

ΜΟΝΑΔΑ ΑΡΙΣΤΕΙΑΣ
ΑΝΟΙΧΤΟΥ ΛΟΓΙΣΜΙΚΟΥ

ΧΑΡΟΚΟΠΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ



ΜΟΝΑΔΕΣ ΑΡΙΣΤΕΙΑΣ
ΑΝΟΙΧΤΟΥ ΛΟΓΙΣΜΙΚΟΥ

RHP



Τι είναι η PHP



- Η PHP (PHP: Hypertext Preprocessor) είναι μια γλώσσα προγραμματισμού ανοικτού λογισμικού ειδικά σχεδιασμένη για δημιουργία σελίδων Web με δυναμικό περιεχόμενο.
- Ο κώδικας PHP ενσωματώνεται μέσα σε HTML σελίδες και εκτελείται από τον Server (που είναι αποθηκευμένες) κάθε φορά που ο χρήστης επισκέπτεται τις σελίδες αυτές.
- Η εκτέλεση ενός κώδικα PHP έχει ως αποτέλεσμα τη παραγωγή κώδικα HTML ο οποίος στέλνεται τελικά στο φυλλομετρητή του χρήστη.

Ιστορία της PHP

- Ο Rasmus Lerdorf το 1995 έγραψε ένα Perl script με σκοπό να διατηρεί μια λίστα στατιστικών για τα άτομα που έβλεπαν το online βιογραφικό του σημείωμα.
- Η γλώσσα αρχικά ονομαζόταν PHP/FI από τα αρχικά Personal Home Page/Form Interpreter.
- Το 1997 η PHP/FI έφθασε στην έκδοση 2.0, βασιζόμενη αυτή τη φορά στη γλώσσα C και αριθμώντας περισσότερους από 50.000 ιστότοπους.
- Οι Andi Gutmans και Zeev Suraski το 1997 ξαναέγραψαν τη γλώσσα από την αρχή δημιουργώντας την εταιρεία Zend (από τα αρχικά των ονομάτων τους), η οποία συνεχίζει μέχρι και σήμερα την ανάπτυξη και εξέλιξη της γλώσσας PHP.
- Τρέχουσα έκδοση: **5.4.33** [18 Σεπτεμβρίου 2014]



Server side scripting languages

- ASP/ASP.NET (*.asp/*.aspx)
- Java via JavaServer Pages (*.jsp)
- PHP (*.php)
- Python
- Ruby
- Perl CGI (*.cgi, *.ipl, *.pl)
- ColdFusion Markup Language (*.cfm)
- και άλλες όχι τόσο διαδεδομένες

© W3Techs.com	usage	change since 1 August 2014
1. PHP	82.0%	-0.2%
2. ASP.NET	17.3%	+0.1%
3. Java	2.7%	
4. ColdFusion	0.7%	-0.1%
5. Perl	0.6%	

percentages of sites

πηγή: <http://w3techs.com/>

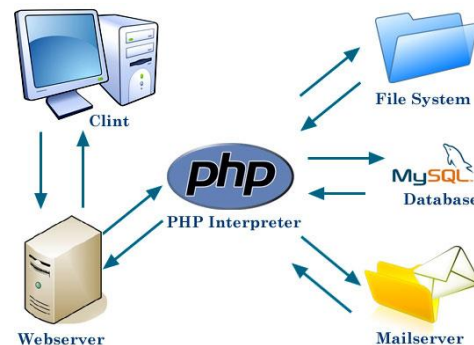
Πλεονεκτήματα

- Υψηλή απόδοση.
- Διασυνδέσεις με πολλά διαφορετικά συστήματα βάσεων δεδομένων (MySQL, Informix, Oracle, Sybase, Solid, PostgreSQL, Generic ODBC, κ.ά.).
- Ενσωματωμένες βιβλιοθήκες για πολλές συνηθισμένες Web διαδικασίες.
- Χαμηλό κόστος.
- Ευκολία μάθησης και χρήσης.
- Μεταφερσιμότητα.
- Διαθεσιμότητα του πηγαίου κώδικα.
- Αρκετά frameworks με καλό documentation



