

Какво ново в

PHP5

- PHP5 и COM
- PHP5 и XML
- PHP5 и SOAP
- PHP5 - обектен модел
- ActivePHP5
- PHP5 и Tidy

www.Бизнес

- ИТ Работа
- Персонализирани портали
- ИТ Иновации 2004

Линукс

- нова версия на Yoper
- Gentoo : интервю

Скъпи приятели,
Ето, че и новият брой е пред очите ви. Издаването му е породено от излизането на версия **5 на езика за програмиране PHP**, който ние в редакцията, а вярвам и вие ценим много. Този брой е разделен на няколко раздела:

PHP 5

Обзор на новия модел на езика, новата вградена функционалност, съвети, приложения на недокументирани функции, са само малка част от това което ви чака в този раздел. Много примери и отговори на въпроси, получени на електронната ни поща.

Линукс

Не сме забравили и любителите на отворения код и на свободния софтуер. Преглед на пакетите и възможностите на Yoper, както и интервю с един от екипа на Getntoo Линукс са изненадите ни за тях.

Бизнес

Известна ви е традицията ни да правим анализи на различни бизнес теми. Сега темата е пазара на работа в ИТ сферата, интересен поглед върху "Персонализираните портали", както и разговор с бившия министър на технологиите на Уганда, който много добре познава страната ни, освен това представяме и новите предложения за наградата ни за "ИТ Иновация"

Надяваме се да намерите в нас едно полезно четиво, което да бъде с вас и през горещото лято.

Благодарим Ви

Екипът на **Списание точка ком**

Този брой беше **компилиран** от:

Главен редактор
Богомил Шопов

Отговорник автори
Христина Георгиева

Реклама
Диян Василев

Дизайн
Мария Иванова

Използвани източници:

Използвани за примери и текстове от сайтовете на Zend, spisanie.com, официалния сайт на PHP. Всички права върху статиите принадлежат на **авторите**. Неподписаните материали са собственост на редакцията. Материалите публикувани **в оригинал** са по изричното настояване на авторите им.

Полезни адреси:

Ако желаете да **пишете** за нас: authors@spisanie.com

Ако имате въпроси относно **интернет технологии**: qanda@spisanie.com

Ако искате да споделите нещо за този брой: edition9@spisanie.com

Ако искате да си поговорим: <http://i-tourist.info/board/>



Новини от

Извора

<http://spisanie.com>

Всички новости от света на интернет технологиите

Съдържание

3 PHP 5 и SOAP

6 PHP и COM

8 ActivePHP

9 Миграция от PHP4 към PHP5 - XML

13 Нов обектен модел на PHP 5

18 Tidy и PHP 5

21 Yoper

23 Интервю с Gentoo

24 Положението на предлагането на ИТ работа у нас

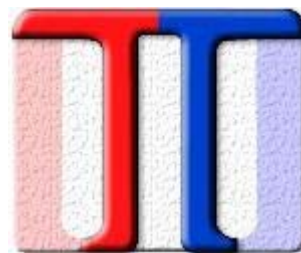
24 Дойде ли времето за персонализирани портали?

26 Награда за 'ИТ Иновации 2004'

Партньори на списанието:



Maguma



PHP Linux Бизнес

<http://spisanie.com>

PHP5 и SOAP

Да “перем” с вграждения “сапун”

Нека да си припомним първо какво беше SOAP и защо толкова много се говори, че голямото предимство на новата версия на PHP е именно вградената функционалност на този протокол в езика.

SOAP

SOAP (Simple Object Access Protocol) представлява протокол базиран на XML предназначен за обмяна на структурирана информация между приложения, на основата на вече съществуващи интернет протоколи (главно HTTP). Спецификацията на SOAP определя точно определени формати използвани от XML съобщенията и как точно трябва да се обработват те.

Уеб услуги

Не може да не стане дума и за уеб услугите, когато говорим за SOAP. Все повече и повече от главни функционалности на интернет сайтове и портали се поемат от тях като стандарт. Интересно е да се каже, че едно от най-малко популярните свойства на SOAP е била поддръжката на RPC (Remote Procedure Calls), но сега тя е една от най-често използваните.

Интеграция

До сега имплементацията на SOAP във PHP ставаше въз основа на писани допълнителни класове на PHP, които симулираха комуникация по стандарта описан за този протокол и не бяха достатъчно ефективни. За радост това вече се промени. Няма да се спирам в момента на самия протокол на обмен на данни, защото това сме го правили в предни броеве на списанието, а и доста обширна дискусия стана по време на конференцията за интернет технологии, организирана от списанието през месец Март, затова преминаваме направо към ..

Примери

За да демонстрираме създаването на елементарен SOAP-клиент, ще ползваме стандартните услуги на сайта Xmethods, разбира се, можете да ползвате друг, но за тестови цели и да разберем как работи новата функционалност в PHP, този е подходящ. Интересно е да се спомене, че когато започва да се учи език за програмиране, първия пример, който се дава е “Hello World”, а когато се започва SOAP пример това е “GetQuote” (взимане на информацията за цената на акциите на световната борса.)

Преди да се започне писането на кода, трябва да “съберем” малко информация за услугата:

Име на метода, URL на който е достъпна услугата, какво връща заявката и какво ѝ се подава.

Нека да разгледаме този пример, и това ако не сте се занимавали досега, най-вероятно е първия ви SOAP клиент. Скоро ще можете да пишете с лекота, дори и SOAP сървър.

Код 1: SOAP клиент

```
<?php

$client = new SoapClient(NULL,
array(
"location" => "http://66.28.98.121:9090/soap",
"uri" => "urn:xmethods-delayed-quotes",
"style" => SOAP_RPC,
"use" => SOAP_ENCODED
));

print($client->__call(
/* Имя SOAP-метода */
"getQuote",
//Параметри
array(
new SoapParam("ibm","symbol")),
array(
"uri" => "urn:xmethods-delayed-quotes",
"soapaction" => "urn:xmethods-delayed-quotes#getQuote"
)). "\n");
?>
```

Друг начин за “вземане” на информация е посредством WSDL файл, който описва услугата. Ето как става това:

```
<?php
$client = new
SoapClient(
"http://services.xmethods.net/soap/urn:xmethods-
delayed-quotes.wsdl");

print($client->getQuote("ibm"));
?>
```

което както забелязвате е по удачният вариант.

За да работи това обаче, е нужна малка промяна на php.ini, която се състои в следното

```
[soap]

soap.wsdl_cache_enabled = "1"
soap.wsdl_cache_dir = "/tmp"
soap.wsdl_cache_ttl = "86400"
```

Сървър

Както ви обещах по-горе, време е да направим един сървър. За целта първо е необходимо да създадем **WSDL** документ, който описва нашата уеб услуга във формат, който клиента (например втория който създадохме) ще разбере.

```
<?xml version='1.0' encoding='UTF-8' ?>
<definitions name='StockQuote'
targetNamespace='http://example.org/StockQuote'
xmlns:tns=' http://example.org/StockQuote '
xmlns:soap='http://schemas.xmlsoap.org/wsdl/soap/'
xmlns:xsd='http://www.w3.org/2001/XMLSchema'
xmlns:soapenc='http://schemas.xmlsoap.org/soap/encoding/'
xmlns:wsdl='http://schemas.xmlsoap.org/wsdl/'
xmlns='http://schemas.xmlsoap.org/wsdl/'>

<message name='getQuoteRequest'>
<part name='symbol' type='xsd:string'/>
</message>
<message name='getQuoteResponse'>
<part name='Result' type='xsd:float'/>
</message>

<portType name='StockQuotePortType'>
<operation name='getQuote'>
<input message='tns:getQuoteRequest'/>
<output message='tns:getQuoteResponse'/>
</operation>
</portType>

<binding name='StockQuoteBinding'
type='tns:StockQuotePortType'>
<soap:binding style='rpc'
transport='http://schemas.xmlsoap.org/soap/http/'>
<operation name='getQuote'>
<soap:operation soapAction='urn:xmethods-delayed-quotes#getQuote'/>
<input>
<soap:body use='encoded' namespace='urn:xmethods-delayed-quotes'
encodingStyle='http://schemas.xmlsoap.org/soap/encoding/'/>
</input>
<output>
<soap:body use='encoded' namespace='urn:xmethods-delayed-quotes'
encodingStyle='http://schemas.xmlsoap.org/soap/encoding/'/>
</output>
</operation>
</binding>

<service name='StockQuoteService'>
<port name='StockQuotePort' binding='StockQuoteBinding'>
<soap:address location='http://[insert real path here]/stockquote1.php'/>
</port>
</service>
</definitions>
```

```
<?php
$quotes = array(
"ibm" => 98.42
);

function getQuote($symbol) {
global $quotes;
return $quotes[$symbol];
}

ini_set("soap.wsdl_cache_enabled", "0"); // отключаем
кэширование WSDL
$server = new SoapServer("stockquote1.wsdl");
$server->addFunction("getQuote");
$server->handle();
?>
```

Като файла, който създадохме по-горе се казва stockquote1.wsdl.

Ето още недин пример, за това как може да се структурира един сървър например, който използва повече от новата функционалност на езика

```
<?php
class QuoteService {
private $quotes = array("ibm" => 98.42);

function getQuote($symbol) {
if (isset($this->quotes[$symbol])) {
return $this->quotes[$symbol];
} else {
throw new SoapFault("Server","Unknown Symbol '$symbol'.");
}
}
}

$server = new SoapServer("stockquote2.wsdl");
$server->setClass("QuoteService");
$server->handle();
?>
```

Това е статична информация, можете разбира се да вземете данните от база данни за да имате по-голяма гъвкавост по отношение на верността на информацията, но това вече не е в техническата част. Може би ви е интересно какво точно представлява комуникацията на горния клиент и сървър погледнато през SOAP.

Това го слагаме на сървъра при нас и правим вече самия SOAP сървър.

