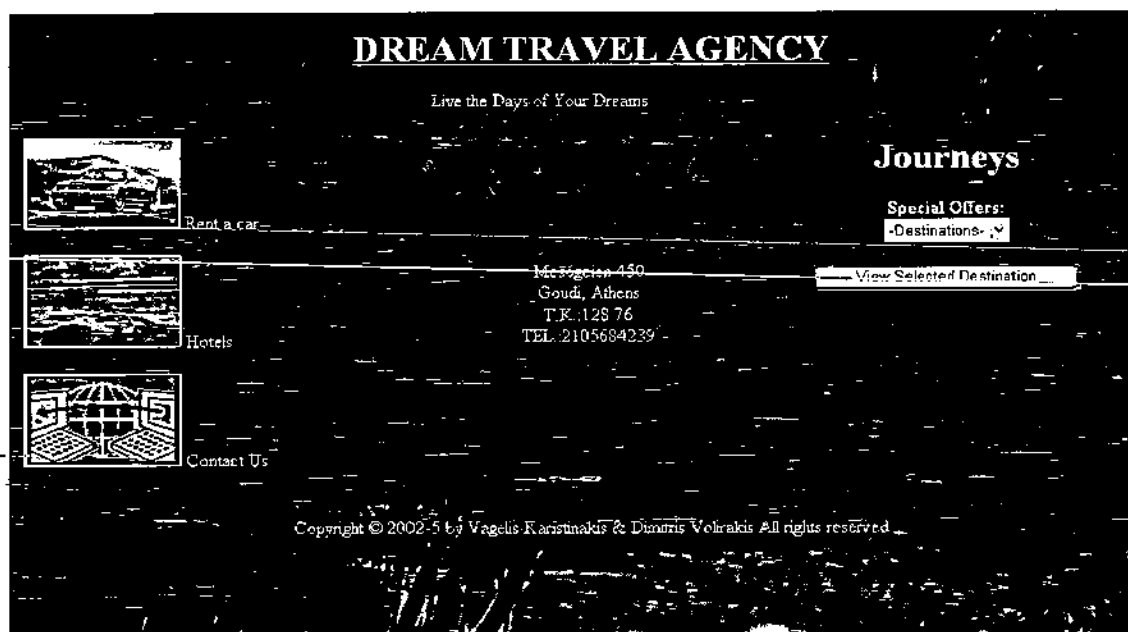


Α.Τ.Ε.Ι. ΠΑΤΡΑΣ
ΣΧΟΛΗ ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΟΙΚΟΝΟΜΙΑΣ
ΤΜΗΜΑ ΔΙΟΙΚΗΣΗΣ ΕΠΙΧΕΙΡΗΣΕΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ :
“ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΡΗΡ. ΥΛΟΠΟΙΗΣΗ
ΣΥΣΤΗΜΑΤΟΣ ΔΙΑΧΕΙΡΙΣΗΣ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ
ΜΕ ΧΡΗΣΗ ΤΟΥ ΠΑΓΚΟΣΜΙΟΥ ΙΣΤΟΥ.”



ΕΙΣΗΓΗΤΗΣ: Κος Δ. Αθανασόπουλος

Σπουδαστές: Καρυστινάκης Ευάγγελος
Βολυράκης Δημήτριος

Πάτρα
Μάιος 2005



ΑΡΙΘΜΟΣ ΕΙΣΑΓΩΓΗΣ	6366
----------------------	------

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΡΟΛΟΓΟΣ.....	1
1. Η γλώσσα HTML.....	2
1.1 Εντολές της HTML.....	3
1.1.1 Ορισμός Παραγράφων.....	3
1.1.2 Αλλαγές Γραμμών.....	3
1.1.3 Παραπομπές (Links).....	3
1.1.4 Τίτλος Σελίδας.....	6
1.1.5 Ορισμός Χρώματος Φόντου & Κειμένου της Σελίδας.....	6
1.1.6 Μορφοποιήσεις Κειμένου.....	7
1.1.7 Επικεφαλίδες.....	8
1.1.8 Στοίχιση.....	9
1.1.9 Οριζόντιες Γραμμές.....	10
1.1.10 Ορισμός Γραμματοσειρών.....	10
1.1.11 Εισαγωγή Εικόνων & Γραφικών.....	11
1.1.12 Δημιουργία Πινάκων.....	11
1.1.13 Δημιουργία Φόρμας Επικοινωνίας.....	12
1.1.14 Δημιουργία Ιστοσελίδας με Πλαίσια.....	13
1.1.15 Σελίδα που καλεί αυτόματα μία άλλη.....	13
1.1.16 Αυτόματη Ανανέωση μιας Σελίδας.....	13
1.1.17 Εισαγωγή ήχου σε σελίδα.....	14
2. MySQL (Structured Query Language).....	15
2.1 Η MySQL ως Σύστημα Διαχείρισης Βάσεων Δεδομένων.....	15
2.2 Οι Τύποι Δεδομένων της MySQL.....	16
2.2.1 Οι Αριθμητικοί Τύποι Δεδομένων.....	16
2.2.2 Τύποι Δεδομένων για Τιμές Ημερομηνίας και Ωρας.....	17
2.2.3 Οι Αλφαριθμητικοί Τύποι Δεδομένων.....	19
2.3 Οι Εντολές της MySQL.....	20
2.3.1 Η Εντολή Δημιουργίας Πινάκων.....	21
2.3.2 Η Εντολή Εισαγωγής Δεδομένων (insert).....	23
2.3.3 Η Εντολή Select.....	25
2.3.4 Η Εντολή Update.....	29
2.3.5 Η Εντολή Join.....	32
2.3.6 Η Εντολή Replace.....	33
2.3.7 Η Εντολή Delete.....	34
3. PHP.....	36
3.1 Σύντομο Ιστορικό της PHP.....	36
3.2 Οι Εντολές της PHP.....	37
3.2.1 Μεταβλητές.....	37
3.2.2 Τύποι Δεδομένων της PHP.....	38
3.2.3 Τελεστές και Εκφράσεις της PHP.....	39
3.2.4 Προτεραιότητα των Τελεστών.....	42
3.2.5 Σταθερές.....	43
3.2.6 Οι Εντολές της PHP.....	44
3.2.7 Βρόγχοι.....	50
3.2.8 Έξοδο από ένα Βρόγχο.....	53
3.3 Συναρτήσεις.....	56
3.3.1 Τι είναι μια συνάρτηση;.....	56

3.3.2 Κλήση Συναρτήσεων.....	57
3.3.3 Ορισμός μιας Συνάρτησης.....	58
3.3.4 Επιστροφή τιμών από τις καθοριζόμενες από τον χρήστη συναρτήσεις.....	60
3.3.5 Προσπέλαση-Μεταβλητών με την Εντολή Global.....	61
3.3.6 Δημιουργία Ανώνυμων Συναρτήσεων.....	62
3.4 Επικοινωνία με βάσεις δεδομένων MySQL μέσω της PHP.....	63
3.4.1 Σύνδεση στην MySQL με την PHP.....	63
4. Apache Server.....	66
4.1 Εγκατάσταση του Apache Server στα Windows.....	66
4.2 Εκκίνηση του Apache.....	73
4.3 Συνεργασία της PHP με τον Apache.....	74
5. Ανάλυση του Script της εργασίας.....	77
6. Συμπεράσματα.....	80
7. Υπόμνημα.....	82
8. Βιβλιογραφία.....	88

ΠΡΟΛΟΓΟΣ

Στο πλαίσιο της παρούσας εργασίας ασχοληθήκαμε με “Βάσεις δεδομένων και PHP.Υλοποίηση συστήματος διαχείρισης βάσεων δεδομένων με χρήση του παγκόσμιου ιστού”. Σύμφωνα με τις οδηγίες του επιβλέποντα καθηγητή κ. Αθανασόπουλου χωρίσαμε την εργασία αυτή σε πέντε στάδια. Καταρχήν, αποκτήσαμε πληροφοριακό υλικό το οποίο περιελάμβανε και τα κατάλληλα υπολογιστικά προγράμματα, τα οποία και εγκαταστήσαμε. Στο δεύτερο στάδιο, ασχοληθήκαμε με το πρώτο –πρακτικό- κομμάτι τη δημιουργία ιστοσελίδων με χρήση της γλώσσας HTML. Αποφασίσαμε να κατασκευάσουμε την ιστοσελίδα ενός εικονικού ταξιδιωτικού γραφείου. Σύμφωνα με τις οδηγίες του επιβλέποντα καθηγητή κατασκευάσαμε την ιστοσελίδα του ταξιδιωτικού γραφείου “DREAM TRAVEL AGENCY”, η οποία περιλαμβάνει μία παραπομπή για την ενοικίαση αυτοκινήτου (rent a car), μία για το κλείσιμο δωματίων σε ξενοδοχείο οποιουδήποτε μέρους του κόσμου (hotels) και μία παραπομπή μέσω της οποίας οι χρήστες θα μπορούν να επικοινωνούν με το ταξιδιωτικό γραφείο (conduct). Στο τρίτο στάδιο, δημιουργήσαμε μία βάση δεδομένων με χρήση της MySQL, η οποία θα περιέχει κάποιες ταξιδιωτικές προσφορές που θα προσφέρει το ταξιδιωτικό γραφείο. Κατά το τέταρτο στάδιο κατασκευάσαμε, με χρήση της γλώσσας PHP, την ιστοσελίδα η οποία θα εμφανίζει στο χρήστη της ταξιδιωτικές προσφορές της επιλογής του, αφού προηγηθεί η ανάλογη αναζήτηση στη βάση δεδομένων. Στο τελευταίο στάδιο ασχοληθήκαμε με την εγκατάσταση της εφαρμογής (web server) η οποία θα επιτρέπει τη λειτουργία των ιστοσελίδων του ταξιδιωτικού γραφείου.

Όλα αυτά θα δημιουργήθηκαν βήμα - βήμα κατά τη διάρκεια αυτής της εργασίας και αφού μεσολάβησε πολλή μελέτη για τη γνώση του θεωρητικού υπόβαθρου που απαιτείται, πριν ξεκινήσει η σχεδίαση και η υλοποίηση της εφαρμογής.

Πιστεύουμε ότι μέσα από τις σελίδες αυτής της εργασίας θα μπορούσαμε να μεταφέρουμε στον αναγνώστη τόσο την εμπειρία μας για τα αντικείμενα με το οποίο ασχοληθήκαμε, όσο και τη βασική γνώση υλοποίησης συστήματος διαχείρισης βάσεων δεδομένων με χρήση του παγκόσμιου ιστού.

1. Η γλώσσα HTML

Η HTML (Hyper Text Markup Language) είναι η πιο συνηθισμένη γλώσσα για τη δημιουργία ιστοσελίδων. Τα αρχεία της έχουν τη μορφή **κειμένου** και την κατάληξη **.htm** ή **.html**. Για τον παραπάνω λόγο, κατά τη δημιουργία ιστοσελίδων με γλώσσα html χρησιμοποιούμε οποιοδήποτε πρόγραμμα μας παρέχει τη δυνατότητα να δημιουργούμε text αρχεία, όπως είναι το σημειωματάριο (notepad) των windows ή ένα πρόγραμμα επεξεργασίας κειμένου (σε περίπτωση που θα χρησιμοποιηθεί το word δεν πρέπει να ξεχάσουμε ότι το αρχείο θα πρέπει να σωθεί με κατάληξη **.txt** δηλαδή text και όχι σε μορφή **.doc** δηλαδή document.)

Όπως όλες οι γλώσσες προγραμματισμού έτσι και η html αποτελείται από εντολές. Ένα έγγραφό της χωρίζεται στην επικεφαλίδα (head), η οποία περιέχει τον τίτλο (title) και στο κυρίως σώμα (body), το οποίο περιέχει όλες τις υπόλοιπες εντολές που πλαισιώνουν την υπό κατασκευή ιστοσελίδα. Κάθε εντολή αποτελείται από την οδηγία (tag) ανοίγματος, το περιεχόμενο της εντολής και την οδηγία κλεισίματος. Η οδηγία ανοίγματος είναι ουσιαστικά το όνομα της εντολής μέσα σε παρένθεση, π.χ <html>,<head>,<body>, ενώ η οδηγία κλεισίματος είναι το όνομα της εντολής με την προσθήκη μιας καθέτου στην αρχή της π.χ </html>,</head>,</body>. Το περιεχόμενο της εντολής τοποθετείται μεταξύ της οδηγίας ανοίγματος και της οδηγίας κλεισίματος.

Σε περίπτωση που επιθυμούμε να δούμε τη μορφή που έχει η σελίδα μας αρκεί να την ανοίξουμε μέσω του προγράμματος φυλλομετρητή (browser) που χρησιμοποιούμε, όπως για παράδειγμα τον Internet Explorer, το Netscape Communicator ή το Mozilla Firefox. Όλες οι μορφοποιήσεις θα πρέπει να δηλωθούν ξεχωριστά την ώρα που γράφουμε το κείμενο γιατί αλλιώς ο φυλλομετρητής θα τις αγνοήσει. Έτσι μέσω ορισμένων εντολών μορφοποιούμε κατάλληλα το κείμενό μας.

1.1 Εντολές της HTML

1.1.1 Ορισμός Παραγράφων

Η εντολή με την οποία η HTML αναγνωρίζει τις παραγράφους είναι η `<P>`, η οποία αφήνει μια κενή γραμμή μεταξύ του κειμένου που βρίσκεται αριστερά της και εκείνου που βρίσκεται δεξιά της.

1.1.2 Αλλαγές Γραμμών

Για τη μεταφορά του υπολοίπου της τρέχουσας γραμμής στην αμέσως επόμενη χρησιμοποιούμε την εντολή `
` η οποία λειτουργεί όπως το πάτημα του πλήκτρου ENTER σε έναν επεξεργαστή κειμένου και σε αντίθεση με την `<P>` δεν αφήνει μια κενή γραμμή. Για να αφήσουμε μια γραμμή κενή με αυτή την εντολή θα πρέπει να τη χρησιμοποιήσουμε δυο φορές (`

`).

Αν θέλουμε να υποχρεώσουμε ένα κείμενο να μείνει ολόκληρο στην ίδια γραμμή, όσο μεγάλο και να είναι, χρησιμοποιούμε την εντολή `<NOBR>`. Επειδή το αποτέλεσμα της εντολής `<NOBR>` μπορεί να μην είναι καλαίσθητο ή είναι ανάγκη να γίνει αλλαγή γραμμής, χρησιμοποιούμε ανάμεσα στην εντολή `<NOBR>`, στο σημείο που θέλουμε να γίνει αλλαγή γραμμής στο κείμενο, την εντολή `<WBR>`. Για παράδειγμα : `<NOBR>κείμενο<WBR>κείμενο</NOBR>`.

1.1.3 Παραπομπές (Links)

Η σύνδεση δυο σελίδων γίνεται κάνοντας κλικ με το ποντίκι μας σε κατάλληλα μορφοποιημένο κείμενο ή εικόνα, έτσι ώστε να μεταφερθούμε από τη μια στην άλλη. Επιτυγχάνεται δε με τις παραπομπές. Η γενική σύνταξη μιας παραπομπής είναι η εξής: ` το κείμενο της παραπομπής `. Όπου URL η θέση κάποιας WEB σελίδας ή η διαδρομή στο δίσκο μας. Οι παραπομπές χωρίζονται στις εξής οκτώ περιπτώσεις :

1. Από μια σελίδα σε μια άλλη, κάπου στο διαδίκτυο.

Στην εικόνα 1.1 της παρούσας εργασίας βλέπουμε δύο σχετικές παραπομπές από τη σελίδα μας (με χρήση εικόνων) σε αντίστοιχες ιστοσελίδες στο διαδίκτυο.

```
dream.txt - Notepad
File Edit Format View Help
<html>...
<body background="c:\ptyxiakh\cancer.jpg" text=white link=yellow vlink=purple>
<font size=9><h1 align=center>DREAM TRAVEL AGENCY</h1></font>
<bgound src="c:\ptyxiakh\Eurithmicks-sweet Dreams.mp3" loop="5">
<p><a href="www.hertz.com" align=left></a>
Rent a car
<p><a href="www.bookhotel.com" align=left></a>
HOTELS
<p><a href="mailto:dream@yahoo.gr" align=left></a>
Contact us</body>
```

Εικ.1.1 Παραπομπή από μία σελίδα σε μία άλλη στο διαδίκτυο.

Τα δύο αυτά links μας παραπέμπουν στην σελίδα της εταιρείας Hertz (www.hertz.com), σε περίπτωση που επιθυμούμε να νοικιάσουμε κάποιο όχημα και στην σελίδα www.bookhotel.com, η οποία περιέχει πληροφορίες και κάνει κρατήσεις για δωμάτια σε ξενοδοχεία σε όλο τον κόσμο.

2. Από μια σελίδα σε μια άλλη που βρίσκεται στον ίδιο φάκελο.

Ο γενικός τύπος της παραπομπής είναι `κείμενο παραπομπής`. Έστω λοιπόν ότι στον φάκελο που έχουμε όλα τα στοιχεία της εργασίας μας, ο οποίος λέγεται *ptyxiaki*, είχαμε άλλη μία HTML σελίδα που θέλουμε να συνδέσουμε με την δικιά μας και την λέγανε *page2.html*. Για να κάνουμε την σύνδεση μέσω της παραπομπής αυτής αρκεί να γράψουμε `Εδώ`, και πατώντας την παραπομπή εδώ μας εμφανίζεται η άλλη σελίδα.

3. Από μια σελίδα σε μια άλλη που βρίσκεται σε υποφάκελο.

Έστω ότι έχουμε την HTML σελίδα μας αποθηκευμένη στην θέση *c:\ptyxiaki\mypage.html*, με το όνομα *mypage.html* και θέλουμε να την συνδέσουμε μέσω παραπομπής με μια δεύτερη σελίδα με το όνομα *page2.html* η οποία βρίσκεται στην θέση *c:\ptyxiaki\others\page2.html*, αρκεί να γράψουμε την εντολή `Εδώ` και κάνοντας κλικ με το ποντίκι μας στην λέξη Εδώ θα μας εμφανιστεί η *page2*.

4. Από μια σελίδα σε μια άλλη που βρίσκεται σε άλλο φάκελο σε ανώτερο επίπεδο.

Αντίστοιχα και σε αυτή την παραπομπή εάν είχαμε τα ίδια στοιχεία με το προηγούμενο παράδειγμα αλλά η page2.html ήταν σε ανώτερο φάκελο και όχι σε υποφάκελο για να παραπεμφθούμε σε αυτή την σελίδα αρκεί να γράφαμε την εντολή `Εδώ`, και κάνοντας κλικ στην λέξη Εδώ θα μας εμφανιζόταν η page2. Η ιδιαιτερότητα της εντολής αυτής είναι ότι δεν χρειάζεται να γράψουμε τον ανώτερο φάκελο γιατί κάθε φάκελος έχει έναν μόνο ανώτερο, οπότε οι δύο τελείες αρκούν.

5. Από μια σελίδα σε μια άλλη που βρίσκεται σε άλλο γενικά φάκελο.

Η παραπομπή αυτή είναι συνδυασμός των δύο προηγούμενων. Δηλαδή έστω ότι η σελίδα μας με όνομα mypage.html είναι αποθηκευμένη στην θέση `c:\ptychiaki\mypage.html` και μία δεύτερη σελίδα με το όνομα page2.html βρίσκεται στον φάκελο `c:\htmlpages\page2.html` και θέλουμε να συνδέσουμε τις δύο αυτές σελίδες μέσω παραπομπής. Αρκεί λοιπόν να γράψουμε την εντολή `Εδώ`.

6. Από ένα σημείο της σελίδας σε ένα άλλο.

Σε αυτήν την περίπτωση πρέπει πρώτα να ονομάσουμε το κείμενο ή τη λέξη που βρίσκεται στο σημείο που θέλουμε να παραπέμψουμε τον αναγνώστη μας, ακολουθώντας την οδηγία: ` κείμενο `, και ακολούθως χρησιμοποιούμε το γενικό τύπο της παραπομπής, όπου αντί για κάποιο URL γράφουμε: `#ONOMA`, δηλαδή το όνομα του σημείου στο οποίο επιθυμούμε να μεταβούμε.

7. Από μια σελίδα σε ένα σημείο μιας άλλης σελίδας.

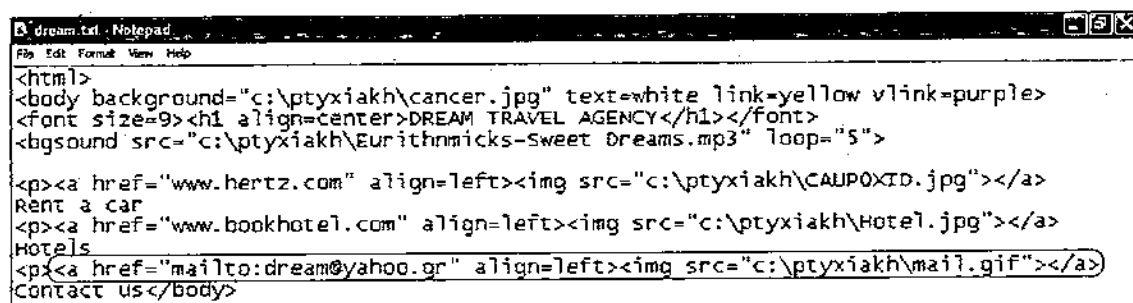
Αυτή η περίπτωση είναι ένας συνδυασμός της 5 και της 6, όπου στη θέση του URL βάζουμε τη διαδρομή στο δίσκο και τη δέση με το όνομα του σημείου που θέλουμε να παραπέμψουμε το χρήστη.

8. Για την αποστολή e-mail.

Για την παραπομπή του αναγνώστη μας σε ένα παράθυρο σύνταξης μηνύματος ηλεκτρονικού ταχυδρομείου (e-mail) με παραλήπτη τη διεύθυνση που επιθυμούμε εμείς, αρκεί να γράψουμε την εντολή:

```
<A HREF="MAILTO:name@provider.προσδιορισμός"> επικοινωνία </A>
```

όπου *name@provider.προσδιορισμός* είναι η διεύθυνση του παραλήπτη του μηνύματος. Στο παράδειγμα της εικόνας 1.2: *dream@yahoo.gr*.



```
dream.txt - Notepad
File Edit Format View Help
<html>
<body background="c:\ptyxiakh\cancer.jpg" text=white link=yellow vlink=purple>
<font size=9><h1 align=center>DREAM TRAVEL AGENCY</h1></font>
<bgsound src="c:\ptyxiakh\Eurithmicks-Sweet Dreams.mp3" loop="5">
<p><a href="www.hertz.com" align=left></a>
Rent a car
<p><a href="www.bookhotel.com" align=left></a>
Hotels
<p><a href="mailto:dream@yahoo.gr" align=left></a>
Contact us</body>
```

Εικ.1.2 Παραπομπή για την αποστολή e-mail.

Μόλις λοιπόν ο χρήστης της σελίδας κάνει κλικ με το ποντίκι του πάνω στην παραπομπή αυτή, αυτομάτως θα ανοίξει το πρόγραμμα που διαθέτει ο υπολογιστής του για τη δημιουργία, αποδοχή και αποστολή e-mail και στη θέση του παραλήπτη θα είναι ήδη γραμμένη η διεύθυνση που προσδιορίσαμε στον κώδικά μας.

1.1.4 Τίτλος Σελίδας

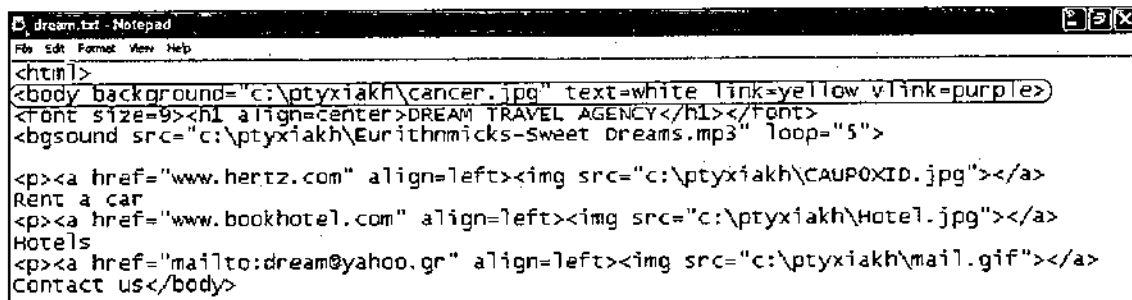
Κάθε ιστοσελίδα πρέπει να έχει έναν τίτλο, αν και αυτό δεν είναι τεχνικά απαραίτητο. Ο τίτλος πρέπει να είναι περιεκτικός και σύντομος έτσι ώστε να γίνει η σωστή ταξινόμηση της σελίδας. Για τον τίτλο χρησιμοποιούμε την εντολή title, η οποία περιέχεται στην εντολή head και συντάσσεται ως εξής :

```
<HEAD><TITLE> ο τίτλος που επιθυμούμε </TITLE></HEAD>
```

1.1.5 Ορισμός χρώματος φόντου και κειμένου της σελίδας

Το χρώμα του φόντου και του κειμένου της σελίδας ορίζεται μέσα στην οδηγία <BODY>, η οποία ανοίγει αμέσως μετά τις <TITLE> και <HEAD>, και κλείνει με την </BODY>, προτελευταία αμέσως πριν την </HTML>. Η σύνταξή της είναι: <BODY BGCOLOR=ΧΡΩΜΑ>. Εάν θέλουμε να αλλάξουμε το χρώμα του

κειμένου γράφουμε: **<BODY TEXT=ΧΡΩΜΑ>**. Επίσης μπορούμε να χρωματίσουμε τις παραπομπές (LINK) που δεν έχουμε ακολουθήσει ακόμα (**<BODY LINK=ΧΡΩΜΑ>**), καθώς και αυτές που έχουμε ήδη ακολουθήσει (**<BODY VLINK=ΧΡΩΜΑ>**) ή και ακόμα αυτές που έχουμε κάνει κλικ πάνω τους αλλά δεν έχουμε αφήσει ακόμα το πλήκτρο του ποντικιού (**<BODY ALINK=ΧΡΩΜΑ>**).



```
dream.txt - Notepad
File Edit Format View Help
<html>
<body background="c:\ptyxiakh\cancer.jpg" text=white link=yellow vlink=purple>
<font size=9><h1 align=center>DREAM TRAVEL AGENCY</h1></font>
<bgsound src="c:\ptyxiakh\Eurithmicks-Sweet Dreams.mp3" loop="5">
<p><a href="www.hertz.com" align=left></a>
Rent a car
<p><a href="www.bookhotel.com" align=left></a>
Hotels
<p><a href="mailto:dream@yahoo.gr" align=left></a>
Contact us</body>
```

Εικ.1.3 Ορισμός χρώματος φόντου και κειμένου της σελίδας.

Ετσι λοιπόν ολοκληρωμένη η εντολή μας θα έχει την ακόλουθη, ή κάποια παρόμοιά της, μορφή: **<BODY BGCOLOR=RED TEXT=BLUE LINK=PINK VLINK=PURPLE ALINK=YELLOW>**. Τα χρώματα υπάρχουν και με τη μορφή ενός δεκαεξαδικού κωδικού (π.χ. το #934CE8 είναι το πράσινο). Εάν χρησιμοποιηθεί στην εντολή ο κωδικός αντί για το όνομα του χρώματος θα πρέπει να γραφεί μέσα σε εισαγωγικά.

1.1.6 Μορφοποιήσεις κειμένου.

Η HTML μας παρέχει τη δυνατότητα να μορφοποιήσουμε το κείμενό μας χρησιμοποιώντας τις κατάλληλες εντολές. Οι μορφοποιήσεις αυτές είναι οι:

- Έντονο (Bold) κείμενο: **...**
- Πλάγιο (Italic) κείμενο: *<I>...</I>*
- Έντονο και πλάγιο (Bold και Italic) μαζί: ***<I>...</I>***
- Υπογραμμισμένο (Underlined) κείμενο: <U>...</U>

```

D dream.txt - Notepad
File Edit Format View Help
<html>
<body background="c:\ptyxiakh\cancer.jpg" text=white link=yellow vlink=purple>
<font size=9><h1 align=center><b><u>DREAM TRAVEL AGENCY</b></u></h1></font>
<bgsound src="c:\ptyxiakh\Eurithmicks-sweet Dreams.mp3" loop=5">
<p><a href="www.hertz.com" align=left></a>
Rent a car
<p><a href="www.bookhotel.com" align=left></a>
Hotels
<p><a href="mailto:dream@yahoo.gr" align=left></a>
Contact us</body>

```

Εικ.1.4 Χρήση μορφοποιήσεων bold και υπογράμμιση.

Στην εικόνα 1.4 απεικονίζεται η επικεφαλίδα που έχουμε βάλει στη σελίδα μας και θα εμφανίζεται πάνω σε αυτή. Όπως μπορούμε να παρατηρήσουμε είναι ανάμεσα στις εντολές `<u>...</u>`, αυτό σημαίνει ότι στην οθόνη μας θα εμφανιστεί με υπογράμμιση και τα γράμματα θα είναι bold (**DREAM TRAVEL AGENCY**).

- Κείμενο γραμμένο με μεγαλύτερα γράμματα από το υπόλοιπο:

`<BIG>...</BIG>`

- Αντίθετη της εντολής BIG: `<SMALL>...</SMALL>`

- Εντολή κατά την οποία κάθε γράμμα διαπερνάται από μια οριζόντια γραμμή (strike through): `<S>...</S>`

- Κείμενο το οποίο τοποθετείται:

1. ελάχιστα πιο κάτω από το επίπεδο των υπόλοιπων γραμμάτων, δηλαδή δείκτης (π.χ. για παρονομαστή κλάσματος – subscript): `_{...}`

2. ελάχιστα πιο πάνω από το επίπεδο των υπολοίπων γραμμάτων, δηλαδή εκθέτης (π.χ. για αριθμητή κλάσματος – superscript): `^{...}`

1.1.7 Επικεφαλίδες

Η εντολή της επικεφαλίδας εκτός από το να την παρουσιάζει στη σελίδα μας καθορίζει επίσης και το μέγεθος των γραμμάτων της. Ξεκινώντας από την μεγαλύτερη σε μέγεθος επικεφαλίδα η εντολή είναι η εξής : `<H1>...</H1>`, `<H2>...</H2>`, ..., `<H6>...</H6>`.

```

dream.txt - Notepad
File Edit Format View Help
<html>
<body background="c:\ptyxiakh\cancer.jpg" text=white link=yellow vlink=purple>
<font size=9><h1 align=center><b><u>DREAM TRAVEL AGENCY</u></b></h1></font>
<bgsound src="c:\ptyxiakh\Eurithmicks-sweet dreams.mp3" loop="5">

<p><a href="www.hertz.com" align=left></a>
Rent a car
<p><a href="www.bookhotel.com" align=left></a>
Hotels
<p><a href="mailto:dream@yahoo.gr" align=left></a>
Contact us</body>

```

Εικ.1.5 Χρήση μορφοποιήσεων δημιουργίας επικεφαλίδων.

Στην εικόνα 1.5 μπορούμε να διακρίνουμε ότι η επικεφαλίδα της σελίδας μας θα είναι με τη μεγαλύτερη δυνατή γραμματοσειρά αφού χρησιμοποιήσαμε την εντολή <h1>...</h1>.

1.1.8 Στοιχίση

Η στοιχίση κειμένου ή φωτογραφιών επιτυγχάνεται μέσω της παραμέτρου **ALIGN**. Αυτό σημαίνει ότι δε λειτουργεί ποτέ μόνη της αλλά πάντοτε μέσα σε μια εντολή.

Για παράδειγμα, για να στοιχίσουμε το κείμενο μιας επικεφαλίδας χρησιμοποιούμε την ALIGN ως εξής: <H1 ALIGN=X>επικεφαλίδα</H1> (όπου X= left, right, center).

```

dream.txt - Notepad
File Edit Format View Help
<html>
<body background="c:\ptyxiakh\cancer.jpg" text=white link=yellow vlink=purple>
<font size=9><h1 align=center>DREAM TRAVEL AGENCY</h1></font>
<bgsound src="c:\ptyxiakh\Eurithmicks-sweet dreams.mp3" loop="5">

<p><a href="www.hertz.com" align=left></a>
Rent a car
<p><a href="www.bookhotel.com" align=left></a>
Hotels
<p><a href="mailto:dream@yahoo.gr" align=left></a>
Contact us</body>

```

Εικ.1.6 Χρήση της align στην σελίδα μας.

Κατά τη δημιουργία του script της εργασίας μας χρειάστηκε πολλές φορές να προσδιορίσουμε τη στοίχιση ορισμένων στοιχείων της. Στην εικόνα 1.6 φαίνονται τέσσερα χαρακτηριστικά παραδείγματα της χρήσης της align.

