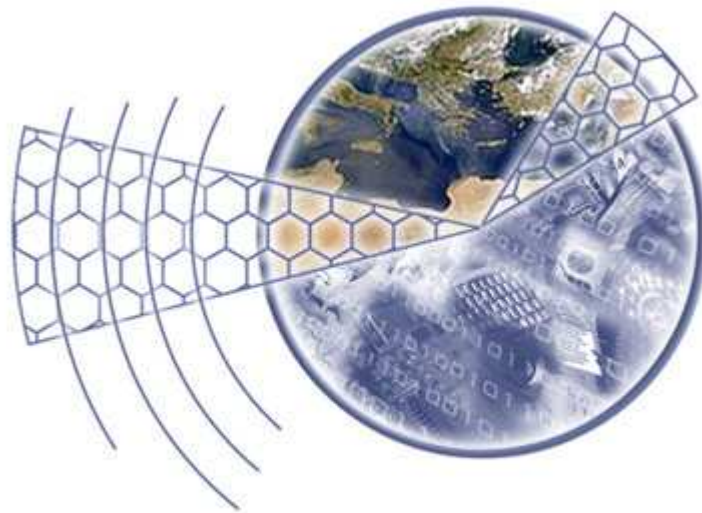




**ΤΕΧΝΟΛΟΓΙΚΟ
ΕΚΠΑΙΔΕΥΤΙΚΟ
ΙΔΡΥΜΑ ΚΡΗΤΗΣ**

**ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ
ΠΟΛΥΜΕΣΩΝ**



**ΔΗΜΙΟΥΡΓΙΑ ΕΝΟΣ ΔΥΝΑΜΙΚΟΥ ΤΟΥΡΙΣΤΙΚΟΥ
ΔΙΚΤΥΑΚΟΥ ΤΟΠΟΥ (TRAVEL PORTAL)
ΧΡΗΣΙΜΟΠΟΙΩΝΤΑΣ ΕΝΑ ΑΝΟΙΧΤΟΥ ΚΩΔΙΚΑ
ΣΥΣΤΗΜΑ ΔΙΑΧΕΙΡΙΣΗΣ ΠΕΡΙΕΧΟΜΕΝΟΥ CMS**

**ΓΙΑΜΟΥΡΗΣ ΓΕΩΡΓΙΟΣ (ΑΜ:557)
ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: ΠΑΠΑΔΟΥΡΑΚΗΣ ΓΕΩΡΓΙΟΣ**

ΗΡΑΚΛΕΙΟ 2010

ΠΕΡΙΕΧΟΜΕΝΑ

Περίληψη.....	3
ΚΕΦΑΛΑΙΟ 1	
1.1 Συστήματα Διαχείρισης Περιεχομένου.....	3
1.2 Joomla, Ένα πλήρες σύστημα διαχείρισης περιεχομένου.....	4
1.3 Το xampp και τα χαρακτηριστικά του.....	6
1.4 Εγκατάσταση του Joomla.....	7
1.5 Ο Δικτυακός περιηγητής Mozilla Firefox 3.6.....	9
1.6 Adobe Photoshop cs3.....	10
ΚΕΦΑΛΑΙΟ 2	
2.1 Η Γλώσσα Προγραμματισμού PHP.....	11
2.2 Τι Μπορεί να Κάνει η PHP.....	12
2.3 Τερματισμός Εντολών.....	15
2.4 Οι Τύποι Δεδομένων της PHP.....	15
2.5 Πίνακες Μίας Διάστασης (Single Dimension Arrays).....	18
2.6 Τα Αντικείμενα (Objects).....	21
2.7 Οι Μεταβλητές (Variables).....	22
2.8 Οι Σταθερές (Constants).....	32
2.9 Οι Εκφράσεις (Expressions).....	34
2.10 Οι Αριθμητικοί Τελεστές.....	37
2.11 Οι Δομές Ελέγχου (Control Structures).....	40
2.12 Συναρτήσεις Οριζόμενες από τον Χρήστη.....	51
2.13 Οι Μεταβλητές Συναρτήσεων.....	53
2.14 Τάξεις και Αντικείμενα.....	54
2.15 Τα Cookies.....	59
ΚΕΦΑΛΑΙΟ 3	
3.1 Ο Κώδικας HTML, ΕΙΣΑΓΩΓΗ.....	63
3.2 Βασικές Γνώσεις HTML.....	64
3.3 Παράδειγμα Δημιουργίας Ιστοσελίδας.....	65
3.4 Βασικές Εντολές.....	66
3.5 Υπερσυνδέσεις – Links.....	67
3.6 Η σελίδα υποδομής.....	73
3.7 Οι Πληροφορίες της Ιστοσελίδας μας.....	75
ΚΕΦΑΛΑΙΟ 4	
4.1 Το Joomla 1.5.8.....	77
4.2 Βασικά εργαλεία Joomla.....	77
4.3 Joomla και δυνατότητες.....	83
ΚΕΦΑΛΑΙΟ 5	
5.1 Το plug-in hotelagent.....	93
5.2 Διαχείριση τόπου διαμονής.....	93
ΚΕΦΑΛΑΙΟ 6	
6.1 Οι frontend χρήστες και η ιστοσελίδα μας.....	100
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	104