Заявка:

```
<?xml version="1.0" encoding="UTF-8" ?>
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/
envelope/"
xmlns:ns1="urn:xmethods-delayed-quotes"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/
encoding/"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/
soap/encoding/">
<SOAP-ENV:Body>
<ns1:getQuote>
<symbol xsi:type="xsd:string">ibm</symbol>
</ns1:getQuote>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Отговор:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="urn:xmethods-delayed-quotes"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/
encoding/">
<SOAP-ENV:Body>
<ns1:getQuoteResponse>
<Result xsi:type="xsd:float">98.42</Result>
</ns1:getQuoteResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Заклучение

SOAP е с много възможности, и то с почти неограничени. Използването му в много сайтове и проекти е само доказателство за това.

Научавайки да работите с него, получавате едно силно оръжие и само въображението ви може да ви води към създаването на все по нови услуги и приложения.

<< Новините в аванс



PHP Състезание, етап 2

От днес стартира втория етап на състезанието по PHP. Той се състои в следното.

- 1) Задава се нова задача с по-лека трудност.
- 2) Всеки участвал в първи етап, може да участва и във втори. По този начин има двойно по-голям шанс да спечели.
- 3) Приемат се решения и на двете задачи.

Задачата за етап 2 е следната:

Изработка на уеб спайдър

- Скриптът да взема първите 5 (пет) линка от дадена страница.
- За всяка страница от първите 5 да взема по още 3 (три) линка.
- За всеки линк да записва следните данни, адрес, заглавие на страницата, мета-таговете: description, keywords и content-type.
- Не се допуска записването на едни и същи линкове.

От екипът на списанието решихме да поощрим някой от състезателите изпратили до тук верен и добре написан код. Те са:

Кирил Митев
Иван Деянов Давидов
Атанас Георгиев Митрев

Те получават **Template Tamer** (Developer edition)

Blog писан на Java: Pebble

Понякога е много трудно да се избере свестен софтуер за "блогване". Едно решение за това базирано на Java излезе с версия 1.5-бета 2 само преди няколко дни. Pebble е със стандартен интерфейс както повечето подобни софтуерни пакети. Поддържа възможност за добавяне на блокове, коментари към тях, статистика, RSS съдържание, снимки и много други. Изискванията са минимум J2SE 1.4 и сървър за уеб приложения поддържащи JSP 1.2 и Servlet 2.3, например Tomcat или Resin.

Повече за Pebble можете да научите от тук: <http://www.simongbrown.com/pebble.html>

Файлови шеринги с PHP

Преди броени дни беше отворен код на бот за търсене по файлови шеринги в мрежата на soulreactor, можете да видите как работи тя <http://soulreactor.com/>

Сорс-кода, можете да изтеглите от <http://soulreactor.com/source/>

Ценното в случая е протокола за търсене, който се намира в файла protocol.php.

PHP и COM

Предимството да имаш Windows

VB

С излязлата вече версия 5 на популярния език за програмиране все повече се появяват в интернетпространството предимствата на тази нова версия пред старата. Разбира се, никой още и не мисли да сменя начина си на работа, поне докато не минат 5-6 месеца за доказване на предимствата на езика.

Връзката на PHP с WMI, обаче дава голяма преднина на windows базираната версия на езика, а заедно с пуснатия от израелската компания програмен продукт, който оптимизира езика за работа на популярната операционна система, разликите между нея и все повече набиращата популярност операционна среда Linux, стават много малки.

WMI

С помощта на WMI (Windows Management Instrumentation) можете да променяте системните настройки на вашата операционна система. Пълнен контрол върху регистъра на Windows би трябвало да звучи много сериозно и това е така. Достъпа до него се осъществява посредством COM обекти, които във версия 5 на PHP са построени на ново, като са избегнати проблемите свързани със сигурността. Ето и пример за приложението (заменете hostname с името на машината ви):

```
$wmi = new COM("WinMgmts:{impersonationLevel=impersonate}
//hostname/root/cimv2")
$drives = $wmi->ExecQuery("Select * from Win32_LogicalDisk");
foreach ($drives as $drive)
{
    if ($drive->MediaType == 12)
    echo "{$drive->Name} {$drive->VolumeName} {$drive-
    >FileSystem} ".FormatBytes($drive->Size)." ".FormatBytes($drive-
    >FreeSpace)."\n";
}
```

Друг по интересен пример може би е следния, който е базиран на примерите за WMI на <http://www.activexperts.com/activmonitor/windowsmanagement/wmi/samples/>. Това ще ви покаже как ва интерпретиратескриптове писани на VB и да ги модифицирате на PHP.

Ето и примера

PHP

```
<?
$wmi = new COM("WinMgmts:{impersonationLevel=impersonate}
//hostname/root/cimv2");
$floppys = $wmi->ExecQuery("Select * from Win32_FloppyDrive");

foreach ($floppys as $floppy)
{
    echo "{$floppy ->Availability} {$drive-> Capabilities }
    {$drive-> Caption} ";
}
?>
```

```
On Error Resume Next
strComputer = "."
Set objWMIService = GetObject("winmgmts:\\." &
strComputer & "\root\cimv2")
Set collItems = objWMIService.ExecQuery("Select * from
Win32_FloppyDrive",,48)
For Each objItem in collItems
Wscript.Echo "Availability: " & objItem.Availability
Wscript.Echo "Capabilities: " & objItem.Capabilities
Wscript.Echo "CapabilityDescriptions: " &
objItem.CapabilityDescriptions
Wscript.Echo "Caption: " & objItem.Caption
Wscript.Echo "CompressionMethod: " &
objItem.CompressionMethod
Wscript.Echo "ConfigManagerErrorCode: " &
objItem.ConfigManagerErrorCode
Wscript.Echo "ConfigManagerUserConfig: " &
objItem.ConfigManagerUserConfig
Wscript.Echo "CreationClassName: " &
objItem.CreationClassName
Wscript.Echo "DefaultBlockSize: " &
objItem.DefaultBlockSize
Wscript.Echo "Description: " & objItem.Description
Wscript.Echo "DeviceID: " & objItem.DeviceID
Wscript.Echo "ErrorCleared: " & objItem.ErrorCleared
Wscript.Echo "ErrorDescription: " &
objItem.ErrorDescription
Wscript.Echo "ErrorMethodology: " &
objItem.ErrorMethodology
Wscript.Echo "InstallDate: " & objItem.InstallDate
Wscript.Echo "LastErrorCode: " & objItem.LastErrorCode
Wscript.Echo "Manufacturer: " & objItem.Manufacturer
Wscript.Echo "MaxBlockSize: " & objItem.MaxBlockSize
Wscript.Echo "MaxMediaSize: " & objItem.MaxMediaSize
Wscript.Echo "MinBlockSize: " & objItem.MinBlockSize
Wscript.Echo "Name: " & objItem.Name
Wscript.Echo "NeedsCleaning: " &
objItem.NeedsCleaning
Wscript.Echo "NumberOfMediaSupported: " &
objItem.NumberOfMediaSupported
Wscript.Echo "PNPDeviceID: " & objItem.PNPDeviceID
Wscript.Echo "PowerManagementCapabilities: " &
objItem.PowerManagementCapabilities
Wscript.Echo "PowerManagementSupported: " &
objItem.PowerManagementSupported
Wscript.Echo "Status: " & objItem.Status
Wscript.Echo "StatusInfo: " & objItem.StatusInfo
Wscript.Echo "SystemCreationClassName: " &
objItem.SystemCreationClassName
Wscript.Echo "SystemName: " & objItem.SystemName
Next
```

Логиката, която следва PHP е същата както на другия език, що се отнася до работата с COM обекти.

“Старият” COM

Това е новото приложение на тези обекти, нека да погледнем малко и старото, което сега му се дава нова възможност за изява. Защо казвам това? След Windows NT, започнаха проблемите със сигурността на достъпа на тези обекти от PHP. Ако си спомняте в един от предните броеве ви показахме как да видите съдържанието на пощенската си кутия, която се обслужва през Outlook, както и да пратите писмо през акаунта си. Това беше невъзможно до преди за излезе PHP 5, защото както знаете сега вече има много подобрения и едното е проблема със сигурността, единственото условие е да бъдете администратор на машината която ползвате и можете да запретнете ръкави и да напишете следния код, който конвертира doc файлове.

```
<?
$word = new COM("word.application") or die("Unable to
instantiate Word");
print "Loaded Word, version {$word->Version}\n";

$word->Visible = 1;
$word->ActivePrinter = "Acrobat Distiller";
$word->Documents->Open($input);

$word->ActiveDocument->PrintOut(0, 0, 0, $output);
$word->Quit();
$word->Release();
$word = null;

?>
```

За целта обаче имате нужда и от Acrobat Distiller.

Друг интересен пример и как да изпратите писмо директно през **Outlook-a** ви:

```
com_load_typelib('Outlook.Application');
$objApp = new COM("Outlook.Application");
$myItem = $objApp->CreateItem(olMailItem);
$a=$myItem->Recipients->Add("your adress here");
$myItem->Subject="Subject";
$myItem->Body="This is a Body Section now.....!";
$myItem->Display();
$myItem->Send();
```

/Мерси на Мах за допълнението/

За съжаление документация за всеки COM обект не може да се намери във вид достъпен на PHP, но това не прези да се види пример писан на VB и от там да се направят съответните изводи, както направихме и по –горе.

COM е обширна област и навлизането в нея може да доведе до много проблеми със сигурността на приложенията, но пък резултатите могат да са впечатляващи.

Например ето този скрипт на Zend, който показва как могат да се прихващат събития :

```
<?php
class IESink {
public $terminated = false;
public function OnQuit() {
$this->terminated = true;
}
}

$ie = new COM("InternetExplorer.Application");
$ie->Visible = true;
$ie->Navigate("http://www.php.net/");

$sink = new IESink;
com_event_sink($ie, $sink, "DWebBrowserEvents2");
while (!$sink->terminated) {
com_message_pump(4000);
}

print "finished!\n";
?>
```

С помощта на COM можете да имате достъп до .net байнърита и да работите с елементите на платформата:

```
<?php
$stack = new DOTNET("mscorlib",
"System.Collections.Stack");

$stack->Push(".Net");
$stack->Push("Hello ");

echo $stack->Pop() . $stack->Pop();
?>
```

Всичко зависи от целите, които си поставите и изберете начина по който да се изпълнят те. COM е едно решение, но не задължително такова.

PHP от страната на клиента

От 1 година вече има PHP библиотека, която дава възможност да се работи с езика от страната на клиента. За сега това е само в интернетексплорер и е по-скоро тестово отколкото да върши работа, но нека да разгледаме няколко примера.

Настройка:

Важно е да се отбележи, че работи библиотеката CAMO във версия пет, преди няколко месеца се провах да работя с нея, като си я дръпнах от CVS-а но уви ефекта беше никакъв.

Стартирайте конзола и напишете следното вътре:

```
C:\>cd php5
C:\PHP5>regsvr32 php5activescript.dll
```

При регистрацията на библиотеката във операционната среда се търси конфигурационния файл намиращ се в system 32 (system) директорията. След това отворете редактор и пишете:

```
<html>
<head>
<title>Тест </title>
</head>
<body>
<script language="ActivePHP5">

$document->write('<h1>Output generated by '. phpversion().
' [' .php_sapi_name().']</h1>');

$document->write('<p>Available objects</p>');

foreach ( array_keys($GLOBALS) as $global_var ) {
if ( is_object($GLOBALS[$global_var]) ) {
$document->write($global_var."<br>");
}
}

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

Запишете го като HTML и стартирайте. Тук не е нужно да имате сървър, скрипта е на същия принцип като JavaScript. За съжаление няма никаква документация за тази библиотека, както и за ползването ѝ, но това не бива да ни спира да експериментираме. Има проект за PHP, което работи на Mozilla, пак като скрипт от страната на клиента. Така, че в това отношение има много проекти, които неминуемо ще доведат до резултат, който ще е полезен за всички.