Μια δεύτερη περίπτωση της align αφορά στη στοίχιση μιας παραγράφου. Σε αυτήν την περίπτωση χρησιμοποιούμε την ALIGN ως εξής: `<P ALIGN=X>κείμενο.`

Μια ακόμη παρόμοια εντολή είναι η `<DIV>...</DIV>`, που χρησιμεύει για να εμφανίσουμε σε όποια πλευρά της σελίδας θέλουμε, είτε επικεφαλίδες είτε κείμενα στο σύνολό τους και συντάσσεται ως εξής:

```
<DIV ALIGN=X><H1>head1</H1><H3>head2</H3><P>κείμενο</DIV>
```

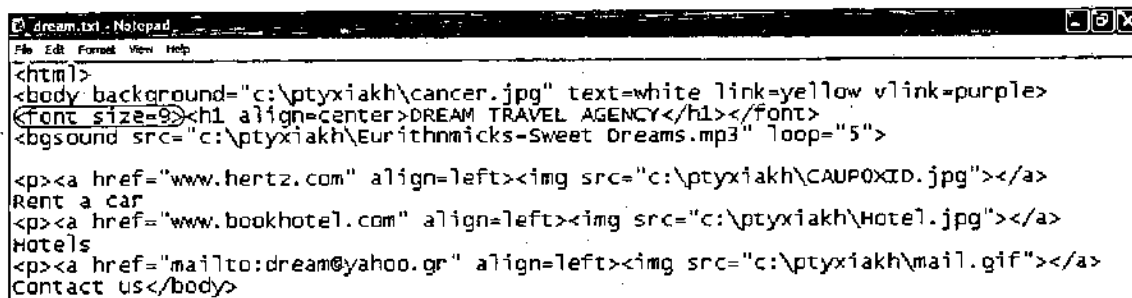
1.1.9 Οριζόντιες γραμμές

Για να σχεδιάσουμε μια οριζόντια γραμμή αρκεί να δώσουμε την εντολή `<HR>`, την οποία δε χρειάζεται να κλείσουμε, αφού στο τέλος της σελίδας σταματάει μόνη της.

Η `<HR>` περιλαμβάνει τις εξής παραμέτρους: **SIZE** για τον αριθμό των pixels της οθόνης, **WIDTH** για το πλάτος της γραμμής, **ALIGN** για να ορίσουμε τη στοίχιση της γραμμής (αριστερά, δεξιά, κέντρο), π.χ. `<HR ALIGN=RIGHT SIZE=8 WIDTH=75%>`.

1.1.10 Ορισμός γραμματοσειρών

Εδώ συναντάμε εντολές που αφορούν στο μέγεθος, στο χρώμα και στο είδος της γραμματοσειράς και είναι αντίστοιχα οι εξής: `...`, `...`, `...`, όπου X π.χ. 2, yellow, times new roman, αντίστοιχα.



```
dream.txt - Notepad
File Edit Format View Help
<html>
<body background="c:\ptyxiakh\cancer.jpg" text=white link=yellow vlink=purple>
<font size=9><h1 align=center>DREAM TRAVEL AGENCY</h1></font>
<bgsound src="c:\ptyxiakh\Eurithmicks-Sweet Dreams.mp3" loop="5">

<p><a href="www.hertz.com" align=left></a>
Rent a car
<p><a href="www.bookhotel.com" align=left></a>
Hotels
<p><a href="mailto:dream@yahoo.gr" align=left></a>
Contact us</body>
```

Εικ.1.7 Ορισμός της γραμματοσειράς της σελίδας μας.

Στην σελίδα που δημιουργήσαμε, στο παράδειγμα της εικόνας 1.7, έχουμε ορίσει το μέγεθος της γραμματοσειράς που χρησιμοποιούμε να είναι οι 9 στιγμές.

1.1.11 Εισαγωγή εικόνων και γραφικών

Για την εισαγωγή εικόνων και γραφικών θα πρέπει να προσέξουμε ώστε τα αρχεία μας να έχουν κατάληξη **.gif** ή **.jpg**, αφού αυτά είναι τα πρότυπα που αναγνωρίζουν οι browsers. Για την εισαγωγή μιας εικόνας σε μια ιστοσελίδα χρησιμοποιούμε την εντολή ****. Για να ρυθμίσουμε σε ποιο σημείο της σελίδας θα εμφανίζεται η εικόνα αρκεί να προσθέσουμε στην εντολή μας την παράμετρο **ALIGN** π.χ. ****. Μπορούμε επίσης να καθορίσουμε την απόσταση του κειμένου από την εικόνα, χρησιμοποιώντας τις παραμέτρους **VSPACE** και **HSPACE**, καθώς και το πλάτος και μήκος της εικόνας, χρησιμοποιώντας τις **width** και **height**. Τέλος μπορούμε να χρησιμοποιήσουμε μια εικόνα σαν παραπομπή γράφοντας την εντολή ****.

```
dream.01 Notepad
File Edit Format View Help
<html>
<body background="c:\ptyxiakh\cancer.jpg" text=white link=yellow vlink=purple>
<font size=9><h1 align=center>DREAM TRAVEL AGENCY</h1></font>
<bgsound src="c:\ptyxiakh\Eurithmicks-sweet Dreams.mp3" loop="5">
<p><a href="www.hertz.com" align=left></a>
Rent a car
<p><a href="www.bookhotel.com" align=left></a>
Hotels
<p><a href="mailto:dream@yahoo.gr" align=left></a>
Contact us</body>
```

Εικ.1.8 Παράδειγμα εισαγωγής εικόνων σε μια σελίδα.

Στις ιστοσελίδες που δημιουργήσαμε έχουμε προσθέσει κάποιες φωτογραφίες οι οποίες βοηθούν στον καλλωπισμό της και πολλές φορές την κάνουν και πιο εύχρηστη. Εμείς σε κάθε παραπομπή έχουμε και από μία εικόνα ώστε να γίνεται πιο κατανοητό το τι ζητάει η κάθε παραπομπή.

1.1.12 Δημιουργία πινάκων

Για τη δημιουργία πινάκων χρησιμοποιούμε την εντολή **<TABLE>**.

Τα βήματα που ακολουθούμε είναι τα εξής:

1. Αποφασίζουμε για το σχήμα του πίνακα

2. Ορίζουμε τον πίνακα με την `<TABLE>...</TABLE>`
3. Ορίζουμε τις γραμμές με την `<TR>...</TR>`
4. Ορίζουμε ένα-ένα τα κελιά της κάθε γραμμής με την `<TH>...</TH>`

για κελιά επικεφαλίδας και με την `<TD>...</TD>` για κελιά δεδομένων.

Στην εντολή `<TABLE>` μπορούμε να προσθέσουμε κάποιες παραμέτρους οι οποίες μορφοποιούν τον πίνακά μας και είναι οι ακόλουθες:

1. Η παράμετρος **BORDER** με την οποία βάζουμε περιθώρια, πχ. `<TABLE BORDER=X>...</TABLE>`.

2. Η παράμετρος **CAPTION**, την οποία χρησιμοποιούμε για να βάλουμε τίτλο στον πίνακά μας, πχ. `<TABLE> <CAPTION>τίτλος</CAPTION>...</TABLE>`.

3. Η παράμετρος **ALIGN** η οποία χρησιμοποιείται για να ορίσουμε σε ποια πλευρά του κελιού θέλουμε να στοιχηθούν τα δεδομένα μας και την χρησιμοποιούμε ως **ALIGN=X** για να επιλέξουμε την μεριά, αριστερά ή δεξιά, που θα γραφούν τα δεδομένα, και ως **VALIGN=X** για να επιλέξουμε αν τα δεδομένα μας θα στοιχηθούν στο πάνω ή στο κάτω μέρος του κελιού.

1.1.13 Δημιουργία φόρμας επικοινωνίας

Μπορούμε να δημιουργήσουμε μια φόρμα επικοινωνίας για τους επισκέπτες μιας ιστοσελίδας, όπου θα μπορούν να παραθέτουν σχόλια και παρατηρήσεις, μέσω της εντολής `<FORM>...</FORM>`, η οποία ρυθμίζεται από τις παραμέτρους **ACTION** που μέσω ενός προγράμματος φροντίζει για τη διαχείριση των δεδομένων που θα δώσει ο χρήστης, **METHOD** που καθορίζει τον τρόπο λήψης των δεδομένων από το πρόγραμμα με το οποίο συνεργάζεται η **ACTION**. Επίσης μέσω της **INPUT**, σε συνδυασμό με την **FORM**, καθορίζουμε και διαμορφώνουμε τους χώρους στους οποίους θα τοποθετήσει ο χρήστης τα δεδομένα του. Η εντολή αυτή αποτελείται από τις παραμέτρους **TYPE** και **NAME**, οι οποίες καθορίζουν το είδος των δεδομένων που θα δοθούν στη φόρμα και το όνομα (λεζάντα) που δίδεται στο πεδίο αυτό αντίστοιχα. Τέλος, υπάρχουν κάποιες παράμετροι, όπως **PASSWORD**, **RADIO**, **CHECKBOX**, **RESET**, **TEXTAREA** και άλλες που μας βοηθούν στην περαιτέρω διαμόρφωση της φόρμας της ιστοσελίδας μας.

1.1.14 Δημιουργία ιστοσελίδας με πλαίσια

Η δημιουργία πλαισίων σε μια ιστοσελίδα μας δίνει τη δυνατότητα να χωρίσουμε τη σελίδα μας σε όσα κομμάτια-τμήματα επιθυμούμε, το καθένα εκ των οποίων αποτελεί μια ξεχωριστή ιστοσελίδα από μόνο του, η οποία περιέχει τα δικά της ξεχωριστά στοιχεία. Θα πρέπει πάντα να κατασκευάζουμε μια σελίδα παραπάνω από αυτές που έχουμε προγραμματίσει να χωρίσουμε την οθόνη μας, καθώς θα χρειαστούμε μια σελίδα (ορίζουσα σελίδα) που θα καθορίζει τη θέση και το μέγεθος των πλαισίων μέσα στα οποία θα τοποθετηθούν οι υπόλοιπες. Στην ορίζουσα σελίδα η οδηγία `<BODY>...</BODY>` αντικαθιστάται από την οδηγία `<FRAMESET>...</FRAMESET>`. Στη συνέχεια η `FRAMESET`, που καθορίζει το μέγεθος των πλαισίων, δέχεται στο εσωτερικό της μια άλλη οδηγία, την `<FRAME>` η οποία καθορίζει το περιεχόμενό τους. Επίσης η `FRAMESET` δέχεται τις ακόλουθες παραμέτρους: την `COLS="X,X,X..."` η οποία καθορίζει τα κάθετα πλαίσια που θα δημιουργηθούν και την `ROWS="X,X,X..."` που με τον ίδιο τρόπο χωρίζει την οθόνη σε οριζόντια πλαίσια. Σε περίπτωση που ο browser του υπολογιστή του χρήστη δεν αναγνωρίζει πλαίσια, με την εντολή `<NOFRAME>...</NOFRAME>`, που την εσωκλείουμε στη `FRAMESET`, ο χρήστης θα λαμβάνει ένα μήνυμα που θα τον πληροφορεί ότι δεν είναι δυνατή η εμφάνιση της σελίδας. Τέλος, άλλες δυο πολύ σημαντικές παράμετροι είναι οι `NAME`, με την οποία ονομάζουμε τα πλαίσια στα οποία έχουμε χωρίσει την οθόνη μας και η `TARGET`, η οποία μέσω της `NAME` εμφανίζει τις παραπομπές στο πλαίσιο που πρέπει να εμφανιστούν.

1.1.15 Σελίδα που καλεί αυτόματα μια άλλη

Πολλές φορές συναντάμε το φαινόμενο μια ιστοσελίδα να αλλάζει διεύθυνση. Για τη διευκόλυνση του επισκέπτη που γνωρίζει την παλιά διεύθυνση αρκεί στην παλιά σελίδα, στην οδηγία `BODY`, να προσθέσουμε την εντολή :

`<BODY onLoad="location.href='http://www.xxxxx.gr'">`, όπου `xxxxx` η νέα διεύθυνση.

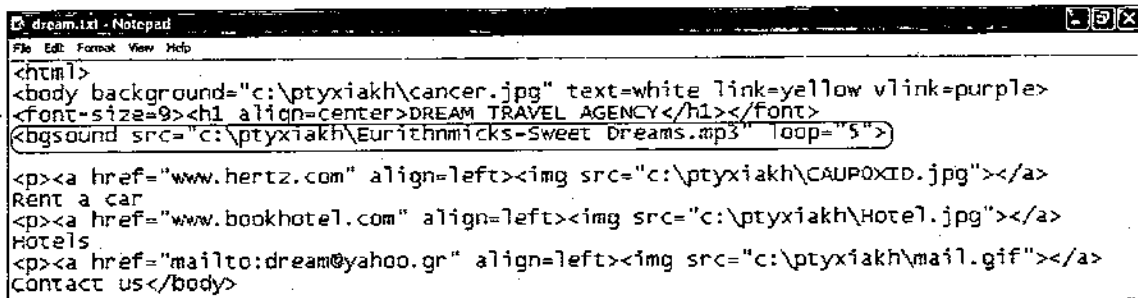
1.1.16 Αυτόματη ανανέωση μιας σελίδας (refresh)

Πολλές φορές υπάρχει ανάγκη να γίνεται ανανέωση των δεδομένων μιας σελίδας το οποίο επιτυγχάνεται με τις οδηγίες που αποκαλούνται **meta tags** και οι οποίες

τοποθετούνται μέσα στην εντολή head αμέσως μετά την title π.χ. **<META HTTP-EQUIV="refresh CONTENT=5;URL=http://host.gr/directory/filename.html">** όπου content=5 σημαίνει ότι η σελίδα ανανεώνεται κάθε 5 δευτερόλεπτα.

1.1.17 Εισαγωγή ήχου σε σελίδα

Σε μια σελίδα μπορούμε πολύ εύκολα να προσθέσουμε ήχο, ο οποίος θα ξεκινάει μόλις φορτωθεί η σελίδα στον browser. Η σύνταξη της οδηγίας αυτής θα είναι **<BGSOUND SRC="file.wav" loop="5">**, όπου file.wav είναι το όνομα του αρχείου και loop=5 σημαίνει πως το κομμάτι θα παιχτεί (επαναληφθεί) πέντε φορές.



```
dream.txt - Notepad
File Edit Format View Help
<HTML>
<body background="c:\ptyxiakh\cancer.jpg" text=white link=yellow vlink=purple>
<font-size=9><h1 align=center>DREAM TRAVEL AGENCY</h1></font>
<bgsound src="c:\ptyxiakh\Eurhythmicks-Sweet Dreams.mp3" loop="5">
<p><a href="www.hertz.com" align=left></a>
RENT a car
<p><a href="www.bookhotel.com" align=left></a>
hotels
<p><a href="mailto:dream@yahoo.gr" align=left></a>
CONTACT us</body>
```

Εικ.1.9 Προσθήκη ήχου στην σελίδα.

Για τη δημιουργία περισσότερων εντυπώσεων, με σκοπό δηλαδή τη σελίδα που δημιουργήσαμε να την θυμούνται οι χρήστες της, θεωρήσαμε πολύ εντυπωσιακό και ότι τραβάει αρκετά την προσοχή, εάν προσθέσουμε ήχο. Εάν η σελίδα μας λοιπόν προσπελαστεί από κάποιο web browser όταν θα εμφανιστεί στην οθόνη μας από πίσω θα ακούγεται το τραγούδι των Eurhythmics το Sweet Dreams. Το τραγούδι θα ακουστεί πέντε φορές γιατί έχουμε ορίσει ότι loop=5.

Στο κεφάλαιο αυτό αναλύσαμε και μάθαμε τα βασικά βήματα για τη δημιουργία μιας HTML σελίδας. Στο κεφάλαιο που ακολουθεί θα δημιουργήσουμε μια βάση δεδομένων η οποία στη συνέχεια θα συνδεθεί με την HTML σελίδα μας μέσω της γλώσσας PHP.

2. MySQL (Structured Query Language).

Η MySQL είναι ένα σύστημα διαχείρισης βάσεων δεδομένων (DBMS – Data Base Management System). Μια βάση δεδομένων είναι μια δομημένη συλλογή στοιχείων. Μπορεί να είναι οτιδήποτε, από έναν απλό κατάλογο αγορών μέχρι και μία συλλογή εικόνων ή απέραντα ποσά στοιχείων σε ένα εταιρικό δίκτυο. Για να έχουμε την ικανότητα να προσθέσουμε, να έχουμε πρόσβαση και να μπορούμε να διαχειριζόμαστε στοιχεία που αποθηκεύονται σε μια βάση δεδομένων σε έναν υπολογιστή, χρειάζεται ένα σύστημα διαχείρισης βάσεων δεδομένων, καθώς και ένας κεντρικός υπολογιστής στο διαδίκτυο στον οποίο θα εγκατασταθεί το κατάλληλο λογισμικό έτσι ώστε να χρησιμοποιηθεί ως web server, δηλαδή ένας υπολογιστής ο οποίος θα αναλάβει την αποθήκευση της βάσης δεδομένων. Στην περίπτωση της παρούσας εργασίας ως web server χρησιμοποιήθηκε το πρόγραμμα Apache. Επίσης για τη δημιουργία και επεξεργασία της βάσης δεδομένων, χρησιμοποιήθηκε το σύστημα διαχείρισης βάσεων δεδομένων MySQL. Σημειώνεται ότι τόσο ο Apache, όσο και η MySQL επιλέχθηκαν διότι είναι ελεύθερα λογισμικά, τα οποία μπορεί να αποκτήσει κάποιος δωρεάν.

Γεδομένου ότι οι υπολογιστές είναι πολύ καλοί στο χειρισμό πληροφοριών σε μεγάλες ποσότητες, τα συστήματα διαχείρισης βάσεων δεδομένων διαδραματίζουν έναν κεντρικό ρόλο στον κόσμο των υπολογιστών, ως αυτόνομες δραστηριότητες, ή ως μέρος άλλων εφαρμογών.

2.1 Η MySQL ως Σύστημα Διαχείρισης Βάσεων Δεδομένων.

Ένα σύστημα δημιουργίας και διαχείρισης βάσεων δεδομένων (όπως είναι η MySQL) αποθηκεύει τα στοιχεία σε χωριστούς πίνακες και δεν τα τοποθετεί απλά όλα μαζί σε ένα μεγάλο αποθηκευτικό χώρο. Αυτό αυξάνει την ταχύτητα και την ευελιξία της βάσης. Οι πίνακες συνδέονται με καθορισμένες σχέσεις δίνοντας έτσι τη δυνατότητα να συνδυαστούν στοιχεία από διάφορους πίνακες όταν αυτό απαιτείται. Το τμήμα SQL στο "MySQL" σημαίνει 'Δομημένη γλώσσα διατύπωσης ερωτήσεων'

ή 'Structured Query Language' και είναι η πιο κοινή τυποποιημένη γλώσσα που χρησιμοποιείται για πρόσβαση στις βάσεις δεδομένων.

Η MySQL όπως όλες οι γλώσσες προγραμματισμού διακρίνεται από κάποιους τύπους δεδομένων και από κάποιες βασικές εντολές για τη δημιουργία και διαχείριση βάσεων δεδομένων. Στο συγκεκριμένο κεφάλαιο θα αναφέρουμε και θα αναλύσουμε τους τύπους δεδομένων της MySQL καθώς και τις βασικές εντολές της και σε γενικό επίπεδο αλλά και όσον αφορά τη συγκεκριμένη πτυχιακή εργασία.

2.2 Οι Τύποι Δεδομένων της MySQL.

Η MySQL χρησιμοποιεί πολλούς και διαφορετικούς τύπους δεδομένων, οι οποίοι ταξινομούνται σε τρεις βασικές κατηγορίες:

1. Οι αριθμητικοί τύποι δεδομένων
2. Οι τύποι ημερομηνίας και ώρας και
3. Οι αλφαριθμητικοί τύποι δεδομένων

2.2.1 Οι Αριθμητικοί Τύποι Δεδομένων

Σε αυτήν την ενότητα θα παρουσιαστούν οι αριθμητικοί τύποι δεδομένων. Κατά την παρουσίασή τους θα χρησιμοποιηθούν οι όροι **προσημασμένος** και **μη-προσημασμένος**. Θεωρήσαμε λοιπόν σωστό να εξηγήσουμε τις δύο αυτές έννοιες. Προσημασμένος είναι ένας θετικός ή αρνητικός ακέραιος αριθμός (δηλαδή φέρει πρόσημο), ενώ μη-προσημασμένος είναι ένας μη αρνητικός ακέραιος αριθμός. Αφού επεξηγήσαμε τις δύο αυτές έννοιες, τώρα θα παρουσιάσουμε τους αριθμητικούς τύπους δεδομένων οι οποίοι είναι οι εξής οχτώ:

- **INT (INTEGER)** – Είναι ένας ακέραιος κανονικού μεγέθους ο οποίος μπορεί να είναι είτε προσημασμένος, είτε μη-προσημασμένος. Εάν είναι προσημασμένος το υποστηριζόμενο πεδίο τιμών είναι από **-2.147.483.648** έως **+2.147.483.648**. Εάν είναι μη-προσημασμένος το υποστηριζόμενο πεδίο τιμών είναι από **0** έως **4.294.967.295**. Το μέγιστο μήκος του είναι **11** ψηφία.

- **TINYINT** – Είναι ένας πολύ μικρός ακέραιος, ο οποίος όπως και ο INT μπορεί να είναι είτε προσημασμένος είτε μη-προσημασμένος, απλά αλλάζει το υποστηριζόμενο πεδίο τιμών, το οποίο για τον TINYINT είναι πολύ μικρό. Δηλαδή, εάν είναι προσημασμένος το υποστηριζόμενο πεδίο τιμών είναι από **-128** έως **+127**

και εάν είναι μη-προσημασμένος το υποστηριζόμενο πεδίο τιμών είναι από 0 έως 255. Το μέγιστο μήκος του είναι 3 ψηφία.

- **SMALLINT** – Είναι ένας μικρός ακέραιος για τον οποίο ισχύουν ό,τι και για τους INT και TINYINT. Η μόνη διαφορά είναι το υποστηριζόμενο πεδίο τιμών το οποίο εάν είναι προσημασμένος είναι από -32.768 έως +32.767 και εάν είναι μη-προσημασμένος από 0 έως 65535 και ότι το μέγιστο μήκος του αριθμού είναι 5 ψηφία.

Στην ίδια ομάδα ανήκουν και οι :

- **MEDIUMINT** (με 9 ψηφία) και

- **BIGINT** (με 11 ψηφία)

- **FLOAT (M,D)** – Είναι ένας αριθμός κινητής υποδιαστολής, ο οποίος δεν μπορεί να είναι μη-προσημασμένος. Με το (M) ορίζεται ο συνολικός αριθμός των ψηφίων (συμπεριλαμβανομένων και των δεκαδικών) και με το (D) ο αριθμός των δεκαδικών ψηφίων. Εάν δεν ορίσουμε τα δεδομένα αυτά, τότε αυτόματα παίρνουν τις τιμές 10,2. Η ακρίβεια μιας τιμής τύπου FLOAT μπορεί να φτάσει τα 24 δεκαδικά ψηφία.

- **DOUBLE (M,D)** – Είναι ένας αριθμός κινητής υποδιαστολής διπλής ακρίβειας, ο οποίος δεν μπορεί να είναι μη-προσημασμένος. Το (M) και το (D) είναι το πλήθος των συνολικών και το πλήθος των δεκαδικών ψηφίων, αντίστοιχα. Σε περίπτωση που δεν ορίσουμε τα (M,D) τότε αυτόματα θα πάρουν τις τιμές (16,4). Η ακρίβεια μιας τιμής τύπου DOUBLE μπορεί να φτάσει τα 53 δεκαδικά ψηφία. Ο όρος **REAL** είναι συνώνυμος του όρου DOUBLE.

- **DECIMAL (M,D)** – Είναι ένας μη-συμπίεσμένος αριθμός κινητής υποδιαστολής, ο οποίος δε μπορεί να είναι μη-προσημασμένος. Μη-συμπίεσμένος αριθμός τύπου DECIMAL σημαίνει ότι κάθε δεκαδικό ψηφίο αντιστοιχεί σε ένα *byte*. Στον αριθμό τύπου DECIMAL τα (M,D) είναι υποχρεωτικό να οριστούν. Το **NUMERIC** είναι ένα συνώνυμο για τον τύπο DECIMAL.

2.2.2 Τύποι Δεδομένων για Τιμές Ημερομηνίας και Ώρας

Η MySQL υποστηρίζει 5 τύπους δεδομένων για τις τιμές ημερομηνίας και ώρας οι οποίοι παρέχουν ένα μεγάλο βαθμό ευελιξίας στο χρήστη που θα εισάγει δεδομένα

τέτοιας μορφής. Η ευελιξία αυτή οφείλεται στο ότι η MySQL ελέγχει μόνο την εγκυρότητα δύο στοιχείων :

1. το ότι η τιμή για το μήνα ανήκει στο πεδίο τιμών 0-12 και
2. η τιμή για τον ημέρα ανήκει στο πεδίο τιμών 0-31.

Οι πέντε τύποι δεδομένων ημερομηνίας και ώρας που υποστηρίζει η MySQL είναι οι εξής :

- **DATE** – Είναι μία τιμή ημερομηνίας με τη μορφή **EEEE-MM-HH**, και δέχεται τιμές μεταξύ 1000-01-01 και 9999-12-31. Για παράδειγμα η ημερομηνία 04 Αυγούστου του 2000 θα αποθηκευτεί ως **2000-08-04**.

- **DATETIME** – Είναι ένας συνδυασμός τιμών ημερομηνίας και ώρας που παίρνει τη μορφή **EEEE-MM-HH ΩΩ:ΛΛ:ΔΔ** και τις τιμές μεταξύ 1000-01-01 00:00:00 και 9999-12-31 23:59:59. Για παράδειγμα, η ώρα 9:30 το βράδυ στις 04 Αυγούστου του 2000 αποθηκεύεται ως **2000-08-04 21:30:00**.

- **TIMESTAMP** – Είναι μια χρονική ένδειξη ανάμεσα στα μεσάνυχτα τις 1^{ης} Ιανουαρίου 1970 και στο έτος 2037. Με την **TIMESTAMP** έχουμε τη δυνατότητα να ορίζουμε πολλαπλά μήκη στο πεδίο τιμών της, το οποίο συσχετίζεται άμεσα με την οποιαδήποτε τιμή αποθηκεύεται μέσα σε αυτό. Το προεπιλεγμένο μήκος για τον τύπο δεδομένων της **TIMESTAMP** είναι 14 ψηφία και αποθηκεύει τιμές της μορφής **EEEEMMHHΩΩΛΛΔΔ**. Η μορφή αυτή δείχνει παρόμοια με την μορφή του τύπου της **TIMESTAMP**, χωρίς τις παύλες ανάμεσα στους αριθμούς. Για παράδειγμα έστω ότι η ώρα είναι 9:30 το βράδυ στις 04 Αυγούστου του 2000, αυτά τα δεδομένα θα αποθηκευτούν ως **20000804213000**. Άλλα μήκη του πεδίου τιμών της **TIMESTAMP** εκτός από το προεπιλεγμένο είναι αυτό με τα 12 ψηφία που έχει μορφή **EEMMHHΩΩΛΛΔΔ**, αυτό με τα 8 ψηφία και μορφή **EEEEMMHH** και τέλος αυτό με τα 6 ψηφία και την μορφή **EEMMHH**.

- **TIME** – Είναι ο τύπος δεδομένων που αποθηκεύουν την ώρα σε μορφή **ΩΩ:ΛΛ:ΔΔ**.

- **YEAR(M)** – Είναι ένας τύπος δεδομένων που αποθηκεύει τα έτη σε διψήφια ή τετραψήφια μορφή. Εάν η παράμετρος του μεγέθους, δηλαδή το (M), οριστεί δύο (πχ **YEAR(2)**) τότε το έτος μπορεί να είναι από 1970 μέχρι και 2069 (70 έως 69), εάν όμως το (M) οριστεί τέσσερα (**YEAR(4)**), τότε το έτος μπορεί να είναι από 1901 έως 2155. Το 4 είναι το προεπιλεγμένο μέγεθος του τύπου **YEAR(M)**.

Οι πιο συχνά χρησιμοποιημένοι τύποι σχετιζόμενοι με ημερομηνία και ώρα είναι συνήθως οι τύποι δεδομένων DATETIME και DATE.

Η τρίτη και τελευταία μορφή τύπων δεδομένων της MySQL είναι οι αλφαριθμητικοί τύποι δεδομένων, που είναι ίσως οι σημαντικότεροι τύποι της MySQL.

2.2.3 Οι Αλφαριθμητικοί Τύποι Δεδομένων

Οι αλφαριθμητικοί τύποι δεδομένων είναι περισσότερο σημαντικοί από τους αριθμητικούς τύπους και τους τύπους ημερομηνίας και ώρας, λόγω του ότι τα περισσότερα δεδομένα σε μια βάση αποθηκεύονται με τη μορφή αλφαριθμητικών (string). Οι πιο κοινοί τύποι δεδομένων για αλφαριθμητικά που υποστηρίζει η MySQL είναι οι εξής :

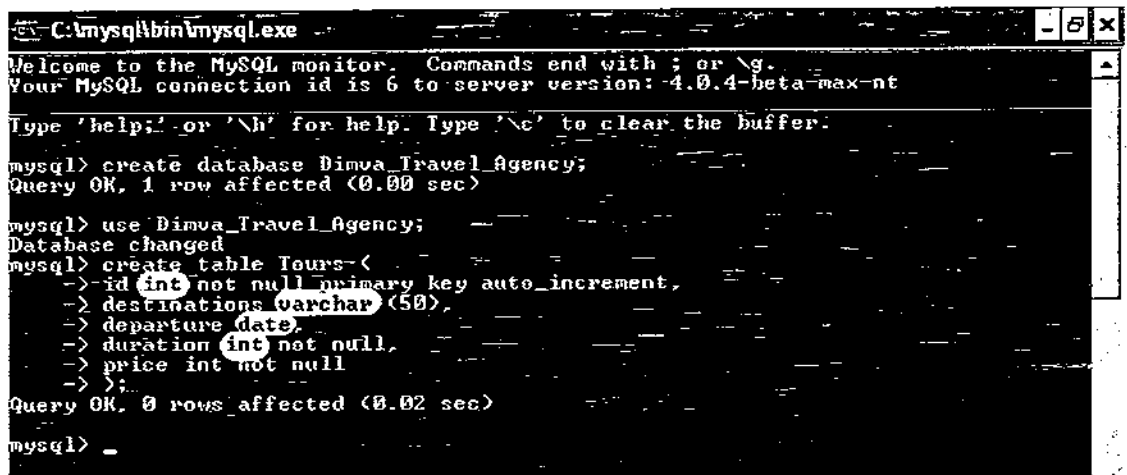
- **CHAR(M)** – Είναι ένα αλφαριθμητικό δεδομένο σταθερού μεγέθους, από 1 έως 255 χαρακτήρες. Έστω ότι έχουμε το CHAR(5), αυτό σημαίνει ότι σε έναν αριθμό θα του προστίθενται κενά διαστήματα στα δεξιά μέχρι να φτάσει στο προσδιοριζόμενο μήκος. Ο ορισμός του μήκους δεν είναι απαραίτητος, αλλά το προεπιλεγμένο μήκος είναι ένας χαρακτήρας.

- **VARCHAR(M)** – Είναι ένα αλφαριθμητικό μεταβλητού μεγέθους και μήκος από 1 έως 255 χαρακτήρες. Ο ορισμός της παραμέτρου του μήκους είναι απαραίτητος όταν δημιουργούμε ένα πεδίο VARCHAR.

- **BLOB** ή **TEXT** – Εδώ αναφερόμαστε σε δύο τύπους αλφαριθμητικών δεδομένων, που και οι δύο έχουν μέγεθος έως 65.535 χαρακτήρες. Ο όρος BLOB είναι ακρωνύμιο του Binary-Large Objects, που μεταφράζεται ως μεγάλα αντικείμενα δυαδικής μορφής. Ο τύπος BLOB χρησιμοποιείται για την αποθήκευση μεγάλων ποσοτήτων δυαδικών δεδομένων, όπως για παράδειγμα εικόνες ή και άλλους τύπους αρχείων. Όσον αφορά τον τύπο TEXT και αυτός χρησιμοποιείται για να αποθηκεύονται μεγάλες ποσότητες δεδομένων. Η μόνη διαφορά μεταξύ τους είναι ότι στα δεδομένα που αποθηκεύονται με την μορφή BLOB γίνεται διαχωρισμός ανάμεσα στους πεζούς και στους κεφαλαίους χαρακτήρες, σε αντίθεση με τα δεδομένα που αποθηκεύονται σε μορφή TEXT. Κανένας από τους δύο τύπους TEXT και BLOB, δεν απαιτεί τον ορισμό της παραμέτρου του μεγέθους. Υπάρχουν βέβαια και έτοιμοι τύποι TEXT και BLOB με καθορισμένο μήκος όπως είναι ο TINYBLOB ή TINYTEXT με μέγεθος μέχρι 255 χαρακτήρες, ο MEDIUMBLOB ή

MEDIUMTEXT με μέγιστο μήκος **16.777.215** χαρακτήρες και τέλος ο **LONGBLOB** ή **LONGTEXT** με **4.294.967.295** χαρακτήρες.

● **ENUM** — Είναι μία απαρίθμηση, δηλαδή μια λίστα. Όταν ορίζουμε έναν τύπο **ENUM** ουσιαστικά δημιουργείται μια λίστα από στοιχεία, από τα οποία πρέπει να επιλεγεί μια τιμή, η οποία μπορεί να είναι και η **NULL** (**NULL=0**/απουσία τιμής). Για παράδειγμα, εάν θέλαμε να ορίσουμε το πεδίο μας να περιέχει μία από τις τιμές 1, 2, 3, τότε θα έπρεπε να ορίσουμε **NULL('1', '2', '3')**, έτσι μόνο αυτές οι τιμές να μπορούν να εισαχθούν στο πεδίο. Ένας τύπος **ENUM** μπορεί να περιέχει μέχρι και **65.535** διαφορετικές τιμές. Ο τύπος **ENUM** ορίζεται ως δείκτης προς μία τιμή, όπως όλοι οι τύποι απαρίθμησης, σε αντίθεση με τον τύπο **SET** που αποθηκεύεται σαν μια πλήρης τιμή, παρά το γεγονός ότι και ο τύπος **SET** ορίζεται και αυτός ως λίστα.



```
C:\mysql\bin\mysql.exe
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6 to server version: 4.0.4-beta-max-nt

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> create database Dimva_Travel_Agency;
Query OK, 1 row affected (0.00 sec)

mysql> use Dimva_Travel_Agency;
Database changed
mysql> create table Tours-<
-> id int not null primary key auto_increment,
-> destinations varchar(50),
-> departure date,
-> duration int not null,
-> price int not null
-> );
Query OK, 0 rows affected (0.02 sec)

mysql> _
```

Εικ.2.1 Τύποι δεδομένων που χρησιμοποιήθηκαν στην εργασία.

