https://stackoverflow.com/questions/54494066/difference-between-create-table-and-create-any-table-privileges

Η . (τελεία) κάνει την συνένωση (αντί για το +)
Δουλεύει και με , (κόμμα)

μέχρι το σφάλμα, εκτελείται ο κώδικας.

μεταβλητές με $ και μετά γράμμα. ΟΧΙ αριθμός.
Υπάρχει και η &_ αλλά είναι για ειδική μεταχείριση

ΕΜΒΕΛΕΙΑ
καθολική ΕΚΤΟΣ προγράμματος
τοπική ΜΕΣΑ στην συνάρτηση

ΣΥΝΑΡΤΗΣΗ
function myTest(){

}

Σε συνάρτηση ΔΕΝ μπορώ να αναφερθώ σε καθολική μεταβλητή

Στο πρόραμμα ΔΕΝ μπορώ να αναφερθώσε μεταβλητή που βρίσκεται εντός συνάρτησης

Μέσα σε συνάρτηση δηλώνω global για να έχω τις συναρτήσεις ως καθολικές
global $x, $y

εναλλακτικά μέσα σε συνάρτηση
$GLOBALS

σουπερ γκλομπαλ
είναι πίνακας όπου αποθηκεύονται οι μεταβλητές ως πίνακας
Ο πίνακας ΔΕΝ έχει index. Η θέση είναι το όνομα της μεταβλητής όπου συσχετίζονται με τις τιμές τους

ΣΤΑΤΙΚΕΣ ΜΕΤΑΒΛΗΤΕΣ ΣΥΝΑΡΤΗΣΕΩΝ (static)
Διατηρούν την τιμή τους.
Όταν ξαναδημιουργηθεί η συνάρτηση, η τιμή της static είναι η τελευταία που είχε αποθηκευτεί στην τελευταία κλήση.
Διατηρείται τιμή από την τελευταία κλήση που είχε αποθηκευτεί.
ΠΡΟΦΑΝΩΣ δεν μπορεί να την καλέσει κάποιος εκτός συνάρτησης.

ΠΙΝΑΚΑΣ (array)
Δήλωση
$cars=array("Ford","BMW","Toyota");

Κλήση
echo "My car is a {$cars[0]}";

ΤΥΠΟΙ ΔΕΔΟΜΕΝΩΝ
String
Integer

var_dump($x) -> τύπος δεδομένων και τι τιμή έχει η μεταβλητή

Float

e3 = 10^3

Boolean (χωρίς αυτάκια γιατί θα γίνει στρινγκ)

```
$x=true;
```

Array
Δεσμεύει τόσους χαρακτήρες όσους είναι η τιμή της μεταβήτής

Object
```
<?php
class Car {
function Car() {        // constructor
$this->model = "VW";
}

// create an object
$herbie = new Car();
// show object properties
echo $herbie->model;
?>
}
```

NULL
Αν θέλω να αλλάξω τύπο σε μια μεταβλητή, μπορώ να το περάσω από null

Resource
Είναι η αποθήκευση μιας αναφοράς σε συναρτήσεις και πόρους εξωτερικούς της PHP
Πχ δεδομένα από βάση δεδομένων

strlen μετράει το μήκος συμβολοσειράς

str_word_count("Hello world!"); καταμέτρηση πλήθους λέξεων σε ένα String

strrev("Hello world!"); αντιστρέφει ένα string

strpos("Hello world!","world"); ψάχνει χαρακτήρα ή κείμενο μέσα σε string

str_replace("world", "Dolly", "Hello world!") αντικαθιστά χαρακτήρες ενός string με άλλους

ΣΤΑΘΕΡΑ
σταθερά, χρησιμοποιήστε τη συνάρτηση define () - παίρνει τρεις παραμέτρους

την σταθερά μπορούμε να την χρησιμοποιήσουμε και μέσα σε συνάρτηση

Σύγκριση === έχει ίδια τιμή και τύπο
Σύγκριση !== δεν έχουν ίδια τιμή ή δεν έχουν ίδιο τύπο

Pre-increment ++$x πρώτα η μεταβλητή αυξάνεται και μετά η μεταβλητή λαμβάνεται στην παράσταση