Ето как се създават функции в ActiveScript

```
<html>
<head>
<title>Тест </title>
</head>
<body>

<script language="ActivePHP5">
function clickit() {
$GLOBALS["window"]->open("http://www.php.net");
}
</script>
<img onclick="clickit();">
</body>
</html>
```

Ето и още един интересен пример, който засега не работи, но автора му се надява да може да има връзка между елементите, които ползва скоро.

```
<HTML>
<HEAD>
<HTA:APPLICATION ID="oHTA"
APPLICATIONNAME="RSS Evaluator"
BORDER="thin"
BORDERSTYLE="normal"
CAPTION="yes"
ICON=""
MAXIMIZEBUTTON="yes"
MINIMIZEBUTTON="yes"
SHOWINTASKBAR="yes"
SINGLEINSTANCE="yes"
SYSMENU="yes"
VERSION="3.5"
WINDOWSTATE="normal"
/>
<script language="ActivePHP5">
function write_file(){
$filename = 'test.txt';
$somecontent = print_r($GLOBALS,true);
if (!$handle = fopen($filename, 'w')) {
print "Cannot open file ($filename)";
exit;
}
if (!fwrite($handle, $somecontent)) {
print "Cannot write to file ($filename)";
exit;
}
print "Success, wrote ($somecontent) to file ($filename)";
fclose($handle);
return "Success";
}
</script>
</HEAD>
<BODY bgcolor="menu" scroll="no">
<button onclick="write_file();return false;">Function Test</a>
</BODY>
```

С помощта на ActivePHP, може да се работи и с WindowsScriptingHost, но коментарите следват. Ако вие откриете някакво приложение на този език, пишете на [редакционният email](#).

Лесна миграция на XML от PHP 4 към PHP 5

Мост между версиите

Александър Алпети

Докато сърфирах из интернет попаднах на много добър пример как може лесно да се мигрира едно приложение ползващо XML от PHP 4 към революционната версия 5 на езика за програмиране. Този пример е разработен от Александър Алпети и логиката е следната.

В PHP 4 XML използва функциите на DOM за работа с XML. За припомняне нека да дадем следния пример:

```
<?php
if ($dom=domxml_open_file('test.xml'))
{
    $root=$dom->document_element();
    $bs=$root->get_elements_by_tagname('body');
    if (count($bs)==1)
    {
        $body=$bs[0];
        $p=$dom->create_element('p');
        $p->append_child($dom->create_text_node('a new paragraph'));
        $body->append_child($p);
        $body->append_child($dom->create_text_node("\n"));
    }
    $dom->dump_file('test.xml',false,false);
}
?>
```

За да мигрира лесно едно приложение писано на старата версия на езика, има нужда от функционалност която да е мост между двете версии, а това става именно с този файл.

Какво да направим първо?

Във всеки файл използващ гореспоменатите функции да проверяваме коя версия на езика има машината изпълняваща кода и в зависимост от резултата да ползва или вградените функции или през моста който ще напишем сега новите такива. Ето го и кода който трябва да напишем

```
<?
if (version_compare(PHP_VERSION,'5','>='))
require_once('dom_xml-php4-to-php5.php');
.....
?>
```

А ето и кода, осигуряващ моста между двете версии. При подробен анализ, може да се схване истинската същност на кода. Отворете редактора на код и запишете:

```
class php4DOMAttr extends php4DOMNode
{
    var $myDOMAttr;
    function php4DOMAttr($aDOMAttr) {$this->myDOMAttr=$aDOMAttr;}
    function Name() {return $this->myDOMAttr->name;}
    function Specified() {return $this->myDOMAttr->specified;}
    function Value() {return $this->myDOMAttr->value;}
}
```

```

class php4DOMCDATASection extends php4DOMNode
{
var $myDOMCDATASection;
function php4DOMCDATASection($aDOMCDATASection) {$this->myDOMCDATASection=$aDOMCDATASection;}
}

class php4DOMDocument
{
var $myDOMDocument;
function php4DOMDocument($filename)
{
$this->myDOMDocument=new DOMDocument();
$this->myDOMDocument->load($filename);
}
function create_attribute($name,$value)
{
$myAttr=$this->myDOMDocument->createAttribute($name);
$myAttr->value=$value;
return new php4DOMAttr($myAttr);
}
function create_cdata_section ($content) {return new php4DOMCDATASection($this->myDOMDocument->createCDATASection($content));}
function create_comment($data) {return new php4DOMElement($this->myDOMDocument->createComment($data));}
function create_element($name) {return new php4DOMElement($this->myDOMDocument->createElement($name));}
function create_text_node($content) {return new php4DOMNode($this->myDOMDocument->createTextNode($content));}
function document_element() {return new php4DOMElement($this->myDOMDocument->documentElement);}
function dump_file($filename,$compressionmode=false,$format=false) {return $this->myDOMDocument->save($filename);}
function dump_mem($format=false,$encoding=false) {return $this->myDOMDocument->saveXML();}
function get_elements_by_tagname($name)
{
$myDOMNodeList=$this->myDOMDocument->getElementsByTagName($name);
$nodeSet=array();
$i=0;
while ($node=$myDOMNodeList->item($i))
{
$nodeSet[]=new php4DOMElement($node);
$i++;
}
return $nodeSet;
}
function html_dump_mem() {return $this->myDOMDocument->saveHTML();}
}

class php4DOMElement extends php4DOMNode
{
function get_attribute($name) {return $this->myDOMNode->getAttribute($name);}
function get_elements_by_tagname($name)
{
$myDOMNodeList=$this->myDOMNode->getElementsByTagName($name);
$nodeSet=array();
$i=0;
while ($node=$myDOMNodeList->item($i))
{
$nodeSet[]=new php4DOMElement($node);
$i++;
}
return $nodeSet;
}
function has_attribute($name) {return $this->myDOMNode->hasAttribute($name);}
function remove_attribute($name) {return $this->myDOMNode->removeAttribute($name);}
function set_attribute($name,$value) {return $this->myDOMNode->setAttribute($name,$value);}
function tagname($name) {return $this->myDOMNode->tagName;}
}

```

```

class php4DOMNode
{
var $myDOMNode;
function php4DOMNode($aDOMNode) {$this->myDOMNode=$aDOMNode;}
function append_child($newnode) {return new php4DOMElement($this->myDOMNode->appendChild($newnode->myDOMNode));}
function append_sibling($newnode) {return new php4DOMElement($this->myDOMNode->parentNode->appendChild($newnode->myDOMNode));}
function attributes()
{
$myDOMNodeList=$this->myDOMNode->attributes;
$nodeSet=array();
$i=0;
while ($node=$myDOMNodeList->item($i))
{
$nodeSet[]=new php4DOMAttr($node);
$i++;
}
return $nodeSet;
}
function child_nodes()
{
$myDOMNodeList=$this->myDOMNode->childNodes;
$nodeSet=array();
$i=0;
while ($node=$myDOMNodeList->item($i))
{
$nodeSet[]=new php4DOMElement($node);
$i++;
}
return $nodeSet;
}
function clone_node($deep=false) {return new php4DOMElement($this->myDOMNode->cloneNode($deep));}
function first_child() {return new php4DOMElement($this->myDOMNode->firstChild);}
function get_content() {return $this->myDOMNode->textContent;}
function has_attributes() {return $this->myDOMNode->hasAttributes();}
function has_child_nodes() {return $this->myDOMNode->hasChildNodes();}
function insert_before($newnode,$refnode) {return new php4DOMElement($this->myDOMNode->insertBefore($newnode->myDOMNode,$refnode->myDOMNode));}
function last_child() {return new php4DOMElement($this->myDOMNode->lastChild);}
function next_sibling() {return new php4DOMElement($this->myDOMNode->nextSibling);}
function node_name() {return $this->myDOMNode->nodeName;}
function node_type() {return $this->myDOMNode->nodeType;}
function node_value() {return $this->myDOMNode->nodeValue;}
function parent_node() {return new php4DOMElement($this->myDOMNode->parentNode);}
function previous_sibling() {return new php4DOMElement($this->myDOMNode->previousSibling);}
function remove_child($oldchild) {return new php4DOMElement($this->myDOMNode->removeChild($oldchild->myDOMNode));}
function replace_child($oldnode,$newnode) {return new php4DOMElement($this->myDOMNode->replaceChild($oldchild->myDOMNode,$newnode->myDOMNode));}
function set_content($text) {return $this->myDOMNode->textContent=$text;}
}

```

```

class php4DOMNodelist
{
var $myDOMNodelist;
var $nodeset;
function php4DOMNodelist($aDOMNodelist)
{
$this->myDOMNodelist=$aDOMNodelist;
$this->nodeset=array();
$i=0;
while ($node=$this->myDOMNodelist->item($i))
{
$this->nodeset[]=new php4DOMELEMENT($node);
$i++;
}
}
}

class php4DOMXPath
{
var $myDOMXPath;
function php4DOMXPath($dom_document) {$this->myDOMXPath=new DOMXPath($dom_document->myDOMDocument);}
function query($eval_str) {return new php4DOMNodelist($this->myDOMXPath->query($eval_str));}
function xpath_register_ns($prefix,$namespaceURI) {return $this->myDOMXPath->registerNamespace($prefix,$namespaceURI);}
}
?>

```

Това може би е по добрият начин за бърза миграция, ако се наложи. Но все пак не бива да забравяме, че ако искаме приложенията да работят на новата версия на езика, трябва да се съобразим с изискванията на новия Zend Engine и да работим по определените нови функции и класове.