Στη συνέχεια θα μελετήσουμε τις εντολές της **MySQL**. Προηγουμένως όμως θα ήταν σωστό να αναφερθεί ότι στη συγκεκριμένη εργασία χρησιμοποιήθηκαν ο αριθμητικός τύπος **int**, ο τύπος για τις τιμές της ημερομηνίας **date** και το αλφαριθμητικό **varchar**, όπως φαίνεται και στην εικόνα 2.1.

2.3 Οι Εντολές της **MySQL**

Αφού χωρίσαμε τους τύπους δεδομένων της **MySQL** σε τρεις βασικές κατηγορίες και αναλύσαμε τι περιέχει η κάθε μία, τώρα πρέπει να αναφερθούμε και στο πιο σημαντικό κομμάτι του προγραμματισμού, που είναι οι εντολές μιας γλώσσας και συγκεκριμένα της **MySQL**. Υπάρχουν πολλές εντολές που θα μπορούσε κανείς να

χρησιμοποιήσει για να δημιουργήσει, να διαχειριστεί και να επεξεργαστεί βάσεις δεδομένων. Εμείς στη συγκεκριμένη πτυχιακή εργασία θα παρουσιάσουμε τις βασικές εντολές της MySQL.

2.3.1 Η Εντολή Δημιουργίας Πινάκων

Η εντολή **CREATE** είναι αυτή που χρησιμοποιείται για τη δημιουργία πινάκων. Για να δημιουργήσουμε όμως έναν πίνακα, θα πρέπει πρώτα να δημιουργήσουμε μία βάση δεδομένων. Αυτό μπορεί να γίνει χρησιμοποιώντας πάλι την εντολή `create`. Έστω για παράδειγμα, ότι επιθυμούμε να δημιουργήσουμε έναν πίνακα με διάφορα δεδομένα, θα πρέπει πρώτα να πληκτρολογήσουμε την εντολή: **create database ονομα_βασης_δεδομένων;** και μετά η MySQL θα μας εμφανίσει το μήνυμα: **Query OK, 1 row affected < 0.00 sec>**.

Στη συνέχεια πληκτρολογώντας: **use ονομα_βασης_δεδομένων;** η MySQL θα μας εμφανίσει το μήνυμα: **Database changed**. Μετά και από αυτή την πολύ σημαντική εγγραφή είμαστε πλέον έτοιμοι να δημιουργήσουμε τον πίνακα.

Η βασική μορφή της εντολής για τη δημιουργία πινάκων είναι: **create table ονομα_πίνακα(όνομα_πεδίου_τύπος_πεδίου,...);**. Το όνομα του πίνακα είναι προσωπική επιλογή του καθενός, αλλά για μεγαλύτερη ευκολία καλό θα ήταν να χρησιμοποιηθεί κάποιο όνομα που να φανερώνει τη χρήση ή το ρόλο του πίνακα. Το ίδιο ισχύει και για τα ονόματα των πεδίων του πίνακα, δηλαδή για το όνομα της κάθε στήλης, θα πρέπει τα ονόματα να είναι περιγραφικά. Για παράδειγμα, εάν θέλουμε να φτιάξουμε μία βάση δεδομένων για ένα ταξιδιωτικό γραφείο, όπως στη συγκεκριμένη εργασία, καλό θα ήταν τον πίνακα να τον ονομάσουμε *Tours* και κάποιο από τα πεδία, που θα δείχνει τους προορισμούς, να το ονομάσουμε *destinations*. Στην παρούσα πτυχιακή, δημιουργήσαμε ένα πίνακα με το όνομα *Tours*, που περιείχε πέντε πεδία, το *id*, το *destinations*, το *departure*, το *duration* και το *price*. Ο τρόπος γραφής και δημιουργίας του πίνακα καθώς και της βάσης δεδομένων φαίνεται στην εικόνα 2.2.

```

C:\mysql\bin\mysql.exe
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 6 to server version: 4.0.4-beta-max-nt
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> create database Dimwa_Travel_Agency;
Query OK, 1 row affected (0.00 sec)

mysql> use Dimwa_Travel_Agency;
Database changed
mysql> create table Tours (
  -> id int not null primary key auto_increment,
  -> destinations varchar (50),
  -> departure date,
  -> duration int not null,
  -> price int not null
  -> );
Query OK, 0 rows affected (0.02 sec)

mysql> _

```

Εικ. 2.2 Δημιουργία βάσης δεδομένων – Δημιουργία πίνακα.

Το πεδίο με τον κωδικό *id* ορίζεται ως **πρωτεύων κλειδί** (primary key) και είναι ένα δεδομένο που απαριθμεί τις γραμμές των υπόλοιπων δεδομένων, για αυτό γράφουμε και το **auto_increment** (αυτόματη αύξηση τιμής), έτσι ώστε η MySQL να δίνει σε κάθε νέο δεδομένο τον επόμενο διαθέσιμο αριθμό. Ο MySQL server ανταποκρίνεται με το μήνυμα Query OK κάθε φορά που εκτελείται με επιτυχία μια εντολή. Σε κάθε άλλη περίπτωση εμφανίζεται ένα μήνυμα σφάλματος.

Το πρωτεύων κλειδί, δηλαδή το *id*, είναι ένα πεδίο που προσθέτουμε στους πίνακες που δημιουργούμε οπού-σε κάθε σειρά να εισάγετε ένας αριθμός μοναδικός ώστε να ξεχωρίζουν αντίστοιχα μοναδικά οι εγγραφές του πίνακα. Οι εγγραφές είναι τα στοιχεία που εισάγουμε στον πίνακα και στην συγκεκριμένη περίπτωση είναι κάποια ταξίδια τα οποία είναι προσφορές που κάνει το ταξιδιωτικό γραφείο που έχουμε δημιουργήσει στους πελάτες του και αφορούν δέκα διαφορετικούς προορισμούς. Σε μία γραμμή του πίνακα θα απεικονίζονται ο αριθμός *id* της εγγραφής, ο προορισμός, η ημερομηνία αναχώρησης, η διάρκεια του ταξιδιού και η τιμή του όπως φαίνεται και στην εικόνα 2.3 που ακολουθεί.

id	destinations	departure	duration	price
1	England	2005-01-10	5	1500
2	France	2005-01-10	7	1700
3	Italy	2005-01-10	3	700
4	Cyprus	2005-01-10	10	1300
5	U.S.A	2005-01-10	7	2200
6	Egypt	2005-01-10	5	1000
7	India	2005-01-10	8	2000
8	Australia	2005-01-10	10	3000
9	Netherlands	2005-01-10	7	1500
10	Germany	2005-01-10	9	1500

Εικ.2.3 Παράδειγμα πίνακα με 10 εγγραφές.

2.3.2 Η Εντολή Εισαγωγής Δεδομένων (INSERT)

Αφού δημιουργήσαμε τον πίνακά μας με την εντολή create πρέπει στη συνέχεια να του προσθέσουμε κάποια δεδομένα, δηλαδή εγγραφές. Καλό βέβαια θα ήταν να αναφέρουμε ότι γράφοντας στην MySQL *show fields from Tours;*, εμφανίζεται ένας πίνακας που μας δείχνει πόσα πεδία και με τι είδους δεδομένα έχουμε δημιουργήσει. Η εντολή που θα χρησιμοποιήσουμε για να εισάγουμε δεδομένα στον πίνακά μας είναι η **INSERT** και η γενική σύνταξή της είναι η:

insert into όνομα_πίνακα(λίστα πεδίων) values (τιμές πεδίων);

Όταν εισάγουμε ένα αλφαριθμητικό δεδομένο πρέπει πάντα να το γράφουμε μέσα σε εισαγωγικά. Εάν το αλφαριθμητικό αυτό περιέχει και από μόνο του εισαγωγικά τότε θα πρέπει εκτός από τα εισαγωγικά που θα του βάλουμε, να χρησιμοποιήσουμε και τον χαρακτήρα \ πριν από κάθε εισαγωγικό που περιέχει το αλφαριθμητικό. Για παράδειγμα, εάν θέλουμε να καταχωρήσουμε το αλφαριθμητικό *Mike said "Hello"*, θα πρέπει να το εισάγουμε γράφοντας:

insert into table_name (column_name) values ("Mike said \"Hello\");

Όταν χρησιμοποιούμε την εντολή insert δύο είναι τα βασικά της μέρη, οι στήλες και οι τιμές. Απαραίτητη για να χρησιμοποιήσουμε την εντολή insert είναι η λίστα των τιμών, ενώ εάν παραλείψουμε τη λίστα των στηλών δεν δημιουργείται πρόβλημα. Το μόνο που χρειάζεται να προσέξουμε είναι ότι πρέπει να δηλώσουμε τα δεδομένα με τη σειρά που έχουμε εισάγει και τα πεδία. Δηλαδή, μπορούμε να εισάγουμε κάποια δεδομένα είτε γράφοντας:

insert into Tours (id, destinations, departure, duration, price) values ('1', 'England', '2005-01-10', '5', '1500');

είτε πάλι γράφοντας αποκλειστικά και μόνο τα δεδομένα που θέλουμε να εισάγουμε, δηλαδή:

insert into Tours values ('1', 'England', '2005-01-10', '5', '1500');

Και στις δύο περιπτώσεις το μήνυμα που θα μας εμφανίσει η MySQL θα πρέπει να είναι το ίδιο: *Query OK, 1 row affected (0.01 sec)*.

Όπως είδαμε προηγουμένως, στην παρούσα πτυχιακή εργασία δημιουργήσαμε μία βάση δεδομένων με το όνομα *Dimva_Travel_Agency*, αφού ασχολούμαστε με ταξιδιωτικό γραφείο, και έναν πίνακα με το όνομα *Tours*, ο οποίος αποτελείται από πέντε πεδία. Τα πέντε αυτά πεδία είναι το *id* που όπως έχω αναφέρει είναι το πρωτεύων κλειδί, είναι το πεδίο *destination*, που περιέχει τους προορισμούς των

ταξιδιών-προσφορών, το πεδίο departure, που περιέχει τις ημερομηνίες αναχώρησης των ταξιδιών, το πεδίο duration το οποίο μας εμφανίζει την διάρκεια παραμονής στον κάθε προορισμό και τέλος το πεδίο price το οποίο περιέχει τις τιμές των προσφορών αυτών.

Χρησιμοποιώντας την εντολή insert προσθέσαμε δεδομένα στον πίνακα αυτό που αφορούσαν σε δέκα ταξιδιωτικές προσφορές του ταξιδιωτικού γραφείου, όπως μπορούμε να δούμε και στην εικόνα 2.3.

```

mysql> show fields from Tours;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id             | int(11)       |      | PRI | NULL     | auto_increment |
| destinations   | varchar(50)   | YES  |     | NULL     |                |
| departure      | date          | YES  |     | NULL     |                |
| duration       | int(11)       |      |     | 0        |                |
| price         | int(11)       |      |     | 0        |                |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> insert into Tours (id, destinations, departure, duration, price) values <
'1', 'England', '2005-01-10', '5', '1500');
Query OK, 1 row affected (0.00 sec)

mysql> insert into Tours (id, destinations, departure, duration, price) values <
'2', 'France', '2005-01-10', '7', '1700');
Query OK, 1 row affected (0.01 sec)

mysql> insert into Tours (id, destinations, departure, duration, price) values <
'3', 'Italy', '2005-01-10', '3', '700');
Query OK, 1 row affected (0.00 sec)

mysql> insert into Tours (id, destinations, departure, duration, price) values <
'4', 'Cyprus', '2005-01-10', '10', '1300');
Query OK, 1 row affected (0.00 sec)

mysql> insert into Tours (id, destinations, departure, duration, price) values <
'5', 'U.S.A', '2005-01-10', '7', '2200');
Query OK, 1 row affected (0.00 sec)

mysql> insert into Tours (id, destinations, departure, duration, price) values <
'6', 'Egypt', '2005-01-10', '5', '1000');
Query OK, 1 row affected (0.00 sec)

mysql> insert into Tours (id, destinations, departure, duration, price) values <
'7', 'India', '2005-01-10', '8', '2000');
Query OK, 1 row affected (0.00 sec)

mysql> insert into Tours (id, destinations, departure, duration, price) values <
'8', 'Australia', '2005-01-10', '10', '3000');
Query OK, 1 row affected (0.00 sec)

mysql> insert into Tours (id, destinations, departure, duration, price) values <
'9', 'Netherlands', '2005-01-10', '7', '1500');
Query OK, 1 row affected (0.00 sec)

mysql> insert into Tours (id, destinations, departure, duration, price) values <
'10', 'Germany', '2005-01-10', '9', '1500');
Query OK, 1 row affected (0.00 sec)

mysql>

```

Εικ.2.3 Εισαγωγή δεδομένων με την χρήση της εντολής insert.

Όπως μπορούμε να διακρίνουμε στην εικόνα 2.3 στο πάνω μέρος έχουμε κάνει χρήση της εντολής `show fields from Tours;` και το αποτέλεσμα ήταν να μας εμφανίσει όλες τις πληροφορίες για τον πίνακα, όσον αφορά τα δεδομένα που έχουμε ορίσει ότι θα εισάγουμε σε κάθε πεδίο και τα ονόματα των ίδιων των πεδίων.

2.3.3 Η Εντολή SELECT

Η εντολή **SELECT** είναι μια εντολή της MySQL η οποία χρησιμοποιείται για την **ανάκτηση** δεδομένων από ένα πίνακα. Η εντολή `select` μπορεί να γίνει από πολύ απλή μέχρι αρκετά πολύπλοκη. Η γενική της σύνταξη, και η πιο απλουστευμένη από τις υπόλοιπες, είναι η: `select εκφράσεις_και_πεδία from όνομα_πίνακα;`

Εάν θέλουμε για παράδειγμα, να δούμε τη μέχρι τώρα πρόοδο της εργασίας μας, δηλαδή τις καταχωρημένες εγγραφές, μπορούμε να γράψουμε την εντολή: `select id, destinations, departure, duration, price from Tours;`

```

C:\mysql\bin\mysql.exe
'6', 'Egypt', '2005-01-10', '5', '1000');
Query OK, 1 row affected (0.00 sec)

mysql> insert into Tours (id, destinations, departure, duration, price) values (<
'7', 'India', '2005-01-10', '8', '2000');
Query OK, 1 row affected (0.00 sec)

mysql> insert into Tours (id, destinations, departure, duration, price) values (<
'8', 'Australia', '2005-01-10', '10', '3000');
Query OK, 1 row affected (0.00 sec)

mysql> insert into Tours (id, destinations, departure, duration, price) values (<
'9', 'Netherlands', '2005-01-10', '7', '1500');
Query OK, 1 row affected (0.00 sec)

mysql> insert into Tours (id, destinations, departure, duration, price) values (<
'10', 'Germany', '2005-01-10', '9', '1500');
Query OK, 1 row affected (0.00 sec)

mysql> select id, destinations, departure, duration, price from Tours;
+----+-----+-----+-----+-----+
| id | destinations | departure | duration | price |
+----+-----+-----+-----+-----+
| 1 | England      | 2005-01-10 | 5       | 1500  |
| 2 | France       | 2005-01-10 | 7       | 1700  |
| 3 | Italy         | 2005-01-10 | 3       | 700   |
| 4 | Cyprus       | 2005-01-10 | 10      | 1300  |
| 5 | U.S.A        | 2005-01-10 | 7       | 2200  |
| 6 | Egypt        | 2005-01-10 | 5       | 1000  |
| 7 | India         | 2005-01-10 | 8       | 2000  |
| 8 | Australia    | 2005-01-10 | 10      | 3000  |
| 9 | Netherlands  | 2005-01-10 | 7       | 1500  |
| 10| Germany      | 2005-01-10 | 9       | 1500  |
+----+-----+-----+-----+-----+
10 rows in set (0.00 sec)

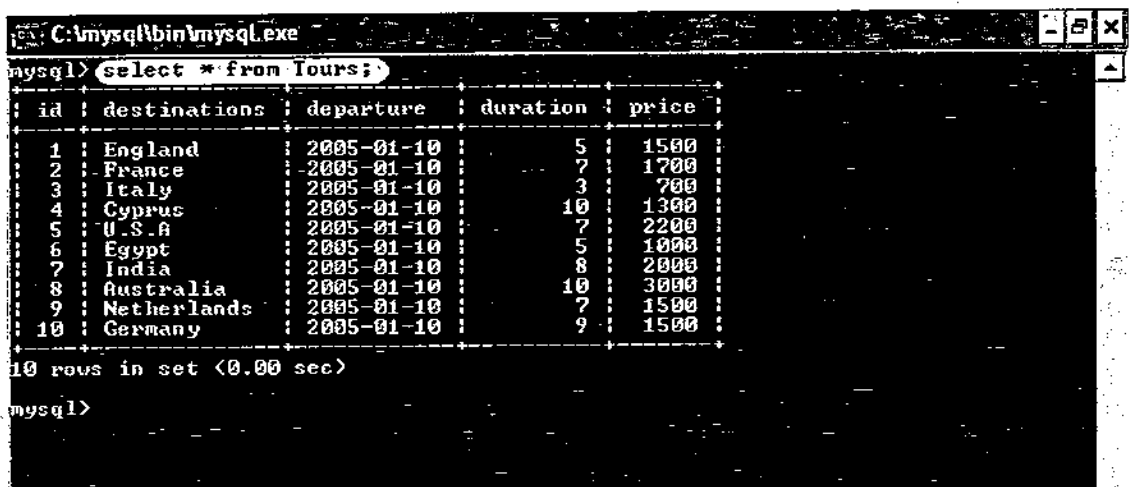
mysql> _

```

Εικ. 2.4 Χρήση της εντολής `select` για να δούμε τι έχουμε φτιάξει.

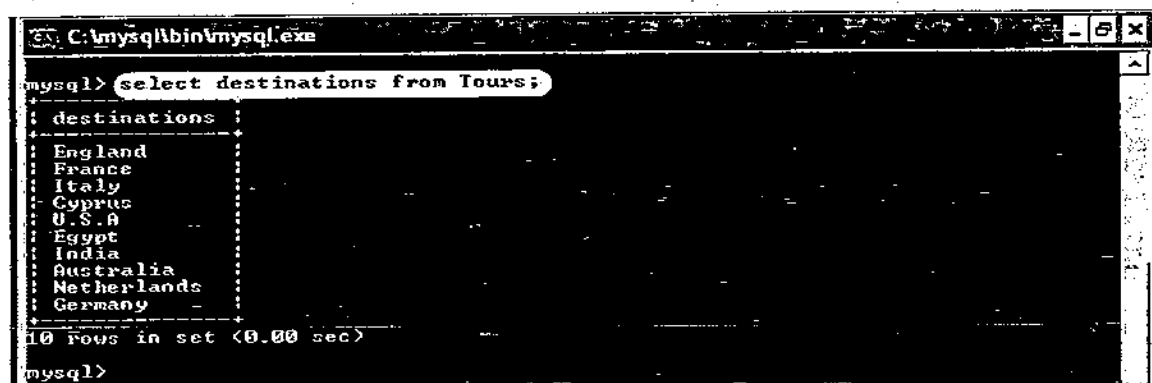
Όπως φαίνεται και στην εικόνα 2.4 αυτή η εντολή μας εμφανίζει όλα τα δεδομένα που έχουμε εισάγει στον πίνακα που έχουμε δημιουργήσει. Δεν είναι αναγκαίο κάθε

φορά που επιθυμούμε να δούμε τα δεδομένα του πίνακα να γράφουμε την εντολή ολόκληρη, ένα πολύ χρήσιμο και βολικό σύμβολο είναι το * (αστερίσκος) το οποίο αντιπροσωπεύει τα πάντα. Οπότε εάν θέλουμε να διαλέξουμε τα πάντα από έναν πίνακα αρκεί να γράψουμε την εντολή με την εξής μορφή: *select * from Tours;*



Εικ.2.5 Χρήση του συμβόλου * για συντομία.

Η εντολή select έχει επίσης τη δυνατότητα να ταξινομεί τα αποτελέσματα που μας εμφανίζει, που μας επιστρέφει στην οθόνη. Για παράδειγμα μπορούμε να διαλέξουμε μόνο μία στήλη από τον πίνακά μας.

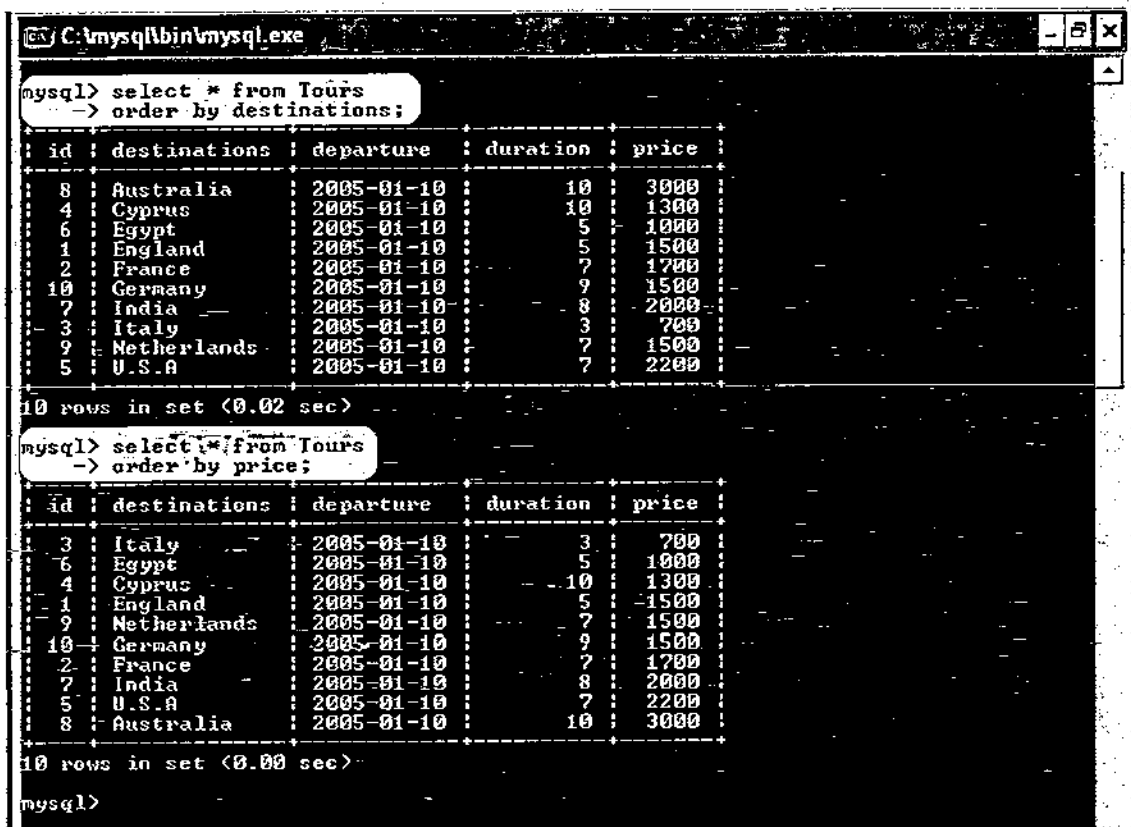


Εικ.2.6 Επιλογή δεδομένων που εμφανίζονται στην οθόνη.

Όπως φαίνεται και στην εικόνα 2.6 γράφοντας select και την στήλη που επιθυμούμε να εμφανίσουμε και μετά from και τον πίνακα η MySQL μας εμφανίζει τα συγκεκριμένα πεδία.

Υπάρχουν και κάποια ορίσματα τα οποία βοηθούν την εντολή select στο να εμφανίσει με καλύτερο τρόπο τα δεδομένα ενός πίνακα.

Το όρισμα **ORDER BY** μας βοηθάει να ταξινομήσουμε τις εγγραφές μας σε κατηγορίες, δηλαδή να εμφανίζονται κατά destinations ή price ή και τις υπόλοιπες στήλες. Η εντολή select μαζί με το όρισμα order by συντάσσεται όπως φαίνεται στην εικόνα 2.7.



```
mysql> select * from Tours
-> order by destinations;

+----+-----+-----+-----+-----+
| id | destinations | departure | duration | price |
+----+-----+-----+-----+-----+
| 8  | Australia   | 2005-01-10 | 10      | 3000  |
| 4  | Cyprus      | 2005-01-10 | 10      | 1300  |
| 6  | Egypt       | 2005-01-10 | 5       | 1000  |
| 1  | England     | 2005-01-10 | 5       | 1500  |
| 2  | France      | 2005-01-10 | 7       | 1700  |
| 10 | Germany     | 2005-01-10 | 9       | 1500  |
| 7  | India       | 2005-01-10 | 8       | 2000  |
| 3  | Italy        | 2005-01-10 | 3       | 700   |
| 9  | Netherlands | 2005-01-10 | 7       | 1500  |
| 5  | U.S.A       | 2005-01-10 | 7       | 2200  |
+----+-----+-----+-----+-----+
10 rows in set (0.02 sec)

mysql> select * from Tours
-> order by price;

+----+-----+-----+-----+-----+
| id | destinations | departure | duration | price |
+----+-----+-----+-----+-----+
| 3  | Italy        | 2005-01-10 | 3       | 700   |
| 6  | Egypt       | 2005-01-10 | 5       | 1000  |
| 4  | Cyprus      | 2005-01-10 | 10      | 1300  |
| 1  | England     | 2005-01-10 | 5       | 1500  |
| 9  | Netherlands | 2005-01-10 | 7       | 1500  |
| 10 | Germany     | 2005-01-10 | 9       | 1500  |
| 2  | France      | 2005-01-10 | 7       | 1700  |
| 7  | India       | 2005-01-10 | 8       | 2000  |
| 5  | U.S.A       | 2005-01-10 | 7       | 2200  |
| 8  | Australia   | 2005-01-10 | 10      | 3000  |
+----+-----+-----+-----+-----+
10 rows in set (0.00 sec)

mysql>
```

Εικ.2.7 Ταξινόμηση αποτελεσμάτων σύμφωνα με την τιμή και τον προορισμό.

Όπως μπορούμε να διακρίνουμε αυτή η μορφή της εντολή select μας εμφανίζει τους προορισμούς κατά αλφαβητική σειρά και τις τιμές από τη μικρότερη προς τη μεγαλύτερη.

Υπάρχει ακόμη το όρισμα **WHERE** το οποίο ακολουθούμενο από μία συνθήκη μπορεί να συγκεκριμενοποιήσει περισσότερο τις πληροφορίες που θα εμφανίζονται στην οθόνη της MySQL, δηλαδή ενώ ως τώρα μάθαμε να κατηγοριοποιούμε κατά στήλη, τώρα η φράση αυτή θα μας βοηθήσει να εμφανίσουμε μόνο όποιες γραμμές επιθυμούμε. Όπως θα δούμε και στην εικόνα 2.8 μπορούμε να εμφανίσουμε για

παράδειγμα τις εγγραφές που ο προορισμός ξεκινάει από *E* ή που η τιμή να κυμαίνεται σε κάποιο συγκεκριμένο όριο (πχ. από 700 € μέχρι 1.500 €).

```
C:\mysql\bin\mysql.exe
mysql> select id, destinations, departure, duration, price from Tours;
+----+-----+-----+-----+-----+
| id | destinations | departure | duration | price |
+----+-----+-----+-----+-----+
| 1  | England      | 2005-01-10 | 5        | 1500  |
| 2  | France       | 2005-01-10 | 7        | 1700  |
| 3  | Italy        | 2005-01-10 | 3        | 700   |
| 4  | Cyprus      | 2005-01-10 | 10       | 1300  |
| 5  | U.S.A       | 2005-01-10 | 7        | 2200  |
| 6  | Egypt       | 2005-01-10 | 5        | 1000  |
| 7  | India       | 2005-01-10 | 8        | 2000  |
| 8  | Australia   | 2005-01-10 | 10       | 3000  |
| 9  | Netherlands | 2005-01-10 | 7        | 1500  |
| 10 | Germany     | 2005-01-10 | 9        | 1500  |
+----+-----+-----+-----+-----+
10 rows in set (0.00 sec)

mysql> select * from Tours where price
-> between 700 and 1500;
+----+-----+-----+-----+-----+
| id | destinations | departure | duration | price |
+----+-----+-----+-----+-----+
| 1  | England      | 2005-01-10 | 5        | 1500  |
| 3  | Italy        | 2005-01-10 | 3        | 700   |
| 4  | Cyprus      | 2005-01-10 | 10       | 1300  |
| 6  | Egypt       | 2005-01-10 | 5        | 1000  |
| 9  | Netherlands | 2005-01-10 | 7        | 1500  |
| 10 | Germany     | 2005-01-10 | 9        | 1500  |
+----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> select * from Tours where destinations like 'E%';
+----+-----+-----+-----+-----+
| id | destinations | departure | duration | price |
+----+-----+-----+-----+-----+
| 1  | England      | 2005-01-10 | 5        | 1500  |
| 6  | Egypt       | 2005-01-10 | 5        | 1000  |
+----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> _
```

Εικ.2.8 Εμφάνιση συγκεκριμένων αποτελεσμάτων με την χρήση της πρότασης where.

Στο όρισμα where μπορούν να χρησιμοποιηθούν κάποια σύμβολα και κάποιοι συνδυασμοί τους τα οποία ονομάζονται τελεστές και μας βοηθούν να συγκεκριμενοποιήσουμε το where ακόμα πιο πολύ. Συγκεκριμένα υπάρχουν δύο κατηγορίες τελεστών, οι οποίες είναι οι τελεστές σύγκρισης και οι λογικοί τελεστές. Οι βασικοί τελεστές σύγκρισης και η σημασία τους εμφανίζονται στον παρακάτω πίνακα.

ΤΕΛΕΣΤΗΣ	ΣΗΜΑΣΙΑ
=	Ίσο με
!=	Διάφορο
<=	Μικρότερο από ή ίσο με
<	Μικρότερο από
>=	Μεγαλύτερο από ή ίσο με
>	Μεγαλύτερο από

Υπάρχει επίσης ο τελεστής **between** ο οποίος είναι χρήσιμος για συγκρίσεις ακεραίων και εμφανίζει τα αποτελέσματα μεταξύ μιας ελάχιστης και μιας μέγιστης τιμής όπως παρατηρήσαμε και στο πρώτο παράδειγμα της εικόνας 2.8.

Όσον αφορά τους λογικούς τελεστές, οι βασικοί είναι ο **AND** (σύζευξη - ΚΑΙ) και ο **OR** (διάζευξη - Ή). Η διαφορά των δύο αυτών τελεστών είναι ότι όταν χρησιμοποιούμε τον τελεστή AND πρέπει οι συγκρίσεις της πρότασης where να αποτιμηθούν σαν αληθείς, ενώ με τον τελεστή OR απαιτείται να είναι αληθής τουλάχιστον μία από τις συγκρίσεις της πρότασης where. Χρησιμοποιώντας την πρόταση where μπορούμε επίσης να συγκρίνουμε αλφαριθμητικά δεδομένα. Πολύ χρήσιμος τελεστής για την ιδιότητα αυτή της where είναι ο **like** ο οποίος χρησιμοποιεί και δύο σύμβολα μπαλαντέρ που είναι το % και το _. Το _ σημαίνει ότι αυτό που ψάχνουμε ταιριάζει με έναν ακριβώς χαρακτήρα μέσα στον πίνακα με τα δεδομένα μας, ενώ το % σημαίνει ότι ταιριάζει με πολλαπλούς χαρακτήρες από τα δεδομένα μας. Ένα παράδειγμα χρήσης του τελεστή like εμφανίζεται στο δεύτερο παράδειγμα της εικόνας 2.8.

2.3.4 Η Εντολή UPDATE

Η εντολή **UPDATE** είναι μια εντολή της MySQL η οποία χρησιμοποιείται για την τροποποίηση του περιεχομένου ενός ή και περισσότερων πεδίων στα δεδομένα ενός πίνακα.

Από τις πιο απλές συντάξεις της update είναι, εάν για παράδειγμα έχουμε δημιουργήσει τον πίνακα και θέλουμε να αλλάξουμε ΟΛΕΣ τις ημερομηνίες αναχώρησης, η:

```
update Tours set departure = '2005-01-11'.
```

Όπως μπορούμε να παρατηρήσουμε και στην εικόνα 2.9 γράφοντας την εντολή αυτή στο περιβάλλον της MySQL αλλάξαμε όλες τις ημερομηνίες αναχώρησης για όλους τους προορισμούς ανεξαιρέτως.

```
C:\mysql\bin\mysql.exe
mysql> select * from Tours;
+----+-----+-----+-----+-----+
| id | destinations | departure | duration | price |
+----+-----+-----+-----+-----+
| 1  | England      | 2005-01-10 | 5        | 1500  |
| 2  | France       | 2005-01-10 | 7        | 1700  |
| 3  | Italy         | 2005-01-10 | 3        | 700   |
| 4  | Cyprus       | 2005-01-10 | 10       | 1300  |
| 5  | U.S.A        | 2005-01-10 | 7        | 2200  |
| 6  | Egypt        | 2005-01-10 | 5        | 1000  |
| 7  | India        | 2005-01-10 | 8        | 2000  |
| 8  | Australia    | 2005-01-10 | 10       | 3000  |
| 9  | Netherlands  | 2005-01-10 | 7        | 1500  |
| 10 | Germany      | 2005-01-10 | 9        | 1500  |
+----+-----+-----+-----+-----+
10 rows in set (0.00 sec)

mysql> update Tours set departure = '2005-01-11';
Query OK, 10 rows affected (0.00 sec)
Rows matched: 10  Changed: 10  Warnings: 0

mysql> select * from Tours;
+----+-----+-----+-----+-----+
| id | destinations | departure | duration | price |
+----+-----+-----+-----+-----+
| 1  | England      | 2005-01-11 | 5        | 1500  |
| 2  | France       | 2005-01-11 | 7        | 1700  |
| 3  | Italy         | 2005-01-11 | 3        | 700   |
| 4  | Cyprus       | 2005-01-11 | 10       | 1300  |
| 5  | U.S.A        | 2005-01-11 | 7        | 2200  |
| 6  | Egypt        | 2005-01-11 | 5        | 1000  |
| 7  | India        | 2005-01-11 | 8        | 2000  |
| 8  | Australia    | 2005-01-11 | 10       | 3000  |
| 9  | Netherlands  | 2005-01-11 | 7        | 1500  |
| 10 | Germany      | 2005-01-11 | 9        | 1500  |
+----+-----+-----+-----+-----+
```

Εικ.2.9 Τροποποίηση δεδομένων με την χρήση της εντολής update.