Βασικές αρχές λειτουργίας

- Η PHP τρέχει σε ένα διακομιστή.
- Ο Web browser ενός χρήστη κάνει μία http:// αίτηση για μια συγκεκριμένη php Web σελίδα. (πχ. <http://www.hua.gr/index.php>)
- Ο Web διακομιστής λαμβάνει την αίτηση για την σελίδα php, ανακαλεί το αρχείο και το περνά στην μηχανή php για επεξεργασία.
- Η μηχανή php αρχίζει την ανάλυση του php κώδικα επικοινωνώντας αν υπάρχει ανάγκη με τον διακομιστή της βάσης δεδομένων.
- Η μηχανή php σταματά την εκτέλεση του προγράμματος επιστρέφοντας στον Web διακομιστή την τελική HTML σελίδα.
- Ο Web διακομιστής περνά την html σελίδα ξανά στο Web browser όπου ο χρήστης μπορεί να διαβάσει το αποτέλεσμα της php σελίδας.



Γράφοντας PHP

Για ν' αναγνωρίσει η PHP ότι ένα κομμάτι κώδικα μέσα σε μία σελίδα HTML είναι PHP πρέπει να περικλείεται ανάμεσα σε εισαγωγικά της μορφής `<?php ... ?>` ή της μορφής `<? ... ?>`. Προτείνεται το πρώτο για λόγους συμβατότητας.

```
<html>
<body>
<strong >
<?php
echo " Hello , " . " <i > " . " World ! " . " </ i > " ;
?>
</strong >
</body>
</html>
```

Online PHP compilers : http://www.compileonline.com/execute_php_online.php
<http://sandbox.onlinephpfunctions.com/>



Σχόλια

Στην PHP χρησιμοποιούμε // για σχόλια σε μια γραμμή και /* και */ για να εισάγουμε ένα μεγάλο μπλοκ σχολίων.

```
<html>
<body>
<?php
// This is a comment
/*
This is
a comment
block
*/
?>
</body>
</html>
```

Μεταβλητές

Οι μεταβλητές χρησιμοποιούνται για αποθήκευση τιμών, όπως αλφαριθμητικά, αριθμοί ή πίνακες. Όταν μια μεταβλητή ορίζεται, μπορεί να χρησιμοποιηθεί πολλές φορές στο script. Όλες οι μεταβλητές στην PHP ξεκινούν με το σύμβολο \$. Ο σωστός τρόπος για να ορίσουμε μια μεταβλητή είναι:

```
$var_name = value ;
```

- Στην PHP μια μεταβλητή δεν χρειάζεται να δηλωθεί πριν χρησιμοποιηθεί.
- Στο προηγούμενο παράδειγμα είδαμε ότι δεν χρειάστηκε να πούμε τι τύπο δεδομένα θα φυλάσσει η μεταβλητή.
- Η PHP μετατρέπει αυτόματα μια μεταβλητή στο σωστό τύπο δεδομένων, αναλόγως με το πως τη χρησιμοποιούμε.

Με τη συνάρτηση **print_r()** τυπώνει την τιμή μιας μεταβλητής σε human readable μορφή **var_dump()** μπορούμε να τυπώσουμε τόσο την τιμή μιας μεταβλητής όσο και τον τύπο της.

Concatenation

Η συνηθέστερη μέθοδος για τα string στην PHP: η ένωση (concatenation). Για να ενώσουμε δύο μεταβλητές, χρησιμοποιούμε την τελεία (.)

Θα έχει σαν αποτέλεσμα το Hello World 1234
Άλλες συναρτήσεις για string είναι:

- strlen()
- strpos()
- strcmp() και άλλες!

```
<?php  
$txt1= "Hello World";  
$txt2= "1234";  
echo $txt1 . " " . $txt2;  
?>
```

Πίνακες

Οι πίνακες πάντα έχουν: key και value. Το key αναφέρεται στο value. Υπάρχουν 3 ειδών πίνακες:

Associative, Numeric, και Multidimensional.

```
\\ index automatically assigned
$cars= array ( " Saab " , " Volvo " , "BMW" , " Toyota " ) ;
\\ index manually assigned
<?php
$cars[0] = "Saab";
$cars[1] = "Volvo";
$cars[2] = "BMW";
$cars[3] = "Toyota";
echo $cars[0] . " and " . $cars[1] . " are Swedish cars . ";
?>
\\Associative Arrays
$ages = array("Peter"=>32, "Quagmire"=>30, "Joe"=>34);
<?php
$ages['Peter'] = "32";
$ages['Quagmire'] = "30";
$ages['Joe'] = "34";
echo " Peter is " . $ages['Peter'] . " years old.";
?>
```