<< Новините в аванс



Нов език за програмиране: Netorque.

Поставени са основите на нов език за програмиране. Той се нарича Netorque и може да бъде видян на сайта : <http://netorque.com/index.html>. Според създателя, езикът е много

лесен за научаване, разполага с прост синтаксис, вградени Perl regex, база данни и засега работи само като FastCGI демон.

Ето и няколко примерни реда от новия език.

```

for usr in user[*]
some code;
some other code;
end;
object obj1
attr1 = 15;
attr2 = "Hello";
attr3 = 0.0;
method(par1, par2)
return par1+par2;
end;
end;

```

Работа с масиви

```

s = set[1, 5, 4, 4, 4]; /* only 1, 5 and 4 will be contained */
b = bag[1, 5, 4, 4, 4]; /* all elements are contained */
os = s.asoset; /* os will contain the sorted elements in s */

```

```

object message
owner: user
if obj.name = "admin"
throw "Administrator not allowed to own a message";
end;
end;
replies: message*
if obj.owner.name = "Peter"
throw "User peter not allowed to respond to this message";
end;
obj.parent = self;
end;
end;

```

Работа с обекти

```

o = new obj1;
o.attr1 += 5;
attr3 += new user;
attr3 -= user[name="admin"];
delete obj1;

```

switch

```

switch str
case "Hello": `Hello, case 1`;
case "world": `Hello, case 2`; break;
default: `Default state`; break;
end;

```


PHP5 - обектен модел

Във версия 5, след много чакане ставаме свидетели на обектно ориентиран модел, който и в основата на всеки език за програмиране. За съжаление до сега на програмистите на PHP им се налагаше да пишат процедурен код и да симулират отношения, който наподобяват обектно ориентираните.

Нека преди да започна да дам един пример с това как е било преди и как ще бъде от сега нататък.

Преди

```
<?php

print "Hello, world.";

?>
```

Сега

```
<?php

class helloWorld {
function myPrint() {
print "Hello, world.";
}
}

$myHelloWorld = new helloWorld();
$myHelloWorld->myPrint();

?>
```

Нека да се запознаем и с нововъведенията, които предлага ZEND Engine, на която основа е “построен” PHP5.

Те са много и разбира се няма да има място в този материал да се запознаем с всички, но ще започнем от най-важните, който променят разбирането за програмиране с този език.

Класове

Кратка дефиниция : Клас в общи линии е сбор от свързани променливи и методи.

Преди да излезе PHP5, също се налагаше да се ползват класове, но за да ги създадете, трябваше да се дефинира функция със същото име като класа, за да може да се инициализира всичко, когато обекта се създаде, или с няколко реда ситуацията е следната:

```
<?php
class Foo
{
var $variable;
function Foo()
{
$this->variable = 'Някаква стойност';
}
}
echo 'Създаваме Foo със стария конструктор на Foo()';
$f = new Foo();
?>
```

Решението, което дадохме в горния код работи, токато не се наложи да смените името на Foo или да имате друг родителски клас на Foo, тогава че се наложи да промените името навсякъде където сте го използвали. В новата версия това е решено, като при декларирането на клас, стандартния начин това да се направи е с извикването на метода **__construct()**:

```
<?php
class Foo
{
var $variable;
/**
 * Constructors now have the common __construct()
 * function.
 */
function __construct()
{
$this->variable = 'Някаква стойност';
}
}
echo 'създаваме Foo с нов __construct()';
$f = new Foo();
?>
```

Много добра възможност е когато класа вече няма да ви трябва, да го унищожите от паметта, това става с метода **__destruct()**

```
<?php
/**
 * Showing the use of a constructor and
 * destructor.
 */
class Foo
{
function __construct()
{
print 'Cera създаваме Foo';
}
function __destruct()
{
print 'Cera унищожаваме обекта Foo';
}
}
$foo = new Foo();
?>
```

Друга интересна функция е `__clone`, която клонира обекта, който създаваме.

Друг интересен пример, да се покаже това е да създадем малко приложение, което да записва във файл резултата от действието си.

```
<?php
/**
 * Class to write to a file.
 *
 * For illustration purposes this does not have a
 * open file pointer.
 */
class FileWriter
{
function __construct()
{
// Отваря файл
}

function __destruct()
{
// Затваря файла
echo '<br />Затварям файла сега.';
}
function write( $str )
{
echo '<br />Ето това ще запишем във файла ' . $str;
}
}
class Logger
{
private $doDebug;

function __construct( FileWriter $writer )
{
$this->doDebug = false;
$this->writer = $writer;
}

function __destruct()
{
$s = 'Application finished at : ' . date( "Y/m/d G:i:s" );
$this->writer->write( $s );
if( $this->doDebug )
{
echo '<br />';
echo 'Debug: ' . $s;
}
}

function doDebug()
{
$this->doDebug = true;
}
}
$writer = new FileWriter();
$logger = new Logger( $writer );
$logger->doDebug();echo 'Executing application';
?>
```

```
<?php
/**
 * Демонстрира клониране на функция
 */

class Foo
{
var $variable;
function __construct()
{
$this->variable = 'конирана променлива';
}
}
echo 'Hi';
$foo = new Foo();
$bar = $foo->__clone();
echo $bar->variable;
?>
```

Разбира се, винаги можете да предефинирате функцията:

```
<?php
/**
 * Demonstrates the use of a custom
 * __clone() function
 */
class Foo
{
var $id;
var $name;
function __construct()
{
print 'Now constructing the Foo object<br />';
$this->id = 1;
$this->name = 'PHP5';
}
function __clone()
{
// Note the use of 'that'
$this->id = $that->id++;
$this->name = $that->name;
}
}
$foo = new Foo();

$bar = $foo->__clone();
echo 'Name : ' . $bar->_name . ' Clone number : ' . $bar->id . '<br />';

$bar2 = $foo->__clone();
echo 'Name : ' . $bar2->_name . ' Clone number : ' . $bar2->id . '<br />';

?>
```

Абстрактни класове

Това може би нищо не говори на хората, който са се занимавали само с PHP програмиране, но ако сте опитвали вкуса на JAVA, знаете за какво става дума.

Ето и един бърз пример за обяснение на какво е това абстрактен клас.

```
<?php
/**
 * Any class with a abstract method in it
 * has to be declared abstract.
 */
abstract class Foo
{
    abstract function toString();
    abstract function version();
}
?>
```

Ако сега наследите този клас и забравите да имплементирате, някоя от функциите ето така:

```
<?php
/**
 * The abstract class we are going to
 * extend.
 */
abstract class Foo
{
    abstract function toString();
    abstract function version();
}

/**
 * The class is not implementing all
 * the abstract functions from Foo, and
 * should therefor be declared abstract itself.
 */
class Bar extends Foo
{
    function toString()
    {
        return 'Class:Bar';
    }
}
$bar = new Bar();
?>
```

Ще се появи следната грешка:

Fatal error: Class bar contains 1 abstract methods and must therefore be declared abstract (foo::version)

С абстрактните класове, можем да сме сигурни, че класът, трябва да бъде направен на подклас, за да бъде използван. Не е позволено да се прави инстанция на абстрактен клас. Тоест, Не можете да направите ето така:

```
<?php
/**
 * You cannot create a object from
 * a abstract class!
 */
abstract class Foo
{
    abstract function toString();
    abstract function version();
}
$f = new Foo();
?>
```

Това ще предизвика следната грешка:

Fatal error: Cannot instantiate abstract class foo

Може би едно от най-интересните неща в сферата на нововъведенията е възможността за използване на интерфейси.

Интерфейси

Определение: Интерфейс, представлява взаимодействие между някакъв интерфейс и имплементираш клас, който трябва да имплементира функция дефинирана в интерфейса. Може би е малко сложно, но в кода по-долу, ще имате възможност да видите на практика определението.

```
<?php
/**
 * A interface is a contract that
 * the implementing class has to
 * create its functions.
 */
interface Foo
{
    function price();
}
class Bar implements Foo
{
    function price()
    {
        return 99;
    }
}
$b = new Bar();
?>
```

Всяка функция декларирана в интерфейса, автоматично става абстрактна, така че ако в класът, не дефинирате тази функция ще видите следната грешка:

Fatal error: Class bar contains abstract methods and must be declared abstract.