Με την εντολή update όμως μπορούμε να κάνουμε και επιλεγμένη τροποποίηση των δεδομένων απλά χρησιμοποιώντας, όπως και στην εντολή select, το όρισμα where. Οπότε η εντολή update, εάν θέλουμε για παράδειγμα να αλλάξουμε έναν προορισμό μπορεί να πάρει τη μορφή:

```
update Tours set destinations = 'United Kingdom' where destinations = 'England';
```

Όπως μπορούμε να παρατηρήσουμε και στην εικόνα 2.10 με τη χρήση της πρότασης where τροποποιήσαμε μόνο μια εγγραφή από τον πίνακα μας και όχι ολόκληρη στήλη (δηλαδή, όλες τις εγγραφές). Αυτό που κάναμε βασικά ήταν να αντικαταστήσουμε μια εγγραφή με μία άλλη.

Έχουμε όμως τη δυνατότητα να κάνουμε και πράξεις για να ικανοποιήσουμε μια τροποποίηση γράφοντας την εντολή update με τη μορφή του ακόλουθου παραδείγματος (βλέπε εικόνα 2.11):

update Tours set price = price - 1 where id = 1;

```

C:\mysql\bin\mysql.exe
mysql>
mysql>
mysql> select * from Tours;
+----+-----+-----+-----+-----+
| id | destinations | departure | duration | price |
+----+-----+-----+-----+-----+
| 1  | England      | 2005-01-10 | 5        | 1500  |
| 2  | France       | 2005-01-10 | 7        | 1700  |
| 3  | Italy        | 2005-01-10 | 3        | 700   |
| 4  | Cyprus       | 2005-01-10 | 10       | 1300  |
| 5  | U.S.A       | 2005-01-10 | 7        | 2200  |
| 6  | Egypt       | 2005-01-10 | 5        | 1000  |
| 7  | India        | 2005-01-10 | 8        | 2000  |
| 8  | Australia   | 2005-01-10 | 10       | 3000  |
| 9  | Netherlands | 2005-01-10 | 7        | 1500  |
| 10 | Germany     | 2005-01-10 | 9        | 1500  |
+----+-----+-----+-----+-----+
10 rows in set (0.00 sec)

mysql> update Tours set destinations = 'United Kingdom' where destinations = 'England';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from Tours;
+----+-----+-----+-----+-----+
| id | destinations | departure | duration | price |
+----+-----+-----+-----+-----+
| 1  | United Kingdom | 2005-01-10 | 5        | 1500  |
| 2  | France       | 2005-01-10 | 7        | 1700  |
| 3  | Italy        | 2005-01-10 | 3        | 700   |
| 4  | Cyprus       | 2005-01-10 | 10       | 1300  |
| 5  | U.S.A       | 2005-01-10 | 7        | 2200  |
| 6  | Egypt       | 2005-01-10 | 5        | 1000  |
| 7  | India        | 2005-01-10 | 8        | 2000  |
| 8  | Australia   | 2005-01-10 | 10       | 3000  |
| 9  | Netherlands | 2005-01-10 | 7        | 1500  |
| 10 | Germany     | 2005-01-10 | 9        | 1500  |
+----+-----+-----+-----+-----+
10 rows in set (0.00 sec)

mysql> _

```

Εικ.2.10 Τροποποίηση δεδομένων με την χρήση της update και της πρότασης where.

```

C:\mysql\bin\mysql.exe
mysql> select * from Tours;
+----+-----+-----+-----+-----+
| id | destinations | departure | duration | price |
+----+-----+-----+-----+-----+
| 1  | England      | 2005-01-10 | 5         | 1500  |
| 2  | France       | 2005-01-10 | 7         | 1700  |
| 3  | Italy        | 2005-01-10 | 3         | 700   |
| 4  | Cyprus      | 2005-01-10 | 10        | 1300  |
| 5  | U.S.A       | 2005-01-10 | 7         | 2200  |
| 6  | Egypt       | 2005-01-10 | 5         | 1000  |
| 7  | India       | 2005-01-10 | 8         | 2000  |
| 8  | Australia   | 2005-01-10 | 10        | 3000  |
| 9  | Netherlands | 2005-01-10 | 7         | 1500  |
| 10 | Germany     | 2005-01-10 | 9         | 1500  |
+----+-----+-----+-----+-----+
10 rows in set (0.16 sec)

mysql> update Tours set price = price - 1 where id = 1;
Query OK, 1 row affected (0.03 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from Tours
->
+----+-----+-----+-----+-----+
| id | destinations | departure | duration | price |
+----+-----+-----+-----+-----+
| 1  | England      | 2005-01-10 | 5         | 1499  |
| 2  | France       | 2005-01-10 | 7         | 1700  |
| 3  | Italy        | 2005-01-10 | 3         | 700   |
| 4  | Cyprus      | 2005-01-10 | 10        | 1300  |
| 5  | U.S.A       | 2005-01-10 | 7         | 2200  |
| 6  | Egypt       | 2005-01-10 | 5         | 1000  |
| 7  | India       | 2005-01-10 | 8         | 2000  |
| 8  | Australia   | 2005-01-10 | 10        | 3000  |
| 9  | Netherlands | 2005-01-10 | 7         | 1500  |
| 10 | Germany     | 2005-01-10 | 9         | 1500  |
+----+-----+-----+-----+-----+
10 rows in set (0.02 sec)

mysql>

```

Εικ.2.11 Ενημέρωση δεδομένων του πίνακα με την εντολή update και την πρόταση where.

Εράφοντας λοιπόν την εντολή με τη μορφή που εμφανίζεται στην εικόνα 2.11 η τιμή που αντιστοιχεί στο ταξίδι στην Αγγλία (εγγραφή με id=1) και είναι 1500 € έγινε 1499 €.

2.3.5 Η Εντολή JOIN

Υπάρχουν πολλοί τρόποι ενώσεων στη MySQL, όλοι εκ των οποίων αναφέρονται στη σειρά με την οποία ενώνονται/συσχετίζονται διαφορετικοί πίνακες μαζί και τη σειρά με την οποία εμφανίζονται τα αποτελέσματα. Ο τύπος της ένωσης ο οποίος μάθαμε, αλλά δεν χρησιμοποιήσαμε στη συγκεκριμένη εργασία για τον λόγο του ότι δεν απαιτήθηκε κατά την υλοποίηση, ενώνει δύο πίνακες μαζί. Για παράδειγμα, έστω ότι είχαμε δημιουργήσει έναν πίνακα με ονόματα φρούτων και ένα με χρώματα, οι οποίοι και οι δύο αποτελούνται από δύο πεδία, τα χρώματα ή τα φρούτα αντίστοιχα και με τον αύξοντα αριθμό των εισαγωγών (id), όπως φαίνονται και στο σχέδιο που ακολουθεί.

id	fruitname
1	apple
2	orange
3	grape
4	banana

Πίνακας
Φρούτων

id	colorname
1	red
2	orange
3	purple
4	yellow

Πίνακας
Χρωμάτων

Η ένωση που θα χρησιμοποιήσουμε λέγεται **εσωτερική ένωση** (inner join) και η γενική σύνταξή της είναι αυτή που απεικονίζεται στο παρακάτω σχέδιο.

```
mysql> select fruitname, colorname from fruit inner join color
      ~> on fruit.id = color.id;
```

fruitname	colorname
apple	red
orange	orange
grape	purple
banana	yellow

Όπως μπορούμε να διακρίνουμε και από το σχέδιο, με την εντολή αυτή ενώσαμε το πεδίο με τα ονόματα των φρούτων, με το πεδίο με τα χρώματα, σύμφωνα με τον αύξοντα αριθμό τους. Εκτός από την inner join υπάρχουν και οι **left join**, **right join**, **cross join**, **straight join** και **natural join**.

2.3.6 Η Εντολή REPLACE

Η εντολή **REPLACE** είναι άλλη μία εντολή τροποποίησης πινάκων της MySQL, η οποία όμως είναι μια συνδυασμένη χρήση της εντολής **delete** (βλέπε 2.3.7) και της εντολής **insert**. Δηλαδή, χρησιμοποιώντας την εντολή replace διαγράφουμε μια εγγραφή και την αντικαθιστούμε εισάγοντας μια άλλη εγγραφή. Η σύνταξη της replace είναι της μορφής:

```
replace into όνομα_πίνακα values (τιμές_στηλών);
```

Στην εργασία μας, για παράδειγμα, ας υποθέσουμε ότι θέλουμε να αντικαταστήσουμε το ταξίδι με αύξοντα αριθμό 1 στην Αγγλία (επειδή καταργήθηκε) με ένα ταξίδι άλλο στην Ρωσία. Η ακόλουθη εντολή θα υλοποιήσει το συγκεκριμένο παράδειγμα: *replace into Tours values ('1', 'Russia', '2005-01-10', '7', '1000');*

```
C:\mysql\bin\mysql.exe
mysql> select * from Tours;
+----+-----+-----+-----+-----+
| id | destinations | departure | duration | price |
+----+-----+-----+-----+-----+
| 1 | England      | 2005-01-10 | 5        | 1500  |
| 2 | France       | 2005-01-10 | 7        | 1700  |
| 3 | Italy         | 2005-01-10 | 3        | 700   |
| 4 | Cyprus       | 2005-01-10 | 10       | 1300  |
| 5 | U.S.A        | 2005-01-10 | 7        | 2200  |
| 6 | Egypt        | 2005-01-10 | 5        | 1000  |
| 7 | India         | 2005-01-10 | 8        | 2000  |
| 8 | Australia    | 2005-01-10 | 10       | 3000  |
| 9 | Netherlands  | 2005-01-10 | 7        | 1500  |
| 10 | Germany      | 2005-01-10 | 9        | 1500  |
+----+-----+-----+-----+-----+
10 rows in set (0.00 sec)

mysql> replace into Tours values
-> ('1', 'Russia', '2005-01-10', '7', '1000');
Query OK, 2 rows affected (0.00 sec)

mysql> select * from Tours;
+----+-----+-----+-----+-----+
| id | destinations | departure | duration | price |
+----+-----+-----+-----+-----+
| 1 | Russia       | 2005-01-10 | 7        | 1000  |
| 2 | France       | 2005-01-10 | 7        | 1700  |
| 3 | Italy         | 2005-01-10 | 3        | 700   |
| 4 | Cyprus       | 2005-01-10 | 10       | 1300  |
| 5 | U.S.A        | 2005-01-10 | 7        | 2200  |
| 6 | Egypt        | 2005-01-10 | 5        | 1000  |
| 7 | India         | 2005-01-10 | 8        | 2000  |
| 8 | Australia    | 2005-01-10 | 10       | 3000  |
| 9 | Netherlands  | 2005-01-10 | 7        | 1500  |
| 10 | Germany      | 2005-01-10 | 9        | 1500  |
+----+-----+-----+-----+-----+
10 rows in set (0.00 sec)

mysql>
```

Εικ.2.12 Αντικατάσταση δεδομένων με την χρήση της εντολής *replace*.

Όπως μπορούμε να παρατηρήσουμε η πρώτη γραμμή του πίνακά μας αντικαταστήθηκε από μια νέα η οποία έχει τη δικιά της τιμή, διάρκεια, ημερομηνία, προορισμό και αύξοντα αριθμό.

2.3.7 Η Εντολή DELETE

Η τελευταία εντολή της MySQL που θα αναφέρουμε είναι η εντολή **DELETE**, της οποίας η γενική σύνταξη και ποιο απλοποιημένη είναι η:

delete from πίνακα;

Η εντολή με τη μορφή αυτή διαγράφει όλες τις εγγραφές του πίνακα που δημιουργήσαμε. Μπορούμε όμως να χρησιμοποιήσουμε και το όρισμα *where*, έτσι

ώστε να συγκεκριμενοποιήσουμε τη διαγραφή που θέλουμε να εκτελέσουμε στα δεδομένα του πίνακά μας. Για παράδειγμα, η `delete from όνομα_πίνακα where όνομα_στήλης = 'εγγραφή'`; διαγράφει τα αποτελέσματα που εμείς ορίσαμε. Στο παράδειγμα της εικόνας 2.13 η εντολή θα πάρει τη μορφή `delete from Tours where price = '700'`;

```
mysql> delete from Tours where price = '700';
Query OK, 1 row affected (0.00 sec)

mysql> select * from Tours;
```

id	destinations	departure	duration	price
1	England	2005-01-10	5	1500
2	France	2005-01-10	7	1700
4	Cyprus	2005-01-10	10	1300
5	U.S.A	2005-01-10	7	2200
6	Egypt	2005-01-10	5	1000
7	India	2005-01-10	8	2000
8	Australia	2005-01-10	10	3000
9	Netherlands	2005-01-10	7	1500
10	Germany	2005-01-10	9	1500

```
9 rows in set (0.00 sec)

mysql>
```

Εικ.2.13 Η εντολή delete σε συνδυασμό με την πρόταση where.

Γράφοντας την εντολή με αυτή τη μορφή σε περιβάλλον MySQL, διαγράφεται από τον πίνακά μας η εγγραφή, δηλαδή στην συγκεκριμένη εργασία το ταξίδι, που η τιμή του είναι 700 €.

Μετά την αναφορά στις βασικές εντολές της MySQL, στο επόμενο κεφάλαιο θα δημιουργήσουμε scripts σε γλώσσα PHP με τα οποία θα συνδέσουμε τις ιστοσελίδες μας με τη βάση δεδομένων μας, δηλαδή μέσω ιστοσελίδας να έχουμε προσπέλαση των δεδομένων της βάσης.

3. PHP.

Η PHP, της οποίας τα αρχικά σήμερα αντιπροσωπεύουν το "PHP: Hypertext Preprocessor" είναι μια ευρέως χρησιμοποιούμενη, ανοιχτού κώδικα, γενικού σκοπού server-side scripting γλώσσα προγραμματισμού, η οποία είναι ειδικά κατάλληλη για ανάπτυξη εφαρμογών για το Web και μπορεί να ενσωματωθεί στην HTML. Η PHP επιτρέπει στους χρήστες της να δημιουργήσουν δυναμικές ιστοσελίδες, δηλαδή ιστοσελίδες οι οποίες επιτρέπουν πρόσβαση σε βάσεις δεδομένων. Οι σελίδες οι οποίες έχουν στο περιεχόμενό τους κώδικα PHP συμπεριφέρονται όπως και οι HTML σελίδες. Στο συγκεκριμένο κεφάλαιο θα μιλήσουμε γενικά για την PHP, τις μεταβλητές της και τις εντολές της, και στο επόμενο κεφάλαιο θα αναλύσουμε τον κώδικα της συγκεκριμένης πτυχιακής εργασίας βήμα προς βήμα, έτσι ώστε να γίνει πιο κατανοητός.

3.1 Σύντομο ιστορικό της PHP.

Η PHP προέρχεται από ένα παλαιότερο προϊόν, ονομασμένο ως PHP/FI. Η PHP/FI δημιουργήθηκε από τον Rasmus Lerdorf το 1994. Αρχικά τη χρησιμοποιούσε για τη δημιουργία απλών scripts τύπου Perl ώστε να μπορεί να δημιουργήσει μια απλή σελίδα στο διαδίκτυο. Η Perl είναι μια γενικής χρήσης γλώσσα προγραμματισμού η οποία αρχικά δημιουργήθηκε για την επεξεργασία κειμένου και τώρα πλέον έχει φτάσει στο σημείο να χρησιμοποιείται για διάφορες εργασίες όπως η διαχείριση συστημάτων, η δημιουργία ιστοσελίδων, ο προγραμματισμός δικτύων και πολλές ακόμα εργασίες. Αυτά τα scripts που δημιούργησε ο Rasmus, τα ονόμασε PHP Tools δηλαδή "Personal Home Page Tools".

Λόγω της ανάγκης για περισσότερη λειτουργικότητα, ο Rasmus έγραψε μια πολύ μεγαλύτερη εφαρμογή σε γλώσσα προγραμματισμού C, που ήταν σε θέση να επικοινωνήσει με βάσεις δεδομένων, και επέτρεψε με αυτόν τον τρόπο στους χρήστες να αναπτύξουν απλές δυναμικές εφαρμογές (απλές σελίδες) στο διαδίκτυο. Ο Rasmus αφού βελτίωσε τον κώδικά του όσο μπορούσε, μετά επέλεξε να το δημοσιεύσει, έτσι ώστε όλοι να μπορούν να το βλέπουν και οποιοσδήποτε να μπορεί να το

χρησιμοποιήσει, καθώς επίσης και να μπορεί να διορθώνει κάποια λάθη (bugs), με αποτέλεσμα δηλαδή να βελτιωθεί ο κώδικας.

Το PHP/FI ως όρος ερμηνεύεται Personal Home Page/Forms Interpreter, και περιλαμβάνει μερικές από τις βασικές λειτουργίες της PHP όπως τη γνωρίζουμε με τη σημερινή της μορφή. Η PHP/FI αποτελούταν από μεταβλητές τύπου Perl, μεταβλητές αυτόματης ερμηνείας φορμών καθώς και σύνταξη τέτοιας μορφής που της επέτρεπε να ενσωματωθεί σε HTML κώδικα.

Μέχρι το 1997, η PHP/FI 2.0, που ήταν η δεύτερη έκδοση της εφαρμογής PHP/FI, υπολογίζεται ότι είχε αρκετές χιλιάδες χρήστες σε όλο τον κόσμο, με περίπου 50.000 σελίδες του διαδικτύου να ισχυρίζονται πως την έχουν εγκαταστήσει, και αυτό το νούμερο αντιστοιχεί περίπου στο 1% των σελίδων του διαδικτύου. Παρά το γεγονός ότι υπήρξαν διάφοροι άνθρωποι που συνέβαλλαν στη βελτίωση του προγράμματος με μερικά μικρά κομμάτια κώδικα, το πρόγραμμα παρέμενε ακόμα στο μεγαλύτερο μέρος του δουλειά ενός ανθρώπου. Στην συνέχεια δημιουργήθηκε η PHP/FI 3.0 και εν τέλει η γνωστή PHP όπως είναι με τη σημερινή της μορφή.

3.2 Οι Εντολές της PHP.

3.2.1 Μεταβλητές

Η PHP όπως και όλες οι γλώσσες προγραμματισμού αποτελείται από μεταβλητές (variables). Μια μεταβλητή είναι μια ειδική θέση την οποία μπορούμε να ορίσουμε εμείς για την αποθήκευση μιας τιμής. Οι μεταβλητές έχουν ζωτική σημασία για τον προγραμματισμό, γιατί χωρίς αυτές θα ήμασταν υποχρεωμένοι να ενσωματώνουμε στον κώδικά μας (script) όλες τις τιμές που χρειαζόμαστε. Οι μεταβλητές μας δίνουν τη δυνατότητα να δημιουργούμε πρότυπα, ορισμούς και συναρτήσεις για την εκτέλεση διάφορων ενεργειών, όπως για παράδειγμα την πρόσθεση, την αφαίρεση, τον πολλαπλασιασμό και τη διαίρεση δύο ή και περισσότερων αριθμών, καθώς και διάφορες άλλες ενέργειες χωρίς να ανησυχούμε για τις τιμές που περιέχουν οι μεταβλητές. Οι τιμές αντιτίθεται στις μεταβλητές μέσω των επιλογών που κάνει ο χρήστης της ιστοσελίδας και προέρχεται από μια ή και περισσότερες βάσεις δεδομένων. Μια μεταβλητή σύμφωνα με την PHP αποτελείται από ένα όνομα της επιλογής μας και μπροστά από το όνομα αυτό μπαίνει σαν πρόθεμα το σήμα του

δολαρίου (\$) . Για παράδειγμα, ας υποθέσουμε ότι θέλουμε να ορίσουμε μια μεταβλητή για κάποιους προορισμούς κάποιων ταξιδιών. Η μεταβλητή αυτή μπορεί να έχει τη μορφή `$destinations`. Τα ονόματα των μεταβλητών μπορούν να περιλαμβάνουν γράμματα, αριθμούς και το χαρακτήρα της κάτω παύλας (`_`) . Οι μεταβλητές δε μπορούν να έχουν διαστήματα, για αυτό το λόγο χρησιμοποιείται και η κάτω παύλα. Επίσης σε μια μεταβλητή πρέπει το όνομα να ξεκινάει από ένα γράμμα ή από το χαρακτήρα της κάτω παύλας, άρα μια λίγο πιο ολοκληρωμένη μεταβλητή στο προηγούμενο παράδειγμα θα έπαιρνε τη μορφή `$destination_names_1`.

Για να τερματιστεί μια εντολή της PHP χρησιμοποιείται ο χαρακτήρας του ελληνικού ερωτηματικού (`;`) και για αυτό το λόγο δε μπορεί να συμπεριληφθεί μέσα στο όνομα μιας μεταβλητής.

Μια μεταβλητή έχει δυο μέρη, το όνομα της μεταβλητής και τις τιμές που παίρνει, τα οποία ενώνονται με διάφορους τελεστές όπως είναι το ίσον (`=`) για να ορίσουν τη σχέση μιας μεταβλητής με την τιμή της. Ας υποθέσουμε λοιπόν, ότι έχουμε μια μεταβλητή με το όνομα `X` η οποία ισούται με `5`, για να την προσθέσουμε στον κώδικα μας, αρκεί να γράψουμε `$X=5;`. Για να εμφανίσουμε στην οθόνη μας την τιμή της μεταβλητής, αρκεί να γράψουμε την εντολή `print $X;` και στην οθόνη μας θα εμφανιστεί μόνο ο αριθμός `5`.

Οι μεταβλητές επομένως, αποτελούνται από το όνομα της μεταβλητής το οποίο και αναλύσαμε, τους τύπους δεδομένων που είναι οι τιμές που δέχονται οι μεταβλητές και τους τελεστές τους ποιους θα αναλύσουμε σε επόμενη ενότητα.

3.2.2 Τύποι Δεδομένων της PHP

Διαφορετικοί τύποι δεδομένων δεσμεύουν διαφορετικά ποσά μνήμης και αντιμετωπίζονται ανάλογα με το πώς θα τους ορίσουμε σε κάθε script. Μερικές γλώσσες προγραμματισμού απαιτούν από τον προγραμματιστή να δηλώνει εξαρχής τους τύπους δεδομένων που θα αποθηκεύει μια μεταβλητή, σε αντίθεση με την PHP η οποία είναι λίγο πιο ελαστική και εξακριβώνει τους τύπους δεδομένων όταν αυτοί ανατίθενται σε μια μεταβλητή. Βέβαια δε θα πρέπει να κάνουμε κατάχρηση της ιδιότητας αυτής, διότι σε ένα μεγάλο script αν αλλάξουμε τους τύπους δεδομένων που δέχεται μια μεταβλητή θα δημιουργηθούν σφάλματα στον κώδικα. Η PHP υποστηρίζει κάποιους στάνταρ τύπους δεδομένων οι οποίοι είναι οι εξής έξι :

INTEGER	π.χ. 5 →	Ακέραιος Αριθμός
DOUBLE	π.χ. 3.287 →	Αριθμοί Κινητής Υποδιαστολής
STRING	π.χ. "Hi" →	Ακολουθία Χαρακτήρων (Συμβολοσειρά)
BOOLEAN	π.χ. true →	True(Αληθές) ή False(Ψευδές)
OBJECT	→	Ένα Υπόδειγμα (Instance) Αντικειμένου μιας Κλάσης
ARRAY	→	Ένα Διατεταγμένο Σύνολο Κλειδιών και Τιμών

Η PHP εκτός από αυτούς τους έξι στάνταρ τύπους δεδομένων υποστηρίζει και δυο ειδικούς τύπους δεδομένων οι οποίοι είναι :

RESOURCE →	Αναφορά σε ένα πόρο πληροφοριών (π.χ. σε μια βάση δεδομένων)
NULL →	Μία μη-αρχικοποιημένη μεταβλητή (δηλ. μια μεταβλητή της οποίας η τιμή είναι απροσδιόριστη)

Οι τύποι της μορφής RESOURCE επιστρέφονται από συναρτήσεις οι οποίες ασχολούνται με εξωτερικές εφαρμογές και εξωτερικά αρχεία. Ο τύπος δεδομένων της μορφής NULL αφορά τις μεταβλητές οι οποίες δεν ισούται με τίποτα είναι δηλαδή απροσδιόριστες .

3.2.3 Τελεστές και Εκφράσεις της PHP

Ένας τελεστής (operator) είναι ένα σύμβολο ή μια σειρά συμβόλων τα οποία όταν χρησιμοποιούνται σε συνδυασμό με τιμές, εκτελούν μια ενέργεια και συνήθως παράγουν μια νέα τιμή. Ο τελεστής μπορεί να τοποθετηθεί ανάμεσα σε δυο τιμές και ανάμεσα από μεταβλητή και τιμή. Όπως μπορούμε να διακρίνουμε, οι τιμές έχουν άμεση σχέση με τους τελεστές και ονομάζονται *τελεστέοι*. Στην πράξη τώρα, αν για παράδειγμα, θέλουμε να προσθέσουμε το 3 και το 2, οι τιμές (τελεστέοι) θα είναι αντίστοιχα το 3 και το 2 ενώ ο τελεστής θα είναι το σύμβολο της πρόσθεσης (+) και μέσω αυτής της εντολής θα πάρουμε μια νέα τιμή που στην προκείμενη περίπτωση είναι το 5. Ο συνδυασμός των τελεστέων με έναν τελεστή για την παραγωγή ενός

αποτελέσματος ονομάζεται έκφραση (expression). Οι πιο χρησιμοποιούμενοι τελεστές είναι :

1. Τελεστής εκχώρησης τιμής

Ο τελεστής αυτός αντιπροσωπεύεται από το σύμβολο ίσον (=) και παίρνει την τιμή του δεξιού τελεστέου του και τον εκχωρεί στον αριστερό. Για παράδειγμα, έστω ότι έχουμε την έκφραση $\$destination = "England"$; . Αυτή η έκφραση σημαίνει ότι η μεταβλητή $\$destination$ περιέχει το αλφαριθμητικό $England$.

2. Αριθμητικοί τελεστές

Οι αριθμητικοί τελεστές εκτελούν αριθμητικές πράξεις και είναι οι εξής :

+	Πρόσθεση →	π.χ. $10+4$ →	14
-	Αφαίρεση →	π.χ. $10-4$ →	6
/	Διαίρεση →	π.χ. $10/4$ →	2,5
*	Πολλαπλασιασμός →	π.χ. $10*4$ →	40
%	Ακέραιο Υπόλοιπο →	π.χ. $10\%4$ →	2

Η πιο ιδιόμορφη περίπτωση είναι αυτή του τελεστή ακέραιου υπόλοιπου (modulus) ο οποίος επιστρέφει το υπόλοιπο της διαίρεσης του αριστερού τελεστέου με του δεξιού.

3. Τελεστής συνένωσης

Ο τελεστής συνένωσης (concatenation) αντιπροσωπεύεται από το σύμβολο της τέλειας (.) και αντιμετωπίζει και τους δυο τελεστέους σαν αλφαριθμητικά, προσαρτώντας το δεξιό στο τέλος του αριστερού. Για παράδειγμα, έστω ότι θέλουμε να εμφανιστεί στην οθόνη μας η έκφραση *HELLO WORLD*, αρκεί να γράψουμε την εντολή *"HELLO"."WORLD"*;

4. Σύνθετοι τελεστές εκχώρησης τιμής

Η PHP υποστηρίζει μερικούς σύνθετους τελεστές οι οποίοι μεταβάλλουν τον αριστερό τελεστέο τους και επιστρέφουν ένα αποτέλεσμα. Οι περισσότερες γλώσσες προγραμματισμού έχουν μόνο έναν τελεστή που είναι το ίσον (=), η PHP όμως κάνει την εξαίρεση. Οι σύνθετοι τελεστές απαρτίζονται από τους αριθμητικούς τελεστές και τον τελεστή συνένωσης που στο τέλος έχουν τον τελεστή ίσον (=). Ο πίνακας που ακολουθεί κάνει μια πιο λεπτομερή προσέγγιση των σύνθετων τελεστών .

$+=$	π.χ. $\$X += 4 \rightarrow$	Ισοδυναμεί με $\$X = \$X + 4 \rightarrow$	8
$-=$	π.χ. $\$X -= 4 \rightarrow$	Ισοδυναμεί με $\$X = \$X - 4 \rightarrow$	0
$/=$	π.χ. $\$X /= 4 \rightarrow$	Ισοδυναμεί με $\$X = \$X / 4 \rightarrow$	1
$*=$	π.χ. $\$X *= 4 \rightarrow$	Ισοδυναμεί με $\$X = \$X * 4 \rightarrow$	16
$\%=$	π.χ. $\$X \% = 4 \rightarrow$	Ισοδυναμεί με $\$X = \$X \% 4 \rightarrow$	0
$.=$	π.χ. $\$X .= "test" \rightarrow$	Ισοδυναμεί με $\$X = \$X . "test" \rightarrow$	"test test"

5. Τελεστές σύγκρισης

Οι τελεστές σύγκρισης όπως υποδηλώνει και το όνομα τους συγκρίνουν δυο τιμές και επιστρέφουν σαν αποτέλεσμα τύπους δεδομένων της μορφής Boolean, δηλαδή **true** αν η σύγκριση είναι αληθής, ή **false** σε κάθε άλλη περίπτωση. Οι τελεστές σύγκρισης αναλύονται στον πίνακα που ακολουθεί και είναι η εξής :

ΣΥΜΒΟΛΑ	ΟΡΙΣΜΟΣ	Έστω ότι $\$X = 3$	ΑΠΟΤΕΛΕΣΜΑ
$==$	Ισοδυναμία. Ο αριστερός τελεστής είναι ισοδύναμος με το δεξιό.	$\$X == 4$	FALSE
$!=$	Μη Ισοδυναμία. Ο αριστερός τελεστής δεν είναι ισοδύναμος με το δεξιό.	$\$X != 4$	TRUE
$===$	Ταυτοποίηση. Ο αριστερός τελεστής είναι ισοδύναμος με το δεξιό και έχουν τον ίδιο τύπο δεδομένων.	$\$X === 4$	FALSE
$>$	Μεγαλύτερο από. Ο αριστερός τελεστής είναι μεγαλύτερος από το δεξιό.	$\$X > 4$	FALSE
$>=$	Μεγαλύτερο ή ίσο με. Ο αριστερός τελεστής είναι μεγαλύτερος ή ίσος από το δεξιό	$\$X >= 4$	FALSE
$<$	Μικρότερο από. Ο αριστερός τελεστής είναι μικρότερος από το δεξιό.	$\$X < 4$	TRUE
$<=$	Μικρότερο ή ίσο με. Ο αριστερός τελεστής είναι μικρότερος ή ίσος από το δεξιό.	$\$X <= 4$	TRUE

6. Λογικοί τελεστές

Οι λογικοί τελεστές ελέγχουν συνδυασμούς λογικών τιμών και επιστρέφουν και αυτοί όπως και οι τελεστές σύγκρισης αποτέλεσμα τύπου Boolean. Οι λογικοί τελεστές είναι οι εξής τέσσερις:

ΣΥΜΒΟΛΑ	ΟΡΙΣΜΟΣ	ΠΑΡΑΔΕΙΓΜΑ	ΑΠΟΤΕΛΕΣΜΑ
ή or	Διάζευξη – Ο αριστερός ή ο δεξιός τελεστέος είναι true.	true false true or false	True
xor	Αποκλειστική Διάζευξη – Ο αριστερός ή ο δεξιός τελεστέος είναι true αλλά όχι ταυτόχρονα και οι δυο.	True xor false	False
&& ή and	Σύζευξη – Ο αριστερός και ο δεξιός τελεστέος είναι true.	true && false true and false	False
!	Άρνηση – Ο ένας και μοναδικός τελεστέος δεν είναι true	! true	False

3.2.4-Προτεραιότητα των Τελεστών

Κατά τη χρήση ενός τελεστή η PHP, αυτός διαβάζει την έκφραση από αριστερά προς τα δεξιά. Ωστόσο για εκφράσεις που περιέχουν πάνω από έναν τελεστές τα πράγματα δυσκολεύουν. Αν δημιουργήσουμε μια έκφραση του τύπου $3+2$ τότε είναι ξεκάθαρη η πράξη που πρέπει να εκτελεστεί. Όταν όμως έχουμε μια έκφραση της μορφής $3+2*5$ δημιουργείται πρόβλημα γιατί αν η PHP εκτελέσει την πράξη αυτή από τα αριστερά προς τα δεξιά θα βγει το αποτέλεσμα 25 που είναι λάθος. Για το λόγο αυτό η PHP χρησιμοποιεί τους τελεστές κατά προτεραιότητα, αν και θα ήταν πολύ πιο λειτουργικό να χρησιμοποιούσαμε παρενθέσεις για την αποφυγή σφαλμάτων σε πιο δύσκολες εκφράσεις.

