



PEAR

PHP Extension and Application Repository

Mocsnik Norbert

Harmadik Magyarországi PHP Konferencia

2005. március 12., Budapest



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>
```



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::uri($this->  
homepage, true);  
}
```



DB_DataObject

```
<?php
$person = DB_DataObject::factory
('person');
$person->get(12);
echo $person->name;
print_r($person);
?>
```



DB_DataObject

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

$person->eyecolor = '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 = DB_DataObject::factory
('person');
$person->get(12);
$original = $person; // clone or copy
$person->name = 'fred';
$person->update($original);
?>
```



DB_DataObject

- ```
<?php
$person = DB_DataObject::factory
('person');
$person->removed = 1;
$person->whereAdd('age > 21');
$person->update();
?>
```



# DB\_DataObject

- `dbname.links.ini`

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



# DB\_DataObject

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

    echo "{$person->name} has
{$person->eyecolor} eyes (RGB:
{$person->_eyecolor->RGB}) ";
}
?>
```




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->scale(250);
$image->addText(
    array('text' => '(c) your name'));
$image->save('./uploads/thumbs/test.jpg');
$image->free();
?>
```



Mail_Mime

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

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

```
$file = '/home/peartest/example.php';
```

```
$img = '/home/peartest/test.png';
```

```
$hdrs = array(  
    'From'      => 'you@yourdomain.com',  
    'Subject' => 'Test mime message'  
);
```



Mail_Mime

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

$mime = new Mail_mime;

$mime->setTXTBody($text);
$mime->setHTMLBody($html);
$mime->addAttachment($file, 'text/plain');
$mime->addAttachment($img);

$body = $mime->get();
$headers = $mime->headers($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';

$mail_queue =& new Mail_Queue(
    $db_options, $mail_options);

$from = 'Chief <user@server.com>';
$to = "admin <user2@server.com>";

[.. Mail_MIME $hdrs, $body ..]

$mail_queue->put(
    $from, $to, $hdrs, $body );
```



Mail_Queue

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



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>';
}
```



Auth

```
$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', 'Jon Wood');
```



Auth_PrefManager

```
require_once 'Auth/PrefManager.php';

$jonName = $prefs->getPref('john',
'fullname');

$billName = $prefs->getPref('bill', '');

echo "John neve $jonName";
echo "Bill neve $billName";
```



Log

```
require_once 'Log.php';

$console = &Log::factory('console',
    '', 'TEST');

$console->log('Logging to the
    console.');
```

```
$file = &Log::factory('file',
    'out.log', 'TEST');

$file->log('Logging to out.log');
```



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



Ajánlott csomagok

- Benchmark
- Cache_Lite
- Config
- Calendar, Date
- ~~HTML_Form~~
- HTML_TreeMenu
- HTML_QuickForm
- DBDO_FormBuilder
- PEAR_ErrorStack
- Services_Google
- DB_Pager



PEAR-re épülő projektek

- Keretrendszer
The Seagull Project
<http://seagull.phpkitchen.com/>



Támogatás

- Kézikönyv, magyar nyelven is
<http://pear.php.net/manual/hu/>
- Felhasználói lista:
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
- 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
- Kérdések?