Post-increment $x++ πρώτα λαμβάνεται η τιμή της μεταβλητής στην παράσταση και μετά αυξάνεται η τιμή της μεταβλητής

. η τελεία συνενώνει μεταβλητές

PHP τελεστές πίνακα
+ ένωση
== ίσοι πίνακες αν έχουν ίδια κλειδιά και ίδιες τιμές
=== ταυτόσημοι πίνακες αν έχουν ίδια κλειδιά και ίδιες τιμές και ίδιο τύπο

Αν σε δυο πίνακες έχω ένα ίδιο κλειδί με διαφορετικές τιμές, τότε στην ένωση καταχωρείται η τιμή του πίνακα που είναι στα αριστερά.

date("H") ημερομηνία

Στην if η πιο συχνά συναντώμενη συνθήκη, την βάζουμε ψηλά.

switch
κάθε case απομονώνουμε με brake

foreach
βρόχος foreach λειτουργεί μόνο σε πίνακες, και χρησιμοποιείται για να «σαρώνει» κάθε ζεύγος κλειδιού / τιμής σε ένα πίνακα

foreach ($array as $value)

Συνάρτηση
function functionName

```
function familyName($fname) {
echo "$fname Refsnes.<br>";
}
```

μια μεταβλητή την χρησιμοποιούμε ως αριθμό μόνο αν έχουμε να κάνουμε πράξη.
αλλιώς καλύτερα string.

μια συνάρτηση
function setHeight($minheight = 50)

αν την καλέσω χωρίς τιμή στην παράμετρο, παίρνει την default.
Αλλιώς θα πάρει την τιμή που θα της εισάγω.

Αν θέλω να επιστρέψει κάτι η συνάρτηση βάζω μέσα το:
return $z;

PHP Πίνακες
Ένας πίνακας είναι μια ειδική μεταβλητή, η οποία μπορεί να κρατήσει περισσότερες από μία τιμή κάθε φορά.

- Πίνακες με δείκτες - Πίνακες με ένα αριθμητικό δείκτη
- Συσχετιζόμενοι πίνακες - Πίνακες με ονομαστικά κλειδιά
- Πολυδιάστατοι πίνακες - Πίνακες που περιέχουν έναν ή περισσότερους πίνακες

ΚΛΑΣΙΚΟ
```
<?php
$cars = array("Volvo", "BMW", "Toyota");
$arrlength = count($cars); // Μετράει τα στοιχεία του πίνακα

for($x = 0; $x < $arrlength; $x++) {
echo $cars[$x];
echo "<br>";
}
?>
```

foreach($age as $x => $x_value)
καθώς διατρέχει τον πίνακα, το $x θα δείχνει το key ενώ το $x_value μας δείχνει την τιμή του key

ΤΑΞΙΝΟΜΗΣΗ ΠΙΝΑΚΩΝ
- sort() - ταξινόμηση πινάκων με αύξουσα σειρά
- rsort()- ταξινόμηση πινάκων με φθίνουσα σειρά
- asort() - ταξινόμηση συσχετιζόμενων πινάκων σε αύξουσα σειρά, σύμφωνα με την τιμή
- ksort() - ταξινόμηση συσχετιζόμενων πινάκων σε αύξουσα σειρά, σύμφωνα με το κλειδί
- arsort() - ταξινόμηση συσχετιζόμενων πινάκων σε φθίνουσα σειρά, σύμφωνα με την τιμή
- krsort() - ταξινόμηση συσχετιζόμενων πινάκων σε φθίνουσα σειρά, σύμφωνα με το κλειδί

πχ
```
sort($cars);
```

ταξινομίζει τον πίνακα $cars

PHP Καθολικές μεταβλητές - superglobals

Οι superglobals μεταβλητές της PHP είναι:
- $GLOBALS
- $_SERVER
- $_REQUEST
- $_POST
- $_GET
- $_FILES
- $_ENV
- $_COOKIE
- $_SESSION