ΠΡΟΤΕΡΑΙΟΤΗΤΑ ΤΕΛΕΣΤΩΝ
1. ++, --, ρητή δήλωση τύπου
2. /, *, %
3. +, -
4. <, <=, >, >=
5. ==, ===, !=
6. &&
7.
8. =, +=, -=, /=, *=, %=, .
9. and
10. xor
11. or

Όπως μπορούμε να διακρίνουμε και από τον πίνακα ο τελεστής `&&` είναι υψηλότερης προτεραιότητας από τον τελεστή `and`, όπως και ο τελεστής `||` από τον τελεστή `or`. Αυτό σημαίνει ότι αν για παράδειγμα, γράψουμε την έκφραση `$X and $Y &&$2`, η PHP θα εκτελέσει πρώτα την έκφραση `$Y&&$2` και με το αποτέλεσμα που θα βγάλει θα εκτελέσει και το `and` με τη `$X`. Αν και όπως προαναφέραμε θα ήταν προτιμότερο την έκφραση να τη γράψουμε ως: `$X and ($Y&&$2)`. Οι δυο αυτές εκφράσεις είναι ισοδύναμες αλλά ο δεύτερος τρόπος είναι πιο ασφαλής για την αποφυγή σφαλμάτων.

3.2.5 Σταθερές

Οι μεταβλητές, όπως προαναφέραμε, είναι ένας τρόπος αποθήκευσης δεδομένων. Οι μεταβλητές έχουν όμως την ευελιξία ότι μπορούμε να αλλάξουμε τις τιμές και τον τύπο δεδομένων τους. Εάν λοιπόν θέλαμε να εργαστούμε με μια τιμή για όλη τη διάρκεια ενός script τότε μπορούμε να χρησιμοποιούμε μια **σταθερά** (constant). Για τη δημιουργία μιας σταθεράς πρέπει να χρησιμοποιούμε την συνάρτηση **define()**. Στην παρένθεση τοποθετούμε το όνομα και την τιμή της σταθεράς, τα οποία θα χωρίζονται μεταξύ τους με ένα κόμμα. Σε αντίθεση με τη μεταβλητή, δε χρειάζεται το πρόθεμα του δολαρίου (\$), συνεπώς αν θέλουμε να ορίσουμε μια σταθερά με το όνομα `destination_name` και την τιμή `England` αρκεί να γράψουμε:

```
define ('DESTINATION_NAME', 'ENGLAND');
```

Η τιμή της σταθεράς μπορεί να είναι ένας αριθμός, ένα αλφαριθμητικό, ή μια λογική (Boolean) τιμή, και το όνομα της πρέπει να αναγράφεται με όλους τους χαρακτήρες κεφαλαίους.

Ας δούμε λοιπόν πως μπορούμε να ορίσουμε μια σταθερά μέσω ενός script.

```

1 <html>
2 <head>
3 <title>Defining a Constant</title>
4 </head>
5 <body>
6 <?php
7 define ("Destination_Name" , "England");
8 print,"Welcome to".Destination_Name;
9 ?>
10 </body>
11 </html>

```

Εικ.3.1 Ορισμός μιας σταθεράς.

Στη γραμμή **8** της εικόνας 3.1, χρησιμοποιήσαμε τον τελεστή συνένωσης για να προσαρτήσουμε την τιμή της σταθεράς στο αλφαριθμητικό *'Welcome to'* και αυτό γίνεται για το λόγο ότι ο μηχανισμός εκτέλεσης της PHP δεν έχει καμία λειτουργία που να κάνει διαχωρισμό μεταξύ μιας σταθεράς και ενός αλφαριθμητικού που περικλείεται σε εισαγωγικά. Εάν αποθηκεύσουμε αυτό το script με το όνομα *define.php* στο φάκελο έγγραφων του Web server μας, στη συγκεκριμένη περίπτωση στο φάκελο **htdocs** του Apache, και τον προσπελάσουμε μέσω του browser, στην οθόνη μας θα εμφανιστεί το κείμενο: *Welcome to England*.

Σημειώνουμε, ότι κάθε script σε PHP αρχίζει και τελειώνει με τα **<?php** και **?>** αντίστοιχα (γραμμές 6 και 9).

3.2.6 Οι Εντολές της PHP

Το σημαντικότερο κεφάλαιο, όσον αφορά την PHP, είναι αυτό των εντολών της. Οι εντολές παρέχουν στη σελίδα τη δυνατότητα να λαμβάνει αποφάσεις. Αυτό σημαίνει ότι την καθιστούν δυνατή να αλλάζει τα αποτελέσματα της ανάλογα με τις ισχύουσες συνθήκες, δηλαδή με τις επιλογές που κάνει ο χρήστης της σελίδας, κάνοντας με αυτόν τον τρόπο τη σελίδα δυναμική.

1. Η εντολή IF

Η εντολή IF παρέχει στον προγραμματιστή τη δυνατότητα να ελέγχει την εκτέλεση μιας άλλης εντολής, ή ενός ολόκληρου τμήματος κώδικα με πολλές εντολές, περικλείοντας τα σε άγκιστρα (**IF {}**). Συγκεκριμένα η εντολή **IF** δέχεται τιμές τύπου Boolean (true ή false) και ανάλογα με το αποτέλεσμα που θα δέχεται εκτελεί την εντολή ή την παρακάμπει αντίστοιχα.

Η γενική μορφή της εντολής είναι η εξής :


```

If (έκφραση ){
    Κώδικας που θα εκτελεστεί αν η έκφραση είναι true
}

```

Παρακάτω ακολουθεί ένα μικρό παράδειγμα. Έστω ότι έχουμε μια μεταβλητή *\$destination* η οποία ισούται με το *England* και θέλουμε, εάν ο χρήστης της σελίδας επιλέξει την Αγγλία να εμφανιστεί στην οθόνη η φράση: *Welcome to England*.

```

1 <html>
2 <head>
3 <title>ΕΝΤΟΛΗ IF</title>
4 </head>
5 <body>
6 <?php
7 $destination="England";
8 if ($destination=="England"){
9     print "Welcome to England";
10 }
11 ?>
12 </body>
13 </html>

```

Εικ.3.2 Η Εντολή if.

Στο script που δημιουργήσαμε χρησιμοποιούμε τον τελεστή σύγκρισης `==`, έτσι ώστε να συγκρίνουμε τη μεταβλητή *\$destination* με το αλφαριθμητικό *England* και αν ταιριάζουν, δηλαδή η έκφραση αυτή είναι *true*, τότε το τμήμα κώδικα που βρίσκεται μέσα στα άγκιστρα θα εκτελεστεί. Στη συγκεκριμένη περίπτωση η έκφραση αυτή είναι *true* γιατί εμείς έχουμε δηλώσει από πριν ότι *\$destination = 'England'*, αλλά σε άλλα μεγαλύτερα και πολυπλοκότερα scripts μπορεί να υπάρχουν παραπάνω από μια τιμές για μια μεταβλητή, άρα η PHP θα πρέπει να βρει την τιμή που της ζητάμε και θα εμφανίσει στην οθόνη μας αυτό που της έχουμε δηλώσει να δείξει. Στην περίπτωση που η έκφραση βγει *false*, το τμήμα του κώδικα μέσα στα άγκιστρα δεν εκτελείται και δεν έχουμε κάποιο αποτέλεσμα.

2. Χρήση της πρότασης `else` στην εντολή `if`

Η πρόταση `else` είναι συμπληρωματική της εντολής `if` και εκτελείται όταν η έκφραση στην εντολή `if` είναι *false*. Δηλαδή η `else` έχει μέσα σε άγκιστρα ένα άλλο κομμάτι κώδικα, όπου εάν η έκφραση είναι *false* θα εκτελεστεί αυτός ο κώδικας.

Η γενική μορφή που παίρνει η `if` σε συνδυασμό με την πρόταση `else` είναι η εξής :

```

if (έκφραση ){
    Ο κώδικας που θα εκτελεστεί αν η πρόταση είναι true
} else {
    Ο κώδικας που θα εκτελεστεί αν η πρόταση είναι false.
}

```

Στο προηγούμενο παράδειγμα μπορούμε να προσθέσουμε την πρόταση `else` και να αλλάξουμε την τιμή της μεταβλητής ώστε να βγει `false`.

```

1 <html>
2 <head>
3 <title>ENTOLI IF ELSE</title>
4 </head>
5 <body>
6 <?php
7 $destination="France";
8 if ($destination=="England"){
9     print "Welcome to England";
10 } else {
11     print "Your destination is $destination";
12 }
13 >>
14 </body>
15 </html>

```

Εικ3.3 Η εντολή if με την χρήση της πρότασης else.

Εάν προσπελάσουμε λοιπόν αυτό το αρχείο-μέσω του browser του υπολογιστή μας, όπως και το προηγούμενο, αφού πρώτα το έχουμε αποθηκεύσει στο φάκελο των έγγραφων (htdocs) του Web server με την κατάληξη .php, τότε στην οθόνη μας θα εμφανιστεί η πρόταση: *"Your destination is France"*.

Όπως παρατηρούμε στη γραμμή 11 δε βάλουμε την τιμή *France* αλλά το όνομα της μεταβλητής, αυτό το κάναμε με το σκεπτικό ότι όποια τιμή και να δώσουμε στη μεταβλητή εκτός από την *England* θα μας βγάλει την πρόταση που προαναφέραμε. Σε αυτήν την περίπτωση δώσαμε τη δυνατότητα στο script μας να λαμβάνει πολυπλοκότερες αποφάσεις.

Η PHP μας δίνει τη δυνατότητα να αποτιμούμε και περισσότερες εκφράσεις σε `true` η `false` την μια μετά την άλλη.

3. Η πρόταση elseif στην εντολή if.

Για να πετύχουμε τον έλεγχο πολλαπλών εκφράσεων μπορούμε να χρησιμοποιήσουμε την πρόταση **elseif** σε συνδυασμό με την **if** και την πρόταση **else**. Δηλαδή, μπορούμε να χρησιμοποιήσουμε μια δομή της μορφής **if...elseif...else**.

Η γενική μορφή της εντολής αυτής είναι η εξής :

```
if (έκφραση) {
    Ο κώδικας που θα εκτελεστεί αν η πρόταση είναι true
} elseif (δεύτερη έκφραση) {
    Ο κώδικας που θα εκτελεστεί εάν η προηγούμενη πρόταση είναι false
} else {
    Ο κώδικας που θα εκτελεστεί σε όλες τις άλλες περιπτώσεις
}
```

Εάν η πρώτη έκφραση είναι *true*, τότε η PHP θα εκτελέσει την ενέργεια που είναι μέσα στα άγκιστρα, ενώ εάν είναι *false* θα προχωρήσει στην πρόταση **elseif**. Στη συνέχεια, εάν στην πρόταση **elseif** η έκφραση είναι *true* τότε η PHP θα εκτελέσει αυτό το κομμάτι του κώδικα που είναι μέσα στα άγκιστρα της **elseif**, ενώ εάν είναι και αυτή *false*, θα εκτελεστεί ο κώδικας που είναι μέσα στα άγκιστρα της πρότασης **else**.

Μέσα σε μία εντολή **if** μπορούμε να συμπεριλάβουμε όσες **elseif** επιθυμούμε.

Ένα παράδειγμα θα ήταν όπως και τα προηγούμενα, απλά θα προσθέσουμε το τμήμα **elseif**.

```
1 <html>
2 <head>
3 <title> Η Protasi ElseIf </title>
4 </head>
5 <body>
6 <?php
7 $destination = "England";
8 if ($destination == "France") {
9     print "I am in France";
10 } elseif ($destination == "Germany") {
11     print "I am in Germany";
12 } else {
13     print "I am neither in France nor in Germany, I am in $destination"
14 }
15 ?>
16 </body>
17 </html>
```

Εικ.3.4 Η εντολής **if** με την χρήση της **elseif** και της **else**.

Δημιουργώντας λοιπόν ένα script σαν και αυτό της εικόνας 3.4, εάν το προσπελάσουμε μέσω του browser του υπολογιστή μας, θα εμφανιστεί στην οθόνη μας η πρόταση «*I am neither in France nor in Germany, I am in England*» αφού και η πρώτη αλλά και η δεύτερη πρόταση είναι *false* και εκτελέστηκε ο κώδικας που ήταν μέσα σε άγκιστρα στο τμήμα *else*.

4. Η εντολή Switch

Η εντολή **switch** μας βοηθάει να αλλάξουμε τη ροή του προγράμματος, ανάλογα με το αποτέλεσμα μιας έκφρασης. Η διαφορά της switch από την if, σε συνδυασμό με την else if, είναι ότι η if μπορεί να εξετάζει πολλαπλές περιπτώσεις, ενώ η switch εξετάζει μια μόνο περίπτωση και ανάλογα με το αποτέλεσμα, όχι μόνο true η false, αλλά αριθμητικό ή και αλφαριθμητικό, εκτελεί διαφορετικό κώδικα.

Η γενική μορφή της switch είναι :

```
switch (έκφραση) {
    case αποτέλεσμα 1:
        ο κώδικας που εκτελείται στο αποτέλεσμα 1
        break;
    case αποτέλεσμα 2:
        ο κώδικας που εκτελείται στο αποτέλεσμα 2
        break;
    default :
        ο κώδικας που θα εκτελεστεί εάν δεν έχει εκτελεστεί κανένας
        προηγούμενος
}
```

Συχνά η έκφραση της switch είναι κάποια μεταβλητή. Μέσα σε κάθε αποτέλεσμα (case) θα βρίσκεται μια εντολή η οποία θα εκτελείται ανάλογα με το αποτέλεσμα.

Δεν πρέπει να ξεχάσουμε στο τέλος κάθε case να γράφουμε και την εντολή **break** γιατί αλλιώς θα εκτελεστούν όλες οι εντολές με τη σειρά που τις είχαμε γράψει.

Σε περίπτωση που δεν έχει εκτελεστεί καμία εντολή και ο κώδικας φτάσει στην εντολή **default**, τότε αυτή εκτελείται.

Εάν για κάποιο λόγο, δυο εντολές ισχύουν ταυτόχρονα, θα εκτελεστεί ο κώδικας αυτής που είναι πρώτη.

Χρησιμοποιώντας τα ίδια δεδομένα με τα προηγούμενα παραδείγματα θα κάνουμε πάλι ένα παράδειγμα με την εντολή switch .

```

1 <html>
2 <head>
3 <title>ENTOLI SWITCH</title>
4 </head>
5 <body>
6 <?php
7 $destination="England";
8 switch ($destination) {
9     case "France":
10     print "I am in France";
11     break;
12     case "Germany":
13     print "I am in Germany";
14     break;
15     default:
16     print "I am neither in France nor in Germany, I am in $destination";
17 }
18 ?>
19 </body>
20 </html>

```

Εικ.3.5 Η εντολή switch.

Στο script που γράψαμε ως παράδειγμα, αν το αποθηκεύσουμε και το προσπελάσουμε όπως και τις προηγούμενες φορές μέσω του Browser του υπολογιστή, θα εμφανιστεί η φράση "*I am neither in France nor in Germany, I am in England*". Εάν αλλάξουμε τη μεταβλητή *\$destination* και την κάνουμε *France* ή *Germany* τότε θα είχαμε τα αποτελέσματα του case "France" και case "Germany" αντίστοιχα.

5. Ο Τριαδικός Τελεστής

Ο τριαδικός τελεστής, δηλαδή το σύμβολο του αγγλικού ερωτηματικού (?) είναι σα μια μορφή εντολής. Δηλαδή ο τριαδικός τελεστής είναι όμοιος με την εντολή if. Αποτελείται από τρία μέρη, εξ' ου και το όνομα του, και αυτά είναι μια έκφραση και δύο κώδικες οι οποίοι θα εκτελεστούν ή ο ένας ή ο άλλος, ανάλογα με το αν η πρόταση βγει true ή false.

Η γενική σύνταξη του τριαδικού τελεστή (?) είναι η εξής:

(έκφραση) ? Κώδικας που θα εκτελεστεί εάν η έκφραση είναι *true* : κώδικας που θα εκτελεστεί αν η έκφραση είναι *false*.

Έστω λοιπόν το script με τα ίδια δεδομένα με το προηγούμενο παράδειγμα:

```

1 <html>
2 <head>
3 <title>TRIADIKOS TELESTIS</title>
4 </head>
5 <body>
6 <?php
7 $destination="England";
8 $text=($destination="France")?"I am in France":"I am not in France but in $destination";
9 print "$text";
10 ?>
11 </body>
12 </html>

```

Εικ.3.6 Παράδειγμα με τριαδικό τελεστή.

Στο script που γράψαμε μπορούμε να παρατηρήσουμε ότι έχουμε μια καινούρια μεταβλητή (*\$text*) που δεν είχαμε στα προηγούμενα παραδείγματα. Αυτό συμβαίνει γιατί για να γράψουμε την εντολή με το τριαδικό τελεστή πρέπει να τη γράψουμε μέσα σε μια μεταβλητή και μετά λέμε στην PHP να εμφανίσει (*print*) στην οθόνη μας τη μεταβλητή (*\$text*) αυτή.

3.2.7 Βρόγχοι

Μέχρι τώρα έχουμε αναφέρει τις αποφάσεις που μπορεί να πάρει ένα script από ένα αποτέλεσμα που θα του δοθεί, τα scripts όμως έχουν και παραπάνω δυνατότητες, όπως να αποφασίζουν το πόσες φορές θα εκτελεστεί ένα κομμάτι κώδικα. Αυτό επιτυγχάνεται με τους βρόγχους. Ένας βρόγχος (loop) επαναλαμβάνει την εκτέλεση ενός μέρους κώδικα. Για τη δημιουργία βρόγχων υπάρχουν ειδικές εντολές όπως είναι η **while**, η **do** και η **for**.

1. Η εντολή **while**

Η δομή της εντολής **while** είναι όμοια με αυτήν της εντολής **if** :

```

While (έκφραση){
    Κώδικας που θα εκτελεστεί αν η έκφραση είναι true
}

```

Όσο η έκφραση της εντολής **while** είναι **true**, το τμήμα του κώδικα εκτελείται συνεχώς. Συνήθως, αλλάζουμε κάτι που επηρεάζει την έκφραση της εντολής **while**, γιατί σε διαφορετική περίπτωση ο κώδικας θα εκτελείται συνεχώς. Η συνεχής εκτέλεση του κώδικα αποκαλείται **iteration** (επανάληψη).

Στο παρακάτω παράδειγμα γίνεται πιο κατανοητή η χρήση της εντολής **while**.

```

1 <html>
2 <head>
3 <title>Entoli While</title>
4 </head>
5 <body>
6 <?php
7 $num=1;
8 while ($num<=12) {
9     print "$num times 2 is".($num*2)."<br>";
10    $num++;
11 }
12 ?>
13 </body>
14 </html>

```

Εικ.3.7 Η εντολή while.

Σε αυτό το παράδειγμα γράψαμε ένα script, που μας βρίσκει τα πολλαπλάσια του δυο από το ένα ως το δώδεκα. Όπως μπορούμε να παρατηρήσουμε στη γραμμή 7 έχουμε ορίσει μια μεταβλητή, τη *\$num* να είναι ίση με τον αριθμό 1. Στη γραμμή 8 ξεκινάμε την εντολή *while* όπου και γράφουμε την έκφραση ότι η μεταβλητή μας πρέπει να είναι μικρότερη ή ίση του 12. Στη συνέχεια στη γραμμή 9 γράφουμε το τμήμα του κώδικα που θα εκτελείται συνεχώς όσο η έκφραση που γράψαμε στην προηγούμενη γραμμή είναι *true*. Η γραμμή 10 είναι ίσως η σημαντικότερη γραμμή όλου του κώδικα. Με την εντολή αυτή που γράψαμε κάναμε την μεταβλητή μας να αυξάνεται κατά ένα, γιατί αν δεν αυξανόταν θα έμενε ίση με ένα και ο κώδικας της *while* θα εκτελούνταν διαρκώς.

Εάν αυτό το script το αποθηκεύσουμε με ένα όνομα της αρέσκειας μας και κατάληξη *.php* στο φάκελο των έγγραφων του web server και το προσπελάσουμε μέσω του browser του υπολογιστή, θα εμφανιστούν στην οθόνη τα πολλαπλάσια του δυο από το ένα ως το δώδεκα με την εξής μορφή:

```

1 times 2 is2
2 times 2 is4
3 times 2 is6
4 times 2 is8
5 times 2 is10
6 times 2 is12
7 times 2 is14
8 times 2 is16
9 times 2 is18
10 times 2 is20
11 times 2 is22
12 times 2 is24

```

Εικ.3.8 Αποτελέσματα της εντολής while.

2. Η Εντολή do...while

Η εντολή **do...while**, θα μπορούσαμε να πούμε ότι, είναι η αντίστροφη εντολή της **while**. Μια σημαντική διάφορα είναι ότι στην εντολή **do...while** το τμήμα του κώδικα εκτελείται πριν από τον έλεγχο για το αν η έκφραση είναι **true** ή **false**.

Η γενική εντολή της **do...while** είναι η εξής:

```
do {  
    Κώδικας που εκτελείται  
} while (έκφραση);
```

Η εντολή **do...while** χρησιμεύει σε περιπτώσεις που ο κώδικας πρέπει να εκτελεστεί **τουλάχιστον** μια φορά.

Στο παράδειγμα που ακολουθεί έχουμε δημιουργήσει ένα βρόγχο **do...while**.

```
1 <html>  
2 <head>  
3 <title>Entoli Do...While</title>  
4 </head>  
5 <body>  
6 <?php  
7 $num=1;  
8 do {  
9     print "The number is: $num <br>\n";  
10    $num++;  
11 } while ($num>0&&$num<3);  
12 >>  
13 </body>  
14 </html>
```

Εικ.3.9 Παράδειγμα με την εντολή Do...while.

Στο script που γράψαμε η εντολή **do...while** ελέγχει αν η μεταβλητή **\$num** είναι μεταξύ μηδέν (0) και τρία (3) (γραμμή 11). Στη γραμμή 7 ορίσαμε την **\$num** ίση με 1, άρα ο κώδικας θα εκτελεστεί γιατί η έκφραση είναι **true** και στην οθόνη μας θα εμφανιστεί η φράση "The number is: 1". Αν η **\$num** ήταν τέσσερα η έκφραση θα ήταν **false** αλλά ο κώδικας θα εκτελούνταν μια φορά και στην οθόνη μας θα εμφανιζόταν η φράση "The number is: 4", γιατί ο κώδικας εκτελείται πριν αποτιμηθεί εάν η έκφραση είναι **True** ή **false**. Στη γραμμή 10, όπως και στο παράδειγμα της εντολής **while** δίνουμε εντολή να αυξηθεί κατά ένα η μεταβλητή, γιατί αλλιώς το αποτέλεσμα του κώδικα δεν θα αλλάξει.

3. Η εντολή for

Η εντολή `for` τυπικά δεν έχει διαφορά από την εντολή `while`. Ουσιαστικά όμως η εντολή `for` είναι συχνά ένας καλύτερος και ασφαλέστερος τρόπος για να πετύχουμε το ίδιο αποτέλεσμα. Το αποτέλεσμα που επιτυγχάνουμε στην `while` και στην `do...while` μπορούμε να το πετύχουμε με την εντολή `for` σε μια μόνο γραμμή κάνοντας έτσι τον κώδικα πιο συμπαγή και κάνοντάς το και δυσκολότερο να γίνουν λάθη, αφού πρέπει μέσα στον κώδικα της `for` να γράφεται και η εντολή αύξησης της μεταβλητής.

Η γενική δομή της `for` είναι η εξής :

```
for (έκφραση αρχικοποίησης;έκφραση ελέγχου;έκφραση τροποποίησης) {  
    Κώδικας που θα εκτελεστεί  
}
```

Ένα αντίστοιχο παράδειγμα με τα προηγούμενα για την εντολή `for` είναι και αυτό της εικόνας 3.10. Εάν αποθηκεύσουμε το script αυτό όπως και τις προηγούμενες φορές και το προσπελάσουμε μέσω του browser του υπολογιστή μας θα εμφανιστεί στην οθόνη το ίδιο ακριβώς αποτέλεσμα του παραδείγματος της εντολής `while`.

```
1 <html>  
2 <head>  
3 <title>Entoli For</title>  
4 </head>  
5 <body>  
6 <?php  
7 for ($num=1;$num<=5;$num++) {  
8     print "$num times 2 is".($num*2)."<br>";  
9 }  
10 ?>  
11 </body>  
12 </html>
```

Εικ.3.10 Παράδειγμα με την χρήση της εντολής `for`.

3.2.8 Έξοδος από ένα Βρόγχο

1. Η εντολή `break`

Σε ένα script το οποίο περιέχει βρόχους, υπάρχει περίπτωση να δημιουργηθούν κάποια σφάλματα κατά την εκτέλεση του.

Για παράδειγμα, σε ένα βρόχο `for` που θα διαιρεί ένα αριθμό (π.χ. το 100) με έναν αυξανόμενο δεύτερο αριθμό ο οποίος θα είναι μικρότερος η ίσος του 10 και θα

ξεκινάει από -3 και τις τιμές θα τις βάζει ο χρήστης μέσω μιας φόρμας, μπορεί να δημιουργηθούν προβλήματα. Δηλαδή, ο χρήστης μπορεί αντί για αριθμό να βάλει κάποιο αλφαριθμητικό, ή από τη στιγμή που ο αριθμός μας ξεκινάει από το -3 σημαίνει ότι ο αριθμός που διαιρείται θα φτάσει η στιγμή που θα διαιρεθεί και με το μηδέν. Για να μπορούμε να βάζουμε ορισμένους περιορισμούς στους βρόχους, έτσι ώστε αν ισχύει ένας από αυτούς να γίνεται έξοδος από το βρόχο χρησιμοποιούμε την εντολή **break**. Έστω λοιπόν ότι έχουμε έναν αριθμό ο οποίος όπως αναφέραμε ξεκινάει από το -3 και αυξάνεται μέχρι το 10 και διαιρεί τον αριθμό 100. Στο συγκεκριμένο παράδειγμα θέλουμε να αποφύγουμε την διαίρεση του 100 με το μηδέν.

```
1 <html>
2 <head>
3 <title>Entoli Break</title>
4 </head>
5 <body>
6 <?php
7 $num=-3;
8 for (;$num<=10;$num++){
9     if ($num==0)
10        break;
11 $temp=100/$num;
12     print "100 divided by $num is...$temp<br>";
13 }
14 ?>
15 </body>
16 </html>
```

Εικ.3.11 Παράδειγμα με την εντολή break.

Ξεκινώντας από την αρχή αυτού του script παρατηρούμε στη γραμμή 7 ότι έχουμε ορίσει την μεταβλητή *\$num* έξω από την εντολή *for*. Αυτό το κάνουμε για να προσομοιώσουμε μια πιθανή περίπτωση κατά την οποία η μεταβλητή αποκτά τις τιμές της από τα δεδομένα που εισάγει ο χρήστης της σελίδας μας, ή από δεδομένα που ανακτώνται από μια βάση δεδομένων. Όπως παρατηρούμε στη γραμμή 8 στην έκφραση της *for* λείπει η πρώτη συνθήκη, στην *for* έχουμε τη δυνατότητα να παραλείπουμε οποιαδήποτε από τις τρεις συνθήκες αλλά τα ελληνικά ερωτηματικά πρέπει να παραμένουν. Στη συνέχεια στη γραμμή 9 χρησιμοποιούμε την εντολή *if* για να ορίσουμε τους περιορισμούς που θέλουμε να ισχύουν στο βρόχο μας, όπου στη συγκεκριμένη περίπτωση είναι η τιμή της *\$num* να μην είναι ίση με το μηδέν. Η εντολή *break* στη γραμμή 10, εάν το αποτέλεσμα της *if* είναι *true*, τερματίζει την εκτέλεση του κώδικα της *for* και η ροή του προγράμματος συνεχίζει κανονικά μετά

το βρόχο. Εάν το script που δημιουργήσαμε το αποθηκεύσουμε, για παράδειγμα με το όνομα *break.php*, στο φάκελο των εγγράφων του web server και το προσπελάσουμε μέσω του browser, στην οθόνη θα εμφανιστεί το παρακάτω αποτέλεσμα :

```
100 divided by -3 is...-33.3333333333333
100 divided by -2 is...-50
100 divided by -1 is...-100
```

Εικ.3.12 Αποτελέσματα εφαρμογής του παραδείγματος της εντολής break.

2. Η εντολή continue

Η εντολή **continue** χρησιμοποιείται και αυτή για τη δημιουργία περιορισμών όσον αφορά στις τιμές που δέχεται ένας βρόχος. Η διάφορα της continue από την break είναι ότι η πρώτη δε σταματάει τη ροή του κώδικα, απλά παρακάμπτει τις μη δεκτές τιμές στον κώδικα του βρόχου.

Έστω λοιπόν, το προηγούμενο παράδειγμα που χρησιμοποιήσαμε την break αλλά αυτή τη φορά η νέα του εκδοχή με την εντολή continue.

```
1 <html>
2 <head>
3 <title>Entoli Continue</title>
4 </head>
5 <body>
6 <?php
7 $num=-3;
8 for (;$num<=10;$num++) {
9     if ($num==0)
10        continue;
11 $temp=100/$num;
12     print "100 divided by $num is...$temp<br>";
13 }
14 ?>
15 </body>
16 </html>
```

Εικ.3.13 Παράδειγμα με την εντολή continue.

Όπως μπορούμε να παρατηρήσουμε στη γραμμή **10** έχουμε αντικαταστήσει την εντολή break με την εντολή *continue*. Η αλλαγή αυτή επιτρέπει στο script μας όταν φτάσει στην τιμή *μηδέν*, που είναι υπό περιορισμό, να την προσπεράσει και να συνεχίσει το script να εκτελείται κανονικά μέχρι τέλους. Εάν αποθηκεύσουμε και αυτό το script όπως και τα προηγούμενα ως *continue.php* στο φάκελο εγγράφων του

web server και το προσπελάσουμε μέσω του browser θα εμφανιστεί στην οθόνη μας το εξής αποτέλεσμα :

```
100 divided by -3 is...-33.3333333333333
100 divided by -2 is...-50
100 divided by -1 is...-100
100 divided by 1 is...100
100 divided by 2 is...50
100 divided by 3 is...33.3333333333333
100 divided by 4 is...25
100 divided by 5 is...20
100 divided by 6 is...16.6666666666667
100 divided by 7 is...14.285714285714
100 divided by 8 is...12.5
100 divided by 9 is...11.1111111111111
100 divided by 10 is...10
```

Εικ.3.14 Αποτελέσματα από το παράδειγμα της εντολής continue.

Συνοψίζοντας, σε αυτήν την ενότητα αναφέραμε τις εντολές που ελέγχουν την ροή εκτέλεσης ενός script, όπως είναι η if, else, else if, switch, while ,do...while και for, τις εντολές δημιουργίας βροχών, καθώς και τις εντολές που περιορίζουν τις τιμές που δέχονται οι κώδικες και δημιουργούν διαφορετικά αποτελέσματα όπως η break και η continue. Στη συνέχεια θα ασχοληθούμε με τις συναρτήσεις και τη διαδικασία επικοινωνίας της σελίδας με βάση δεδομένων MySQL μέσω της PHP .

3.3 Συναρτήσεις

3.3.1 Τι είναι μια Συνάρτηση;

Σε αυτήν την ενότητα θα ασχοληθούμε και θα αναλύσουμε τις **συναρτήσεις**. Μια συνάρτηση, εάν θέλουμε να θέσουμε έναν ορισμό, είναι ένα αυτόνομο τμήμα κώδικα το οποίο μπορούν να καλούν τα scripts που δημιουργούμε. Όταν καλείται μια συνάρτηση εκτελείται ο κώδικας της. Μια συνάρτηση δέχεται τιμές και τις χειρίζεται με τέτοιο τρόπο ανάλογα με τη συνάρτηση και τα δεδομένα που θα της δώσουμε. Όταν ο κώδικας της συνάρτησης εκτελεσθεί, τότε αυτή με τη σειρά της επιστρέφει ένα αποτέλεσμα στο script που την κάλεσε .

3.3.2 Κλήση Συναρτήσεων

Υπάρχουν δυο ειδών συναρτήσεις, οι εγγενείς στη γλώσσα της PHP και αυτές που μπορούμε να ορίσουμε μόνοι μας. Οι εγγενείς συναρτήσεις είναι έτοιμες συναρτήσεις τις οποίες χρησιμοποιούμε για συγκεκριμένες εργασίες που είναι ορισμένες να εκτελούν. Η δομή τους είναι, αρχικά το όνομα της συνάρτησης και εν συνεχεία ένα ζεύγος παρενθέσεων, μέσα στο οποίο περνάμε πληροφορίες, τα επονομαζόμενα **ορίσματα** (arguments). Μια συνάρτηση μπορεί να απαιτεί παραπάνω από ένα ορίσματα για να εκτελέσει την εργασία που είναι προγραμματισμένη να κάνει. Σε αυτήν την περίπτωση τα ορίσματα γράφονται μέσα στο ζεύγος παρενθέσεων και χωρίζονται μεταξύ τους με κόμματα. Η γενική μορφή των εγγενών συναρτήσεων είναι: **όνομα_συνάρτησης (\$ορισμα1 , \$ορισμα2);**.

Μια συνάρτηση την οποία έχουμε συναντήσει σχεδόν σε όλα τα παραδείγματα μέχρι τώρα είναι η **print ()**. Η συνάρτηση **print ()** είναι ένα ειδικό είδος συνάρτησης αφού το ζεύγος παρενθέσεων δεν είναι υποχρεωτικό, δηλαδή η **print "Hello"** και η **print (Hello)** είναι το ίδιο έγκυρες συναρτήσεις και η εκτέλεση τους αποφέρει ακριβώς το ίδιο αποτέλεσμα. Η **print** όμως είναι η εξαίρεση στον κανόνα αφού όλες οι άλλες συναρτήσεις απαιτούν το ζεύγος των παρενθέσεων.