Δομές Ελέγχου : If

```
<?php
date_default_timezone_set ("Europe/ Athens");
$t=date("H") ;
if ($t<"10")
{
echo "Have a good morning!";
}
elseif ($t<"20")
{
echo "Have a good day!";
}
else
{
echo "Have a good night!" ;
}
?>
```

Δομές Ελέγχου : while

```
<html>
<body>
<?php
$i=0;
while ($i <= 5)
{ echo "Number : " . $i . "<br/>";
$i++;
}
?>
</body>
</html>
```

```
<html>
<body>
<?php
$i=1;
do
{
  $i++;
  echo "The number is " . $i . "<br/>";
}
while ($i <= 5);
?>
</body>
</html>
```

Δομές Ελέγχου : for - foreach

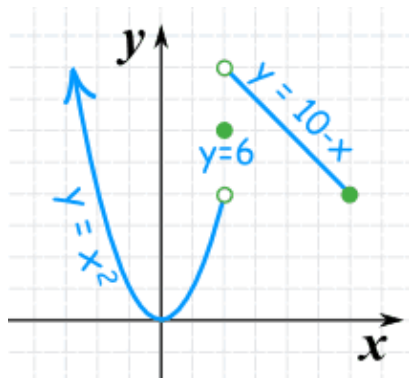
```
<html>
<body>
<?php
for ( $i =1; $i <=5; $ i ++ )
{
echo "Hello World! <br/>";
}
?>
</body>
</html>
```

```
<html>
<body>
<?php
$x=array ( "one", "two", "three" ) ;
foreach ( $x as $value )
{
echo $value . "<br/>";
}
?>
</body>
</html>
```

Δομές Ελέγχου : switch

```
<?php
$favcolor= "red" ;
switch ($favcolor)
{
case "red":
echo "Your favorite color is red!";
break;
case "blue":
echo "Your favorite color is blue!";
break;
case "green":
echo "Your favorite color is green!";
break;
default:
echo "Your favorite color is neither red, blue , or green!";
}
?>
```

Συναρτήσεις



```
<html>
<body>
<?php
function add($x,$y)
{ $total = $x + $y ;
return $total;
}
echo " 1 + 16 = " . add(1,16);
?>
</body>
</html>
```

Συναρτήσεις: Εμβέλεια

Πεδίο δράσης (εμβέλεια) Μεταβλητών

- Τοπικές Μεταβλητές
- Καθολικές Μεταβλητές

```
<html>
<body>
<?php
function add($x,$y)
{
    global $z; // Global variable.
    $total = $x + $y + $z;
    return $total;
}
$z = 5;
echo " 1 + 16 = " . add(1,16);
?>
</body>
</html>
```

Φόρμες

Επιτρέπουν στους χρήστες να εισάγουν δεδομένα

Input tags:

- Text Fields
- Radio Buttons
- Checkboxes

Οι PHP μεταβλητές **`$_GET`** και **`$_POST`** χρησιμοποιούνται για να ανακτήσουν δεδομένα από φόρμες που συμπληρώνονται από χρήστες.

```
<html>
<body>
<form action="welcome.php" method="get">
Name: <input type="text" name="name"/>
Age : <input type="text" name="age"/>
<input type="submit"/ >
</form>
</body>
</html>
```

```
<html>
<body>
Welcome <?php echo $_POST["name"]; ?> . <br/>
You are <?php echo $_POST["age"]; ?> years old.
</body>
</html>
```

PHP \$_GET

- Η μεταβλητή \$_GET είναι ένας πίνακας από ονόματα μεταβλητών και τιμές που στέλνονται μέσω της μεθόδου HTTP GET.
- Χρησιμοποιείται για να συλλέξει τιμές από μια φόρμα που χρησιμοποιεί τη μέθοδο get. Η πληροφορία που στέλνεται μέσω της HTTP GET μεθόδου είναι ορατή στον καθένα (φαίνεται στο address bar του browser!) και υπάρχει περιορισμός στην ποσότητα πληροφορίας που μπορεί να αποσταλλεί (max 2000 χαρακτήρες).
- Προσοχή! Δεν μπορούμε να στέλνουμε passwords ή άλλη ευαίσθητη πληροφορία με τη GET. Όμως όταν θέλουμε να θέσουμε έναν σελιδοδείκτη είναι χρήσιμη η GET.