PHP $GLOBALS

Η $GLOBALS είναι μια PHP σούπερ καθολική μεταβλητή που χρησιμοποιείται για πρόσβαση σε καθολικές μεταβλητές από οπουδήποτε

```
$x = 75;
$y = 25;

function addition() {
$GLOBALS['z'] = $GLOBALS['x'] + $GLOBALS['y'];
}
addition();
echo $z;
```

PHP $_SERVER

Η $_SERVER είναι μια PHP σούπερ global μεταβλητή που περιέχει πληροφορίες σχετικά με τις κεφαλίδες, διαδρομές και το PHP σενάριο.

```
<?php
echo $_SERVER['PHP_SELF'];
echo "<br>";
echo $_SERVER['SERVER_NAME'];
echo "<br>";
echo $_SERVER['SERVER_SOFTWARE'];
echo "<br>";
echo $_SERVER['HTTP_HOST'];
echo "<br>";
echo $_SERVER['HTTP_REFERER'];
echo "<br>";
echo $_SERVER['HTTP_USER_AGENT'];
echo "<br>";
echo $_SERVER['SCRIPT_NAME'];
?>
```

PHP $_REQUEST

Η PHP $_REQUEST χρησιμοποιείται για τη συλλογή των δεδομένων μετά την υποβολή μιας φόρμας HTML.

```
form
τα attributes είναι:
method ---> get / post (get γρήγορη αλλά λιγότερο ασφαλής. post πιο ασφαλής αφού κάνει επεξεργασία και είναι πιο αργή
οταν κάνουμε φόρμες καλύτερα να βάζουμε post
action ----> αν όλα πάνε καλά με την φόρμα σε σχέση με τον browser, τι θα γίνει μετά;
ποιος αναλαμβάνει δράση;

κάθε πεδίο, σημαντικό είναι να βάλουμε το attribute name="fname" αι δεύτερο να βάλουμε το input type="text" για είσοδο κειμένου
ΚΟΥΜΠΑΚΙ ΥΠΟΒΟΛΗ <input type="submit">


<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
// collect value of input field
$name = $_REQUEST['fname'];
if (empty($name)) {
echo "Name is empty";
} else {
echo $name;
}
}
?>
```

PHP $ _REQUEST

Κρατάει τα στοιχεία από μια φόρμα από get/post

53: Μπορεί να μην έχει και τον κώδικα PHP αλλά να υπάρχει σε άλλο αρχείο.

στο action θέλει το όνομα του αρχείου που θα επεξεργαστεί την φόρμα. Όμως εδώ έχει εναλλακτική προσέγγιση.
έχει php κώδικα που παράγει το όνομα του αρχείου που θα επεξεργαστεί. αυτό γίνεται από το $_SERVER που θα γίνει στο πεδίο
PHP_SELF

<?php echo $_SERVER['PHP_SELF'];?>
Καλεί τον εαυτό του!!! (σελ 59)

Αυτό γίνεται έτσι γιατί στον server που θα το κάνουμε host, θα εμφανίζονται με σωστό τρόπο.

στην φόρμα get πιο γρήγορη αλλά ευάλωτη ενώ η post είναι πιο αργή αλλά πιο ασφαλής

if ($_SERVER["REQUEST_METHOD"] == "POST")
εάν έρθουν τα στοιχεία με post

$name = $_REQUEST['fname'];
καταχωρεί στην μεταβλητή name ότι λάβει από την φόρμα.

Αλλά το πιο σωστό είναι:
$name = $_POST['fname'];

Η $_GET['subject'] μας δίνει πρόσβαση στο URL του a href (παράδειγμα 55)

ΑΡΧΕΙΟ 56 SOS (σελίδα 62)

$ _GET Είναι ένας πίνακας από μεταβλητές που περνάνε στο τρέχον script μέσω των παραμέτρων URL.
$ _POST Είναι ένας πίνακας από μεταβλητές που περνάνε στο τρέχον script με τη μέθοδο HTTP POST.

SOS PHP Εγκυρότητα φόρμας

Παράδειγμα 57
Σε radiobutton, checkbox το name είναι ίδιο. Δημιουργείται πίνακας με index θέση που αντιστοιχεί στην κάθε επιλογή.
Για να ξεχωρίσουν αυτά τα δυο, βάζουμε value για να γνωρίζουμε τι έχει επιλέξει ο χρήστης.