Ένα παράδειγμα εγγενούς συνάρτησης είναι η **abs()** η οποία απαιτεί να της δώσουμε ένα προσημασμένο αριθμό και αυτή θα μας επιστρέψει την απόλυτη τιμή του .

```
1 <html>
2 <head>
3 <title>Synartisi ABS</title>
4 </head>
5 <body>
6 <php
7 $num=-3;
8 $num2=abs($num);
9 print $num2;
10 >>
11 </body>
12 </html>
```

Εικ.3.15 Παράδειγμα συνάρτησης abs.

Αυτό το script αν αποθηκευτεί και προσπελαστεί, όπως έχουμε αναφέρει σε προηγούμενα παραδείγματα, θα μας εμφανίσει στον υπολογιστή μας τον αριθμό 3. Υπάρχουν και άλλες εγγενείς συναρτήσεις της PHP όπου μπορεί κανείς να τις μελετήσει και μάθει μέσω κάποιου βιβλίου εκμάθησης της PHP .

3.3.3 Ορισμός μιας Συνάρτησης

Όπως αναφέραμε εκτός από τις εγγενείς συναρτήσεις, οι οποίες είναι έτοιμες συναρτήσεις, υπάρχουν και οι συναρτήσεις που δημιουργούμε εμείς .

Ο γενικός τύπος των συναρτήσεων αυτών είναι:

```
function όνομα_συνάρτησης ($όρισμα1, $όρισμα2) {  
    κώδικας της συνάρτησης ;  
}
```

Το όνομα της συνάρτησης όπως και στις μεταβλητές δεν επιτρέπεται να έχει κενά και συνήθως στη θέση των κενών χρησιμοποιούμε το χαρακτήρα της κάτω παύλας. Επίσης μπορούμε να παρατηρήσουμε ότι πριν το όνομα της συνάρτησης πρέπει να γράψουμε την λέξη **function**, αυτό γίνεται για να καταλαβαίνει η PHP ότι αυτό που γράφουμε είναι συνάρτηση που τη δημιουργούμε εμείς. Ύστερα υπάρχει το ζεύγος των παρενθέσεων μέσα στο οποίο γράφουμε τα ορίσματα, όμως και σε περιπτώσεις που δεν έχουμε ορίσματα οι παρενθέσεις είναι αναγκαίες.

Ας εξετάσουμε λοιπόν δυο παραδείγματα, ένα που η συνάρτηση δεν έχει ορίσματα και ένα που έχει .

```
1 <html>  
2 <head>  
3 <title>Synartisi Xwris Orismata</title>  
4 </head>  
5 <body>  
6 <?php  
7 function bighello(){  
8     print "<hl>HELLO!</hl>";  
9 }  
10 bighello();  
11 ?>  
12 </body>  
13 </html>
```

Εικ.3.16 Παράδειγμα συνάρτησης χωρίς ορίσματα.

```

1 <html>
2 <head>
3 <title>Synartisi Me Orismata</title>
4 </head>
5 <body>
6 <?php
7 function printBR($txt){
8     print "$txt<br>\n";
9 }
10 printBR("This is line 1");
11 printBR("This is line 2");
12 printBR("and this is line 3");
13 ?>
14 </body>
15 </html>

```

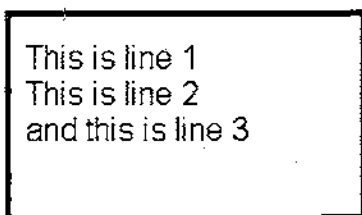
Εικ.3.17 Παράδειγμα συνάρτησης με ορίσματα.

Στο πρώτο παράδειγμα η συνάρτηση *bighello()* απλώς εμφανίζει στην οθόνη μας το αλφαριθμητικό "HELLO !" μορφοποιημένο με το στοιχείο <h1> της HTML. Δηλαδή, αν το αποθηκεύσουμε για παράδειγμα με το όνομα *printhello.php* στο φάκελο εγγράφων του web server και το προσπελάσουμε μέσω του browser του υπολογιστή στην οθόνη μας θα εμφανιστεί:



Εικ.3.18 Αποτελέσματα από το παράδειγμα της συνάρτησης *bighello()*.

Το δεύτερο παράδειγμα αν το αποθηκεύσουμε στο φάκελο εγγράφων του web server και το προσπελάσουμε μέσω του browser, στην οθόνη μας θα εμφανιστεί η εξής οθόνη :



Εικ.3.19 Αποτελέσματα από το παράδειγμα της συνάρτησης *printBR()*.

Στο script του δεύτερου παραδείγματος σαν όρισμα της συνάρτησης στη γραμμή 7 θέσαμε τη μεταβλητή *\$txt*, οπότε ότι γράφουμε μέσα στη συνάρτηση *printBR ()* θα αποθηκεύεται στη μεταβλητή *\$txt*. Στη γραμμή 8 εκτυπώνουμε τις τιμές της μεταβλητής *\$txt* προσθέτοντας το στοιχείο **
** της HTML και το χαρακτήρα αλλαγής γραμμής \n (new line).

3.3.4 Επιστροφή Τιμών από τις Καθοριζόμενες από το Χρήστη Συναρτήσεις

Στα προηγούμενα παραδείγματα δείξαμε πως μπορούμε να εμφανίσουμε στην οθόνη μας ένα αλφαριθμητικό μέσω κάποιας συνάρτησης. Έχουμε όμως τη δυνατότητα να εισάγουμε και τιμές σε μια συνάρτηση και να μας εμφανίζεται στην οθόνη μας κάποιο αποτέλεσμα από μια διεργασία των τιμών αυτών. Αυτό το πετυχαίνουμε με την εντολή **return**, η οποία μετά την εκτέλεση της συνάρτησης μας επιστρέφει το ανάλογο αποτέλεσμα. Το παράδειγμα που ακολουθεί θα προσπαθήσει να κάνει πιο κατανοητή τη λειτουργία της **return**.

```
1 <html>
2 <head>
3 <title>Entoli Return</title>
4 </head>
5 <body>
6 <?php
7 function addnums($first, $second){
8     $result=$first+$second;
9     return $result;
10 }
11 print addnums(3, 5);
12 </?>
13 </body>
14 </html>
```

Εικ.3.20 Παράδειγμα εντολής return.

Εάν εκτελέσουμε το παραπάνω script στην οθόνη θα εμφανιστεί το νούμερο 8. Αυτό θα συμβεί γιατί έχουμε ορίσει εμείς στο script να κάνει πρόσθεση των δυο ορισμάτων της *addnums()*. Τα ορίσματα αυτά αποθηκεύονται σε δυο μεταβλητές, τη *\$first* και τη *\$second* και το αποτέλεσμα της πρόσθεσης αποθηκεύεται σε μια τρίτη μεταβλητή, τη *\$result*.

3.3.5 Προσπέλαση Μεταβλητών με την Εντολή Global

Συνήθως οι μεταβλητές που ορίζονται έξω από μια συνάρτηση δεν είναι προσπελάσιμες από αυτήν. Έστω το παράδειγμα :

```
1 <html>
2 <head>
3 <title>Mi Prospelasimes Metablites</title>
4 </head>
5 <body>
6 <?php
7 $num=3;
8 function numberofrounds(){
9     print "The number of rounds is $num <br>";
10 }
11 numberofrounds();
12 ?>
13 </body>
14 </html>
```

Εικ.3.21 Παράδειγμα μη προσπελάσιμης μεταβλητής.

Εαν αποθηκεύσουμε το παραπάνω script έστω με το όνομα *test.php* στο φάκελο εγγράφων του web server και το προσπελάσουμε μέσω του browser που διαθέτει ο υπολογιστής μας, στην οθόνη θα εμφανιστεί η φράση “*The number of rounds is*”. Όπως μπορούμε δηλαδή να καταλάβουμε, η συνάρτηση *numberofrounds()* δεν έχει πρόσβαση στη μεταβλητή *\$num* και γενικά η μεταβλητή *\$num* είναι κενή όταν η συνάρτηση προσπαθεί να εκτυπώσει την τιμή της. Έχουμε όμως τη δυνατότητα να προσπελάσουμε μια μεταβλητή μέσω μιας συνάρτησης και ας μην την έχουμε παράσει σαν όρισμα της. Σε αυτό μας χρησιμεύει η εντολή **global**. Στο προηγούμενο παράδειγμα θα προσθέσουμε την εντολή **global** για να έχουμε το επιθυμητό αποτέλεσμα.

```
1 <html>
2 <head>
3 <title>Mi Prospelasimes Metablites</title>
4 </head>
5 <body>
6 <?php
7 $num=3;
8 function numberofrounds(){
9     global $num;
10    print "The number of rounds is $num <br>";
11 }
12 numberofrounds();
13 ?>
14 </body>
15 </html>
```

Εικ.3.22 Παράδειγμα εντολής global.

Όπως παρατηρούμε στη γραμμή 9 με την εντολή `global` μπροστά από τη μεταβλητή `$num` υποχρεώνουμε τη συνάρτηση `numberofrounds()` στην οποία είχαμε δηλώσει την `$num` να αναφερθεί στη μεταβλητή αυτή, η οποία έχει οριστεί έξω από τη συνάρτηση. Εάν αυτό το script το αποθηκεύσουμε στο φάκελο εγγράφων του web server, για παράδειγμα ως `global.php` και το προσπελάσουμε μέσω του browser, τότε στην οθόνη μας θα εμφανιστεί η φράση “The number of rounds is 3”.

3.3.6 Δημιουργία Ανώνυμων Συναρτήσεων

Υπάρχουν συναρτήσεις οι οποίες δεν παίρνουν κάποιο όνομα, αλλά αποθηκεύονται σε μεταβλητές, ή περνιούνται σε άλλες συναρτήσεις και για αυτό το λόγο αναφέρονται γενικά ως ανώνυμες συναρτήσεις. Η PHP διαθέτει τη συνάρτηση `create_function()` για τη δημιουργία των ανώνυμων συναρτήσεων.

Η `create_function()` απαιτεί δυο ορίσματα, αλφαριθμητικά. Το πρώτο όρισμα πρέπει να περιέχει μια λίστα μεταβλητών, χωρισμένες μεταξύ τους με κόμματα και το δεύτερο όρισμα πρέπει να περιέχει τον κώδικα της συνάρτησης. Έστω λοιπόν ένα παράδειγμα για την πληρέστερη κατανόηση των ανώνυμων συναρτήσεων .

```
1 <html>
2 <head>
3 <title>Anonimes Synartiseis</title>
4 </head>
5 <body>
6 <?php
7 $anon=create_function('$a,$b', 'return $a+$b;');
8 print $anon(3,5);
9 ?>
10 </body>
11 </html>
```

Εικ.3.23 Παράδειγμα ανώνυμης συνάρτησης.

Όπως μπορούμε να παρατηρήσουμε, στη γραμμή 7 έχουμε χρησιμοποιήσει αποστρόφους για το πέρασμα των ορισμάτων της `create_function`. Αυτό το κάναμε για να είναι απλούστερη η συνάρτηση και να μη χρειαστεί να βάλουμε εισαγωγικά ή το σήμα της αντίθετης κάθετου (backslash) που θα την έκαναν πιο πολύπλοκη. Εάν τώρα αποθηκεύσουμε και αυτό το script με τη σειρά του στο φάκελο των εγγράφων του web server (`anon.php`), και το προσπελάσουμε μέσω του browser του υπολογιστή μας, τότε θα εμφανιστεί στην οθόνη μας το αποτέλεσμα της πρόσθεσης της

μεταβλητής $\$a$ (που είναι ίση με τρία) και της $\$b$ (που είναι ίση με πέντε), το οποίο είναι ο αριθμός 8 .

Σε αυτήν την ενότητα ασχοληθήκαμε με την ανάλυση συναρτήσεων της PHP. Αναλύσαμε όλα τα βασικά σημεία που αφορούν τις συναρτήσεις, όπως τα είδη των συναρτήσεων, πως δημιουργούμε και πως ενσωματώνουμε συναρτήσεις μέσα σε ένα script. Στην ενότητα που ακολουθεί θα αναφερθούμε στην επικοινωνία μιας βάσης δεδομένων MySQL με την σελίδα μας μέσω της PHP .

3.4 Επικοινωνία με Βάσεις Δεδομένων MySQL μέσω της PHP

3.4.1 Σύνδεση στην MySQL με την PHP

Αφού έχουμε δημιουργήσει μια βάση δεδομένων MySQL και επιθυμούμε μια ιστοσελίδα μας να καλεί κάποιες πληροφορίες από τη βάση αυτή, το πρώτο πράγμα που πρέπει να κάνουμε είναι να χρησιμοποιήσουμε την PHP για να **συνδεθούμε** στη βάση μας. Η βάση δεδομένων θα πρέπει να έχει κάποιο όνομα (όνομα_βάσης), κάποιο όνομα χρήστη (username) και κάποιο κωδικό πρόσβασης (password), τα οποία θα πρέπει να γνωρίζουμε για να κάνουμε τη σύνδεση. Στη σύνδεση και την επικοινωνία της βάσης μας με την σελίδα θα μας βοηθήσουν κάποιες συναρτήσεις της PHP.

1. Χρήση της `mysql_connect()`

Η συνάρτηση `mysql_connect()` είναι η πρώτη που πρέπει να καλέσουμε για να συνδεθούμε στην MySQL. Η γενική σύνταξη της `mysql_connect()` είναι η: `mysql_connect (όνομα_host, όνομα_χρήστη, κωδικός_πρόσβασης);`.

Η συνάρτηση αυτή επιστρέφει ένα δείκτη σύνδεσης (connection index) εάν η σύνδεση είναι επιτυχής, ή false εάν η σύνδεση δεν υλοποιηθεί. Συνήθως τη συνάρτηση της σύνδεσης τη γράφουμε μέσα σε μια μεταβλητή `$conn` για περισσότερη ευκολία, σε περίπτωση που θέλουμε αργότερα να κλείσουμε τη σύνδεση και μετά ξανά να την ανοίξουμε.

Ας δούμε λοιπόν ένα παράδειγμα σύνδεσης όπου το όνομα `_host` (δηλαδή, ο web server) είναι το `localhost`, το όνομα `_χρήστη` είναι `vag` και ο κωδικός πρόσβασης είναι `dim`.

```
1 <?php
2 $conn=mysql_connect("localhost","vag","dim");
3 echo $conn;
4 >>
```

Εικ.3.24 Σύνδεση της βάσης με την σελίδα μέσω της PHP.

Στη γραμμή 3 έχουμε δώσει την εντολή να μας εμφανίσει στην οθόνη μας το αποτέλεσμα της σύνδεσης, δηλαδή τον κωδικό της μεταβλητής `$conn`. Εάν λοιπόν αποθηκεύσουμε αυτό το script με το όνομα, για παράδειγμα, `mysql_connect.php` στο φάκελο των εγγράφων του web server και το προσπελάσουμε μέσω του browser τότε στην οθόνη μας θα πρέπει να εμφανιστεί μια πρόταση σαν την *Resource id # 1*. Η ανάλυση της επικοινωνίας της MySQL με σελίδες μέσω της PHP είναι ένα κεφαλαίο που επιδέχεται πολλή ανάλυση. Εμείς στη συγκεκριμένη ενότητα έχουμε αναλύσει μόνο τη συνάρτηση της σύνδεσης και θα αναφερθούμε στη δημιουργία ερωτημάτων που είναι ίσως τα δυο πιο σημαντικά θέματα του κεφαλαίου που λέγεται σύνδεση.

2. Επιλογή Βάσης Δεδομένων και Εκτέλεση Ερωτημάτων

Για την εκτέλεση ερωτημάτων, η συνάρτηση που θα χρησιμοποιήσουμε είναι η `mysql_query()`. Εάν γνωρίζουμε τη `mysql_query()` για τη δημιουργία ερωτημάτων SQL στην MySQL, σημαίνει ότι γνωρίζουμε πάνω από τα μισά που αφορούν τα ερωτήματα τέτοιου είδους. Επειδή σε ένα script μπορούμε να καλούμε πληροφορίες από περισσότερες από μια βάσεις δεδομένων mysql, χρησιμοποιούμε για την επιλογή αυτή τη συνάρτηση `mysql_select_db()` της οποίας η γενική σύνταξη είναι:

`mysql_select_db (όνομα_βάσης_δεδομένων, δείκτης_σύνδεσης);`

Στη συγκεκριμένη ενότητα θα αναφέρουμε διαφορετικά παραδείγματα από την εργασία που ασχοληθήκαμε, αφού το επόμενο κεφαλαίο ασχολείται αποκλειστικά με την ανάλυση του script της συγκεκριμένης εργασίας.

Ας γράψουμε ένα παράδειγμα για να δούμε πως διαλέγουμε τη βάση δεδομένων μας και πως θέτουμε ένα ερώτημα.

```

1 <?php
2 $conn=mysql_connect("localhost","username","password");
3 mysql_select_db ("DB", $conn);
4 $sql="CREATE TABLE testTable (id int not null primary key auto_increment,
5 testField varchar(75))";
6 $result=mysql_query($sql, $conn);
7 echo $result;
8 ?>

```

Εικ.3.25 Παράδειγμα για την επιλογή βάσης και την δημιουργία ερωτήματος.

Επειδή το συγκεκριμένο παράδειγμα είναι λίγο πολύπλοκο θα το αναλύσουμε για την καλύτερη κατανόηση του. Στη γραμμή **2** συνδεθήκαμε με την MySQL. Το ονομα_χρηστη και ο κωδικός_πρόσβασης που βάλαμε είναι τυχαία. Ύστερα επιλέξαμε στη γραμμή **3** τη βάση δεδομένων μας, την *DB*, της οποίας το όνομα είναι και αυτό τυχαίο. Στη συνέχεια, στις γραμμές **4** και **5**, δημιουργήσαμε στη *mysql* ένα δοκιμαστικό πίνακα, τον *testTable* με δυο πεδία, το *id* και το *testField*. Στο τέλος του script ορίσαμε τη μεταβλητή *\$result* η οποία διενεργεί ένα ερώτημα *mysql_query*, το οποίο ρωτάει ποιο είναι το αποτέλεσμα της μεταβλητής *\$sql* (η οποία δημιούργησε τον πίνακα) μέσω της σύνδεσης *\$conn*. Η συνάρτηση *mysql_query* επιστρέφει τιμές true ή false, στη συγκεκριμένη περίπτωση θα επιστρέψει *1* εάν το αποτέλεσμα είναι true και *0* εάν το αποτέλεσμα είναι false. Στο παράδειγμα που μόλις αναφέραμε, το αποτέλεσμα είναι false, δηλαδή 0, γιατί τα ονόματα που προαναφέραμε είναι τυχαία και ως αποτέλεσμα δεν μπορεί να γίνει η σύνδεση.

Στο κεφαλαίο αυτό αναφέραμε τα βασικά γνωρίσματα της PHP. Εάν κάποιος γνωρίζει αυτά που προαναφέρθηκαν είναι σίγουρο ότι μπορεί να κάνει μια καλή αρχή στη δημιουργία δυναμικών σελίδων. Στο κεφαλαίο που ακολουθεί θα αναλύσουμε τον κώδικα της πτυχιακής μας εργασίας.

4. Apache Server

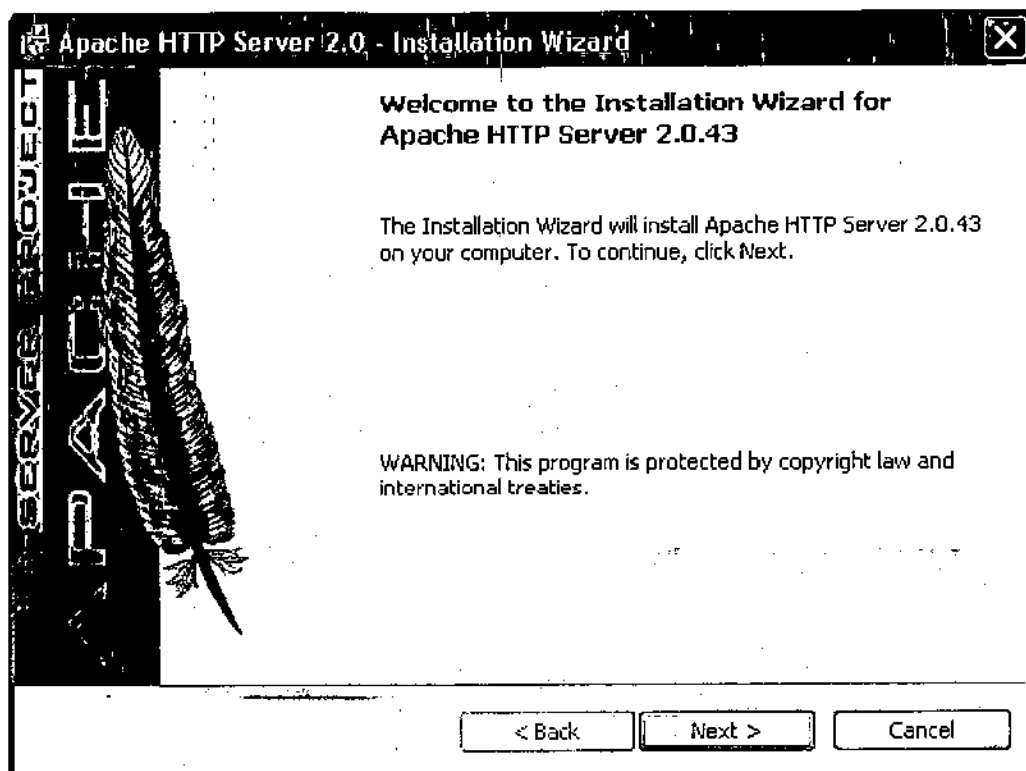
4.1 Εγκατάσταση του Apache Server στα Windows

Ξεκινώντας θα ήταν σωστό να αναφέρουμε τι είναι ένας web server. Καταρχήν, ένας web server είναι ένα πρόγραμμα το οποίο τρέχει σε έναν υπολογιστή (ο οποίος επίσης λέγεται web server). Ο web server περιμένει αιτήσεις από διάφορους browser (από χρήστες του διαδικτύου) και προσφέρει τα απαραίτητα αρχεία (html αρχεία, εικόνες, flash animations κ.τ.λ) στους χρήστες τους. Ο Apache web server είναι ένα ελεύθερο – ανοικτό λογισμικό, το οποίο διατίθεται δωρεάν (free open source) και λειτουργεί πολύ καλά σε πολλά λειτουργικά συστήματα, όπως τα Windows και το Linux. Συνήθως, αποκαλείται απλά Apache.

Η πρώτη έκδοση του Apache ήταν βασισμένη στον NCSA httpd Web server και κατασκευάστηκε το 1995 από μια ομάδα περίπου 20 εθελοντών προγραμματιστών, γνωστοί με το όνομα Apache Group. Επειδή ο πηγαίος κώδικας είναι διαθέσιμος, ο καθένας μπορεί να υιοθετήσει τον Apache για συγκεκριμένες ανάγκες, και υπάρχει και μια μεγάλη δημόσια βιβλιοθήκη με προσθήκες (add-ons) του Apache. Η πρωτότυπη έκδοση του Apache είχε γραφτεί για να λειτουργεί σε UNIX.

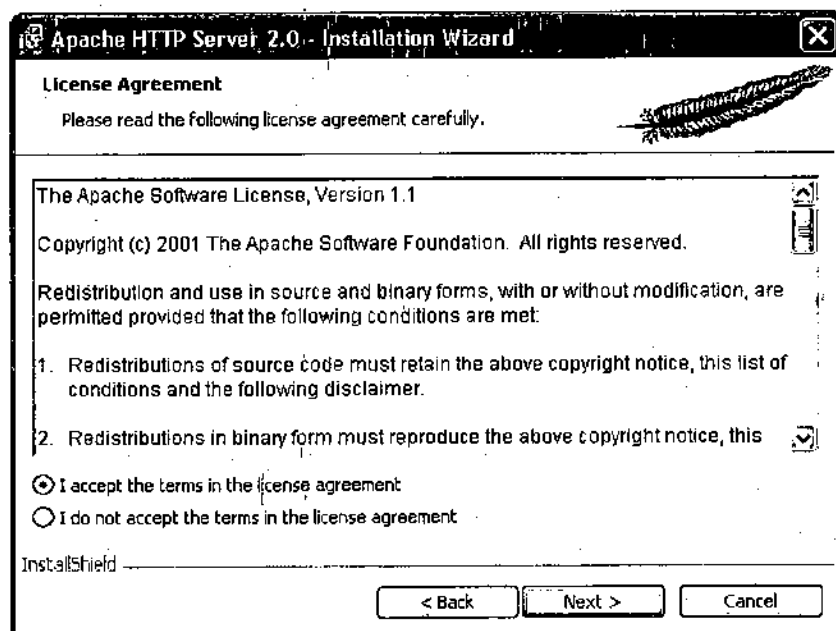
Το όνομα του είναι αφιερωμένο στους Αμερικάνους ιθαγενείς ινδιάνους, τους Apache, μια φυλή γνωστή για την αντοχή και τα προσόντα της στον πόλεμο. Εάν σκεφτούμε ότι περίπου το 70% των web server που τρέχουν στο διαδίκτυο είναι Apache, μπορούμε να καταλάβουμε γιατί είναι ο δημοφιλέστερος web server σε χρήση σήμερα.

Πριν εγκαταστήσουμε τον Apache θα πρέπει να διασφαλιστεί ότι στο σύστημά μας δεν τρέχει ήδη ένας web server. Καλό θα είναι να καταργήσουμε ή να απενεργοποιήσουμε την εγκατάσταση και λειτουργία των υπάρχοντων servers. Αφού επιλέξουμε το εικονίδιο εγκατάστασης του apache server, το οποίο στη συγκεκριμένη περίπτωση έχει το όνομα `'apache_2.0.43-win32-x86-no_ssl.exe'` θα ακολουθήσουμε τα στάδια της διαδικασίας που μας εμφανίζει ο οδηγός εγκατάστασης (installation wizard). Η πρώτη οθόνη που θα εμφανιστεί είναι μια οθόνη καλωσορίσματος όπως φαίνεται και στην εικόνα 4.1.



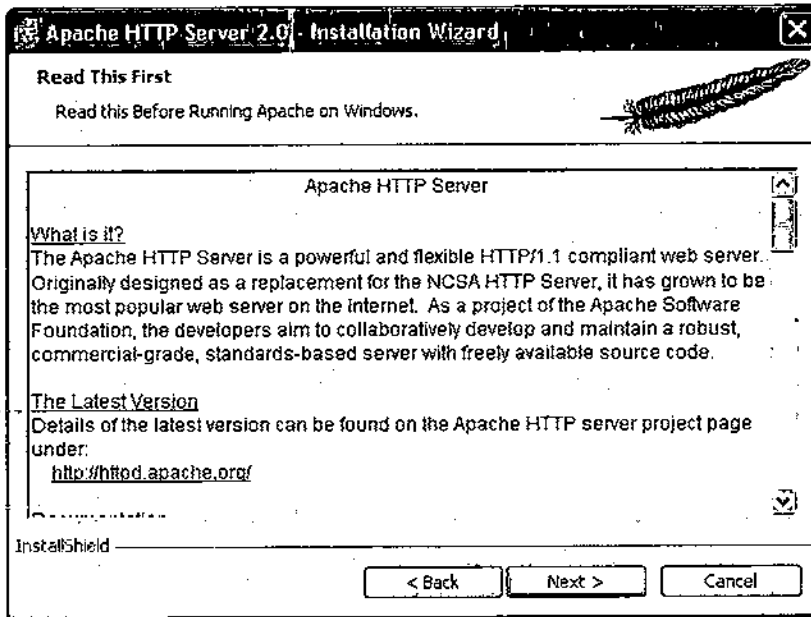
Εικ.4.1 Οθόνη καλωσορίσματος.

Κάνοντας κλικ στο *Next* συνεχίζουμε με τη διαδικασία της εγκατάστασης, όπου στην οθόνη που ακολουθεί θα μας ζητηθεί να αποδεχτούμε τους όρους της συμφωνίας άδειας χρήσης του προγράμματος όπως φαίνεται και στην παρακάτω εικόνα.



Εικ.4.2 Όροι άδειας χρήσης.

Αφού αποδεχθούμε τους όρους, επιλέγοντας την αντίστοιχη πρόταση (*I accept the terms in the license agreement*), κάνοντας κλικ στο *Next* περνάμε στο επόμενο στάδιο στο οποίο μας εμφανίζεται μια οθόνη εισαγωγής στον apache server με γενικές πληροφορίες, όπως τι είναι, που χρησιμεύει, ποια έκδοση είναι και που μπορούμε να το βρούμε.



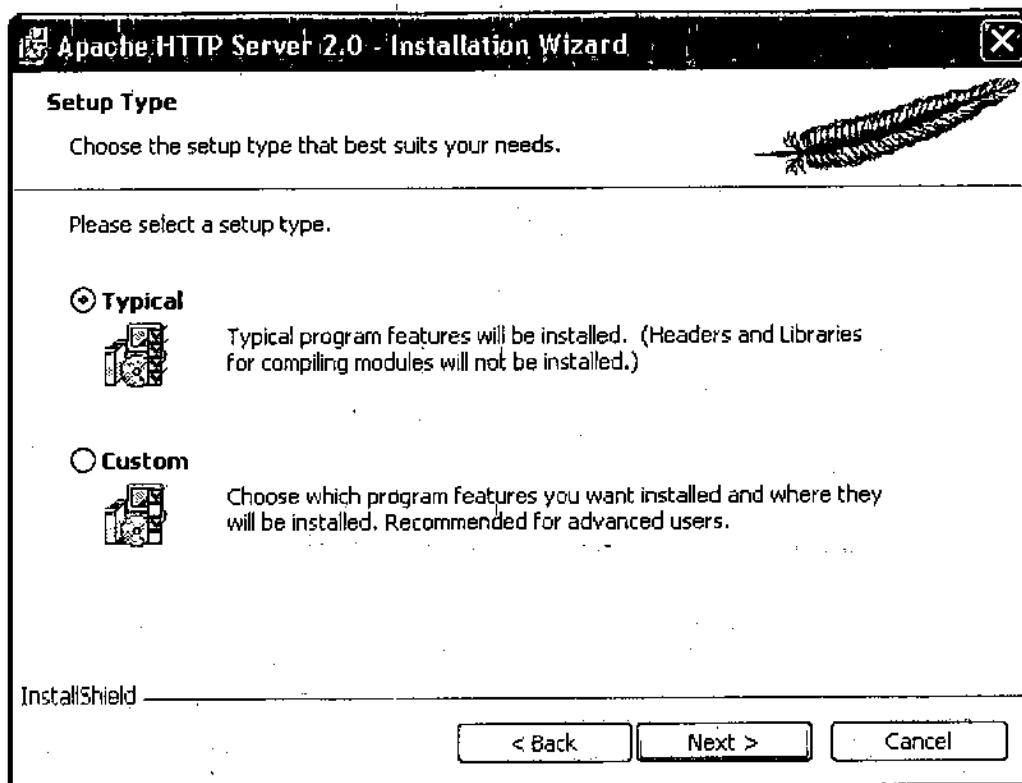
Εικ.4.3 Πληροφορίες για τον Apache server.

Στη συνέχεια, μέσω μιας φόρμας, μας ζητάει κάποιες βασικές πληροφορίες για τον υπολογιστή, όπως η διεύθυνση δικτύου για το server, το όνομα domain και όνομα του server, καθώς και η διεύθυνση ηλεκτρονικού ταχυδρομείου του διαχειριστή του server.

Εικ.4.4 Φόρμα για τις πληροφορίες του server.

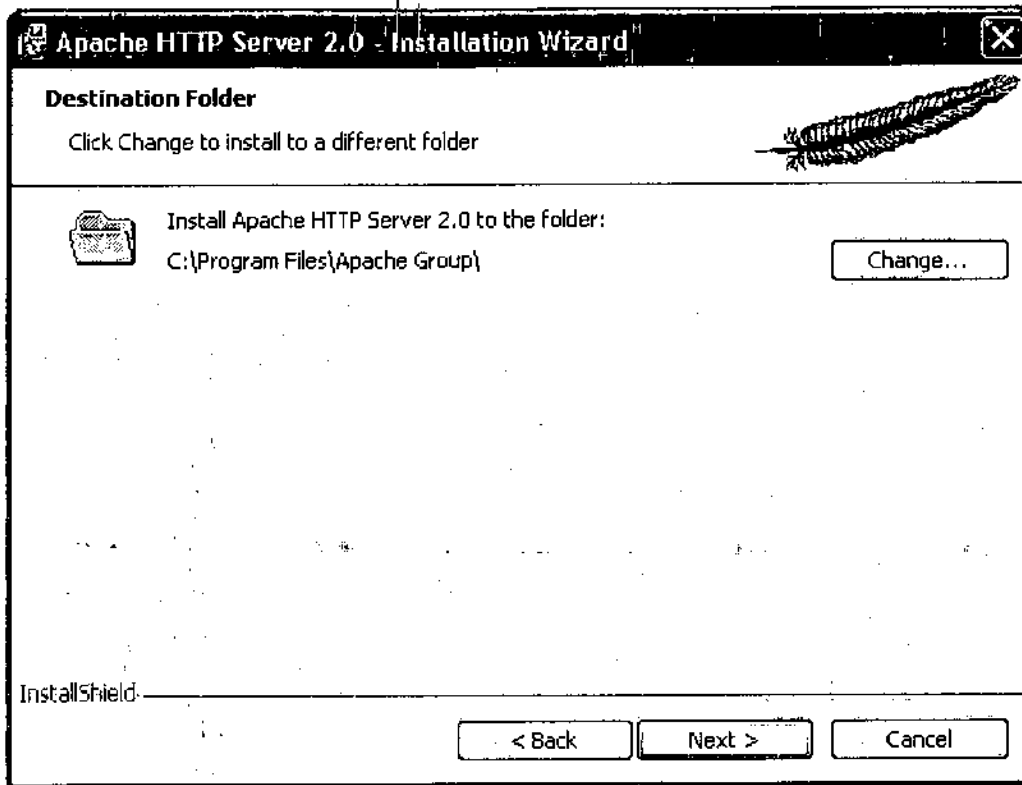
Το όνομα του server θα είναι το όνομα που θα χρησιμοποιούν οι πελάτες (client) υπολογιστές για να τον προσπελάσουν και η διεύθυνση ηλεκτρονικού ταχυδρομείου του διαχειριστή (administrator) θα προστίθεται στα μηνύματα σφάλματος, έτσι ώστε οι επισκέπτες να ξέρουν πώς να επικοινωνήσουν μαζί μας όταν προκύπτουν προβλήματα. Στη συγκεκριμένη περίπτωση οι πληροφορίες που έχουμε συμπληρώσει είναι υποθετικές, εάν είχαμε το συγκεκριμένο ταξιδιωτικό γραφείο.

