

Introduction to PHP

ApacheCon Europe

London, England. October 23-35, 2000.

Rasmus Lerdorf <rasmus@php.net>

<http://conf.php.net/ac/>

- History and Numbers
- Characteristics
- Language Overview
- Advanced Concepts
- Future

A complete server-side dynamic application:

```
Hello <? echo "World" ?>
```

Output:

Hello World

A somewhat more useful example:

```
<? echo $HTTP_USER_AGENT ?>
```

Output:

Handling simple data coming from a form took something like this to do in C:

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <string.h>

#define ishex(x) (((x) >= '0' && (x) <= '9') || ((x) >= 'a' && (x) <= 'f') || \
                ((x) >= 'A' && (x) <= 'F'))

int htoi(char *s) {
    int    value;
    char   c;

    c = s[0];
    if(isupper(c)) c = tolower(c);
    value=(c >= '0' && c <= '9' ? c - '0' : c - 'a' + 10) * 16;

    c = s[1];
    if(isupper(c)) c = tolower(c);
    value += c >= '0' && c <= '9' ? c - '0' : c - 'a' + 10;

    return(value);
}

void main(int argc, char *argv[]) {
    char *params, *data, *dest, *s, *tmp;
    char *name, *age;

    puts("Content-type: text/html\r\n");
    puts("<html><header><title>Form Example</title></header>");
    puts("<body><h1>My Example Form</h1>");
    puts("<form action=\"form.cgi\" method=\"POST\">");
    puts("Name: <input type=\"text\" name=\"name\">");
    puts("Age: <input type=\"text\" name=\"age\">");
    puts("</form>");

    data = getenv("QUERY_STRING");
    if(data) {
        params = data; dest = data;
        while(*data) {
            if(*data=='+') *dest=' ';
            else if(*data == '%' && ishex(*(data+1)) && ishex(*(data+2))) {
                *dest = (char) htoi(data + 1);
                data+=2;
            } else *dest = *data;
            data++;
            dest++;
        }
        *dest = '\\0';
        s = strtok(params, "&");
        do {
            tmp = strchr(s, '=');
            if(tmp) {
```

```
        *tmp = '\0';
        if(!strcmp(s,"name")) name = tmp+1;
        else if(!strcmp(s,"age")) age = tmp+1;
    }
    while(s=strtok(NULL,"&"));

    printf("Hi %s, you are %s years old\n",name,age);
}
puts("</body></html>");
}
```

Perl became an obvious choice because it was made for text processing. The same thing in Perl using CGI.pm:

```
use CGI qw(:standard);
print header;
print start_html('Form Example'),
      h1('My Example Form'),
      start_form,
      "Name: ", textfield('name'),
      p,
      "Age: ", textfield('age'),
      p,
      submit,
      end_form;
if(param()) {
    print "Hi ",em(param('name')),
          "You are ",em(param('age')),
          " years old";
}
print end_html;
```

Much easier both to read and to write, at least to people with a bit of a programming background.

I wanted an HTML-centric approach. The same script in PHP became:

```
<html><header><title>Form Example</title></header>
<body><h1>My Example Form</h1>
<form action="form.phtml" method="POST">
Name: <input type="text" name="name">
Age: <input type="text" name="age">
</form>
<?if($name):>
Hi <?echo $name?>, you are <?echo $age?> years old
<?endif?>
</body></html>
```

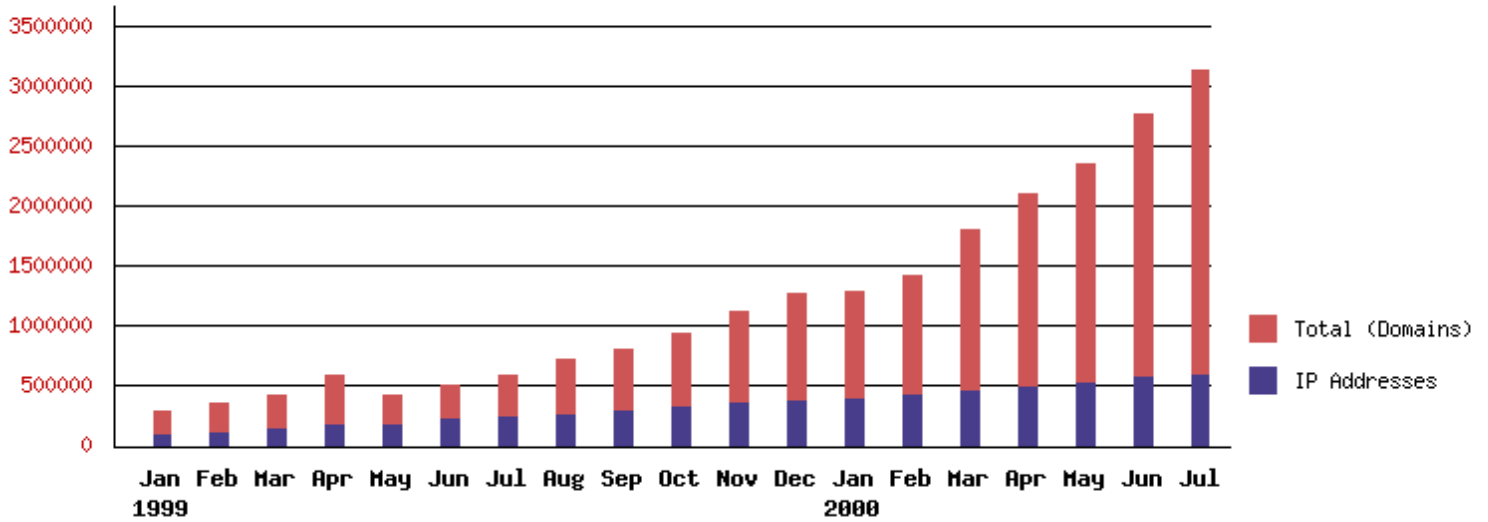
A block of raw HTML followed by the minimum amount of logic possible.

Dates

- Conceived in fall of 1994
- PHP Version 1 in spring 1995
- PHP Version 2 1995-1997
- PHP Version 3 1997-2000
- PHP Version 4 in Q2 2000

Development Team (162 people with CVS commit access)

- Zeev Suraski and Andi Gutmans in Israel
- Shane Caraveo in Florida
- Stig Bakken in Norway
- Andrei Zmievski in Lincoln, Nebraska
- Sascha Schumann in Dortmund, Germany
- Thies C. Arntzen in Hamburg, Germany
- Jim Winstead in Los Angeles
- Sam Ruby in Raleigh, NC
- Rasmus Lerdorf in San Francisco



Source: [Netcraft](#)
July 2000

3,121,918 Domains, **598,213** IP Addresses