<input type="submit" name="submit" value="Submit">
Αντί για Submit βάζω αυτό που θέλω να φαίνεται στο κουμπί.

Από την γραμμή 44-55 σβήνω. Είναι μόνο για την περίπτωση ανάπτυξης

htmlspecialchars () συνάρτηση για να οχειρωθεί ο κώδικας του server από επιθέσεις με scripts.

αντικαθιστά του < και > με άλλους παρόμοιους και ακυρώνει την λειτουργικότητα πχ του script.

stripslashes($data)
βγάζαει τις καθέτους. Η κάθετη αλλάζει την εμβέλεια και αλλαζει το directory.

trim($data)
σβήνει κενούς χαρακτήρες και tab πριν και μετά σε αυτά που έχουν εισαχθεί.

ορισμός μεταβλητών
$name = $email = $gender = $comment = $website = "";

test_input είναι μια συνάρτηση δικιά μας για να καθαρίσουμε τα δεδομένα (όπως είναι από 19-26).

Παράδειγμα 58
Βάζει css με error

Μήνυμα σφάλματος: βάζω μεταβλητές
$nameErr = $emailErr = $genderErr = $websiteErr = "";

Εμφανίζουμε δίπλα στο πεδίο τον κώδικα
* <?php echo $nameErr;?>
για να εμφανιστεί το τυχόν μήνυμα σφάλματος

για radiobutton
* <?php echo $genderErr;?>

Έλεγχος αν ένα πεδίο είναι άδειο
if (empty($_POST["name"]))

Για τα πεδία που δεν είναι υποχρεωτικά, τα περνάμε από έλεγχο για την επιτάχυνση του κώδικα
```
if (empty($_POST["website"])) {
     $website = "";
   } else {
     $website = test_input($_POST["website"]);
   }
```

να μην περάσει από τον έλεγχο του test_input

ΠΑΡΑΔΕΙΓΜΑ 59

!preg_match ελέγχει την τιμή μιας μεταβλητής βάση μιας "μάσκας" (συντακτικό)

Έλεγχος με αλφαβητικό και κενό
(!preg_match("/^[a-zA-Z]*$/",$name)

Για το email κάνω έλεγχο:
(!filter_var($email, FILTER_VALIDATE_EMAIL)

Για website
(!preg_match("/\b(?:(?:https?|ftp):\/\/|www\.)[-a-z0-9+&@#\/%?=~_|!:,.;]*[-a-z0-9+&@#\/%=~_|]/i",$website))

ΠΑΡΑΔΕΙΓΜΑ 60
<input type="text" name="name" value="<?php echo $name;?>">

δείχνει στο πεδίο αυτό που έχει πληκτρολογήσει ο χρήστης.

για checkbox/radiobutton
<input type="radio" name="gender" <?php if (isset($gender) && $gender=="female") echo "checked";?> value="female">Female

πρέπει να βάλω ποιο είναι checked

ΠΑΡΑΔΕΙΓΜΑ 60b

Προσθέσαμε περιοχή για σφάλματα του browser.
 Error Messages Area:

Τον έλεγχο της φόρμας το κάνει ο browser
Name: <input type="text" name="name" value="<?php echo $name;?>" required>

με το required

ΔΕΝ υπάρχει για το box/radio.
Μπορούμε να το κάνουμε με Javascript ή όπως πριν με έλεγχο με php

Το submit το κάνουμε με
<input type="submit" name="submit" value="Submit" onclick="return checkForm();">

για τον έλεγχο. return συνάρτηση()

γραμμή 39
if (mail.indexOf('uom') == -1)

τσεκάρει εάν υπάρχει στην μεταβλητή mail το uom. Αν δεν υπάρχει, επιστρέφει -1.
Αν υπάρχει επιτρέφει true

καθαρίζω την περιοχή
clearElement("problemArea");

mysqli πιο απλή στην σύνταξη αλλά δουλεύει μόνο σε mysql. Αν θέλουμε να αλλάξουμε πχ σε Oracle, πρέπει να ξαναγράψουμε τις εντολές.