Συνεχίζοντας, η επόμενη οθόνη έχει δύο επιλογές που αφορούν στον τύπο εγκατάστασης του προγράμματος όπως βλέπουμε και στην εικόνα 4.5. Με την τυπική (typical) εγκατάσταση, εγκαθιστούμε τα δυαδικά αρχεία του apache και τα αρχεία πληροφοριών τεκμηρίωσης, αλλά δεν εγκαθίστανται τα header αρχεία και οι βιβλιοθήκες. Αυτή είναι η προτιμώμενη επιλογή (recommended), εκτός και εάν κάποιος επιθυμεί να μεταγλωττίσει δικές του ρουτίνες όπου θα πρέπει να διαλέξει την άλλη επιλογή (custom).



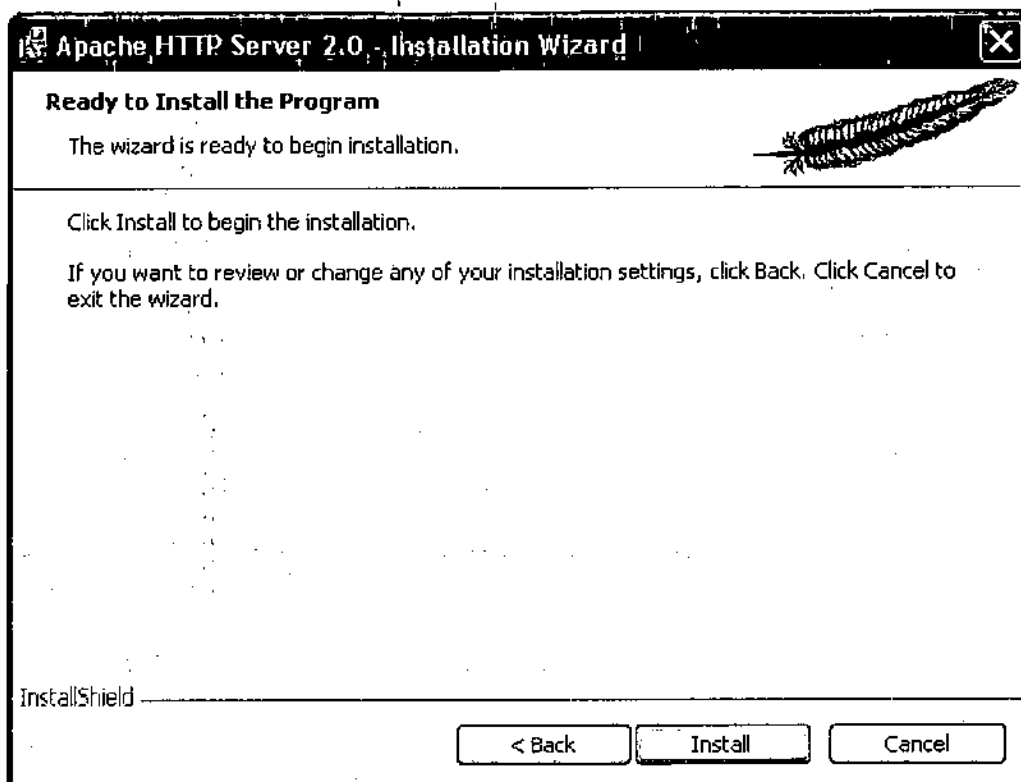
Εικ.4.5 Επιλογή τύπου εγκατάστασης του apache.

Στον εξειδικευμένο (custom) τύπο εγκατάστασης μας δίνεται η δυνατότητα επιλογής είτε της εγκατάστασης με τα header αρχεία ή με τα αρχεία πληροφοριών τεκμηρίωσης. Ακολούθως στην επόμενη οθόνη μας δίνεται η δυνατότητα να επιλέξουμε το φάκελο μέσα στο σύστημά μας που θα εγκατασταθεί το πρόγραμμα. Ο Apache μπορεί να αποθηκευτεί οπουδήποτε στο σύστημα του υπολογιστή μας, αλλά προτιμότερο θα ήταν να εγκατασταθεί στον προτεινόμενο από το πρόγραμμα κατάλογο όπως φαίνεται και στην εικόνα 4.6.

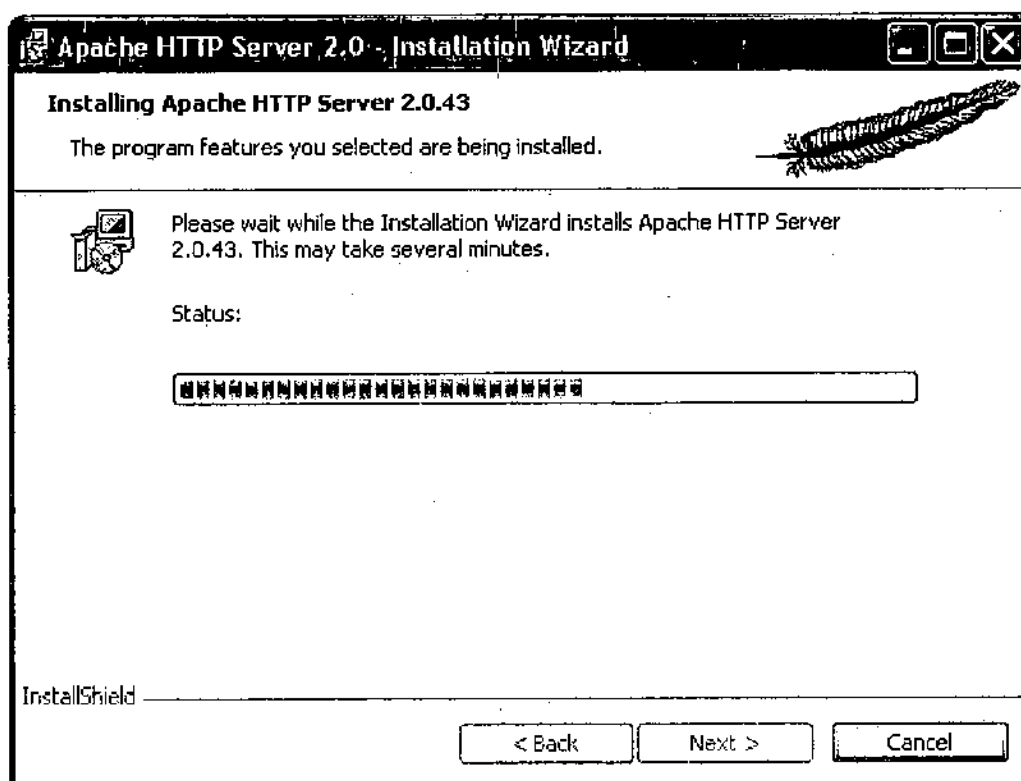


Εικ.4.6 Επιλογή καταλόγου εγκατάστασης.

Στις εικόνες 4.7 και 4.8 φαίνονται τα δύο τελευταία στάδια της εγκατάστασης του Apache στον υπολογιστή μας, τα οποία αντίστοιχα μας εμφανίζουν την επιλογή της εγκατάστασης και την πρόοδό της.

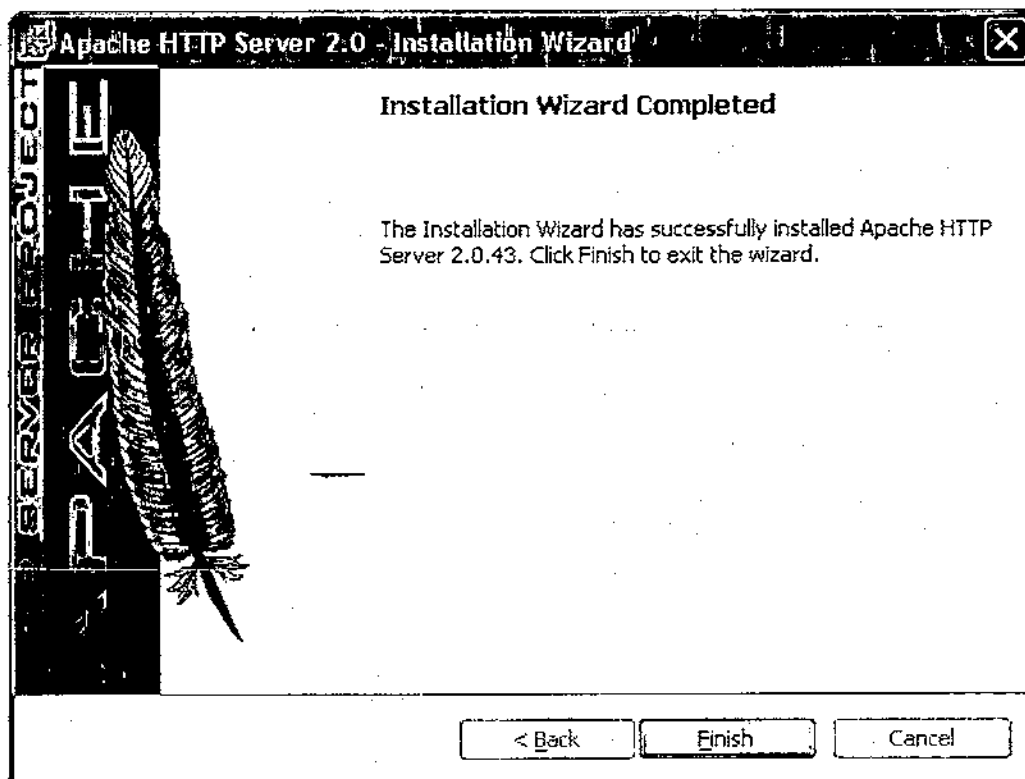


Εικ.4.7 Τελευταία εικόνα πριν την εγκατάσταση.



Εικ.4.8 Πρόοδος εγκατάστασης.

Μετά από τα στάδια που αναφέραμε και αφού ολοκληρωθούν επιτυχώς, μας εμφανίζεται και η τελική εικόνα, που είναι η παρακάτω, η οποία επιβεβαιώνει ότι η εγκατάσταση ολοκληρώθηκε επιτυχώς.

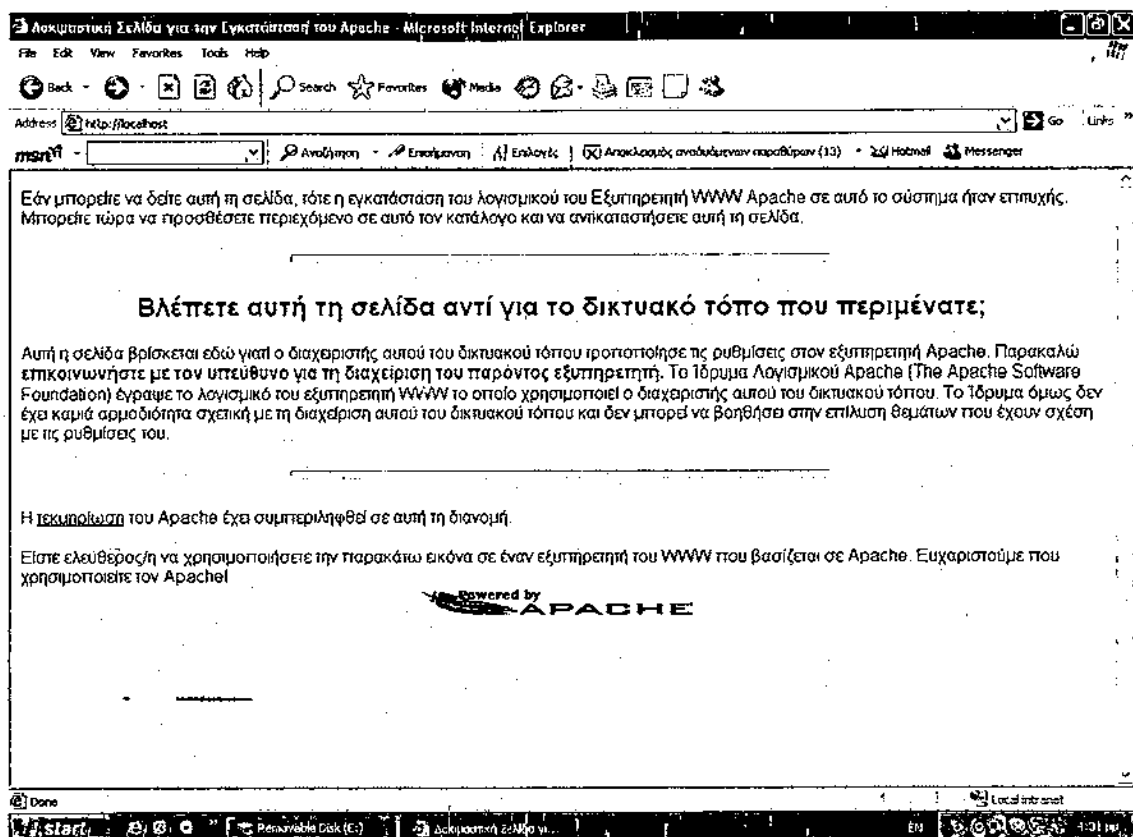


Εικ.4.9 Τελική εικόνα μετά την εγκατάσταση.

Κάνοντας κλικ στην επιλογή *Finish* οριστικοποιούμε την εγκατάσταση του προγράμματος.

4.2 Εκκίνηση του Apache.

Για να εκκινήσουμε τον Apache στα Windows, κάνουμε κλικ στη σύνδεση **Start Apache** στην ενότητα **Control Apache** του μενού **Start** (εάν το Apache έχει εγκατασταθεί σαν μια υπηρεσία, τότε θα πρέπει να εκκινηθεί η υπηρεσία του Apache) και αν όλα έχουν εξελιχθεί ομαλά, μπορούμε να προσπελάσουμε τον Apache χρησιμοποιώντας μια εφαρμογή φυλλομετρητή (Browser), όπου και θα μας εμφανιστεί η προεπιλεγμένη σελίδα που παρουσιάζεται παρακάτω.



Εικ.4.10 Οθόνη επιβεβαίωσης επιτυχίας της εγκατάστασης.

4.3 Συνεργασία της PHP με το Apache

Η συνεργασία της PHP με τον Apache προϋποθέτει την εγκατάσταση των αρχείων της PHP στα Windows και την μεταφορά τους σε συγκεκριμένες θέσεις, κάτι το οποίο γίνεται ακολουθώντας τα παρακάτω απλά στάδια:

1. Επιλέγουμε το συμπιεσμένο (zip) αρχείο της PHP και αφού το μεταφέρουμε στο σύστημά μας, κάνοντας διπλό κλικ επάνω του, το αποσυμπιέζουμε. Αφού τα βασικά αρχεία της έχουν πλέον αποσυμπιεστεί, αυτό που χρειάζεται είναι να μετακινήσουμε ορισμένα από αυτά στις σωστές θέσεις:
2. Στον κατάλογο εγκατάστασης της PHP, εντοπίζουμε το αρχείο **php.ini-dist** και το μετονομάζουμε σε **php.ini**.
3. Μετακινούμε το αρχείο **php.ini** στον κατάλογο **C:\WINDOWS**, ή σε οποιαδήποτε θέση τοποθετούμε όλα τα .ini αρχεία στο σύστημά μας.
4. Μετακινούμε το αρχείο **php4ts.dll** στον κατάλογο **C:\WINDOWS\SYSTEM**, ή σε όποια θέση τοποθετούμε τα .dll αρχεία μας.

Αφού έχουμε πλέον εγκαταστήσει και τα αρχεία της PHP, για να διασφαλίσουμε ότι θα συνεργαστεί ομαλά με τον Apache θα πρέπει να προσθέσουμε ορισμένες καταχωρίσεις στο αρχείο διαμόρφωσης του τελευταίου, `httpd.conf`. Σε αυτό το αρχείο αφού εντοπίσουμε την ακόλουθη ενότητα:

```
# Example:
# LoadModule foo_module modules/mod_foo.so
#
LoadModule access_module modules/mod_access.so
LoadModule actions_module modules/mod_actions.so
LoadModule alias_module modules/mod_alias.so
LoadModule asis_module modules/mod_asis.so
LoadModule auth_module modules/mod_auth.so
#LoadModule auth_anon_module modules/mod_auth_anon.so
#LoadModule auth_dbm_module modules/mod_auth_dbm.so
#LoadModule auth_digest_module modules/mod_auth_digest.so
LoadModule autoindex_module modules/mod_autoindex.so
#LoadModule cern_meta_module modules/mod_cern_meta.so
LoadModule cgi_module modules/mod_cgi.so
#LoadModule dav_module modules/mod_dav.so
#LoadModule dav_fs_module modules/mod_dav_fs.so
LoadModule dir_module modules/mod_dir.so
LoadModule env_module modules/mod_env.so
#LoadModule expires_module modules/mod_expires.so
#LoadModule file_cache_module modules/mod_file_cache.so
#LoadModule headers_module modules/mod_headers.so
LoadModule imap_module modules/mod_imap.so
LoadModule include_module modules/mod_include.so
#LoadModule info_module modules/mod_info.so
LoadModule isapi_module modules/mod_isapi.so
LoadModule log_config_module modules/mod_log_config.so
LoadModule mime_module modules/mod_mime.so
#LoadModule mime_magic_module modules/mod_mime_magic.so
#LoadModule proxy_module modules/mod_proxy.so
#LoadModule proxy_connect_module modules/mod_proxy_connect.so
#LoadModule proxy_http_module modules/mod_proxy_http.so
#LoadModule proxy_ftp_module modules/mod_proxy_ftp.so
LoadModule negotiation_module modules/mod_negotiation.so
#LoadModule rewrite_module modules/mod_rewrite.so
LoadModule setenvif_module modules/mod_setenvif.so
#LoadModule spelling_module modules/mod_spelling.so
#LoadModule status_module modules/mod_status.so
#LoadModule unique_id_module modules/mod_unique_id.so
LoadModule userdir_module modules/mod_userdir.so
#LoadModule usertrack_module modules/mod_usertrack.so
#LoadModule vhost_alias_module modules/mod_vhost_alias.so
#LoadModule ssl_module modules/mod_ssl.so
```

Στο τέλος της θα προσθέσουμε τα ακόλουθα:

```
LoadModule php4_module c:/php-version/sapi/php4apache2.dll
```

Στη συνέχεια θα αναζητήσουμε την ακόλουθη ενότητα:

```
#  
# AddType allows you to add to or override the MIME configuration  
# file mime.types for specific file types.  
#
```

Και θα προσθέσουμε τις ακόλουθες γραμμές:

```
AddType application/x-httpd-php .php .phtml .html
```

```
AddType application/x-httpd-php-source .phps
```

Τελειώνοντας με αυτά αποθηκεύουμε το αρχείο αυτό και επανεκκινούμε τον Apache Server. Από εδώ και στο εξής η PHP αποτελεί μέρος του Apache WebServer.

Οι παραπάνω προσθήκες διασφαλίζουν ότι ο μηχανισμός εκτέλεσης της PHP θα διερμηνεύει τα αρχεία με επεκτάσεις **.php**, **.phtml**, **.html**. Οποιαδήποτε αρχεία έχουν επέκταση **.phps** θα εξάγονται σαν πηγαία αρχεία PHP, δηλαδή ο πηγαίος κώδικας θα μετατρέπεται σε HTML και η σύνταξή του θα επισημαίνεται με χρώματα, κάτι το οποίο θα μας βοηθήσει στην αποσφαλμάτωση των script μας.

5. Ανάλυση του script της εργασίας

Έχοντας παρουσιάσει τα προηγούμενα κεφάλαια που αφορούσαν στην HTML, στη MySQL, στην PHP και στον Apache web server, στο κεφάλαιο αυτό θα αναλύσουμε το script της πτυχιακής εργασίας μας.

Το script είναι γραμμένο σε HTML με χρήση κώδικα PHP, το οποίο κάνει και τη σύνδεση της ιστοσελίδας με τη βάση δεδομένων που έχουμε δημιουργήσει στη MySQL. Το script που έχουμε δημιουργήσει είναι συνολικά 124 σειρές, ξεκινώντας την ανάλυση θα δούμε στην εικόνα που ακολουθεί τις πρώτες 11 σειρές.

```
1 <html>
2 <head>
3 <title>Dream Travel Agency</title>
4 </head>
5 <body background="c:\ptychiakh\cancer.jpg" text=white link=yellow vlink=white>
6 <font size=9><h1 align=center><b><u>DREAM TRAVEL AGENCY</b></u></h1></font>
7 <p>
8 <marquee direction="right" behavior="alternate">Live the Days of Your Dreams</marquee></p>
9
10 <bgsound src="c:\ptychiakh\Eurithmicks-Sweet Dreams.mp3" loop="5">
11
```

Εικ.5.1 Η αρχή του script.

Όπως μπορούμε να διακρίνουμε αυτό το αρχικό κομμάτι του κώδικα είναι σε μορφή HTML. Στην πρώτη γραμμή δηλώνουμε ότι ξεκινάμε τη συγγραφή κώδικα HTML γράφοντας την εντολή ανοίγματος `<html>`. Στην συνέχεια, στη δεύτερη γραμμή γράφουμε την εντολή ανοίγματος του “κεφαλιού” του κώδικά μας τη `<head>`, η οποία κλείνει στη γραμμή 4 γράφοντας `</head>`. Μέσα στη `<head>...</head>` έχουμε τοποθετήσει τον τίτλο της σελίδας μας. Ανάμεσα στο άνοιγμα και στο κλείσιμο της εντολής `<title>...</title>` τοποθετείται ο οποίος είναι και το όνομα του «φανταστικού» ταξιδιωτικού γραφείου (Dream Travel Agency) του οποίου την σελίδα αποφασίσαμε να δημιουργήσουμε. Ο τίτλος αυτός δεν εμφανίζεται πάνω στη σελίδα μας, αλλά στη γραμμή τίτλου του παραθύρου του φυλλομετρητή (web browser) ενός χρήστη, κάθε φορά που αυτός συνδέεται με την ιστοσελίδα του ταξιδιωτικού μας γραφείου (πληκτρολογώντας τη διεύθυνση του γραφείου, πχ. www.dreamta.gr). Συνεχίζοντας, στη γραμμή 5 ανοίγουμε το κυρίως σώμα του κώδικά μας με την εντολή `<body>`, στην οποία έχουμε ορίσει διάφορα στοιχεία που αφορούν στην εμφάνιση της σελίδας μας. Στη συγκεκριμένη περίπτωση δηλαδή, μια

εικόνα η οποία τοποθετείται ως φόντο στην ιστοσελίδα, το χρώμα του κειμένου, το χρώμα των παραπομπών πρίν κάνουμε κλίκ (*link*) με το ποντίκι πάνω τους και το χρώμα των παραπομπών αφού έχουμε κάνει κλίκ (*vlink*) με το ποντίκι μας πάνω σε αυτές.

Αναλύοντας τώρα τη γραμμή 6 μπορούμε να διακρίνουμε ότι σε αυτή τη γραμμή δημιουργήσαμε τον τίτλο που θα εμφανίζεται στο πάνω μέρος της ιστοσελίδας μας. Γράφοντας την εντολή `...` δηλώνουμε το μέγεθος της γραμματοσειράς του τίτλου που θα βάλουμε, το οποίο βέβαια συνδυάζεται με την εντολή `<h1 align = center>...</h1>`, η οποία μας δηλώνει ότι ο τίτλος θέλουμε να είναι με μεγάλα γράμματα και εκτός αυτού ότι η στοίχιση του τίτλου θέλουμε να είναι στο κέντρο. Τέλος, τον τίτλο τον έχουμε τοποθετήσει ανάμεσα από τις εντολές `...` και `<u>...</u>` οι οποίες μετατρέπουν τα γράμματα σε έντονα (bold) και τα υπογραμμίζουν αντίστοιχα.

Ακολούθως, στη γραμμή 8 με χρήση της `<marquee>...</marquee>` έχουμε δημιουργήσει μια μαρκίζα η οποία γράφει "*Live the Days of your Dreams*", η οποία μετακινείται προς τα δεξιά και μόλις φτάσει στην άκρη της σελίδας γυρίζει πίσω, κάτι το οποίο κάνει επ'άπειρον. Χρησιμοποιώντας την εντολή `<p>...</p>` δηλώνουμε ότι θέλουμε να αλλάξουμε παράγραφο και για αυτό τον λόγο η μαρκίζα που δημιουργούμε τοποθετείται κάτω από τον τίτλο. Τελειώνοντας με αυτό το κομμάτι του script τη γραμμή 9 την έχουμε αφήσει κενή χωρίς αυτό να σημαίνει κάτι, απλά το κάναμε για δική μας ευκολία έτσι ώστε ο κώδικας να είναι πιο ευανάγνωστος για μας.

Στη γραμμή 10 με την εντολή `<bgsound>` έχουμε ορίσει ένα μουσικό αρχείο το οποίο θα ακούγεται αυτόματα, όταν κάποιος προσπελάσει τη σελίδα μας. Με την παράμετρο `loop = 5` που έχουμε τοποθετήσει μέσα στην `bgsound` δηλώνουμε ότι το μουσικό αρχείο θα εκτελεστεί για 5 συνεχόμενες φορές, εάν κάποιος αφήσει την σελίδα μας ανοιχτή στο πρόγραμμα του φυλλομετρητή του.

Στην επόμενη εικόνα αναλύουμε τον κώδικα των γραμμών 12 ως και 14 όπου και χρησιμοποιούμε PHP στο script μας.

```
12 <?php  
13 $conn = mysql_connect("localhost", "vag", "dim") or die (mysql_error());  
14 mysql_select_db("dimva travel agency", $conn) or die (mysql_error());
```

Εικ.5.2 Εισαγωγή στον κώδικα της PHP και σύνδεση με τη βάση δεδομένων μας.

Με το που ξεκινάμε τον κώδικα php στη γραμμή 12, στην επόμενη γραμμή κάνουμε τη σύνδεση του script της ιστοσελίδας με τη βάση δεδομένων μας, με χρήση πάντα της PHP. Για να γίνει η σύνδεση χρειάζεται να δηλώσουμε το όνομα του web server, το username και το password για την πρόσβαση στα δεδομένα της MySQL, τα οποία στην συγκεκριμένη περίπτωση είναι *localhost*, *vag* και *dim* αντίστοιχα. Στη συνέχεια κάνουμε επιλογή της βάσης δεδομένων στην οποία θα γίνει πρόσβαση, με την εντολή *mysql_select_db* και επιλέγουμε τη συγκεκριμένη βάση, η οποία στην περίπτωση αυτής της πτυχιακής είναι η *dimva_travel_agency*. Όπως μπορούμε να διακρίνουμε και στην εντολή της σύνδεσης και στην εντολή επιλογής της βάσης δεδομένων έχουμε γράψει ότι εάν το script δε μπορεί να υλοποιήσει τις δύο αυτές εργασίες να μας βγάλει μήνυμα λάθους της MySQL.

Στη συνέχεια έχουμε χωρίσει τον κώδικά μας σε δύο τμήματα. Το πρώτο αφορά στο να μην φαίνονται τα αποτελέσματα του πίνακα της βάσης δεδομένων μας (`$_POST[op] != "view"`) και το δεύτερο στο να φαίνονται (`$_POST[op] == "view"`).

```

15 if ($_POST[op] != "view") {
16     <table width=100%><tr><td width=30%>;
17     <td align=left><img src=\\CAUROXID.jpg\\></td>
18     Rent a car";
19     <td align=left><img src=\\Hotel.jpg\\></td>
20     Hotels";
21     <td align=left><img src=\\mail.gif\\></td>
22     Contact Us</td>;
23
24     <td align=center valign=center>Mesogeion 450<br>Gaudi, Athens<br>T.K.:128 76<br>TEL.:2105684239</td>
25

```

Εικ.5.3 Όταν η συνθήκη είναι να μην φαίνονται τα αποτελέσματα της βάσης.

Όπως φαίνεται και στην εικόνα 5.3 η συνθήκη της γραμμής 15 ορίζει να μη φαίνονται τα αποτελέσματα της βάσης δεδομένων μας. Οπότε δηλώνουμε ότι εάν *\$post != view* δηλαδή δεν είναι ίσον με *view*, δε βλέπουμε τα αποτελέσματα της βάσης δεδομένων μας και ισχύουν όλα τα παρακάτω μέχρι να αλλάξει η συνθήκη αυτή. Δηλαδή, στην οθόνη μας εμφανίζονται καταρχήν τρεις παραπομπές, που είναι η παραπομπή για την ενοικίαση αυτοκινήτου **Rent a car**, που μας παραπέμπει στη σελίδα γνωστής εταιρίας ενοικίασης αυτοκινήτων (www.hertz.com), η παραπομπή για την ενοικίαση δωματίου σε κάποιο ξενοδοχείο σε οποιοδήποτε μέρος του κόσμου **Hotels**, που κάνοντας κλικ πάνω της μεταβαίνουμε στη σελίδα www.bookhotels.com και η παραπομπή **Contact us**, η οποία ανοίγει το πρόγραμμα αποδοχής, δημιουργίας και αποστολής e-mail. Επίσης, όπως παρατηρούμε στη γραμμή 24 έχουμε

δημιουργήσει ένα κομμάτι κειμένου όπου σύμφωνα με την εντολή που του έχουμε δώσει `<td width=20% align=center valign=center>` θα εμφανιστεί στο κέντρο της οθόνης μας. Το κομμάτι αυτό είναι η φανταστική διεύθυνση του ταξιδιωτικού γραφείου που έχουμε δημιουργήσει. Οι εντολές `td width`, `align` και `valign` είναι εντολές της HTML τις οποίες τις έχουμε χρησιμοποιήσει μέσα στην PHP.

```

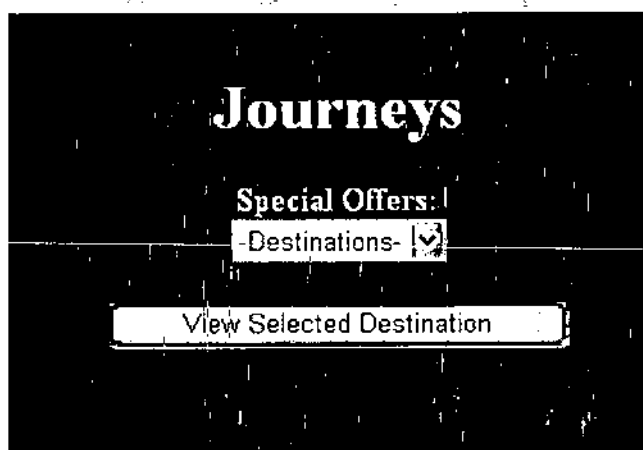
26 <td width=30% align=center valign=top><h1>Journeys</h1>;
27 $get_list = "select id, concat_ws(',', destinations) as display_destinations from Tours order by id";
28 $get_list_res = mysql_query($get_list) or die(mysql_error());
29 if (mysql_num_rows($get_list_res) < 1) {
30     $display_block .= "<p><em>Sorry, no records to select!</em></p>";
31 }
32 else {
33     $display_block .= "    <form method=\\"post\\"action=\\"$ _SERVER[PHP_SELF]\\">
34 <p><strong>Special Offers:</strong><br>
35 <select name=\\"sel_id\\">
36 <option value=\\"\">-Destinations-</option>;
37
38     while ($recs = mysql_fetch_array($get_list_res)) {
39         $id = $recs['id'];
40         $display_destinations = stripslashes($recs['display_destinations']);
41         $display_block .= "<option value=\\"$id\\">$display_destinations</option value>";
42     }
43
44     $display_block .= "
45 </select>
46 <input type=\\"hidden\\"name=\\"op\\"value=\\"view\\">
47 <p><input type=\\"submit\\"name=\\"submit\\"
48 value=\\"View Selected Destination\\"></p>
49 </FORM></td></tr></table>;
50
51
52 }
53
54

```

Εικ.5.4 Συνέχεια της συνθήκης (\$ POST[op] !="view").