ΠΕΡΙΛΗΨΗ

Η παρούσα εργασία πραγματοποιήθηκε στα πλαίσια της πτυχιακής άσκησης του προπτυχιακού φοιτητή Γιαμούρη Γεώργιου, φοιτητή του Τεχνολογικού Εκπαιδευτικού Ιδρύματος Ηρακλείου Κρήτης στο τμήμα Εφαρμοσμένης Πληροφορικής και Πολυμέσων. Επιβλέπων καθηγητής ήταν ο κ. Παπαδουράκης Γεώργιος.

Αντικείμενο της εργασίας αυτής είναι η δημιουργία ενός δυναμικού τουριστικού δικτυακού τόπου (travel portal) για την νήσο Νάξο, χρησιμοποιώντας ένα ανοιχτού κώδικα σύστημα διαχείρισης περιεχομένου (cms) το Joomla.

Στο πρώτο κεφάλαιο περιγράφονται το Joomla καθώς και τα υπόλοιπα προγράμματα που χρησιμοποιήθηκαν. Στο δεύτερο κεφάλαιο βλέπουμε τις λειτουργίες που μπορεί να πραγματοποιήσει backend χρήστης όπως εισαγωγή δεδομένων, εισαγωγή πολυμέσων, επεξεργασία κειμένου, και άλλες βασικές λειτουργίες του Joomla. Στο τρίτο κεφάλαιο παρουσιάζουμε το HotelAgent το κύριο plug-in που μας βοήθησε να προσθέσουμε και να διαχειριστούμε τόπους διαμονής και έτσι υλοποιήσουμε την ιστοσελίδα μας. Τέλος στο τέταρτο κεφάλαιο ένας frontend χρήστης μπορεί να μάθει περισσότερα για την περιήγηση στην ιστοσελίδα, όπως η αναζήτηση διαμονής και η ενημέρωση για τη Νάξο.

ΚΕΦΑΛΑΙΟ 1

1.1 ΣΥΣΤΗΜΑΤΑ ΔΙΑΧΕΙΡΙΣΗΣ ΠΕΡΙΕΧΟΜΕΝΟΥ

Ο όρος Content Management Systems (CMS, Συστήματα Διαχείρισης Περιεχομένου) αναφέρεται στις εφαρμογές που επιτρέπουν στον πελάτη να διαχειρίζεται το δικτυακό του περιεχόμενο, όπως κείμενα, εικόνες, πίνακες κ.λπ., με εύκολο τρόπο, συνήθως παρόμοιο με αυτόν της χρήσης ενός κειμενογράφου. Οι εφαρμογές διαχείρισης περιεχομένου επιτρέπουν την αλλαγή του περιεχομένου χωρίς να είναι απαραίτητες ειδικές γνώσεις σχετικές με τη δημιουργία ιστοσελίδων ή γραφικών, καθώς συνήθως τα κείμενα γράφονται μέσω κάποιων online WYSIWYG ("What You See Is What You Get") html editors, ειδικών δηλαδή κειμενογράφων, παρόμοιων με το MS Word, που επιτρέπουν τη μορφοποίηση των κειμένων όποτε υπάρχει ανάγκη.