PHP \$_POST

- Η μεταβλητή \$_POST είναι ένας πίνακας από ονόματα μεταβλητών και τιμές που στέλνονται μέσω της μεθόδου HTTP POST.
- Χρησιμοποιείται για να συλλέξει τιμές από μια φόρμα που χρησιμοποιεί τη μέθοδο post. Η πληροφορία που στέλνεται μέσω της POST μεθόδου είναι αόρατη και δεν έχει όριο μεγέθους για την πληροφορία που αποστέλλεται. (Καθορίζεται μέσω της μεταβλητής post_max_size στο αρχείο php.ini)

```
<html>
<body>
<form action="welcome.php" method="post">
Name: <input type="text" name="name"/>
Age : <input type="text" name="age"/>
<input type="submit"/>
</form>
</body>
</html>
```

```
<html>
<body>
Welcome <?php echo $_POST["name"]; ?> . <br/>
You are <?php echo $_POST["age"]; ?> years old.
</body>
</html>
```

PHP \$_REQUEST

- Η μεταβλητή \$_REQUEST είναι ένας πίνακας που κρατά τα περιεχόμενα των πινάκων \$_GET, \$_POST και \$_COOKIE.
- Χρησιμοποιείται για να συλλέξει τιμές από μια φόρμα που χρησιμοποιεί τη μέθοδο get και post.

```
<html>
<body>
Welcome <?php echo $_REQUEST["name"]; ?>!<br/>
You are <?php echo $_REQUEST["age"]; ?> years old.
</body>
</html>
```

Έλεγχος τιμών των μεταβλητών

Για να ελέγξετε αν μία μεταβλητή έχει πάρει τιμή κάντε χρήση της συνάρτησης `empty()`. Η `empty()` ελέγχει αν η μεταβλητή έχει σαν τιμή (`empty string` , `0`, `NULL` ή `False`) . Επιστρέφει `FALSE` αν η μεταβλητή έχει `non-empty` ή `non-zero` τιμή.

Η `isset()` ελέγχει αν η μεταβλητή έχει πάρει τιμή, η οποία μπορεί να είναι ακόμη και (`False` , `0` ή `Empty string`) , αλλά όχι `NULL`. Επιστρέφει `TRUE` αν η μεταβλητή υπάρχει και `FALSE` σε οποιαδήποτε άλλη περίπτωση.

```
<?php
$var = 0 ;
//Evaluates to true because $var is empty
if (empty($var)) {
echo '$var is either 0, empty, or not set at all'; }
// Evaluates as true because $var isset
if (isset($var)) {
echo ' $var is set even though it is empty'; }
?>
```

Με την εντολή `unset()` μπορείτε να ελευθερώσετε μια μεταβλητή `$_POST` από την τιμή που έχει πάρει.

```
unset ( $_POST['variable_name'] );
```

Cookies

Τα cookies χρησιμοποιούνται για να αναγνωρίζουμε χρήστες. Είναι ένα μικρό αρχείο το οποίο ο server ενσωματώνει στον υπολογιστή του χρήστη. Κάθε φορά που ο ίδιος υπολογιστής αιτείται μια σελίδα μέσω του browser, στέλνεται και το cookie μαζί. Με την php μπορούμε να δημιουργούμε και να ανακτούμε τις τιμές ενός cookie.

```
setcookie(name,value,expire,path,domain);
```

Πρέπει να μπει πριν το <html> tag

```
<?php  
setcookie("user","Alex Porter",time () + 3600);  
?>  
<html>  
....
```



Ανάκτηση τιμής ενός cookie

Χρησιμοποιείται η PHP μεταβλητή \$_COOKIE

```
<html>
<body>
<?php
if (isset($_COOKIE["user"]))
echo "Welcome" . $_COOKIE["user"] . "!"<br/>";
else
echo "Welcome guest!"<br/>";
// A way to view all cookies
print_r($_COOKIE);
?>
</body>
</html>
```



Διαγραφή cookie

```
setcookie("user","",time() -
3600);
```

Sessions



Η session (σύνοδος) αποτελεί τον τρόπο με τον οποίο μπορούμε να αποθηκεύσουμε πληροφορία (με τη μορφή μεταβλητών) ώστε να μπορεί να χρησιμοποιηθεί μεταξύ πολλών σελίδων (username, shopping items κτλ).