С абстрактните класове, вие можете да екстендвате само по един клас, но с интерфейса, няма ограничение като това.

```
<?php
/**
 * Demonstrates that multiple inheritance
 * is now allowed.
 */
interface Foo
{
function price();
}
interface Bar
{
function name();
}

/**
 * You can also extend this with a third class.
 */
class FooBar implements Foo, Bar
{
function price()
{
return 99;
}
function name()
{
return 'NuCleuZ';
}
}

$b = new FooBar();
echo $b->name();
?>
```

Ето и друг пример за това

```
<?php
/**
 * A User interface that defines two functions
 * that we need for the 'printUser()' function
 */
interface User
{
function name();
function level();
}

class Author implements User
{
private $name;
private $level;
function __construct($name)
{
$this->name = $name;
$this->level = 10;
}
function name()
{
return $this->name;
}
function level()
{
return $this->level;
}
}
```

```
/**
 * A reader doesn't have a level
 */
class Reader implements User
{
private $name;
private $level;
function __construct($name)
{
$this->name = $name;
$this->level = 0;
}
function name()
{
return $this->name;
}
function level()
{
return $this->level;
}
}

// Create two User objects:
$me = new Author('NuCleuZ');
$you = new Reader('Anonymous');

// Now print their name:
printUser($me);
printUser($you);
```

```
/**
 * This functions requires a User object
 * to be passed in as parameter:
 */
function printUser(User $u)
{
echo '<li>' . $u->name() . ', Level : ' . $u->level() . '</li>';
}
?>
```

Когато обекта имплементира интерфейс, то интерфейса също става собственик на родителския клас:

```
<?php
interface Foo
{
function a();
}

class Bar
{
function b()
{
}
}
```

```

/**
 * Class FooBar does now have 3 types:
 * FooBar, Bar and Foo
 */
class FooBar extends Bar implements Foo
{

function __construct()
{
echo 'Constructing FooBar<br />';
}
function a()
{
// Do nothing for the moment
}
}

// All the following are now evaluated to true:
$f = new FooBar();
if($f instanceof Bar)
{
echo '$f is a Bar object<br />';
}

if($f instanceof Foo)
{
echo '$f is a Foo object<br />';
}

if($f instanceof FooBar)
{
echo '$f is a FooBar object<br />';
}

// You can also pass the $f object to 3 different functions that
// take specific objects as parameters:
passInFoo( $f );
function passInFoo( Foo $f )
{
echo '$f passed in as a Foo object<br />';
}

passInBar( $f );
function passInBar( Bar $f )
{
echo '$f passed in as a Bar object<br />';
}

passInFooBar( $f );
function passInFooBar( FooBar $f )
{
echo '$f passed in as a FooBar object<br />';
}
?>

```

```

<?php
/**
 * Custom class that will catch a
 * NumberFormatException:
 */
class NumberFormatException
{
function __construct($exception)
{
$this->exception = $exception;
}
}
/**
 * All exceptions has a toString() function:
 */
function toString()
{
return 'A NumberFormatException occurred : ' . $this->exception;
}
}
$i = 'Hello!';
// A try block:
try
{
/**
 * If the $i variable isn't a in we
 * throw a new NumberFormatException:
 */
if(!is_int($i))
{
throw new NumberFormatException('$i is wrong!!');
}
else
{
echo 'No exception caught!';
}
}
}
/**
 * Finally catch any errors that occurred within the try-catch
 * block
 */
catch (NumberFormatException $exception)
{
echo $exception->toString();
}
?>

```

Докато `$i` не е от числов тип, блока `catch` ще се изпълнява. Това прави възможно да прихванете тази грешка и да я обработите както желаете. Разбира се, има още много примери, които само ще намерите, както и много нова функционалност. Ние ви даваме насоката, а пътя ще намерите сами.

Използвани за материали от [phpvolcano](http://phpvolcano.com). Материалът е част от обширната статия за PHP 5 публикувана в PCMagazine.

Малко ще се отплеснем и ще преминем на една конструкция, която беше крайно време да се появи и в този език. Става дума за

Try Catch

С помощта ѝ, вече може да се прихважат грешки и да се подават към браузера по различен начин, логиката е следната

Tidy и PHP5 - Нека бъде чист код!

Ивайло Гроздев

1. Въведение

Да, PHP5 вече е факт. Хубава седмица. Всичко сега е фокусирано около новите възможности на езика, добавянето на дългоочаквани нови функции, превръщането на слабите му страни в силни. И по моето мнение, целият шум е оправдан - стъпката напред е наистина голяма. Новият обектен модел е номер Едно, поддръжката за XML е вдигната на едно съвсем друго ниво, вече си имаме MySQLi и SQLite, можем да си обработваме и изключения. Аз обаче избрах да ви обърна внимание на една от не толкова "одумваните" новости в PHP5 - а именно разширението Tidy, служещо за нещо доста полезно - да пази кода на страниците ви чист.

Като човек, който по принцип държи на чистия изходен код и на който уеб стандартите са известна слабост, няма как Tidy да не ми е симпатичен. Омръзнало ми е да гледам разбъркани тагове, ненужен код, липса на каквото и да е покритие на основните стандарти - гледка, която е доста често срещана, особено когато става въпрос за някои редактори, генериращи HTML. С Tidy в PHP5 вече имаме инструмент, който да редуцира "мръсния" изходен код до минимум.

2. Що е Tidy

HTML Tidy е безплатен open source инструмент за почистване и поправяне на HTML, XHTML и XML документи. Проектът е стартиран от Dave Raggett, но в момента поддръжката е прехвърлена на група разработчици, чиято страница може да видите на <http://tidy.sourceforge.net/>. Самата програма служи за отстраняване на често срещаните грешки в маркър кода на страниците, като може да бъде използвана и като инструмент за конвертиране на HTML в XHTML, който изцяло спазва правилата на добре оформения XML. Досега HTML Tidy е успешно интегриран в множество софтуерни продукти, които продуцират HTML код. Достъпни са версии за различни платформи и на различни езици.

Създадена е и библиотеката TidyLib, чиято основна цел е полесната съвместна работа на програмата с други продукти. Какво може да прави Tidy? Tidy открива и поправя неправилни или липсващи затварящи тагове, преподрежда за атрибутите или "/" в затварящите елементи, добавя липсващи елементи при списъци или таблици, грижи се за добрия изглед и четимостта на кода. Може да създаде изцяло покриващ стандартите маркър код...

3. Какво ни трябва

И така, стигнахме и до PHP. Първото, за което трябва да се погрижим, ако искаме да използваме ext/tidy, е разширението да е инсталирано, заредено и да ни чака да му кажем какво да прави...

Ако тестватена Win32, всичко което трябва да се направи е да се откомментира съответния ред във файла php.ini:

extension=php_tidy.dll

Ако обаче сте на друга платформа, трябва да компилирате разширението. Самото разширение е част от официалната сорс дистрибуция, но също така е нужна и споменатата библиотека TidyLib, която е достъпна на Sourceforge страницата на проекта [<http://tidy.sourceforge.net>]. Освен това трябва да компилирате PHP с опцията **--with-tidy**.

4. Основни положения

Използването на Tidy е всъщност изключително лесно, като се има предвид, че основната работа се върши кажи-речи с две-три функции. Ще ги разгледаме последователно. Илюстрация на стандартния начин на работа:

```
$tidy = tidy_parse_file('myfile.html');
tidy_clean_repair($tidy);
echo tidy_get_output($tidy);
```

На три реда имаме кода, необходим за цялостния "ремонт" на един документ. Нека обърнем внимание на първата функция

```
resource tidy_parse_file ( string filename [, mixed config [,
string encoding [, bool use_include_path]]])
```

Тук единственият задължителен параметър е filename, а именно името на файла, който ще бъде парсван. За него може да бъде даден както относителен път, така и URL (**разбира се, ако опцията allow_url_fopen е със стойност On във вашето php.ini**). Вторият параметър може да бъде или асоциативен масив, съдържащ необходимите конфигурационни опции, или пътят до специален Tidy конфигурационен файл, откъдето съответните настройки да бъдат заредени. Повече за конфигурацията може да прочетете по-нататък в статията. Третият параметър определя каква да е кодировката на изходния код. Последният параметър (use_include_path) указва дали файлът, който трябва да бъде парсван, трябва да бъде търсен и в include_path-а на PHP.

Освен, че зарежда файла, **tidy_parse_file()** също така и извършва някои основни поправки в кода, като преподреждане на тагове и коригиране на други синтактични грешки. Същинската работа обаче се върши от функцията **tidy_clean_repair()**, която следва директивите на зададената конфигурация и която е способна да създаде напълно покриващ стандартите код.

bool tidy_clean_repair (resource tidy)

Функцията приема единствен параметър Tidy ресурс. За да визуализираме вече готовия поправен документ, ни трябва функцията tidy_get_output(), която връща ремонтния код:

string tidy_get_output (resource tidy)

Освен файлове, към Tidy парсера могат да бъдат подавани и низове. Това става чрез функцията tidy_parse_string(), която наподобява много съответната функция за парсване на файл:

resource tidy_parse_string (string input [, mixed config [, string encoding]])**5. Който както го харесва**

Забелязва се тенденция, новите разширения в PHP5 да имат двойно API - в процедурен стил и в обектно-ориентиран стил. Tidy също не прави изключение. Следният код е напълно равностоен на по-горе дадения пример с използване на функции, които приемат Tidy ресурс:

```
$tidy = new tidy();
$tidy->parseFile('myfile.html');
$tidy->cleanRepair();
echo $tidy;
```

Двата стила могат спокойно да бъдат и смесвани, което означава, че следното е напълно възможно:

```
$tidy = tidy_parse_file('myfile.html');
$tidy->cleanRepair();
echo tidy_get_output($tidy);
```

Аз лично никога не бих смесвал обектно-ориентиран с процедурен код, но съм длъжен да ви обърна внимание и на тази възможност. Въпрос на предпочитание е кой от двата стила да използвате. Ако ползвате ООП, трябва да запомните, че параметрите, подавани на методите са същите, с изключение на Tidy ресурсите, които няма нужда да се подават, след като вече става дума за манипулации с обекти. Също така имената на методите са в така наречения studyCaps стил, т.е. вместо tidy_parse_file() има parseFile() и т.н.

6. Може и с буфер

Винаги може да си спестите писане с използването на output buffer. Трябва да предадете низът "ob_tidyhandler" на функцията ob_start() и всичкият output оттук нататък ще минава първо през Tidy за поправка. Това например е много полезно примерно ако ползвате template engine като Smarty.

```
ob_start("ob_tidyhandler");
// Следва кода...
```

Автоматичното използване на Tidy може да бъде настроено и в php.ini. Опцията tidy.clean_output трябва да бъде сложена на On, а пътят до конфигурационния файл по подразбиране за всички документи се задава с опцията tidy.default_config:

[Tidy]

; The path to a default tidy configuration file to use when using tidy

tidy.default_config = /usr/local/lib/php/tidy.cfg

; Should tidy clean and repair output automatically?

; WARNING: Do not use this option if you are generating non-html content

; such as dynamic images

tidy.clean_output = On

Обърнете внимание на забележката - не бива да използвате този подход, ако ще генерирате нещо различно от (X)HTML съдържание, пример за това са графиките, динамично генерирани от GD библиотеката.

