



PHP

עבר, הווה ועתיד

נובמבר 2007

שחר עברון <shahar.evron@gmail.com>

קבוצת משתמשי PHP בישראל <http://php-israel.org>

- ברוכים הבאים!

- מי אני?

- משתמש ב-PHP כחמש שנים

- עובד בזנד טכנולוגיות כשנתיים וחצי

- מי אתם?

- משתמש/ת מנוסה ב-PHP?

- מתכנת/ת ב-PHP 5.x?



מה זה PHP ?

PHP היא שפת תכנות
(סקריפטינג) רבת משתמשים
המיועדת במיוחד לפיתוח ל-
Web, אם כי אפשר להשתמש
בה גם לצרכים אחרים



מה זה PHP?

- PHP עצמה היא תוכנה חופשית / קוד פתוח, והרוב הגדול של התוכנה הכתובה ב-PHP היא חופשית
- PHP רצה בהצלחה על הרבה (מאד) פלטפורמות
- PHP היא שפה גמישה מאוד וקלה מאוד ללימוד
- חוץ מ-Web, אפשר לעשות עוד דברים עם PHP:

CLI -

Desktop applications -

Embedded -

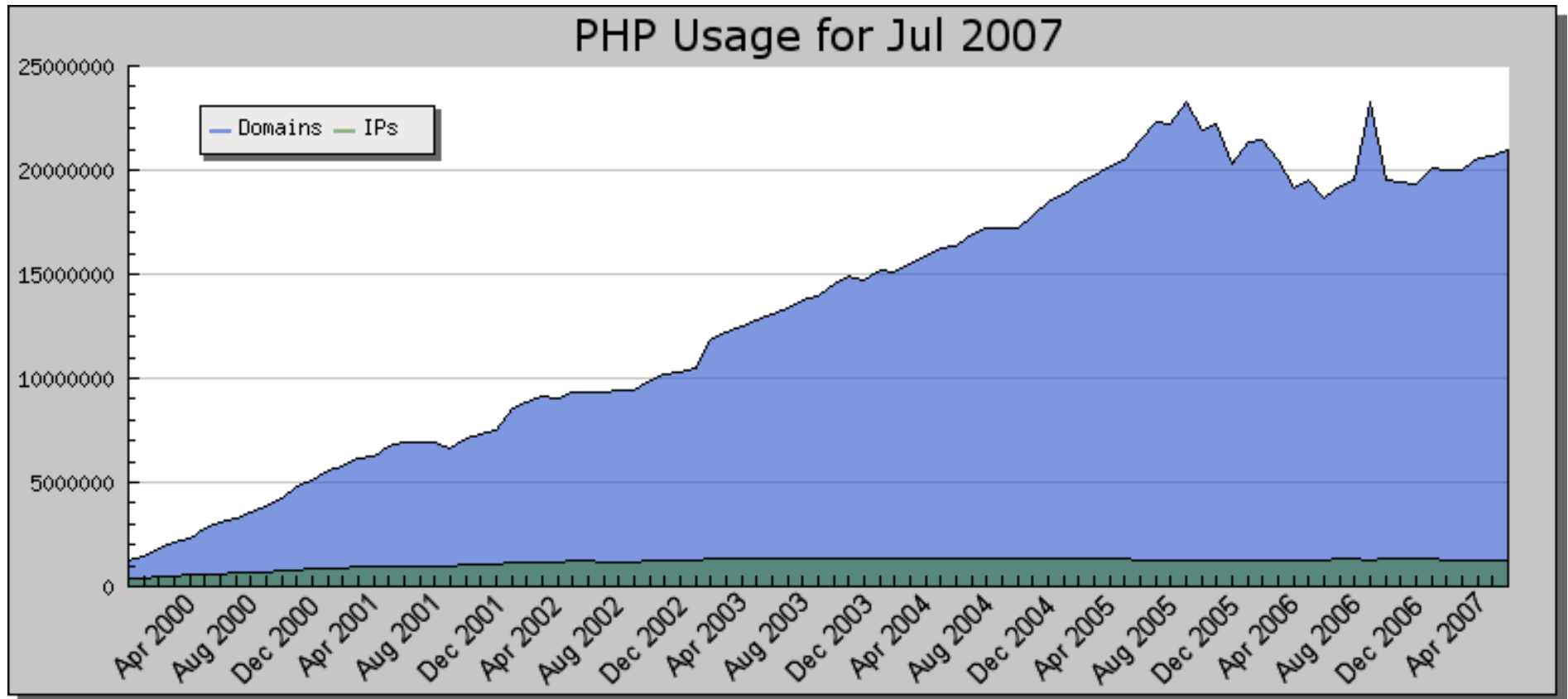


מי משתמש ב- PHP ?