18,169,498 Domains queried
PHP installed on **17.2%** of all domains

Source: [SecuritySpace.com](#)
July 2000
Apache Module Report

1,268,923 Apache Servers surveyed
418,011 (32.94%) PHP
248,388 (19.57%) Frontpage
126,503 (9.97%) mod_perl
112,304 (8.85%) mod_ssl
93,391 (7.36%) OpenSSL
43,086 (3.40%) ApacheJServ

Platforms (current):

- UNIX (all variants)
- Win32 (NT/W95/W98/W2000)
- QNX
- Mac (WebTen)
- OS/2
- BeOS

Platforms (on the way):

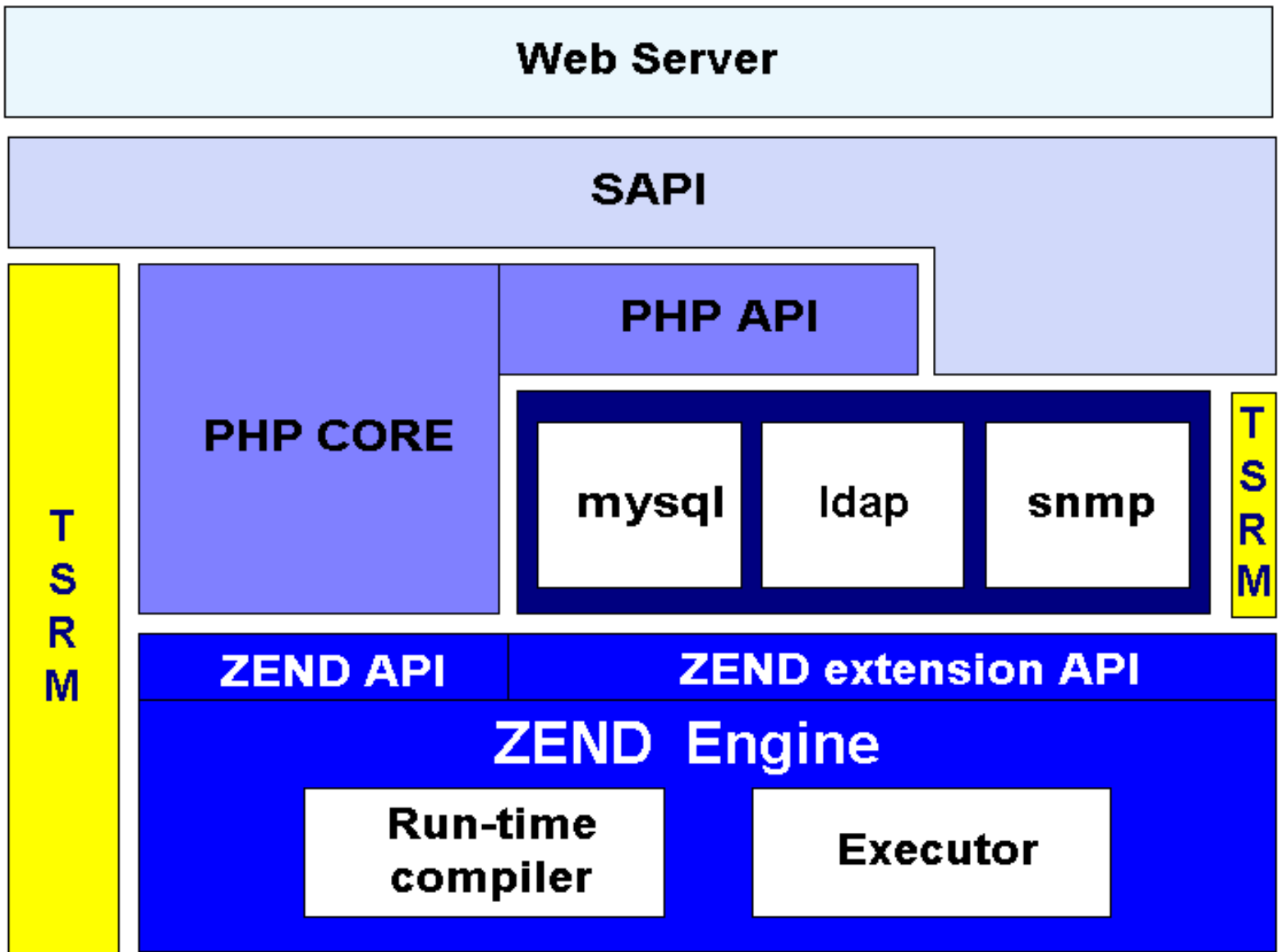
- OS/390
- AS/400

Server Interfaces:

- Apache module (UNIX, Win32)
- CGI
- fhttpd module
- ISAPI module (IIS, Zeus)
- NSAPI module (Netscape iPlanet)
- Java servlet
- AOLServer
- Roxen module

Server Interfaces (Planned):

- Apache 2.0 module
- WSAPI module (O'Reilly WebSite)
- phttpd module
- thttpd module



HTML Embedding

- SGML style: `<? code ?>`
- XML style: `<?php code ?>`
- ASP style: `<% code %>`
- Javascript style: `<script language="php"> code </script>`