PDO είναι πιο σύνθετος τρόπος αλλά δίνει ανεξαρτησία του DBMS.

σύνδεση με MySQL (mysqli αντικειμενοστραφής)

```
<?php
$servername = "localhost";
$username = "user";
$password = "";
// Οι 3 μεταβλητές για να κάνουμε την σύνδεση


// Create connection
$conn = new mysqli($servername, $username, $password); //αντικειμενοστρεφής σύνδεση

// Check connection (κάνουμε έλεγχο αν έγινε η σύνδεση
if ($conn->connect_error) { // Υπάρχει connection error?
die("Connection failed: " . $conn->connect_error); // Αφού έχει λάθος, να σταματήσει η σύνδεση και δείχνει μήνυμα (echo)
}
echo "Connected successfully";
?>
```

ΜΗ ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗΣ ΣΥΝΔΕΣΗ

```
<?php
$servername = "localhost";
$username = "user";
$password = "";

// Create connection
$conn = mysqli_connect($servername, $username, $password);

// Check connection
if (!$conn) {
die("Connection failed: " . mysqli_connect_error()); // Το μήνυμα λάθους το παίρνουμε από την συνάρτηση mysqli_connect_error
}

echo "Connected successfully";
?>
```

ΜΕ PDO (με try / catch)

```
<?php
$servername = "localhost";
$username = "user";
$password = "";

try {
    $conn = new PDO("mysql:host=$servername;dbname=test", $username, $password); // αντικειμενοστρεφής σύνδεση. ΠΙΟ περίπλοκη σύνταξη (θέλει server, όνομα βάσης)
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION); // ΣΤΑΝΤΑΡ ΓΡΑΜΜΗ για προσπάθεια σύνδεσης
    echo "Connected successfully";
    }
catch(PDOException $e)
    {
    echo "Connection failed: " . $e->getMessage(); // εμφάνιση μηνύματος σφάλματος
    }
```

```
?>
```

Κλείστε τη σύνδεση
Η σύνδεση αυτή θα κλείνει αυτόματα όταν τελειώνει το σενάριο. Για να κλείσετε τη σύνδεση πριν, χρησιμοποιήστε τα ακόλουθα:

Παράδειγμα (mysqli αντικειμενοστρεφής)

```
$conn->close();
```

Παράδειγμα (mysqli διαδικαστική)

```
mysqli_close($conn);
```

Παράδειγμα (PDO)

```
$conn = null;
```

Δημιουργήστε μια βάση δεδομένων MySQL χρησιμοποιώντας
mysqli και PDO

Αφού άνοιξα σύνδεση... (mysqli αντικειμενοστραφής)

```
// Create database
$sql = "CREATE DATABASE myDB"; // Δημιουργώ μια μεταβλητή με την εντολή δημιπουργίας
if ($conn->query($sql) === TRUE) { // Δοκιμή να τρέξει το κουίρι
echo "Database created successfully";
} else {
echo "Error creating database: " . $conn->error; // Το σφάλμα είναι μόνο αν τυχόν δεν δημιουργηθεί η βάση
}
$conn->close();
```

Στο παράδειγμα 4 το δημιουργεί γιατί ο χρήστης είναι ο root και έχει δικαιώματα να δημιουργήσει μια βάση.

Στο παράδειγμα 4b προσπαθεί να δημιουργήσει ο user που δεν έχει δικαιώματα να δημιουργήσει μια βάση.

Στο παράδειγμα 4c αν δεν υπάρχει η βάση, να την φτιάξει (IF NOT EXISTS)

```
// Create database
$sql = "CREATE DATABASE IF NOT EXISTS myDB2";
if ($conn->query($sql) === TRUE) {
    echo "Database created successfully / was already created";
} else {
    echo "Error creating database: " . $conn->error;
}
```

ΜΕ ΤΗΝ ΑΛΛΗ ΜΟΡΦΗ mysqli (διαδικαστική)

```
// Create connection
$conn = mysqli_connect($servername, $username, $password);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

// Create database
$sql = "CREATE DATABASE myDB";
if (mysqli_query($conn, $sql)) { // Διαφορετικό statement
    echo "Database created successfully";
} else {
    echo "Error creating database: " . mysqli_error($conn);
}
```

