_name in $dest_dir";  
} elseif ($file->isMissing()) {  
    echo "No file selected";  
} elseif ($file->isError()) {  
    echo $file->errorMsg();  
}
```



# Image\_Transform

- ```
<?php
require_once 'Image/Transform.php';
$image = Image_Transform::factory('GD');
$image->load('./uploads/full/test.jpg');
$image->setMaxLength(250);
$image->addText(
    array('text' => '(c) your name'));
$image->save
('./uploads/thumb/test.jpg');
$image->free();
?>
```



# Mail\_Mime

- `$text = 'Text version of email';`

```
$html = '<html><body>HTML  
version</body></html>';
```

```
$file = '/home/peartest/example.php';  
$crlf = "\n";  
$hdrs = array(  
    'From'      => 'you@yourdomain.com',  
    'Subject'  => 'Test mime message'  
);
```



# Mail\_Mime

- `require_once 'Mail/mime.php';`

```
$mime = new Mail_mime($crlf);
```

```
$mime->setTXTBody($text);
```

```
$mime->setHTMLBody($html);
```

```
$mime->addAttachment($file,  
'text/plain');
```

```
$body = $mime->get();
```

```
$hdrs = $mime->headers($hdrs);
```



# Mail

- ```
require_once 'Mail.php';  
$mail =& Mail::factory('mail');  
$mail->send('example@example.com',  
$hdrs, $body);
```



# Mail\_Queue

- ```
$db_options['type']           = 'db';  
$db_options['dsn']           =  
'mysql://user:password@host/database';  
$db_options['mail_table'] =  
'mail_queue';
```

```
$mail_options['driver']      = 'smtp';  
$mail_options['host']       =  
'your_server_smtp.com';  
$mail_options['port']       = 25;  
$mail_options['auth']       = false;  
$mail_options['username']   = '';  
$mail_options['password']   = '';
```





# Mail\_Queue

- ```
require_once 'base.php';  
require_once 'Mail/Queue.php';  
  
$from = 'Chief <user@server.com>';  
$to = "admin <user2@server.com>";  
[.. Mail_Queue $hdrs, $body ..]  
  
$mail_queue =& new Mail_Queue(  
    $db_options, $mail_options);  
$mail_queue->put(  
    $from, $to, $hdrs, $body );
```



# Mail\_Queue

- ```
require_once 'base.php';  
require_once 'Mail/Queue.php';  
  
$max_amount_mails = 50;  
$mail_queue =& new Mail_Queue(  
    $db_options, $mail_options);  
$mail_queue->sendMailsInQueue(  
    $max_amount_mails);
```



# Auth

- ```
require_once "Auth/Auth.php";  
function loginFunction()  
{  
    echo "<form method=\"post\"  
action=\"index.php\">";  
    echo "<input type=\"text\"  
name=\"username\">";  
    echo "<input type=\"password\"  
name=\"password\">";  
    echo "<input type=\"submit\">";  
    echo "</form>";  
}
```

- ```
$dsn =  
"mysql://user:password@localhost/db";  
  
$a = new Auth(  
    "DB", $dsn, "loginFunction");  
  
$a->start();  
if ($a->getAuth()) {  
    /**  
    * site output  
    */  
}
```



# Auth\_PrefManager

- ```
require_once 'Auth/PrefManager.php';  
$options = array(  
    'table' => 'user_prefs');  
$prefs = new Auth_PrefManager(  
    $dsn, $options);  
  
$prefs->setDefaultValue(  
    'fullname', 'New User');  
$prefs->setValue(  
    'john', 'fullname', 'John Wood');
```



# Auth\_PrefManager

- `require_once 'Auth/PrefManager.php';`  
  
`$johnName = $prefs->getPref('john',  
'fullname');`  
  
`$billName = $prefs->getPref('bill',  
'');`
- `echo "John neve $johnName";`  
`echo "Bill neve $billName";`



# Log

- ```
require_once 'Log.php';  
$conf = array(  
    'mode' => 0600,  
    'timeFormat' => '%X %x');  
$log = &Log::singleton(  
    'file', '/tmp/user.log',  
    'test1', $conf, LOG_INFO);  
  
for ($i = 0; $i < 12; $i++) {  
    $log->log('Foo: $i', LOG_INFO);  
}  
  
$log->close();
```



# Log

- Log\_composite
- Log\_console
- Log\_display
- Log\_error\_log
- Log\_file
- Log\_mail
- Log\_mcal
- Log\_null
- Log\_sql
- Log\_sqlite
- Log\_syslog
- Log\_win





# PEAR-re épülő projektek

- Keretrendszer  
The Seagull Project  
<http://seagull.phpkitchen.com/>
- HTML\_FlexyFramework
- Sablon motor  
Savant  
<http://phpsavant.com/>
- stb.



# Részvétel a fejlesztésben

- Új csomagok fejlesztésével
- Meglévő csomagok továbbfejlesztésével
- Bug-ok keresésével, bejelentésével
- Bejelentett bug-ok felülvizsgálatával
- Dokumentációírással
- Dokumentációfordítással
- Levelezőlistákon támogatás nyújtásával



# PEPr

- PEAR Package Proposals
- <http://pear.php.net/pepr/>
- Csomag-előterjesztések
- Új csomagok indítványozása
- Draft, Proposed, Call for votes, Finished



# Bug Tracking System

- <http://pear.php.net/bugs/>
- Hibák nyomkövetése, kommentálása
- Tallózás az ismert bug-ok adatbázisában
- Felismert új bug-ok jelentése



# Támogatás

- Kézikönyv, magyar nyelven is  
<http://pear.php.net/manual/>
- Felhasználói lista:  
[pear-general@lists.php.net](mailto:pear-general@lists.php.net)
- Fejlesztői listák:  
[pear-dev](mailto:pear-dev@lists.php.net), [pear-doc](mailto:pear-doc@lists.php.net), [pear-qa](mailto:pear-qa@lists.php.net), [pear-core@lists.php.net](mailto:pear-core@lists.php.net)
- EFNet, #PEAR csatorna
- <http://pear.php.net/support.php>



# Linkek

- <http://pear.php.net>
- <http://www.phpkitchen.com> (tutorials)
- <http://norbert.mocsnik.hu>
- [norbert@mocsnik.hu](mailto:norbert@mocsnik.hu)
- Kérdések?