```
<HTML><HEAD>
  <TITLE>Search results for "<?php print $query; ?>"</TITLE>
</HEAD>
<BODY>
```

Traditional CGI Programming

```
#!/usr/bin/perl

print "<HTML><HEAD>\n";
print " <TITLE>Search results for \"\$query\"</TITLE>\n";
print "</HEAD>\n";
print "<BODY>\n";
```

Syntax and switching modes

```
<? if(strpos($_SERVER['HTTP_USER_AGENT'], "MSIE")) { ?>  
<b>You are using Internet Explorer</b>  
<? } else { ?>  
<b>You are not using Internet Explorer</b>  
<? } ?>
```

Output:

You are not using Internet Explorer

Handling Forms

```
<form action="action.php" method="POST">
Your name: <input type="text" name="name"><br>
You age: <input type="text" name="age"><br>
<input type="submit">
</form>
```

Your name:

Your age:

action.php:

```
Hi <?echo $name?>.
You are <?echo $age?> years old.
```

C-Like Syntax

```
<?
for ($loop = -5; $loop < 5; $loop++) {
    if ($i < 0) {
        echo "-";
    } elseif ($i > 0) {
        echo "+";
    }
    echo "$loop<BR>\n";
}

while(--$loop) {
    switch($i % 2) {
        case 0:
            echo "Even<BR>\n";
            break;
        case 1:
            echo "Odd<BR>\n";
            break;
    }
}

do {
    echo "$loop<BR>";
} while (++$loop < 10);
?>
```

Basic data types

- numbers (integers and real)
- strings
- booleans

Dynamic typing

- Don't have to declare types
- Automatic conversion done

```
<? echo 5 + "1.5" + "10e2" ; ?>
```

Output:

1006.5

Ordered arrays

```
<?
$a[0] = 1;
$a[1] = "foo";
$a[] = 1.57;
?>
```

Associative arrays

```
<?
$catch_it['cat'] = "mouse";
$catch_it['dog'] = "cat";
?>
```

Manipulating

- **Sorting:** sort(), rsort(), ksort(), usort()
- **Traversal:** reset(), end(), next(), each(), current(), key()

Array Manipulation using built-in Array Cursor

```
<? $arr = array(1, 'foo', 1.57, 'cat'=>'mouse', 'dog'=>'mailman'); ?>
```

```
<?
while(list($k,$v) = each($arr)) {
    echo "\$arr[$k] = $v<br>\n";
}
?>
```

Output:

```
$arr[0] = 1
$arr[1] = foo
$arr[2] = 1.57
$arr[cat] = mouse
$arr[dog] = mailman
```

| | |
|----------------|----------------|
| current(\$arr) | 1 |
| next(\$arr) | foo |
| current(\$arr) | foo |
| prev(\$arr) | 1 |
| end(\$arr) | mailman |
| current(\$arr) | mailman |
| key(\$arr) | dog |
| reset(\$arr) | 1 |
| each(\$arr) | array(0,1) |
| each(\$arr) | array(1,'foo') |
| each(\$arr) | array(2,1.57) |

```
<?
function log_data($user, $data) {
    mysql_query("INSERT INTO userdata VALUES ('".
                uniqid()."', '$user', '$data')");
}
?>
```

Pass-by-reference

```
<? log_data($PHP_AUTH_USER, &$data); ?>
```

Default values

```
<?
function header($title="Default Title") {?>
    <HTML><HEAD><TITLE>
    <? echo $title ?>
    </TITLE></HEAD><BODY><?
}
?>
```

```
<?class Cart {
    var $items;
    function add_item($artnr, $num) {
        $this->items[$artnr] += $num;
    }
} ?>
```

Inheriting a class with a Constructor

```
<?class NamedCart extends Cart {
    var $owner;
    function NamedCart($name) {
        $this->owner = $name;
    }
} ?>
```

Invocation

```
<? $cart = new NamedCart("PenguinGear");
    $cart->add_item(170923, 2);?>
```