- אנחנו ואתם, ועוד הרבה מאוד אנשים
- PHP היא כנראה שפת ה- Web הפופולרית ביותר, והיא אחד הכוחות המניעים את השינויים ב- Web.
- חלק גדול מאתרי ה- Web הפופולריים ביותר מונעים ע"י PHP:
- Yahoo!, Flickr, Facebook, Digg, Wikipedia ...
- ... אבל גם בנקים, ממשלות וחברות ענק בינ"ל



מי משתמש ב-PHP ?



מקור: Netcraft



עבר: ההיסטוריה בקצרה...

• **1995:** רסמוס לרדורף משחרר את PHP/FI

- תחילת הדרך: אוסף של סקריפטים ב-Perl
- עם הזמן הפרוייקט הוסב ל-C, ונוספו יכולות גישה לבסיסי נתונים, שאפשרו יצירת תוכן דינמי

• **1997:** PHP/FI 2.0

- כמה אלפי משתמשים ברחבי העולם
- כ- 50,000 (1%) משרתי האינטרנט



עבר: ההיסטוריה בקצרה...

• **1997**: אנדי גוטמנס וזאב סורסקי כותבים מחדש את **PHP 3.0** כפרוייקט לימודי לטכניון

- הגרסה הראשונה ש"מזכירה" את **PHP** של היום

- שם חדש: **PHP: Hypertext Preprocessor**

- מפתחים רבים נוספים לפרוייקט

- שיפורי תחביר, **OO**

- מאות אלפי אתרים (**10%**)

- שחרור סופי: יוני **1998**





עבר: ההיסטוריה בקצרה...

- **1998:** אנדי וזאב מתחילים לכתוב מחדש את הליבה של PHP: כך נוצר ה-Zend Engine

- שיפור ביצועים וארכיטקטורה

- שיפור מודולריות

- **PHP 4.0** משוחרר במאי 2000

- Sessions

- Output Buffering

- תמיכה בשרתים נוספים מלבד Apache





עבר: ההיסטוריה בקצרה...



- יולי 2004: PHP 5.x משוחרר

- Zend Engine 2.0

- Object model - כתיבה מחדש של ה-

- PHP 5.1 משוחרר בנובמבר 2005

- שיפורי ביצועים ותיקוני באגים רבים מאוד

- PDO כברירת מחדל

- PHP 5.2 משוחרר בנובמבר 2006

- שיפורים בניהול הזיכרון ושיפורי ביצועים נוספים

- Filter, JSON, Upload hooks



אז מה עוד אפשר
לעשות עם PHP 5.x?

(חוץ מ-OOP?)

```
try {
    $db = new PDO('mysql:host=localhost;dbname=mydb',
                 'user',
                 'xxxxx');

} catch (PDOException $e) {
    echo "Error connecting to DB: {$e->getMessage()}";
    exit;
}

$stmt = $db->prepare('INSERT INTO employees (name, salary)
                    VALUES(?, ?)');

foreach ($employeeList as $emp) {
    $stmt->execute(array($emp['name'], $emp['salary']));
}
```



XML: SimpleXML

```
$xml = <<<EOML
<bookmarks>
  <bookmark>
    <name>CNN World News</name>
    <url>http://cnn.com</url>
  </bookmark>
  <bookmark>
    <name>Planet PHP</name>
    <url>http://planet-php.org</url>
  </bookmark>
</bookmarks>
EOML;

$bookmarks = new SimpleXMLElement($xml);
foreach($bookmarks->bookmark as $bmark) {
    echo "{$bmark->name}: {$bmark->url}\n";
}
```

```
// continuing from previous example...
$domBookmarks = dom_import_simplexml($bookmarks);

$domDoc = new DOMDocument();
$domBookmarks = $domDoc->importNode($domBookmarks, true);
$domDoc->appendChild($domBookmarks);

foreach ($domDoc->getElementsByTagName('bookmark') as $bmark)
{
    $rank = $domDoc->createElement('ranking', '5');
    $bmark->appendChild($rank);
}

echo $domDoc->saveXML();
```



Web Services: SOAP

```
// SOAP Server
class myServiceClass
{
    public function doServiceAction($foo, $bar)
    {
        // ...
    }
}

// Set up SOAP server
$server = new SoapServer('myService.wsdl');
$server->setClass('myServiceClass');

// Handle request
$server->handle();
```



Web Services: SOAP

```
// SOAP client
$myWsdL = 'http://www.example.com/service/myService.wsdl';
$client = new SoapClient($myWsdL);

try {
    $myData = $client->doServiceAction('foo', 'bar');
} catch (SoapFault $fault) {
    echo "Error getting data from remote server: " .
        $fault->getMessage();

    exit;
}
```