Στη γραμμή 26 ρυθμίζουμε τη θέση του κομματιού αυτού του script μέσα στη σελίδα μας, όπως κάναμε και στη γραμμή 16 για τις τρεις παραπομπές και στην γραμμή 24 για τη διεύθυνση του ταχυδρομικού γραφείου. Το κομμάτι αυτό μας δημιουργεί στην οθόνη μας μία φόρμα με προσφορές ταξιδιών τις οποίες τις αντλεί από τη βάση δεδομένων που έχουμε δημιουργήσει με τη MySQL και με την οποία συνδέεται η ιστοσελίδα μας μέσω της εντολής, που είδαμε παραπάνω, της γραμμής 14. Στη συγκεκριμένη βάση μας έχουμε αποθηκεύσει δέκα προσφορές, για δέκα διαφορετικούς προορισμούς με τις τιμές τους, τη διάρκεια του ταξιδιού και τις μέρες αναχώρησης. Αναλυτικότερα, στη γραμμή 27 δίνουμε την εντολή στο script να

τραβήξει από τη βάση μας τη λίστα *id* (κωδικοί ταξιδιών σε προσφορά) και *destinations* (αντίστοιχοι προορισμοί) ταξινομημένη σύμφωνα με τη σειρά τις λίστας *id*, δηλαδή αριθμητικά. Στη συνέχεια κάνουμε ένα ερώτημα (*mysql_query*) και ζητάμε ότι εάν τα αποτελέσματα της λίστας είναι μικρότερα από 1 να μας εμφανίσει το μήνυμα “*Sorry, no records to select!*”, δηλαδή ότι δεν υπάρχουν ταξίδια σε προσφορά αυτήν την περίοδο. Αυτό βέβαια είναι μια απλή δικλίδα προστασίας. Ακολούθως, στη γραμμή 32 μέχρι και τη γραμμή 36 δίνουμε την εντολή ότι εάν δεν ισχύει το παραπάνω, δηλαδή τα αποτελέσματα της λίστας να είναι μικρότερα του 1, τότε να δημιουργήσει μία φόρμα όπως της εικόνας 5.5.



Εικ.5.5 Η φόρμα που δημιουργείται στις γραμμές 32 έως 36.

Δηλαδή, έχουμε ήδη δημιουργήσει από τη γραμμή 26 τον τίτλο: “*Journeys*” και στις γραμμές 34 έως και 36 δημιουργούμε τον υπότιτλο “*Special Offers:*”. Επίσης, δημιουργούμε και τη λίστα επιλογών “*-Destinations-*” και το οποίο θα μας παρουσιάζει τους ταξιδιωτικούς προορισμούς. Στις γραμμές 38 έως και 43 γράφουμε τον κώδικα που αναλαμβάνει να εμφανίζει τα περιεχόμενα της λίστας επιλογών, δηλαδή τους συγκεκριμένους ταξιδιωτικούς προορισμούς ταξινομημένους σύμφωνα με τον κωδικό τους.

Τελειώνοντας, με αυτό το κομμάτι του κώδικα, στις γραμμές 44 έως και 53 δημιουργούμε το κουμπί που βρίσκεται στο τέλος της φόρμας μας και γράφει πάνω του “*View Selected Destination*” και το οποίο καταχωρεί την επιλογή μας (δηλαδή, τον προορισμό που επιλέγουμε από τη λίστα *-Destinations-*) και μας φέρνει τα ανάλογα αποτελέσματα.

Ο τρόπος όμως που εμφανίζονται τα αποτελέσματα αφορά την δεύτερη συνθήκη στην οποία ζητάμε αυτό ακριβώς, δηλαδή να φαίνονται τα αποτελέσματα.

```
55 elseif ($_POST[op] == "view") {
56     if ($_POST[sel_id]==0) {
57         $display_block .= "<table width=100%><tr><td align=center><h1>No Destination Selected</h1>";
58         $display_block .= "<strong>Please select a destination</strong></td></tr></table>"; }
59     else {
60
61         $get_Tours = "select concat_ws(',',destinations) as display_destinations
62         from Tours where id = $_POST[sel_id]";
63         $get_Tours_res = mysql_query($get_Tours);
64         $display_destinations = stripslashes(mysql_result($get_Tours_res,0,'display_destinations'));
65
66         $get_destinations = "select destinations from Tours where id = $_POST[sel_id]";
67         $get_destinations_res = mysql_query($get_destinations);
68
69         if (mysql_num_rows($get_destinations_res) > 0) {
70             $display_block .= "<p><strong>Destination:</strong><br> <ul>";
71             $display_block .= "<h1>$display_destinations</h1>";
72             $display_block .= "</ul>";
73         }
74     }
```

Εικ.5.6 Κομμάτι της συνθήκης ($\$_POST[op] == "view"$).

Στη γραμμή 55 θέτουμε την άλλη πλευρά του θέματος, κατά την οποία τα αποτελέσματα της επιλογής ενός προορισμού από τη φόρμα που δημιουργήσαμε, εμφανίζονται στην οθόνη μας. Οπότε γράφουμε την συνθήκη ($\$_POST[op] == "view"$). Αφού ορίσουμε τη συνθήκη, προσθέτουμε κομμάτια script με το τι θέλουμε να εμφανίζεται στην οθόνη μας. Δηλαδή, εάν ο χρήστης της σελίδας αντί να επιλέξει κάποιο προορισμό από τη λίστα, πατήσει το κουμπί της φόρμας στην επιλογή “-Destinations-”, τότε σύμφωνα με τις γραμμές 56 έως 58 το id του προορισμού που διάλεξε είναι μηδέν και δεν αντιστοιχεί σε προορισμό και θα του εμφανιστεί στην οθόνη του ξανά ο τίτλος και η μαρκίζα που έχουμε δημιουργήσει, όπως επίσης και το φόντο με την μουσική όπως ακριβώς τα έχουμε ρυθμίσει, αλλά όλα τα υπόλοιπα θα εξαφανιστούν και θα εμφανιστεί στη θέση τους στη μέση της οθόνης ένα μήνυμα με έναν μεγάλο τίτλο που θα λέει “No Destination Selected” (δηλαδή, δεν επιλέχθηκε προορισμός), έναν πολύ μικρότερο που θα λέει “Please select a destination” (δηλαδή, παρακαλώ επιλέξτε έναν προορισμό) και τέλος κάτω από αυτούς τους δύο τίτλους θα υπάρχει ένα link που θα λέει “select another” (δηλαδή, επιλέξτε έναν άλλο) και θα επαναφέρει το χρήστη στην αρχική οθόνη για να επιλέξει κάποιον προορισμό. Το αποτέλεσμα αυτής της διαδικασίας φαίνεται στην εικόνα 5.7.

DREAM TRAVEL AGENCY

Live the Days of Your Dreams

No Destination Selected

Please select a destination

[select another](#)

Copyright © 2002-5 by Vagelis Karistinakis & Dimitris Voltrakis All rights reserved

Εικ.5.7 Τι εμφανίζεται όταν δεν έχει επιλεγεί κάποιος προορισμός.

Συνεχίζοντας, στη γραμμή 59 δηλώνουμε ότι εάν δεν ισχύει το παραπάνω, δηλαδή η μη επιλογή προορισμού, θα ισχύουν τα παρακάτω. Συγκεκριμένα, στις γραμμές 61 έως και 72 δίνουμε την εντολή στο script να καλέσει τη λίστα των προορισμών από τη βάση δεδομένων μας η οποία είναι η *Dimna_Travel_Agency* και από τον πίνακά μας τον *Tours* και ταξινομώντας τους αριθμητικά σύμφωνα με το πεδίο *id*. Μόλις ο χρήστης της σελίδας επιλέξει κάποιον προορισμό, ο προορισμός αυτός σύμφωνα με το *id* του θα εμφανιστεί πάνω αριστερά στην οθόνη μας με μεγάλα και bold γράμματα και ακριβώς από πάνω του ένας μικρότερος τίτλος που θα λέει “*Destination :*” όπως μπορούμε να διακρίνουμε και στην εικόνα 5.8.



Εικ.5.8 Εμφάνιση του πεδίου Destination.

Συνεχίζοντας, η συγγραφή του script αρχίζει και γίνεται πιο εύκολη, γιατί εφόσον καταφέραμε να εμφανίσουμε την πρώτη επιλογή (Destination), οι υπόλοιπες

πληροφορίες είναι αντιγραφή του κώδικα, απλά αλλάζουμε τα ονόματα των πεδίων του πίνακα της βάσης δεδομένων μας από όπου παίρνουμε τις πληροφορίες.

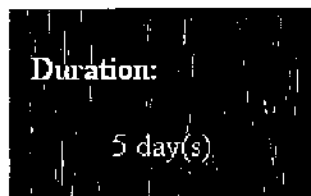
```
75 $get_Tours = "select concat_ws(' ',departure) as display_departure from Tours where id = $_POST[sel_id]";
76 $get_Tours_res = mysql_query($get_Tours);
77 $display_departure = stripslashes(mysql_result($get_Tours_res,0,'display_departure'));
78
79 $get_departure = "select departure from Tours where id = $_POST[sel_id]";
80 $get_departure_res = mysql_query($get_departure);
81 if (mysql_num_rows($get_Tours_res) > 0) {
82     $display_block .= "<P><strong>Departure Date:</strong><br> <ul>";
83     $display_block .= "$display_departure";
84     $display_block .= "</ul>";
85 }
86
87
88 $get_Tours = "select concat_ws(' ',duration) as display_duration from Tours where id = $_POST[sel_id]";
89 $get_Tours_res = mysql_query($get_Tours);
90 $display_duration = stripslashes(mysql_result($get_Tours_res,0,'display_duration'));
91
92 $get_duration = "select duration from Tours where id = $_POST[sel_id]";
93 $get_duration_res = mysql_query($get_duration);
94
95
96 if (mysql_num_rows($get_Tours_res) > 0) {
97     $display_block .= "<P><strong>Duration:</strong><br> <ul>";
98     $display_block .= "$display_duration";
99     $display_block .= " day(s)</ul>";
100 }
101
```

Εικ.5.9 Εμφάνιση ημερομηνίας αναχώρησης και διάρκειας του ταξιδιού.

Στην εικόνα 3.9 μπορούμε να δούμε πόσο όμοιος είναι ο κώδικας για την εμφάνιση των στοιχείων *Αναχώρηση* (departure) και *Διάρκεια* (duration) με τον κώδικα για τους προορισμούς. Με τη δημιουργία του κώδικα αυτού, κάτω από τον προορισμό στα αποτελέσματα της επιλογής του χρήστη θα εμφανιστούν η ημερομηνία αναχώρησης και η διάρκεια του ταξιδιού.

Στις σειρές 75 έως και 81 καλούμε από τη βάση μας να μας εμφανίσει την ημερομηνία αναχώρησης που διάλεξε ο χρήστης. Δηλαδή, επειδή όλα στη βάση μας κατανέμονται σύμφωνα με το πεδίο id θα μας εμφανίσει την ημερομηνία της οποίας το id ταιριάζει με το id του προορισμού που επέλεξε ο χρήστης της σελίδας. Ο τρόπος εμφάνισης ορίζεται στις γραμμές 83 έως 85 όπου δημιουργούμε έναν τίτλο που λέει "*Departure Date:*" και από κάτω εμφανίζεται η ημερομηνία αναχώρησης. Διαβάζοντας στη συνέχεια το script, μπορούμε να καταλάβουμε ότι την ίδια διαδικασία ακολουθούμε για να τραβήξουμε τα στοιχεία της διάρκειας του ταξιδιού από τον πίνακα της βάσης μας. Η μόνη διαφορά είναι στην εμφάνιση, γιατί στην ημερομηνία αναχώρησης και στους προορισμούς απλά αφήναμε τις πληροφορίες από τον πίνακα της βάσης μας να μπαίνουν αυτούσιες στην σελίδα μας. Τώρα την τελευταία γραμμή για τη διάρκεια, δηλαδή τη γραμμή 100, δεν την αφήνουμε κενή αλλά αφήνουμε ένα κενό και μετά γράφουμε τη λέξη *day(s)*. Με την πρόσθεση αυτής

της λέξης, καταφέραμε να κάνουμε απλά την σελίδα μας λίγο πιο ευπαρουσίαστη, χωρίς αυτό να σημαίνει πως αν δεν τη γράφαμε θα επηρεαζόταν το script.



Εικ.5.10 Εμφάνιση της διάρκειας.

Όπως μπορούμε να δούμε και στην εικόνα 10, εάν για παράδειγμα η διάρκεια του ταξιδιού που επέλεξε ο χρήστης της σελίδας είναι πέντε μέρες, τότε στην οθόνη μας κάτω από τον προορισμό και την ημερομηνία αναχώρησης θα εμφανιστεί η πληροφορία της εικόνας 5.10 με τη μορφή που απεικονίζεται.

Αφού έχουμε εμφανίσει τους προορισμούς πάνω αριστερά στην οθόνη μας, μετά από κάτω εμφανίσαμε την ημερομηνία αναχώρησης και τη διάρκεια του ταξιδιού, τώρα αυτό που απομένει είναι να εμφανιστεί κάτω από όλα αυτά η τιμή της προσφοράς. Η τιμή θα εμφανιστεί και αυτή με τον ίδιο τρόπο που εμφανίστηκαν και οι προηγούμενες πληροφορίες στην οθόνη μας. Η εμφάνιση της τιμής όπως θα δούμε και στην εικόνα 5.11, θέλει και αυτή το σήμα του € να εμφανίζεται δίπλα από την τιμή. Με τον ίδιο τρόπο που προσθέσαμε τη λέξη day(s) στη διάρκεια του ταξιδιού, θα προσθέσουμε και το σήμα του Ευρώ μετά την τιμή. Δηλαδή, αρκεί στη γραμμή 112 να αφήσουμε ένα κενό και να βάλουμε το σήμα του Ευρώ, όπως φαίνεται δηλαδή και στην εικόνα 5.11.

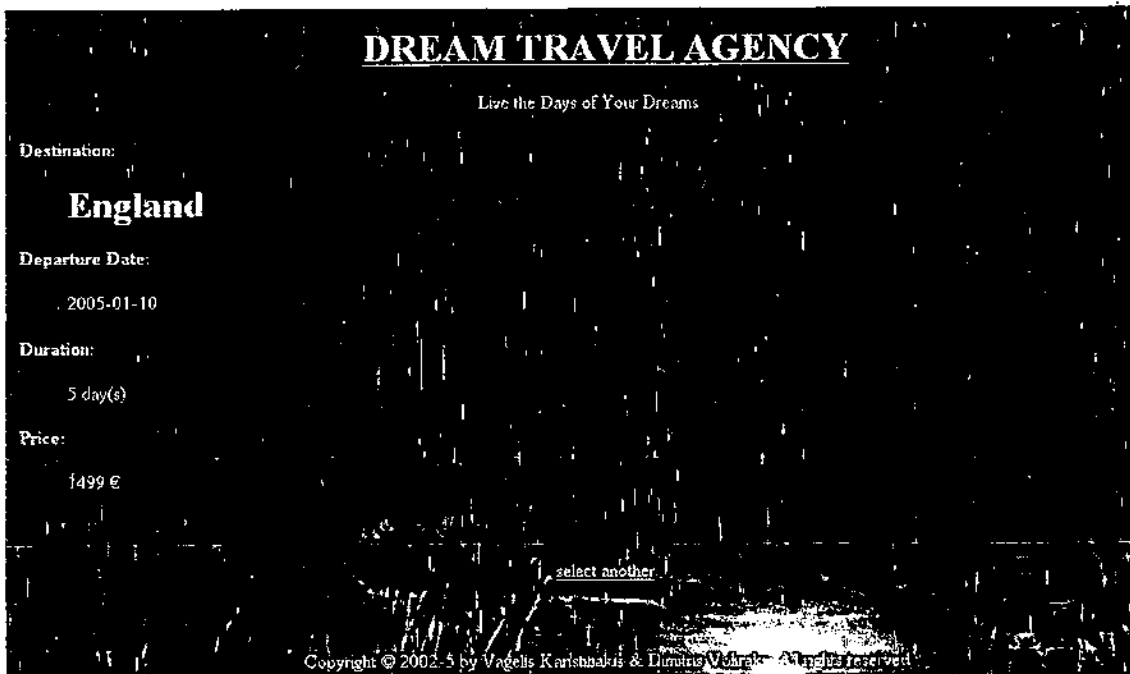
```

102 $get_Tours = "select concat_ws(' ',price) as display_price from Tours where id = $_POST[sel_id]";
103 $get_Tours_res = mysql_query($get_Tours);
104 $display_price = stripslashes(mysql_result($get_Tours_res,0,'display_price'));
105
106 $get_price = "select price from Tours where id = $_POST[sel_id]";
107 $get_price_res = mysql_query($get_price);
108 if (mysql_num_rows($get_price_res) > 0 ) {
109
110     $display_block .= "<P><strong>Price:</strong><br> <ul>";
111     $display_block .= "$display_price";
112     $display_block .= " €</ul>";
113 }
114
115 $display_block .= "<br><br><P align=center>
116 <a href='\"$_SERVER[PHP_SELF]\"'>select another</a></p>";
117
118 $display_block .= "<br><br><P align=center valign=bottom>Copyright © 2002-5 by Vagelis Karistinakis & Dimitris Volirakis
119 All rights reserved" ;
120 ?>
121
122 <? print $display_block;?>
123 </body>
124 </html>

```

Εικ.5.11 Η εμφάνιση της τιμής και το τέλος του κώδικα.

Αφού εμφανίσαμε όλα τα δεδομένα που επιθυμούσαμε στην οθόνη μας, στις γραμμές 115 και 116 δημιουργήσαμε στο κάτω μέρος της σελίδας μας στο κέντρο μία παραπομπή η οποία λέει *select another* (δηλαδή, επιλέξτε άλλο) και εάν την επιλέξει ο χρήστης μεταφέρεται στην αρχική εικόνα της σελίδας μας για να επιλέξει άλλον προορισμό. Στην οθόνη μας λοιπόν, εάν κάποιος χρήστης επιλέξει την Αγγλία σαν προορισμού του ταξιδιού του θα εμφανιστεί η εικόνα 5.12.



Εικ.5.12 Εμφάνιση αποτελέσματος στην οθόνη από την επιλογή κάποιου προορισμού.

Πριν κλείσουμε την PHP προσθέσαμε και μία φράση που λέει ότι εμείς δημιουργήσαμε τη σελίδα αυτή με όλα τα νόμιμα δικαιώματα (copyright). Τη φράση αυτή την προσθέσαμε γιατί είναι μια φράση που εμφανίζεται σχεδόν σε όλες τις σελίδες στον παγκόσμιο ιστό, και εμφανίζεται σε όλες μορφές που παίρνει η σελίδα μας στο κάτω μέρος της.

Αυτό που απέμεινε να εξηγήσουμε είναι ότι κάθε τι που επιθυμούσαμε να εμφανίσουμε στην οθόνη μας το εμφανίζαμε μέσω της μεταβλητής *display_block* και για αυτό το λόγο στη γραμμή 122 δίνουμε την εντολή να τυπωθεί η μεταβλητή αυτή στην οθόνη μας. Τέλος, κλείνουμε το σώμα του script με την εντολή `</body>` και ολόκληρο το script με την εντολή `</html>`.

Ο κώδικάς μας είναι τώρα έτοιμος και ολοκληρωμένος και το μόνο που χρειάζεται είναι να τον αποθηκεύσουμε στον κατάλογο των εγγράφων του web server μας και να τον προσπελάσουμε μέσω του web browser του υπολογιστή μας.

6. ΣΥΜΠΕΡΑΣΜΑΤΑ

Έχοντας πλέον ολοκληρώσει την πτυχιακή εργασία που μας ανατέθηκε θα ήταν σωστό να ανακεφαλαιώσουμε και να δούμε τι ακριβώς έχουμε φτιάξει. Ξεκινήσαμε με την εκμάθηση της HTML και ως φυσικό επόμενο τη δημιουργία ιστοσελίδων με χρήση HTML. Δημιουργήσαμε την ιστοσελίδα ενός φανταστικού ταξιδιωτικού γραφείου με το όνομα “DREAM TRAVEL AGENCY” σύμφωνα με τις οδηγίες του υπεύθυνου καθηγητή Κ^ου Αθανασόπουλου Δημήτρη.

Στη συνέχεια ασχοληθήκαμε με την εκμάθηση της MySQL και συνεχίσαμε με τη δημιουργία μίας βάσης δεδομένων MySQL. Αφού η σελίδα μας αφορούσε ταξιδιωτικό γραφείο θεωρήσαμε σωστό να δημιουργήσουμε μία βάση η οποία θα αποτελούταν από προσφορές για ταξίδια στο εξωτερικό σε διαφορετικούς προορισμούς.

Μετά από την ολοκλήρωση και αυτού του μέρους της εργασίας, συνεχίσαμε με τη γλώσσα PHP και με την υλοποίηση του πιο δύσκολου κομματιού της πτυχιακής, την δημιουργία δηλαδή σελίδας PHP, η οποία θα ενσωματώνεται στην ήδη υπάρχουσα HTML σελίδα και θα τη συνδέει με τη βάση δεδομένων μας, από όπου και θα αντλεί πληροφορίες, ανάλογα με τις επιθυμίες του χρήστη. Σύμφωνα λοιπόν με τα δεδομένα αυτά, αποφασίσαμε να δημιουργήσουμε μία φόρμα η οποία θα εμφανίζει στο χρήστη της σελίδας τους προορισμούς της βάσης δεδομένων μας και αφού επιλέξει τον προορισμό της αρεσκείας του θα μεταφέρεται κατά κάποιον τρόπο σε μία άλλη σελίδα η οποία θα του εμφανίζει την προσφορά που υπάρχει για τον προορισμό που επέλεξε, την οποία θα την τραβάει η σελίδα μας από την βάση δεδομένων μας.

Έχοντας λοιπόν ολοκληρώσει όλα τα παραπάνω θεωρούμε ότι έχουμε λάβει κάποιες βασικές γνώσεις όσον αφορά τη δημιουργία μίας δυναμικής σελίδας στον παγκόσμιο ιστό και θεωρούμε ότι αυτή η γνώση μας διεύρυνε τους ορίζοντές μας αφού γνωρίσαμε και ένα περιβάλλον εντελώς διαφορετικό από αυτό το οποίο σπουδάσαμε και ασχοληθήκαμε με αυτό. Για την εργασία αυτή αφιερώσαμε αρκετό χρόνο αφού χρειάστηκε να μάθουμε να εγκαθιστούμε έναν web server (apache) καθώς και τον χρόνο που χρειάστηκε για την εκμάθηση τριών γλωσσών προγραμματισμού. Ξεκινήσαμε μαθαίνοντας πώς να δημιουργούμε HTML κώδικα και δημιουργήσαμε μια απλή σελίδα, συνεχίσαμε με την εκμάθηση της γλώσσας για

την δημιουργία της βάσης MySQL και τέλος μάθαμε πώς να γράφουμε PHP. Ήταν μια πρωτόγνωρη εμπειρία για μας η επιλογή της εργασίας αυτής αφού έχει πολύ μικρή σχέση με το αντικείμενο που σπουδάσαμε, αλλά το ενδιαφέρον μας για τον τομέα των υπολογιστών μας βοήθησε να την φέρουμε εις πέρας. Η βοήθεια του κυρίου Αθανασόπουλου ήταν πολύτιμη καθώς είναι γνώστης του αντικειμένου και μας βοήθησε στην κατανόηση εξειδικευμένων γνώσεων. Η εμπειρία αυτή ήταν επίσης πολύτιμη αφού μας έμαθε να μην υποτιμούμε τις δυνάμεις μας, καθώς και να μην τις υπερεκτιμούμε όταν ερχόμαστε για πρώτη φορά σε επαφή με έναν νέο χώρο.

7. ΥΠΟΜΝΗΜΑ

• Ο ΚΩΔΙΚΑΣ

```
1 <html>
2 <body background="c:\ptyxiakh\cancer.jpg" text=white link=yellow vlink=white>
3 <font size=9><h1 align=center><b>UDREAM TRAVEL AGENCY</b></h1></font>
4 <p>
5 <marquee direction="right" behavior="alternate">Dream Travel Agency</marquee></p>
6
7 <bgound src="c:\ptyxiakh\Eurithnicks-Sweet Dreams.mp3" loop="5">
8
9 <?php
10 $conn = mysql_connect("localhost", "vag", "dim") or die (mysql_error());
11 mysql_select_db("dimva_travel_agency", $conn) or die (mysql_error());
12 if ($_POST[op] != "view") {
13     $display_block .= "<table width=100%><tr><td width=90%>";
14     $display_block .= "<p><a href='\"http://www.hertz.com\"' align=left><img src='\"CAUFOXID.jpg\"'></a>
15 Rent a car";
16     $display_block .= "<p><a href='\"http://www.bookhotel.com\"' align=left><img src='\"Hotel.jpg\"'></a>
17 Hotels";
18     $display_block .= "<p><a href='\"mailto:dream@yahoo.gr\"' align=left><img src='\"mail.gif\"'></a>
19 Contact Us</td>";
20
21     $display_block .= "<br><br><p align=center>Mesogeion 450<br>Goudi, Athens<br>T.K.:128 76<br>TEL.:2105684239";
22     $display_block .= "<td width 20%></td><td width=30% align=center valign=top><h1>Journeys</h1>";
23
24     $get_list = "select id, concat_ws(',', destinations) as display_destinations from Tours order by id";
25     $get_list_res = mysql_query($get_list) or die(mysql_error());
26     if (mysql_num_rows($get_list_res) < 1) {
27         $display_block .= "<p><em>Sorry, no records to select!</em></p>";
28     }
29     else {
30         $display_block .= " <form method='post' action='\"$_SERVER{PHP_SELF}\">
31 <p><strong>Special Offers:</strong><br>
32 <select name='\"sel_id\"'>
33 <option value='\"\"'>-Destinations-</option>";
34
35         while ($recs = mysql_fetch_array($get_list_res)) {
36             $id = $recs['id'];
37             $display_destinations = stripslashes($recs['display_destinations']);
38             $display_block .= "<option value='\"$id\"'>$display_destinations</option value>";
39         }
40
41         $display_block .= "
42 </select>
43 <input type='\"hidden\"' name='\"op\"' value='\"view\"'>
44 <p><input type='\"submit\"' name='\"submit\"'
45 value='\"View Selected Destination\"'></p>
46 </FORM></td></tr></table>";
47     }
48 }
49
50 elseif ($_POST[op] == "view") {
51     if ($_POST[sel_id]==0) {
52         $display_block .= "<table width=100%><tr><td align=center><h1>No Destination Selected</h1>";
53         $display_block .= "<strong>Please select a destination</strong></td></tr></table>"; }
54     else {
55
56         $get_Tours = "select concat_ws(',', destinations) as display_destinations
57 from Tours where id = $_POST[sel_id]";
58 $get_Tours_res = mysql_query($get_Tours);
59 $display_destinations = stripslashes(mysql_result($get_Tours_res,0,'display_destinations'));
60
61 $get_destinations = "select destinations from Tours where id = $_POST[sel_id]";
62 $get_destinations_res = mysql_query($get_destinations);
63
64 if (mysql_num_rows($get_destinations_res) > 0) {
65     $display_block .= "<p><strong>Destination:</strong><br> <u>";
66     $display_block .= "<h1>$display_destinations</h1>";
```

```

67     $display_block .= "</ul>";
68 }
69
70 $get_Tours = "select concat_ws(' ',departure) as display_departure from Tours where id = $_POST[sel_id]";
71 $get_Tours_res = mysql_query($get_Tours);
72 $display_departure = stripslashes(mysql_result($get_Tours_res,0,'display_departure'));
73
74 $get_departure = "select departure from Tours where id = $_POST[sel_id]";
75 $get_departure_res = mysql_query($get_departure);
76 if (mysql_num_rows($get_Tours_res) > 0 ) {
77     $display_block .= "<P><strong>Departure Date:</strong><br> <ul>";
78     $display_block .= "$display_departure";
79     $display_block .= "</ul>";
80 }
81
82
83 $get_Tours = "select concat_ws(' ',duration) as display_duration from Tours where id = $_POST[sel_id]";
84 $get_Tours_res = mysql_query($get_Tours);
85 $display_duration = stripslashes(mysql_result($get_Tours_res,0,'display_duration'));
86
87 $get_duration = "select duration from Tours where id = $_POST[sel_id]";
88 $get_duration_res = mysql_query($get_duration);
89
90
91 if (mysql_num_rows($get_Tours_res) > 0 ) {
92     $display_block .= "<P><strong>Duration:</strong><br> <ul>";
93     $display_block .= "$display_duration";
94     $display_block .= " day(s)</ul>";
95 }
96
97 $get_Tours = "select concat_ws(' ',price) as display_price from Tours where id = $_POST[sel_id]";
98 $get_Tours_res = mysql_query($get_Tours);
99 $display_price = stripslashes(mysql_result($get_Tours_res,0,'display_price'));
100
101 $get_price = "select price from Tours where id = $_POST[sel_id]";
102 $get_price_res = mysql_query($get_price);
103 if (mysql_num_rows($get_Tours_res) > 0 ) {
104     $display_block .= "<P><strong>Price:</strong><br> <ul>";
105     $display_block .= "$display_price";
106     $display_block .= "</ul>";
107 }
108 }
109 }
110 $display_block .= "<br><br><P align=center>";
111 <a href="\$_SERVER[PHP_SELF]">select another</a></p>";
112 }
113 $display_block .= "<br><br><P align=center>Copyright © 2002-5 by Vagelis Karistinakis & Dimitris Voli
114 >>";
115
116 <? print $display_block;?>
117 </body>
118 </html>

```

• Η ΒΑΣΗ

```

C:\mysql\bin\mysql.exe
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6 to server version: 4.0.4-beta-max-nt

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> create database Dimva_Travel_Agency;
Query OK, 1 row affected (0.00 sec)

mysql> use Dimva_Travel_Agency;
Database changed
mysql> create table Tours (
  -> id int not null primary key auto_increment,
  -> destinations varchar(50),
  -> departure date,
  -> duration int not null,
  -> price int not null
  -> );
Query OK, 0 rows affected (0.02 sec)

mysql> show fields from Tours;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id         | int(11)       |      | PRI | NULL     | auto_increment |
| destinations | varchar(50)   | YES  |     | NULL     |                |
| departure  | date          | YES  |     | NULL     |                |
| duration   | int(11)       |      |     | 0        |                |
| price     | int(11)       |      |     | 0        |                |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>

```

```

mysql> insert into Tours (id, destinations, departure, duration, price) values (<
'1', 'England', '2005-01-10', '5', '1500');
Query OK, 1 row affected (0.00 sec)

mysql> insert into Tours (id, destinations, departure, duration, price) values (<
'2', 'France', '2005-01-10', '7', '1700');
Query OK, 1 row affected (0.01 sec)

mysql> insert into Tours (id, destinations, departure, duration, price) values (<
'3', 'Italy', '2005-01-10', '3', '700');
Query OK, 1 row affected (0.00 sec)

mysql> insert into Tours (id, destinations, departure, duration, price) values (<
'4', 'Cyprus', '2005-01-10', '10', '1300');
Query OK, 1 row affected (0.00 sec)

mysql> insert into Tours (id, destinations, departure, duration, price) values (<
'5', 'U.S.A', '2005-01-10', '7', '2200');
Query OK, 1 row affected (0.00 sec)

mysql> insert into Tours (id, destinations, departure, duration, price) values (<
'6', 'Egypt', '2005-01-10', '5', '1000');
Query OK, 1 row affected (0.00 sec)

mysql> insert into Tours (id, destinations, departure, duration, price) values (<
'7', 'India', '2005-01-10', '8', '2000');
Query OK, 1 row affected (0.00 sec)

mysql> insert into Tours (id, destinations, departure, duration, price) values (<
'8', 'Australia', '2005-01-10', '10', '3000');
Query OK, 1 row affected (0.00 sec)

mysql> insert into Tours (id, destinations, departure, duration, price) values (<
'9', 'Netherlands', '2005-01-10', '7', '1500');
Query OK, 1 row affected (0.00 sec)

mysql> insert into Tours (id, destinations, departure, duration, price) values (<
'10', 'Germany', '2005-01-10', '9', '1500');
Query OK, 1 row affected (0.00 sec)

```



```
mysql> select id, destinations, departure, duration, price from Tours;
```

id	destinations	departure	duration	price
1	England	2005-01-10	5	1500
2	France	2005-01-10	7	1700
3	Italy	2005-01-10	3	700
4	Cyprus	2005-01-10	10	1300
5	U.S.A	2005-01-10	7	2200
6	Egypt	2005-01-10	5	1000
7	India	2005-01-10	8	2000
8	Australia	2005-01-10	10	3000
9	Netherlands	2005-01-10	7	1500
10	Germany	2005-01-10	9	1500


```
10 rows in set (0.00 sec)
```


```
mysql> _
```


◦ Η ΣΕΛΙΔΑ

DREAM TRAVEL AGENCY

Live the Days of Your Dreams

 Rent a car

 Hotels

 Contact Us

Journeys

Special Offers:
Destinations-▼

View Selected Destination

Messogeion 450
Gouli Athens
TEL 12376
TEL 2105684239

Copyright © 2002-5 by Vagelis Katsinakis & Dimitris Volrakis All rights reserved

DREAM TRAVEL AGENCY

Live the Days of Your Dreams

No Destination Selected

Please select a destination

select another

Copyright © 2002-5 by Vagelis Katsinakis & Dimitris Volrakis All rights reserved

DREAM TRAVEL AGENCY

Like the Days of Your Dreams

Destination:

England

Departure Date:

2005-01-10

Duration:

5 day(s)

Price:

1489 €

[select another](#)

Copyright © 2002-5 by Vagelis Karisthakis & Diminis Volagakis. All rights reserved

8. ΒΙΒΛΙΟΓΡΑΦΙΑ

1. **PHP5 and MySQL Bible**,
Tim Converse and Joyce Park, Wiley Publishing, ISBN: 0-7645-5746-7
2. **Teach Yourself PHP, MySQL and Apache All in One**
Julie Meloni, Sams Publishing, ISBN: 0-672-32725-2
3. **PHP and MySQL Web Development**,
Luke Welling and Laura Thomson, Sams Publishing
4. **PHP/MySQL Programming for the Absolute Beginner**
Andy Harris, Premier Press, ISBN: 1931841322
5. **HTML The Definitive Guide**,
Chuck Musciano and Bill Kennedy, O'Reilly and Associates,
ISBN: 1-56592-492-4
6. **Teach Yourself PHP, MySQL and Apache in 24 Hours**,
Julie C. Meloni, Sams Publishing , ISBN: 0-6723-2489-X
7. **MySQL/PHP Database Applications**,
Jay Greenspan and Brad Bulger, M&T Books, ISBN: 0-7645-3537-4