```
<? echo date("M d, Y H:i:s", time()); ?>
```

Sep 13, 2000 07:20:02

```
<?
$dday = mktime(0, 0, 0, 9, 15, 2000);
$diff = $dday - time();
$days = $diff / 86400;
if ($days >= 1) {
    printf( "Only %d day%s left!<BR>\n", $days, $days == 1 ? "" : "s");
} else {
    $hours = $diff / 3600;
    if ($hours >= 1) {
        printf( "Only %d hour%s left!<BR>\n", $hours, $hours == 1 ? "" : "s");
    } else {
        echo "The Olympics are here!<BR>\n";
    }
}
?>
```

Only 1 day left!

Calendar Conversions

- Converts between calendars (Julian, Gregorian, Mayan, etc)

substr

```
<?
  $str = "Fast String Manipulation";
  echo substr($str,0,4) . substr($str,-9);
?>
```

Fastipulation

explode

```
<?
  $a = explode(":", "This:string:has:delimiters.");
  while (list($value) = each($a)) {
    if (strcmp($value, "has") == 0) {
      echo "had ";
    } else echo $value." ";
  }
?>
```

This string had delimiters.

Posix-style

```
<?
echo ereg_replace('will be ([:alpha:]]+)',
                 'has been \1',
                 'This string will be modified.');
```

This string has been modified.

Perl-style

```
<?
echo preg_replace('/will be ([\w\s]+)/',
                 'has been \1',
                 'This string will be modified.');
```

This string has been modified.

HTML

```
<?echo htmlspecialchars("This & that are <problems>.");?>
```

This & that are <problems>.

Base64

```
<?
echo ($encoded = base64_encode(
    "A rose by any other name..."));
echo ($decoded = base64_decode($encoded));
?>
```

QSBYb3NIIGJ5IGFueSBvdGhlciBuYW11Li4u

A rose by any other name...

Others

MD5, DES, TripleDES, Blowfish, 3-WAY, SAFER-SK64, SAFER-SK128, TWOFISH, TEA, RC2, GOST in CBC, OFB, CFB and ECB cipher modes as well as RC6 and IDEA


```
<PRE><? readfile("/proc/cpuinfo");?></PRE>
```

Output:

```
processor           : 0
vendor_id          : GenuineIntel
cpu family         : 6
model              : 6
model name         : Mobile Pentium II
stepping           : 10
cpu MHz            : 397.053378
cache size         : 256 KB
fdiv_bug           : no
hlt_bug            : no
sep_bug            : no
f00f_bug           : no
coma_bug           : no
fpu                : yes
fpu_exception      : yes
cpuid level        : 2
wp                 : yes
flags               : fpu vme de pse tsc msr pae mce cx8 sep mtrr pge mca cmov pat pse36
mmx fxsr
bogomips           : 792.99
```

Reading from a file

```
<?
$file = fopen("sample.txt", "r");
while (!feof($file)) {
    echo fgets($file, 1024), "<BR>";
}
?>
```

Reading from a URL

```
<? $file = fopen("http://www.php.net/file.txt", "r"); ?>
```

Writing to a file

```
<?
$file = fopen("agent.log", "a");
fputs($file, $_HTTP_USER_AGENT."\n");
?>
```

Guestbook Example

A very simple guestbook example to illustrate basic file handling.

```
<html><head><title>My Guestbook</title></head>
<body>
<h1>Welcome to my Guestbook</h1>
<h2>Please write me a little note below</h2>
<form action="<?echo $PHP_SELF?>" method="POST">
<textarea cols=40 rows=5 name=note wrap=virtual></textarea>
<input type=submit value=" Send it ">
</form>
<?if(isset($note)) {
    $fp = fopen("/tmp/notes.txt", "a");
    fputs($fp, nl2br($note). '<br>');
    fclose($fp);
}
?><h2>The entries so far:</h2>
<? @ReadFile("/tmp/notes.txt") ?>
</body></html>
```

This is what it looks like: [Guestbook Example](#)

Creating a PNG with a TrueType font

```
<?
Header("Content-type: image/png");
$im = ImageCreate(630,80);
$blue = ImageColorAllocate($im,0x5B,0x69,0xA6);
$white = ImageColorAllocate($im,255,255,255);
$black = ImageColorAllocate($im,0,0,0);
ImageTTFText($im, 60, 0, 10, 55, $black, "CANDY.TTF", $text);
ImageTTFText($im, 60, 0, 6, 52, $white, "CANDY.TTF", $text);
ImagePNG($im);
?>