Αντίθετα με τα cookies, η σχετική πληροφορία των μεταβλητών δεν αποθηκεύεται στον υπολογιστή των χρηστών. Επίσης διαφέρει από τις υπόλοιπες μεταβλητές με την έννοια ότι δεν τις περνάμε ξεχωριστά σε κάθε σελίδα, αλλά τις ανακτούμε από τη σύνοδο (session), την οποία ανοίγουμε στην αρχή κάθε σελίδας. Ωστόσο, αυτού του είδους οι πληροφορίες είναι προσωρινές και συνήθως διαγράφονται πολύ σύντομα μόλις ο χρήστης αφήσει το website που χρησιμοποιεί sessions.

session start

Για να μπορέσουμε να αποθηκεύσουμε πληροφορίες στη σύνοδο, πρέπει πρώτα απ' όλα να ξεκινήσουμε τη σύνοδο, και αυτό γίνεται όσο πιο νωρίς είναι δυνατό στον κώδικα, πριν οποιοδήποτε HTML κείμενο σταλεί.

```
<?php
session_start ( ) ;
$_SESSION['views']=1; // store session data
echo "Pageviews = " . $_SESSION['views']; //retrieve data
?>
```

το αποτέλεσμα του παραπάνω κώδικα είναι: Pageviews = 1
Προκειμένου να πάρουμε την τιμή μιας μεταβλητής συνόδου, καλό θα ήταν να δούμε πρώτα αν έχει πάρει τιμή. Αυτό γίνεται με τη συνάρτηση isset

```
<?php
session_start();
if (isset($_SESSION['views']))
$_SESSION['views']=$_SESSION['views'] + 1;
else
$_SESSION['views']=1;
echo "views = " . $_SESSION['views'];
?>
```

αντίστοιχα υπάρχει και η
συνάρτηση unset.

session destroy

Αν θέλουμε να διαγράψουμε ορισμένα δεδομένα από τη σύνοδο, μπορούμε να χρησιμοποιήσουμε την unset()

```
<?php  
unset($_SESSION['views']);  
?>
```

Μπορούμε να τελειώσουμε μια σύνοδο, χρησιμοποιώντας τη συνάρτηση session_destroy.

```
<?php  
session_start();  
session_destroy();  
?>
```



Όταν γίνει αυτό, σημαίνει ότι όλα τα δεδομένα που αποθηκευτηκαν στη σύνοδο θα χαθούν!

include - require

Μπορούμε να εισάγουμε τα περιεχόμενα ενός PHP αρχείου σε ένα άλλο, πριν ο server ξεκινήσει να τα εκτελεί. Αυτό γίνεται με τις:

- include() σε περίπτωση σφάλματος δημιουργεί μια προειδοποίηση, αλλά ο κώδικας συνεχίζεται να εκτελείται
- require() σε περίπτωση σφάλματος δημιουργεί ένα fatal error και ο κώδικας σταματάει να εκτελείται

Που χρησιμοποιούνται;

```
<?php include("header.php");?>  
<h1>Welcome to my home page!</h1>  
<p>Some text.</p>
```

Error Handling

Η διαχείριση σφαλμάτων είναι σημαντικό μέρος της ανάπτυξης web εφαρμογών.

Αν ο κώδικας δεν ελέγχει για τυχόν σφάλματα, υπάρχει πιθανότητα να προκύψουν θέματα ασφάλειας δεδομένων. Μέθοδοι αντιμετώπισης σφαλμάτων:

- δηλώσεις τύπου "die()"
- αναφορά σφάλματος
- καθορισμός τύπων σφαλμάτων



```
<?php
$file=fopen("welcome.txt","r");
?>
```

```
Warning : fopen (welcome.txt) [function.fopen]:failed to open stream
:
```

Warning: fopen() [function.fopen]: failed to open stream: No such file or directory in C:\xampp\htdocs\test.php on line 2

```
<?php
if (!file_exists("welcome.txt")) { die ("File not found"); }
else
{ $file=fopen("welcome.txt","r"); } ?>
```

File not found

Debugging



Τι κάνουμε για debugging!