Οι αλλαγές του site μπορούν να γίνουν από οποιονδήποτε υπολογιστή που είναι συνδεδεμένος στο Διαδίκτυο, χωρίς να χρειάζεται να έχει εγκατεστημένα ειδικά προγράμματα επεξεργασίας ιστοσελίδων, γραφικών κ.λπ. Μέσω ενός απλού φυλλομετρητή ιστοσελίδων (browser),

ο χρήστης μπορεί να συντάξει ένα κείμενο και να ενημερώσει άμεσα το δικτυακό του τόπο.

1.2 JOOMLA - ΕΝΑ ΠΛΗΡΕΣ ΣΥΣΤΗΜΑ ΔΙΑΧΕΙΡΙΣΗΣ ΠΕΡΙΕΧΟΜΕΝΟΥ

Γεμάτο δυνατότητες αλλά και ταυτόχρονα εξαιρετικά ευέλικτο και φιλικό, το Joomla! είναι η εφαρμογή που χρειάζεστε για να δημοσιεύσετε στο διαδίκτυο μια προσωπική ιστοσελίδα, αλλά και έναν ολόκληρο εταιρικό δικτυακό τόπο. Είναι προσαρμόσιμο σε περιβάλλοντα επιχειρηματικής κλίμακας όπως τα intranets μεγάλων επιχειρήσεων ή οργανισμών. Οι δυνατότητες επέκτασής του είναι πρακτικά απεριόριστες.

Το Joomla! είναι εφαρμογή ανοικτού κώδικα. Τι σημαίνει αυτό;

Η χρήση του Joomla! είναι απολύτως δωρεάν. Είστε ελεύθεροι να το χρησιμοποιήσετε, να το τροποποιήσετε και να διερευνήσετε τις δυνατότητές του χωρίς να πρέπει να πληρώσετε κάποια άδεια χρήσης σε κανέναν.

Πώς λειτουργεί;

Το Joomla εγκαθίσταται σε έναν κεντρικό υπολογιστή, τον web server. Ο χρήστης, εσείς δηλαδή, έχετε πρόσβαση στο περιβάλλον διαχείρισης μέσω ενός browser, όπως είναι ο Internet Explorer ή ο Firefox.

Από τη στιγμή που είμαστε ο διαχειριστής, μπορούμε να προσθέσουμε οποιοδήποτε κείμενο ή γραφικό, και έτσι να δημιουργήσουμε τις ιστοσελίδες μας.

Τι μπορώ να κάνω με Joomla;

Αυτό εξαρτάται από εσάς. Οι δυνατότητες χρήσης του Joomla! περιορίζονται μόνον από τη δημιουργικότητά σας. Κάποιοι χρησιμοποιούν το Joomla! για διασκέδαση, για τη κατασκευή μιας προσωπικής ή οικογενειακής ιστοσελίδας. Ενσωματώνοντας διάφορα, διαθέσιμα δωρεάν, πρόσθετα εργαλεία / επεκτάσεις, διευρύνονται οι δυνατότητες και λειτουργίες του Joomla!, κάνοντάς το μια εφαρμογή αξιόπιστη για την ανάπτυξη σοβαρών εταιρικών δικτυακών τόπων.

Πώς θα μπορούσα να διαφοροποιήσω την εικόνα του Joomla ;

Το Joomla! χρησιμοποιεί μια ισχυρή templating engine που σας δίνει τη δυνατότητα να χρησιμοποιήσετε το δικό σας, εξατομικευμένο, template. Για το σκοπό αυτό μπορείτε είτε να δημιουργήσετε δικό σας template, είτε να κατεβάσετε από το διαδίκτυο ένα από τα εκατοντάδες (περίπου 500) που διατίθενται δωρεάν, είτε να αγοράσετε κάποιο από αυτά που πωλούνται, είτε να παραγγείλετε την κατασκευή ενός σε έναν επαγγελματία. Από τη στιγμή που θα δημιουργηθεί το δικό σας template, δεν χρειάζονται παρά τρία κλικ, ώστε το περιεχόμενο του δικτυακού σας τόπου να προσαρμοστεί στη μορφή του.