7. Конфигурацията

Основният (и всъщност единствен) начин да управляваме работата на Tidy е конфигурацията. Както споменах конфигурационните опции могат да се предават на парсващите функции или като асоциативен масив, или като път до текстов файл, който ги съдържа. Примери за двата вида използване:

```
$config = array('output-xhtml' => true, 'clean' => true);
$tidy = tidy_parse_file('myfile.html', $config);
[myfile.cfg]
output-xhtml: yes
clean: yes
```

```
[test.php]
<? ....
$tidy = tidy_parse_file('myfile.html', 'tidy.cfg');
... ?>
```

Конфигурационните опции на Tidy са много и позволяват достатъчно добър контрол върху това, което се прави с документите. Ще обърна внимание само на най-основните от тях:

- add-xml-decl (Тип: Boolean, Стойност по подразбиране: no) - добавя xml декларация в началото на документа;
- alt-text (String, -none-) - дефинира текст, който по подразбиране да се добавя в alt="" атрибутите на IMG елементите;
- clean (Boolean, no) - указва дали някои презентационни тагове да бъдат заменени със съответните стилови правила;
- doctype (DocType, auto, Възможни стойности: auto, omit, strict, loose, transitional, друга настройка (низ)) - указва Doctype декларацията на документа;
- drop-empty-paras (Boolean, yes) - премахва празни параграфи;
- drop-font-tags (Boolean, no) - премахва FONT таговете;
- hide-comments (Boolean, no) - указва дали коментарите да бъдат премахнати от изходния код;
- input-xml (Boolean, no) - ако входните данни са в XML формат, трябва чрез тази опция да се настрои използването на XML парсер вместо поправящия синтактични грешки HTML парсер;
- join-classes (Boolean, no) - комбинира няколко класа в един, ако е установено, че за един елемент има няколко класа;



Българската асоциация на софтуерните компании – БАСКОМ проведе работна среща на компании от софтуерния

бранш във Варна

Българската асоциация на софтуерните компании – БАСКОМ (www.basscom.org) проведе на 9 и 10-ти юли дискусия за състоянието и посоките за развитие на българската софтуерна индустрия.

В срещата се включиха водещи фирми от региона – **Pi-consult, DAN Electronics, Euro Risk, Kytex, Tonegan, Svet Solutions Interactive, Black Sea Consulting Group, Triada Soft, E-stream и Clarima** и 12 фирми, членове на БАСКОМ.

Участниците дискутираха водещата роля на сектора за развитието на България по пътя на трансформирането в "икономика на знанието". Остро бе поставен въпроса за конкретните стъпки, които бизнеса и държавата трябва да предприемат за устойчивото развитие на сектора като: подготовка на специалисти, индустриалните треднове, обща маркетингова стратегия и качеството на образованието в момента.

В този аспект се дискутира мисията на БАСКОМ като професионална организация за подобряване на бизнес средата и насърчаване на бизнес етиката. Регионалният център на Европейският софтуерен институт в България – ЕСИ Център – България (www.esicenter.bg) представи програмата си до края на годината за повишаване на конкурентноспособността на индустрията. Фирмите проявиха специален интерес към въвеждането на световни стандарти за софтуерни процеси CMMI и курса на ЕСИ в тази област, които ще се проведе в края на септември 2004 г.

Приоритетите за представяне на индустрията на международни изложения бяха обсъдени с директора на Агенция за развитие на информационните и комуникационни технологии - Орлин Кузов.

Програмата за участието на България с национален щанд на предстоящото през октомври месец изложение Systems в Мюнхен беше представена от Лукас фон Цалингер, GTZ (Германско дружество за техническо сътрудничество).

В заключение от дискусията Георги Шарков, председател на БАСКОМ, изтъкна значението на интегрирания подход за развитието на бранша и утвърждаване на неговия имидж в рамките на Стратегия и план за действие за конкурентно способност на България на световните ИКТ пазари.

- join-styles (Boolean, yes) - комбинира няколко стила в един, ако е установено, че за един елемент има няколко стила;
- output-xhtml (Boolean, no) - указва дали изходния код да бъде XHTML;
- output-html (Boolean, yes) - указва дали изходния код да бъде HTML;
- output-xml (Boolean, no) - указва дали изходния код да бъде XML;
- word-2000 (Boolean, no) - спестява главоболия със създадени с микрософтовия продукт документи.

- indent (Auto/Boolean, no) - указва дали да има индентация (поделените да се намират на известно разстояние навътре в реда спрямо своите родители);
- indent-attributes (Boolean, no) - дали всеки атрибут да е на нов ред;
- indent-spaces (Integer, 2) - указва колко празни полета да има при индентация;
- wrap (Integer, 68) - указва броя на символите на ред (кода ще е автоматично в word-wrap).

Разбира се, има и още доста опции за по-фина настройка, като пълния списък можете да намерите на <http://tidy.sourceforge.net/docs/quickref.html>.

8. А кирилица?

Бях споменал, когато ставаше въпрос за парсващите функции, че може да се избира encoding-а на изходния код. Тогава не посочих възможните кодировки. Всъщност възможностите са следните: ascii, latin0, latin1, raw, utf8, iso2022, mac, win1252, ibm858, utf16, utf16le, utf16be, big5 и shiftjis. Да, правилно виждате - няма win1251... Ефектът е такъв, че ако не сме дали стойност за кодировката, на страниците ни на български ще се виждат само разбъркани символи. Единственото работещо решение за кирилица тук е да се използва стойност за encoding параметъра "raw", по този начин всъщност указваме да не се правят промени със символите и да се пазят символните данни "сурови".

9. Заключение

Ако имате проблеми със своя маркъп код, новото PHP5 разширение Tidy е едно много приятно решение за вас.

С малко усилия вече могат да се постигнат задоволителни успехи по отношение на чистота и правилността на кода. Това прави Tidy една от множеството полезни придобивки в новата версия на езика. Приятно ползване!

Ивайло Гроздев

Yoper*Версия 2***Поддържан хардуер**

Следните процесори се поддържат от системата

- Pentium II
- Pentium Pro/ Celeron
- Pentium III /Celeron Coppermine
- Pentium 4 /Xeon
- Athlon/Duron/K7 (Athlon-xp и Athlon-mp)
- Cyrix M2

Какви са предимствата на Yoper?

- Бърза инсталация
- Пакети от Slackware (installpkg, removepkg, xfree86setup).
- Raidtools от Redhat.
- Kudzu Hardware разпознаване от RedHat.
- Стартъп скриптове от RedHat.
- Mozilla от Netscape Ltd.
- OpenOffice от Sun.
- Hwsetup от Knoppix.

Сваляне

Можете да си го свалите отивайки на този адрес:

<http://gd.tuwien.ac.at/opsys/linux/yoper/>

Големина : **675 МБ**



Сигурно понякога сте се впускали в експерименти за избор на нова операционна система или просто от любопитство сте тествали някоя нова дистрибуция.Ето поредното предложение , което може и да ви е интересно.

Преди няколко дни излезе новата версия 2 на Yoper - операционна система с много добри показатели, оптимизирана за PC с 686 или по бърз процесор.Тя е главно десктоп базирана операционна система за разлика от Redhat или Mandrake.

За какво можете да ползвате Yoper?

Терминални сървъри (Ydesktop terminal server)

Работни станции (Ydesktop)

Системи за домашно ползване (Ydesktop)

Клъстерни системи (Yminimal)

Системи за разработка, които имат възможността да работят с много софтуерни пакети

Крайни сървъри, рутери или фаеруоли

Поддържани пакети (част)

3ddesktop 0.2.7
 3ddiag 0.716
 AfterStep 1.8.11
 DBI 1.42
 GConf 2.6.1
 Geramik 0.27
 Gtk-Perl 0.7009
 HTML-Parser 3.35
 Linux-PAM 0.77
 MAKEDEV 1.7
 MesaLib 6.0.1
 NVIDIA 5336
 Net-Daemon 0.37
 ORBit2 2.10.2
 PIRPC 0.2017
 QtPixmap 0.28
 Storable 2.12
 Video DVDRip
 WindowMaker 0.80.2
 XFree86 4.4.0
 XML-Dumper 0.71
 XML-Parser 2.34
 Ygnome 2.6.1
 Ykde 1.0.0
 a2ps 4.13
 aalib 1.4.0
 acidrip 0.12
 alsa-lib 1.0.5
 alsa-utils 1.0.5
 amarok 1.0.1



```

apt 0.5.15cnc6
arts 1.2.3 1.2.91
aspell 0.50.5
at-spi 1.5.2
atk 1.7.0
audiofile 0.2.6
autoconf 2.59
automake 1.8.5
automake14 1.4p6
automake16 1.6.3
avifile 0.8
bash 2.05b
bc 1.06
beecrypt 3.1.0
bin86 0.16.15
bind 9.2.3
binutils 2.15
bison 1.875
bluefish 0.13
bochs 2.1.1
bug-buddy 2.6.1
bzip2 1.0.2
cdrtools 2.01a32
cdparanoia alpha9.8
cdrdao 1.1.8
chkconfig 1.3.9
chkfontpath 1.10.0
compat-gcc 3.3
control-center 2.6.1
coreutils 5.2.1
courier-imap 3.0.4
cpio 2.5
cups 1.1.20
curl 7.11.0
cvs 1.11.16
cvsgraph 1.4.0
cyrus-sasl 2.1.18
db 4.1.25
db3 3.2.9
desktop-data-SuSE 9.1
desktop-file-utils 0.4
dhcp 3.0pl2
dhcpcd 1.3.22pl4
dialog 0.9b
dietlibc 0.23
diffutils 2.8.1
distcc 2.16
divx4linux 20030428
docbook-dtds 1.0
docbook-style-dsssl 1.78
docbook-style-xsl 1.65.1
docbook-utils 0.6.14
dosfstools 2.10
doxygen 1.3.7
dvd+rw-tools 5.19.4.9.7
dvdrttools 0.1.5
eject 2.0.13
elfutils 0.95
emacs 21.3

```

Екипът разработващ тази операционна среда ви дава възможността да добавяте собствени пакети към дистрибуцията посредством анонимен CVS достъп. Сайта на проекта е: <http://www.yoper.com/>

WebDevMagazine: Please introduce yourself.

Sven Vermeulen: My name is Sven Vermeulen and I'm currently the project lead of the Gentoo Documentation Project (GDP) [1] which, as the name implies, takes care of the Gentoo Documentation [2]. It's a great position to be in because one cannot write documentation without a good understanding of the subject.