<IMG src="txt.php?text=<?echo urlencode($text)?>">
```

CreateFrom and Bounding Box Math

```
<?
Header("Content-type: image/png");
$font = "php.ttf";
if(!$si) $si = 80;
$im = ImageCreateFromPNG("php-blank.png");
$tsize = imagettfbbox($si,0,$font,$text);
$dx = abs($tsize[2]-$tsize[0]);
$dy = abs($tsize[5]-$tsize[3]);
$x = ( imagesx($im) - $dx ) / 2;
$y = ( imagesy($im) - $dy ) / 2 + 3*$dy/4;
$blue = ImageColorAllocate($im,0x5B,0x69,0xA6);
$white = ImageColorAllocate($im,255,255,255);
$black = ImageColorAllocate($im,0,0,0);
ImageTTFText($im, $si, 0, $x, $y, $white, $font, $text);
ImageTTFText($im, $si, 0, $x+2, $y, $white, $font, $text);
ImageTTFText($im, $si, 0, $x, $y+2, $white, $font, $text);
ImageTTFText($im, $si, 0, $x+2, $y+2, $white, $font, $text);
ImageTTFText($im, $si, 0, $x+1, $y+1, $black, $font, $text);
ImagePNG($im);
?>

<IMG src="txt2.php?text=<?echo urlencode($text)?>&si=<?echo $si?>">
```

Text:

Size:

```
<?
Header("Content-type: application/pdf");
$pdf = PDF_open();
pdf_set_info_author($pdf, "Uwe Steinmann");
PDF_set_info_title($pdf, "Test for PHP wrapper of PDFlib 3.0.1");
PDF_set_info_author($pdf, "Name of Author");
pdf_set_info_creator($pdf, "See Author");
pdf_set_info_subject($pdf, "Testing");
PDF_begin_page($pdf, 595, 842);
PDF_add_outline($pdf, "Page 1");
pdf_set_font($pdf, "Times-Roman", 30, "winansi");
pdf_set_text_rendering($pdf, 1);
PDF_show_xy($pdf, "Times Roman outlined", 50, 750);
pdf_moveto($pdf, 50, 740);
pdf_lineto($pdf, 330, 740);
pdf_stroke($pdf);
PDF_end_page($pdf);
PDF_close($pdf);
?>
```

[Click here to run above test script](#)

[And here is a cool dynamic clock PDF document](#)

- GD (GIF, JPEG, PNG, WBMP)
- LDAP
- SNMP
- IMAP (POP, NNTP)
- FTP
- MCAL
- IMSP
- IPTC
- BC (arbitrary precision math)
- Hyperwave
- XML parser
- PDF generation
- FDF (PDF forms)
- System V Semaphores and Shared memory
- DCOM (Win32 only)
- Java Connectivity
- gettext (GNU internationalization)
- zlib (compressed IO)
- Charset/text conversion (UTF-8, Cyrillic, Hebrew)
- Browser capabilities extension
- EXIF
- SWF (Flash)
- ASPELL/PSPELL
- MCRYPT
- Cybercash
- Recode
- Readline
- WDDX
- NIS
- YAZ (Z39.50 client)
- Payflow Pro
- CCVS (Credit Card Verification System)

```
<?
    $frame = new Java("java.awt.Frame", "PHP 4");
    $button = new Java("java.awt.Button", "Hello World");
    $frame->add("North", $button);
    $frame->validate();
    $frame->pack();
    $frame->visible = True;

    $thread = new Java("java.lang.Thread");
    $thread->sleep(10000);

    $frame->dispose();
?>
```



```
<?
  session_start();
  if (!session_is_registered("mySession")) {
    session_register("mySession");
  }
?>
```

Session Data Backends

```
<?
  session_set_save_handler("myOpen", "myClose",
                          "myRead", "myWrite",
                          "myDestroy", "myGC");
  session_module_name("mm");

  session_module_name("files");
?>
```

SQL

- Adabas D
- Empress
- IBM DB2
- Informix
- Ingres
- Interbase
- Frontbase
- mSQL
- Direct MS-SQL
- MySQL
- ODBC
- Oracle (OCI7,OCI8)
- PostgreSQL
- Raima Velocis
- Solid
- Sybase

Others

- dBase
- filePro (read-only)
- dbm (ndbm, gdbm, Berkeley db)

PHP scripts that talk to databases all look similar to the code below. Connect to the database, select a database, send a query and loop through the results.