Ποιες είναι οι δυνατότητες του Joomla που ξεχωρίζουν;

Από πού να αρχίζουμε σε αυτήν την απάντηση; Το Joomla! έχει τόσα πολλά χαρακτηριστικά γνωρίσματα που είναι δύσκολο να αναφερθούμε σε όλα. Επιγραμματικά, αναφέρουμε τις πιο δημοφιλείς δυνατότητες: Μπορούμε να προσθέσουμε περιεχόμενο στον ιστότοπό μας από

οποιοδήποτε υπολογιστή διαθέτει σύνδεση στο διαδίκτυο. Πληκτρολογήστε το κείμενο, ανεβάστε τις φωτογραφίες, δημοσιεύστε!. Χρησιμοποιήστε το Joomla! για να συνεργαστείτε με όσους μπορούν να συμβάλλουν στην επεξεργασία του περιεχομένου. Δώστε τη δυνατότητα σε συνεργάτες σας να δημοσιεύσουν περιεχόμενο, στο πλαίσιο ασφαλώς των αρμοδιοτήτων τους. Είναι εύκολο. Μπορούν να το κάνουν από κάθε υπολογιστή με σύνδεση στο διαδίκτυο. Επίσης μπορούμε να δημοσιεύσουμε απεριόριστες σελίδες, χωρίς να περιοριστούμε από το Joomla! Χωρίς κανένα πρόβλημα, μπορούμε να κάνουμε αναζητήσεις περιεχομένου και να τις αρχειοθετήσουμε. Η εφαρμογή υποστηρίζει τα διαφημιστικά banners: προωθήστε δικά σας προϊόντα και υπηρεσίες, ή χρησιμοποιήστε τα ως διαφημιστικό μέσο για τρίτους. Forum, photo galleries, βιβλιοθήκες αρχείων, βιβλία επισκεπτών και φόρμες επικοινωνίας: λίγες από τις εκατοντάδες πρόσθετες εφαρμογές που κυκλοφορούν για το Joomla!

1.3 ΤΙ ΕΙΝΑΙ ΤΟ ΧΑΜΡΡ ΚΑΙ ΤΙ ΠΕΡΙΕΧΕΙ

Το ΧΑΜΡΡ είναι ένα χρήσιμο βοήθημα που μας δίνει την δυνατότητα να κάνουμε τον υπολογιστή μας ένα web server και όχι μόνο. Αυτό είναι απαραίτητο για να εγκαταστήσουμε το Joomla. Η εγκατάστασή του είναι πολύ απλή. Το μόνο που έχουμε να κάνουμε είναι να τρέξουμε το αρχείο που μπορούμε να βρούμε δωρεάν στο διαδίκτυο μιας και το ΧΑΜΡΡ όπως και το Joomla είναι OpenSource προγράμματα.

Η έκδοση του ΧΑΜΡΡ για Windows που χρησιμοποιήσαμε περιέχει τα παρακάτω χρήσιμα εργαλεία:

- + Apache 2.2.2
- + MySQL 5.0.21

- + PHP 5.1.4 + PHP 4.4.2-pl1 + PEAR
- + PHP-Switch win32 1.0
- + XAMPP Control Version 2.3
- + XAMPP Security 1.0
- + SQLite 2.8.15
- + OpenSSL 0.9.8b
- + phpMyAdmin 2.8.1
- + ADOdb 4.80
- + Mercury Mail Transport System v4.01b
- + FileZilla FTP Server 0.9.16c
- + Webalizer 2.01-10
- + Zend Optimizer 3.0.0

1.4 ΕΓΚΑΤΑΣΤΑΣΗ JOOMLA

Καταρχήν, για να τρέξει το Joomla χρειάζεται ένας server (Apache), php και mysql. Αντί να τα σετάρουμε όλα ένα ένα, κατεβάζουμε το Xampp (<http://www.apachefriends.org/download.php?xampp-win32-1.6.7-installer.exe>) (υπάρχουν και άλλα, αλλά εγώ χρησιμοποιώ αυτό αφού έχει εκδόσεις και για linux,mac os).