I'm also positioned as project co-lead for the Gentoo Public Relations Project [3] but this project still needs to find it's way; it has a fairly low activity record. But work is on the way to improve the PR. Due to my position as GDP project lead I'm one of the 'Gentoo Managers', a group of Gentoo project leads that coordinates the Gentoo Development and communicates directly with each other to fix issues quickly when they occur. Apart from this coordination group we also have the 'Gentoo Trustees'. Those are an initial group of developers that are forming the Gentoo Foundation, a legal entity that allows us to protect the Gentoo logo and trademark, to have financial power and be able to 'own' hardware and such. I'm one of those trustees as well.

Okay, enough Gentoo positioning, you asked me to introduce myself I'm turning 22 in a few months, live in Belgium (Europe), more precisely in a touristic city called Bruges. My day-time activities are studying (I'm majoring in Software Engineering at the University of Ghent) and researching (I'm the project lead of an Internet technologies research group). I try to spend as much time as possible with my girlfriend Eveline and enjoy going out to some local pubs (as long as they're not too noisy). And finally, when I'm alone, not working, studying or chatting, I spend some time in front of my television, watching DVDs.

WebDevMagazine: When the work will begin on the new portage-system ?

Sven Vermeulen: Good question. The first work on Portage-ng (the name assigned to the new portage system) happened a long time ago when Daniel Robbins started outlining the roadmap for it's development [4]. As one cannot create software without requirements, the first step was to document all feature requests [5] (without going into the technical specifications or language-specific stuff). Although this step seems trivial, it resulted in a long feature list and in big discussions. **Sadly this is also where the development stopped.** Many people gave their vision on the new portage-ng resulting in big threads on the mailinglist, but too few people stripped those discussions into summaries for inclusion in the feature request document. Currently many developers are asking to speed up the project again so I'm sure it won't take long before the discussions start heating up, but currently development is progressing slowly.

This might sound strange, but the reason is fairly simple: the current Portage developers (Nicholas Jones, Jason Stubbs and a few other devs who will now probably kill me because I don't mention them) are integrating many of the portage-ng features in the current Portage. So the need for portage-ng is only felt by those that need functionality which cannot be integrated in the current Portage.

WebDevMagazine: Will it be more META i.e. will we be able easily use it under different Linux-OSes ?

Sven Vermeulen: Yes, the design specification of portage-ng mentions that it should easily integrate in other systems, not only other Linux distributions, but also other operating systems such as BSD, MacOS. It should also allow for seamless addition of functionality, so the magic word of "modularity" is frequently heard on the mailinglist.

WebDevMagazine: Will there be configuration utility - like debian ? Mostly for semi-automated configuration after emerging more complex packages ? I mean different than etc-update..

Sven Vermeulen: This is highly probable but not related to the development of portage-ng. We are already working on several tools that will make managing a Gentoo system a breeze. This includes a Gentoo Installer [6] (with features such as remote installation management, batch installation, graphical front-ends, ...), Gentoo-specific configuration tools [7] (runlevel configuration, system administration, portage configuration, networking configuration, ...) and feature-specific configuration tools (such as virtual hosting).

If there is no configuration tool for your favorite action (for instance semi-automated configuration) it is not because we don't want it, but because noone started developing one.

Sven Vermeulen: Work is already on the way to create a LiveCD with a graphical interface, autodetection scripts, applications and more. It will be catalyst-based because Catalyst allows to easily create those LiveCDs. It's just a matter of changing the specification file et voila. Kudos to the Catalyst developers for making such a powerful tool! If I recall correctly, such a graphical LiveCD will be available with our 2004.3 release. No hard promises though; we know we can create such LiveCDs easily but many other factors come into play when we want to release them (hosting resources, multiple architecture support, documentation aspects, ...) and there was no final saying on this yet.

WebDevMagazine: What is the "purpose of meaning" :"))

Sven Vermeulen: The purpose and meaning of Gentoo? To provide choices to the users, to give them control over every aspect of their system by not forcing them to work one way or the other but to give them the ability to choose their path. To evolve quickly and remain flexible so it integrates easily in any environment. To keep an open and attractive development model for developers and contributors from all over the world. And of course to blow other distributions their socks off performance-wise when they challenge Gentoo.

* Въпросите са зададени благодарение на помощта на raptor@tvskat.net

Ситуацията на пазара на търсене и предлагане на работа в ИТ сферата

По последни данни има разширение към периферията на страната.

С все по навлизащите инвестиции у нас в сферата на информационните технологии и телекомуникацията се разширява и предлагането на работа в този сектор. Голяма част от инвеститорите искат да “избегнат” стандарта на София и насочват усилия към други градове където с много по-малки разходи разработват продукт, който се продава на същата цена. Явлението което се наблюдава, може да се нарече “вътрешен аутсорсинг” и със сигурност не е полезен за икономиката в направлението в тази сфера. Ако се спрем на понятието единен стандарт, който някои организации желаят да наложат у нас, може би ще реши проблема. Открит остава въпроса обаче за спазването на този стандарт. От разговори с собственици на малки фирми, работещи в тези градове останах с впечатлението, че цената на труда в тези градове се определя от условията на живот там и те предпочетат да дават много по-малки заплати от средното равнище. **Някои фирми прибягват и до понятието наречено “разсрочено плащане на заплати”, което говори само за себе си.**

Търсене

Ако погледнем данните от таблицата в дясно както и на другия портал за работа, можем да си направим следния извод Градовете в който се търсят най-много ИТ специалисти са

София

Пловдив

Варна

Русе

Стара загора

Нека да погледнем тази таблица, която дава общ поглед върху това какво е положението за изминалата година:

Град	Процент
Благоевград	1%
Бургас	3%
Добрич	1%
Габрово	0%
Хасково	1%
Монтана	0%
Пазарджик	0%
Перник	0%
Плевен	1%
Пловдив	3%
Разград	0%
Русе	0%
Шумен	0%
Силистра	0%
Сливен	0%
Смолян	0%
София	77%
Софийска област	1%
Стара Загора	0%
Търговище	0%
Варна	5%
Велико Търново	0%
Видин	0%
Враца	0%
Ямбол	0%
други в България	1%

Едва ли трябва да се учудваме от този факт, понеже влязоха доста инвестиции в тези региони, както и в тях се появиха доста инициативни предприемачи, които успяха да наложат свои проекти и да отворят работни места. Няма да се спирам сега на имена на фирми и проекти, но бих искал да кажа, че идеята да се прави бизнес в по-слаби икономически райони заслужава и похвала и различен поглед от страна на институциите. От друга страна погледнато обаче изтичането на персонал от тези центрове към по-стабилните икономически райони се дължи в много голяма степен на ниското заплащане и на факта, че работата никъде не е много по различна, а само заплащането. Повечето специалисти се местят именно поради тези причини и това води до един сериозен проблем, а именно до изтичането на знания от периферията към центъра. Когато една фирма не може да задържи специалистите при себе си и се налага всеки път да взима нови и нови за обучение, губи и самата компания губи и развитието на отрасъла в този регион, колкото и елементарно да звучи.

Компаниите попадат в ситуацията на безизходица, породена обаче от собствената им политика, нежеланието им да повишават заплатите води до загуби на самата фирма, вместо печалба.

Цифрите

С този лек анализа на състоянието в страната минавам към статистиката, любезно предоставена от **ITJobs.bg** и **JobTiger.bg**, два сайта определящи пазара на такива ресурси у нас.

Не трябва да забравяме, че това са платени обяви което не отразява реалното предлагане на работа, но в процентно съотношение дава да се разбере, че основното ядро на търсене и предлагане се намира в София.

Румъния

Ако направим един мост със северната ни съседка, положението не е много по различно, силните икономически райони привличат много специалисти, докато слабите в това отношение издъхват. Ако направим паралел обаче и в заплащането, тука излизат големите разлики по отношение на нашата страна. Нека само да ви припомним, че страната е избрана за официална база на много големи компании за тази част на Европа, в която влизаме и ние. Политиката на страната за привличане на инвестиции работи много добре, за разлика от нашата.

Анализ

Гледайки в далечно бъдеще, може би има светлина за оправяне в тази част на бизнеса, но засега се налага да се съобразяваме с ситуацията която сме попаднали. Все повече специалисти се наемат, но липсват данни за това за да може да се направи пълен анализ, но ще продължаваме да следим за вас положението в страната от гледна точка на статистиката и на анализирането на този дял от пазара на работа. Благодарим за съдействието на порталите JobTiger.bg, Itjobs.bg, itjobs.ro. Ако разполагате с данни, които можем да използваме, моля свържете се с нас.

Дойде ли време за персонализираните интернет портали?

Преди време излезе една статия в вестник "eWeek", в която се помъчих да изясня понятието "Персонализиран портал". Сега продължавам с резюме на предната и малко анализи по темата. Едва ли скоро ще дойде времето, когато телевизора ще ви познае и ще ви каже от какво имате нужда или ще ви даде информацията от която имате нужда. Това важи с пълна сила и за другата електронна медия - радиото. Има начин обаче една медия, която може да служи така както имаме желание и да ни дава информация от каквато имаме нужда. Нека да разгледаме следната ситуация.

Ситуация

Влизаме в голям портал и четем новини за европейското по футбол (примерно), след това отиваме на други сайтове, пак се връщаме на този сайт и пак четем новини от тема спорт и времето. На третото влизане сайта зарежда информацията от която имаме нужда само за нас и то без да я търсим.

Как е досега?

Потребителя си създава акаунт, чрез който може да управлява информацията която иска да получава. Ако имаме обаче много любими сайтове, тази операция става безмислена и изисква много работа от страна на потребителя. Ако се замислим в тази посока, потребителя сам да си избира, какво да чете и като това да си купи вестник и да прочете само малка част от страниците а другото да използва за други цели, което първо прави цялата тази операция излишна, втори води до разход на средства.