```
<?
mysql_pconnect("db.server.com", "username", "password");
mysql_select_db("products");
$result = mysql_query("SELECT * FROM details");
if ($result) {
    echo "<TABLE>\n";
    echo "<TR><TH>Name</TH><TH>Description</TH></TR>\n";
    while ($a = mysql_fetch_array($result)) {
        echo "<TR><TD>$a[name]</TD> ", "<TD>$a[descr]</TD></TR>";
    }
    echo "</TABLE>";
} else {
    echo "<P>Nothing to see here.";
}
?>
```

SQL'izing the Guestbook Example

Recall our file-driven guestbook example from earlier. We are going to convert this into an SQL-driven guestbook by first creating a database, then a schema for the table where we will store the data and then we will modify the code.

```
<h1>Welcome to my Guestbook</h1>
<h2>Please write me a little note below</h2>
<form action="<? echo $PHP_SELF?>" method="POST">
<textarea cols=40 rows=5 name="note" wrap=virtual></textarea>
<input type="submit" value=" Send it ">
</form>
<?
if(isset($note)) {
    $fp = fopen("notes.txt", "a");
    fputs($fp, nl2br($note). "<br>");
    fclose($fp);
}
?>
<h2>The entries so far:</h2>
<? @ReadFile("notes.txt") ?>
```

Create a database

```
mysqladmin create mydb
```

Create a Schema

```
CREATE TABLE comments (
    id int(8) DEFAULT '0' NOT NULL auto_increment,
    comment text,
    ts datetime,
    PRIMARY KEY (id)
);
```

SQL'izing the Guestbook Example

Here we add the necessary code to store our guestbook comments in an SQL database

```
<html><head><title>My Guestbook</title></head>
<body>
<h1>Welcome to my Guestbook</h1>
<h2>Please write me a little note below</h2>
<form action="<? echo $PHP_SELF?>" method="POST">
<textarea cols=40 rows=5 name="note" wrap=virtual></textarea>
<input type="submit" value=" Send it ">
</form>
<?
mysql_connect("localhost");
if(isset($note)) {
    $ts = date("Y-m-d H:i:s");
    mysql_db_query("mydb","insert into comments values (0,'$note','$ts')");
}
?>
<h2>The entries so far:</h2>
<? $result = mysql_db_query("mydb","select * from comments");
    while($row=mysql_fetch_row($result)) {
        echo $row[0] . " " . $row[1] . " " . $row[2] . "<br>\n";
    } ?>
</body></html>
```

This is what it looks like: [SQL Guestbook Example](#)

A database abstraction layer is bundled with PHP 4. In the example below, the only thing you would need to change to use a different database is the **odbc** word on the third line.

```
<?
require_once 'DB';
$db = DB::connect('odbc://rasmus:pw@host/my_db');
$stmt = $db->prepare('select * from my_table');
$result = $db->execute($stmt);
while($row = $db->fetchrow($result)) {
    while(list($key, $value) = each($row)) {
        echo "$field: $value<br>\n";
    }
}
$db->disconnect();
?>
```

- Better OO (syntax fixes, op. overloading, superclass ref.)
- Modular, self-contained extensions
- Database abstraction layer (ala Perl's DBI)
- Reference counting
- Thread safe core
- More server modules through SAPI
- foreach()
- Output buffering
- Support for a real debugger
- Support for caching compiled code
- Support for an optimizer
- Java ties

New in Version 4

Resources



Home Page: <http://www.php.net>

These Slides: <http://conf.php.net>

Manual: <http://www.php.net/manual/>

Tutorial: <http://www.php.net/tut.php>

Books: <http://www.php.net/books.php>