Δημιουργια (PDO)

```
try { // Εντολές σύνδεσης
$conn = new PDO("mysql:host=$servername;dbname=myDB", $username, $password); // ΠΡΟΣΟΧΗ!!! Πρέπει να υπάρχει μια βάση για να συνδεθεί
// set the PDO error mode to exception
$conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
$sql = "CREATE DATABASE myDBPDO"; // Μια μεταβλητή θα πάρει την εντολή δημιουργίας
// use exec() because no results are returned
$conn->exec($sql); // Execute ΕΝΤΟΛΗ για το PDO
echo "Database created successfully<br>";
}

catch(PDOException $e)
{
echo $sql . "<br>" . $e->getMessage(); // Εμφανίζει τα σφάλματα...
}
```

ΠΑΡΑΔΕΙΓΜΑ 7 (mysqli) ΔΗΜΙΟΥΡΓΙΑ ΠΙΝΑΚΑ

Βάζω και την βάση που θα συνδεθώ

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "myDB";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// sql to create table (η δυσκολία είναι να είναι αυτό συντακτικά καλά)
$sql = "CREATE TABLE MyGuests (
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
firstname VARCHAR(30) NOT NULL,
lastname VARCHAR(30) NOT NULL,
email VARCHAR(50),
reg_date TIMESTAMP
)";

ΚΛΕΙΣΙΜΟ
$conn->close();
```

SQL8.php (mysqli διαδικαστική)

```
ΣΥΝΔΕΣΗ
$conn = mysqli_connect($servername, $username, $password, $dbname);

ΕΛΕΓΧΟΣ
if (mysqli_query($conn, $sql)) {

ΚΛΕΙΣΙΜΟ
mysqli_close($conn);
```

SQL9.php (PDO)

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "myDBPDO";
try {
```

```
$conn = new PDO("mysql:host=$servername;dbname=$dbname", $username,
$password);
// set the PDO error mode to exception
$conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
```

ΚΛΕΙΣΙΜΟ

```
$conn = null;
```

ΕΙΣΑΓΩΓΗ ΣΤΟΙΧΕΙΩΝ ΣΕ ΠΙΝΑΚΑ

```
INSERT INTO table_name (column1, column2, column3,...)
VALUES (value1, value2, value3,...);
```

ΠΑΡΑΔΕΙΓΜΑ 10

Πρέπει να δημιουργήσω σωστά την γραμμή εισαγωγής

```
$sql = "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com')";
```

Με τους άλλους 2 τρόπους απλά αλλάζει ο τρόπος εισαγωγής.

Σύνταξη Δημιουργήστε cookies
setcookie(name, value, expire, path, domain, secure, httponly);

Μόνο η παράμετρος όνομα απαιτείται

PHP Δημιουργία / Ανάκτηση ενός cookie
```
<?php
$cookie_name = "user";
$cookie_value = "John Doe";
setcookie($cookie_name, $cookie_value, time() + (86400 * 30), "/"); // 86400 = 1
day
?>
<html>
<body>

<?php
if(!isset($_COOKIE[$cookie_name])) {
echo "Cookie named '" . $cookie_name . "' is not set!";
} else {
echo "Cookie '" . $cookie_name . "' is set!<br>";
echo "Value is: " . $_COOKIE[$cookie_name];
}
?>
```

Τροποποίηση τιμής ενός cookie

```
<!DOCTYPE html>
<?php
$cookie_name = "user";
$cookie_value = "Alex Porter";

setcookie($cookie_name, $cookie_value, time() + (86400 * 30), "/");
?>
<html>
<body>

<?php
if(!isset($_COOKIE[$cookie_name])) {
echo "Cookie named '" . $cookie_name . "' is not set!";
} else {
echo "Cookie '" . $cookie_name . "' is set!<br>";
echo "Value is: " . $_COOKIE[$cookie_name];
}
?>
```