Персонализираните портали не са замислени много отдавна като идея. Тази идея обаче изисква страшно много време и средства за да бъде реализирана, так аче да работи правилно. За съжаление у нас няма лаборатории, които да тестват и обсъждат такива нововъведения и какви ползи биха довели за крайния клиент, за рекламодателя и за собствениците на такива портали. Тук не става дума за малки сайтове, които няма какво да предложат като комплекс от информация. Първата стъпка обаче е направена и то стъпка в положително направление. Порталите у нас започнаха разделянето на информацията на канали, като лидер в това отношение е нетинфо, следван от news.bg, що се отнася до изцяло новинарски ориентирани портали. Следващата стъпка ще е персонализираната информация, която е напълно логична според мен. Забелязват се сигурно, че говорим само за сайтове, който са у нас. Това е така защото в страната има много качествени продукти, които могат да дадат начало на една нова страница в предоставянето на информация и на услуги на потребителите.

Услуги

Като говорих за предните аспекти на персонализацията не се спрях на услугите, тук повечето от нещата значително съвпадат, но не и що се отнася до сигурността, предоставянето на информация не изисква кой знае каква сигурност между портала и клиента, докато предоставянето на услуги минава и през проблема със сигурността на данните, разплащанията и анонимността на потребителя в известни граници. За това тук не можем да говорим за пълна персонализация, но като частичен ефект, това може да се постигне с предлагането на специфична информация за услуги, а след идентификацията на потребителя да се пристъпи към същинското извършване на предоставяне на услугата.

Какво трябва да се направи

Ако говорим за близките 2-3 години, това няма да бъде резлизирано, поне не в същността в която трябва да работи един такъв портал. Не бива да се пренебрегват и защитата на данните, който са косвено събрани от портала и обвързани за всеки потребител, което е основна пречка за забавянето на технологията. Друга причина, която споменах и малко по-горе и многото време което трябва да се отдели за програмиране на един такъв портал и ресурсите, който поглъщат едни такива дейности. Преди това обаче трябва информацията, която потребителя иска да получи да бъде набавена, разбира се, което порталите разбират и много добре и в последно време правят наистина добре по мой лични наблюдения.

Заклучение

Една такава стъпка би довела до полезност на определени интернет портали. Засега портал е място където има много неща и за малко или много време се намира това което интересува потребителя или клиента. С минаването към следващата стъпка, тази на персонализацията ще се избегне излишното лутане и търсене от страна на по – малкото запознатите потребители. Това е все едно, че си купувате вестник с информация само каквато ви е интересна. Няма да се спирам сега на 'борбата' на електронните медии с традиционните печатни, защото всеки тип медия си има своите предимства и недостатъци и всеки тип медия си има своите читатели. Ако мога с няколко думи да обобща цялата схема на един такъв портал, това би било най-доброто решение за един интернет портал, на следващите 4-5 години, а защо не и по-скоро.

Малко отклонение

След известни наблюдения, бизнеса в страната предприе логичната стъпка на консолидация на идеи с целта да се направи нещо ново и полезно за интернет пространството у нас. Похвално е, разбира се да има нови инициативи, а и когато новаторството влиза в него напредъка ще е по-голям. Всичко в интернет у нас е свързано, не може и няма да има проекти, който излизат много напред с идеи, както е и в световното интернет пространство.

Концептуален модел е Gmail, който просто провокира изблика на много нови услуги и променяне на модела на действие на сегашните.

Струва си и да се замислим защо се влагат толкова много средства в пропагандирането на аутсорсинг идеята, а не се вложат малко средства в софтуерни лаборатории и такива по стратегии за интернет, който да дават мнения и по-важните решения по точно тези проблеми. Защо у нас големи корпорации като IBM, Microsoft, Oracle, и други имат само представителство, което се грижи за продажбата на продуктите, а не спонсорира или подкрепят една такава лаборатория, както е в 'белите страни'. Можем да се надяваме само, че доброто желание на самите интернет портали ще достигнат до вярното решение.

Сайтове регистрирани за наградата за "ИТ Иновации" 2004

Има много добри предложения

PCMagazine България и Списание точка ком **инициират създаването на наградата ИТ Иновация на годината.** Целта на конкурсът е да се подкрепи инициативността във сферата на информационните технологии и да се стимулира създаването на нови проекти. Категориите са следните:

Корпоративен продукт
Система внедрена в корпоративния бизнес, други решения.

Военна информационна система
Система внедрена в отбранителната индустрия.

Електронно обучение
e-learning, системи за електронно обучение, образование, виртуални общности в тази сфера.

e-government приложение
Продукти приложими в електронното правителство.

Софтуерен продукт
Софтуерен продукт, framework, операционна система.

Интернет сайт
B2B сайт, посреднически сайт, информационен сайт.

Хардуерен продукт
Нов продукт или осъвършенстван стар.

Игри и софтуер за обучение и отмора
Всякакъв тип игри, включително за мобилни телефони.

Мобилно приложение
Приложение за мобилни устройства.

ИТ Личност
Личност със заслуги за развитието на тези сфери. В този брой сме се спрели на сайтовете, които засега са сев регистрирали за тази висока награда, и така предложенията са следните:

**Zone4Web.com**

Проектът **Zone4Web.com** представлява контент мениджмънт система, чиято цел е да събере и организира информационни ресурси (статии, скриптове и програми, новини, акценти, анонси, помощна документация и др.) в помощ на професионалистите от ИТ сектора в България. Сайтът е насочен предимно към програмисти, разработчици на приложения за уеб и уеб дизайнерите. Разнообразието и гъвкавостта на предлаганата информация и документация предполага и разширяване на тематиката в зависимост от нуждите на потребителите му.

Разработването на сайта започна през 2003 г. от Пламен Железов, а по-късно се включиха Пламен Георгиев и Станислав Лечев. Използваните технологии са Apache Web Server, PHP 4.3.4, JavaScript, HTML и MySQL 4.0.15a.

Bgproekti.com

BG Проекти уеб сайт, който свързва специалисти (фирми или частни лица) от различни области, с клиенти, търсещи техните услуги.

Потребителите, които желаят да се регистрират като специалисти, могат да представят накратко своята дейност, както и информация за разработени от тях проекти. От своя страна клиентите, търсещи изпълнител за дадена задача или проект, могат да публикуват описание, и да изчакат фирмите и специалистите да направят своите оферти; така клиентите избират най-добрата оферта или се свързват с тези, които са направили добри предложения.

Сайтът BG Проекти (www.bgproekti.com) не е първия по рода си в България, по според специалисти е първия сериозен опит в тази насока. Макар и само от няколко месеца, той има вече над 120 регистрирани специалисти в различни области (предимно уеб дизайн, софтуер и инженеринг), а почти всяка седмица се публикува нов проект. Според инж. Георги Златев (управител на фирма "МетаВизия" ООД - създател на BG Проекти), този тип сайтове са популярни в чужбина и често използвани от доставчиците на услуги. Той смята, че пред BG Проекти има отлични перспективи, и в бъдеще сайта може да се превърне в услуга, даваща значителни възможности на българските фирми и специалисти да намерят нови поръчки и клиенти. Сайтът е създаден от фирма МетаВизия - гр. Русе. Фирмата се занимава с уеб дизайн от 2001 г., като преди това е създавала няколко подобни сайта.

Друг проект, който залужава голямо внимание е наречен **Laffer**. Той запазва тенденциите за уеб базираност и е за обмяна на бързи съобщения по ICQ, MSN, AOL, Yahoo мрежите – нещо много необходимо, особено след все повече променяните протоколи на тези услуги. Laffer е написан на HTML и JavaScript, използва DOM 1,2 и PHP. В момента с Laffer може да разменяте текстови съобщения, да извличате и визуализирате списъка с приятели и да виждате кога даден потребител е на линия.

Всъщност Laffer проекта е много повече. Той се състои от две основни части - Laffer уеб клиента и IM Proxy сървър.

Laffer уеб клиентът е HTML уеб приложение, използващо JavaScript и DOM. Не го бъркайте с Java аplet или Flash приложение. За своята работа уеб клиента използва PHP и MySQL за база от данни.

Уеб клиентът в момента предлага възможност за използването на интерфейса на различни езици. Към момента се поддържат преводи на български и английски език.

Част от предимствата, които дава уеб клиента са:

- * Не се нуждае от Java или Flash. За да го използвате се нуждаете единствено от модерен уеб браузър

- * Сигурен е. Поне дотолкова доколкото е сигурен уеб браузър, който използвате.

- * Използва порт 80 и може да се използва от офиси или места където е

забранено използването на стандартните клиенти за обмяна на бързи съобщения

За да работи уеб клиента се нуждае и от "двигател". Основният "двигател" на проекта е IM Proxy (IMP) сървър. IMP е сървър, който се грижи да поддържа постоянна връзка към отдалечения сървър за мигновени съобщения и който доставя съобщенията на ползвателите на Laffer уеб клиента.

Сървърът е написан на Perl и е проектиран така, че предлага възможност да се свързвате с различни мрежи като ICQ, MSN, Yahoo и др. С негова помощ можете да разменяте съобщения, да получавате списъка с вашите приятели и информация за техния статус.

Освен основните части към Laffer проекта е добавен и експериментален сървър, който изпълнява роля, сходна на сървърите занимаващи се с обмяна на бързи съобщения, като ICQ сървърите например.

Всички части в проекта се разпространяват под GPL лиценз и Laffer е свободен софтуер <http://fsa-bg.org/bin/view.cgi/Fsosite/FreeSoftware>.

Основен разработчик на проекта е **Славей Караджов**, който е и създател на сайта на Linux-bg.org, а повече информация за Лафер може да получите от: <http://laffer.sf.net>.

Ако знаете някой проект, който заслужава да бъде отличен, регистрирайте го за наградата за "ИТ Иновации 2004" на адрес <http://itaward.spisane.com>. Право на участие имат само сайтове/идеи или проекти, които са стартирали в периода от 01.04.2003.

Крайния срок за регистрация е 15.09.2004.

Тематиката , трябва да е свързана със сферата на информационните технологии, телекомуникациите, интернет културата и всичко, което е свързано с горните области.

Ако сайта/проекта е стартирал по рано от обявения срок, но във срока е стартирал нова идея, която влиза в горепосочените критерии се допуска за участие.

Участие могат да вземат интернет сайтове, софтуерни продукти и всичко свързано с горните области.

Сайтове с порнографско, незаконно съдържание и всички проекти, които НЕ влизат в допустимите категории, няма да бъдат вземани под внимание.



Най-новото от света на

PHP

<http://spisane.com>

Всички новости от света на интернет технологиите