SPL: Standard PHP Library

• אוסף של מחלקות (Classes) וממשקים (Interfaces) שנועדו לפתור בעיות נפוצות

- <http://www.php.net/~helly/php/ext/spl/>

```
// SPL Directory Iterator
$musicDir = new DirectoryIterator('/home/shahar/Music');

foreach ($musicDir as $song) {
    if ($song->isDir()) continue;
    echo "$song - Last played: " .
        date("Y-m-d h:i:s", $song->getATime()) . "\n";
}
```



SPL: Standard PHP Library

```
// SPL Interfaces: Iterator, Countable
```

```
class PhotoAlbum implements Iterator, Countable
{
    // ... a bunch of code here ...
}
```

```
$album = new PhotoAlbum('/home/shahar/Albums/India');
```

```
// Access the object using the 'count()' function
echo "The {$album->getTitle()} contains " . count($album) .
    " photos.<br />";
```

```
// Iterate using 'foreach()'
foreach ($album as $photo) {
    echo htmlspecialchars($photo->getTitle()) .
        '';
}
```

The PHP logo, featuring the lowercase letters 'php' in a bold, black, sans-serif font, set against a blue oval background with a white border.

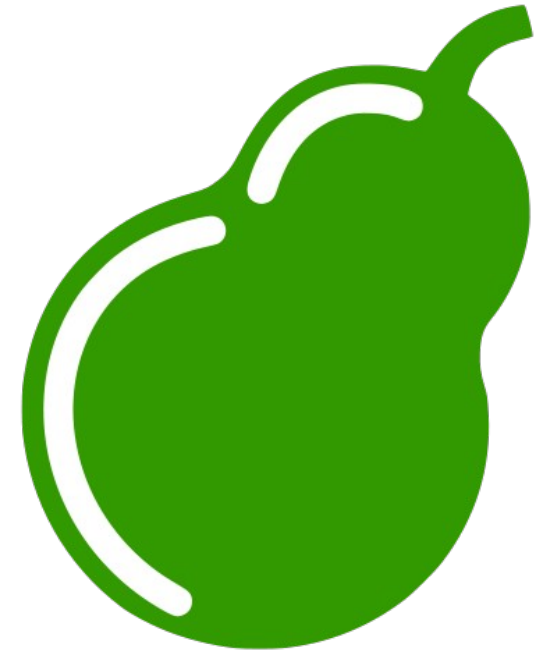
PHP Frameworks



<http://cakephp.org>



<http://framework.zend.com>



<http://pear.php.net>

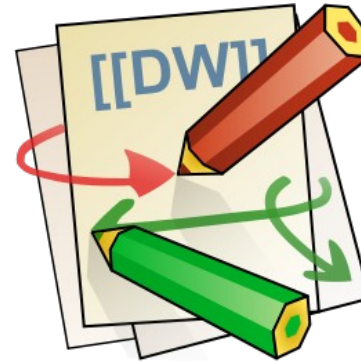
<http://symfony-project.com>



<http://ez.no/ezcomponents>

php

אפליקציות פופולריות





PHP 4.x מת!

- 13/7/2007 - הכרזה על "סוף חיים" ל- PHP4

- בדיוק 3 שנים אחרי ש- PHP5 שוחרר

- החל מה-31/12/2007 רק עידכוני אבטחה קריטיים

- הפיתוח יפסק לחלוטין ב- 8/8/2008





העתיד: PHP 5.3 ו-6.0

• PHP 5.3 מתוכנן לתחילת 2008 (כנראה)

Namespaces –

Partial Unicode Support –

Late Static Binding –

Class::__callStatic() –

• PHP 6.0 ?

– “It's ready when it's ready”

– Full Unicode Support



העתידי: PHP 5.3 - 6.0

Namespaces: לפני... •

```
class Zend_Http_Client
{
    const METHOD_POST = 'POST';

    public function request($method)
    {
        // ...
    }

    static public function encodeFormData()
    {
        // ...
    }
}

$client = new Zend_Http_Client();
$client->request(Zend_Http_Client::METHOD_POST);
$body = Zend_Http_Client::encodeFormData();
```



העתיד: PHP 5.3 ו-6.0

```
namespace Zend::Http; // <--
class Client
{
    const METHOD_POST = 'POST';

    public function request($method)
    {
        // ...
    }

    static public function encodeFormData()
    {
        // ...
    }
}
```

• Namespaces : אחרים...

```
use Zend::Http::Client as Client; // <--
$client = new Client();
$client->request(Client::METHOD_POST);
$body = Client::encodeFormData();
```



שאלות?



This work is licensed under the Creative Commons Attribution-Share Alike 2.5 Israel License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/2.5/il/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.