Διαγραφή Cookie
Για να διαγράψετε ένα cookie, χρησιμοποιήστε τη συνάρτηση setcookie() με ημερομηνία
λήξης στο παρελθόν:
Παράδειγμα (php_65.php)
```
<!DOCTYPE html>
<?php
// set the expiration date to one hour ago
setcookie("user", "", time() - 3600);
?>
<html>
<body>
<?php
echo "Cookie 'user' is deleted.";
?>
</body>
</html>
```

Ελέγξτε εάν είναι ενεργοποιημένα τα cookies

```
<!DOCTYPE html>
<?php
setcookie("test_cookie", "test", time() + 3600, '/');
?>
<html>
<body>
<?php
if(count($_COOKIE) > 0) {
echo "Cookies are enabled.";
} else {
echo "Cookies are disabled.";
}
?>
</body>
</html>
```

Ξεκινήστε μια PHP συνεδρία

```
<?php
// Start the session
session_start();
?>
<!DOCTYPE html>

<html>
<body>
<?php
// Set session variables
$_SESSION["favcolor"] = "green";
$_SESSION["favanimal"] = "cat";
echo "Session variables are set.";
?>
</body>
</html>
```

Μπορώ και να διαβάσω και να εμφανίσω τις τιμές μεταβλητών SESSION

```
<?php
// Echo session variables that were set on previous page
echo "Favorite color is " . $_SESSION["favcolor"] . ".<br>";
echo "Favorite animal is " . $_SESSION["favanimal"] . ".";
?>
```

Η print_r Καλεί μια for/each για να εκτυπώσει τον πίνακα και τα keys αλλά και τα values

```
<?php
print_r($_SESSION);
?>
```

Τροποποίηση PHP μεταβλητών συνεδρίας

```
$_SESSION["favcolor"] = "yellow";
```

Διακόψτε μια PHP συνεδρία

```
<?php
// remove all session variables
session_unset();
// destroy the session
session_destroy();
echo "All session variables are now removed, and the session is destroyed."
?>
```

PHP Πάρτε το ID της Τελευταίας Εγγραφής που εισήχθη

ΠΑΡΑΔΕΙΓΜΑ 13 (mysqli αντικειμενοστραφής)

```
if ($conn->query($sql) === TRUE) {
$last_id = $conn->insert_id;  // Με την insert_id παίρνω το τελευταίο που μπήκε
echo "New record created successfully. Last inserted ID is: " . $last_id;
}
```

SQL14.php (mysqli διαδικαστικά)

```
if (mysqli_query($conn, $sql)) {
$last_id = mysqli_insert_id($conn); // Κρατάμε με το mysqli_insert_id
echo "New record created successfully. Last inserted ID is: " . $last_id;
}
```

SQL15.php (PDO)

```
$last_id = $conn->lastInsertId();
echo "New record created successfully. Last inserted ID is: " . $last_id;
```

Εισαγωγή πολλαπλών εγγραφών σε MySQL χρησιμοποιώντας mysqli και PDO

SQL16.php (mysqli αντικειμενοστραφής)

```
$sql = "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com');";

$sql .=

$sql .=
```

Θέλουμε παραπάνω από μια φορά να εκτελεστεί

```
if ($conn->multi_query($sql) === TRUE)
```

SQL17.php (mysqli διαδικαστική)

Θέλουμε παραπάνω από μια φορά να εκτελεστεί

```
if (mysqli_multi_query($conn, $sql))
```

SQL18.php (PDO)

```
try {
$conn = new PDO("mysql:host=$servername;dbname=$dbname", $username,
$password);
// set the PDO error mode to exception
$conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

// begin the transaction
$conn->beginTransaction(); // Επειδή θέλουμε πολλές εντολές

// our SQL statememtns ότι θέλουμε να προσθέσουμε
$conn->exec("INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com')");
$conn->exec("INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('Mary', 'Moe', 'mary@example.com')");
$conn->exec("INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('Julie', 'Dooley', 'julie@example.com')");
```

```
// commit the transaction
$conn->commit(); // Πρέπει να τελειώσει με commit
echo "New records created successfully";
}

catch(PDOException $e)
{

// roll back the transaction if something failed
$conn->rollback(); // Αν γίνει σφάλμα, τότε πρέπει να ακευρώσει ότι πρόσθεσε.

echo "Error: " . $e->getMessage();
}
```