- Κοιτάμε τον κώδικα!
- Κοιτάμε το log του server (σε unix συνήθως είναι στο /var/log/apache2/error.log).
- Χρησιμοποιούμε echo, print_r, var_dump για να δούμε αν μεταβλητές ή αντικείμενα έχουν τις τιμές που θέλουμε.
- Ελέγχουμε αν όλα τα απαραίτητα πακέτα είναι εγκατεστημένα στον server.
- Χρησιμοποιούμε το logging σύστημα του framework.

PHP Class

Οι κλάσεις αποτελούν τη δομή των δεδομένων και των ενεργειών και χρησιμοποιούν αυτές τις πληροφορίες για να χτίσουν αντικείμενα. Πάνω από ένα αντικείμενα μπορεί να δημιουργηθεί από μια κλάση, καθένα από τα οποία είναι ανεξάρτητα μεταξύ τους.

```
<?php
class MyClass
{
    public $prop1 = "I'm a class property!";
    public function setProperty($newval)
    { $this->prop1 = $newval; }
    public function getProperty()
    { return $this->prop1 . "<br/>"; }
}
$obj = new MyClass;
echo $obj->prop1;  ?>
```

Οι κλάσεις μπορούν να έχουν properties, methods, constructors, destructors.

Constructors- Destructors

```
<?php
class MyClass
{
    public $prop1 = "I'm a class property!";
    public function __construct()
    { echo 'The class "', __CLASS__, '" was
initiated!<br />'; }
    public function setProperty($newval)
    { $this->prop1 = $newval; }
    public function getProperty()
    { return $this->prop1 . "<br />"; }
}

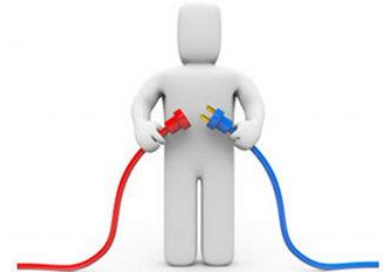
// Create a new object
$obj = new MyClass;
// Get the value of $prop1
echo $obj->getProperty();
// Output a message at the end of the file
echo "End of file.<br />";
?>
```

Όταν ένα αντικείμενο δημιουργείται, πολλές φορές χρειάζεται να γίνουν κάποια πράγματα. Για το σκοπό αυτό η PHP παρέχει τη μέθοδο **__construct()** η οποία καλείται αυτόματα κάθε φορά που δημιουργείται ένα αντικείμενο.

Επίσης όταν «καταστρέφεται» ένα αντικείμενο μπορούμε να «τακτοποιούμε» (π.χ. να κλείνουμε μια σύνδεση με τη βάση) πράγματα χρησιμοποιώντας τη μέθοδο **__destruct()**.

Σύνδεση με βάση δεδομένων

```
<?php
$con = mysql_connect("localhost","user","password");
if (!$con)
{
    die('Could not connect: ' . mysql_error());
}
?>
```



Σε περίπτωση αποτυχημένης σύνδεσης ο χρήστης πρέπει να ειδοποιείται.

Δημιουργία βάσης - εισαγωγή πίνακα

```
<?php
    $con = mysql_connect("localhost","user","
password");
    if (!$con)
    {
        die('Could not connect: ' . mysql_error());
    }

    if (mysql_query("CREATE DATABASE my_db",$con))
    {
        echo "Database created";
    }
    else
    {
        echo "Error creating database: ".mysql_error();
    }

    mysql_close($con);
?>
```

```
<?php
    $con = mysql_connect("localhost","user","
password");
    // Select the database
    mysql_select_db("my_db", $con);

    // Create the table
    $sql = "CREATE TABLE phone_book
    (
        personID int NOT NULL,
        PRIMARY KEY(personID),
        LastName varchar(20) NOT NULL,
        FirstName varchar(20),
        Address varchar(30),
        Age int,
        Phone varchar(10) )";

    mysql_query($sql,$con);

    mysql_close($con);
?>
```

Επιλογή δεδομένων από πίνακα

```
<?php
    // Select the database
    mysql_select_db("my_db", mysql_connect
("localhost","user","password"));

    $result = mysql_query("SELECT * FROM phone_book
WHERE age>30");

    while($row = mysql_fetch_array($result))
    {
        echo $row['LastName'] . " " . $row['FirstName'] . "
" .
            $row['Age'] . "<br>";
    }
?>
```