Η εγκατάσταση του είναι πολύ απλή (next, yes,next). Θα δημιουργηθεί ένας φάκελος C:\xampp όπου εκεί βρίσκονται όλα....

Αφού τελειώσει η εγκατάσταση, πάμε στη διεύθυνση <http://localhost> για να σιγουρευτούμε ότι όλα πήγαν κατ' ευχήν.

Ωραία, τώρα πρέπει να κατεβάσουμε και το Joomla (http://joomlancode.org/gf/download/frsrelease/8102/29421/Joomla_1.5.5-Stable-Full_Package.zip). Η τελευταία έκδοση είναι η 1.5.5 η οποία βγήκε μόλις προχτές. Αφού κατέβει το zip το ξεζιπάρουμε μέσα στο φάκελο C:\xampp\htdocs. Εννοείται μέσα σε ένα φάκελο, στον οποίο μπορείτε να δώσετε ότι όνομα θέλετε (π.χ mycoolsite). Εχουμε δηλαδή μέχρι τώρα ένα φάκελο C:\xampp\htdocs\mycoolsite και εκεί μέσα είναι όλα τα αρχεία του joomla.

Ωραία, και τώρα ήρθε η ώρα για την εγκατάσταση... Πρώτα πρέπει να φτιάξουμε τη βάση (δεδομένων) μας. Πάμε στη διεύθυνση <http://localhost/phpmyadmin/>. Εκεί στο πεδίο Create new database πρέπει να κάνουμε 2 πράγματα. Πρώτα βάζουμε ένα όνομα για τη βάση (π.χ mycoolsite_db) και επιλέγουμε απο το διπλανό dropdown collation utf8_unicode_ci (είναι το τελευταίο).

Επόμενο βήμα είναι αυτό καθεαυτό το σετάρισμα του Joomla. Πάμε λοιπόν στη διεύθυνση <http://localhost/mycoolsite> στην οποία θα εμφανιστεί η πρώτη σελίδα του setup. Απλά επιλέγουμε τη γλώσσα η οποία θα χρησιμοποιηθεί στην εγκατάσταση, στην περίπτωση μας τα αγγλικά.

Στη δεύτερη σελίδα γίνεται ένας έλεγχος αν έχουμε ότι χρειάζεται για να συνεχίσουμε.

Στην τρίτη σελίδα διαβάζουμε τους όρους χρήσης (...). Πατάμε next

Στην τέταρτη σελίδα συμπληρώνουμε τα στοιχεία της βάσης.
Host name: localhost

Username: root

Password: κενό (αφού δεν έχουμε βάλει κάποιο στο στήσιμο του phpmyadmin)

Database Name: mycoolsite_db (ή όποιο άλλο όνομα έχετε επιλέξει)
Στην πέμπτη σελίδα είναι το ftp configuration, το οποίο προς το παρόν δεν μας χρειάζεται. Πατάμε next

Στην έκτη σελίδα συμπληρώνουμε το όνομα της ιστοσελίδας μας (το οποίο μπορεί να αλλάξει και αργότερα), το email που θα χρησιμοποιήσουμε σαν admin, και το administrator password που καλό θα είναι να το γράψετε κάπου γιατί εδώ δεν έχει forgot my password και λοιπές δικαιολογίες.

Επίσης υπάρχει ένα κουμπάκι Install Sample Data το οποίο αν πατήσουμε το joomla θα εισάγει στη βάση μας κάποια default άρθρα, menu και modules. Αν όχι, το site θα είναι εντελώς άδειο! Αν είναι η πρώτη σας φορά (...) πατήστε το για να δείτε τι παίζεται με τις ρυθμίσεις

κτλ, αλλά ίmho μαθαίνει κανείς καλύτερα όταν βάζει ένα ένα κομμάτι και βλέπει τι γίνεται. Με τα sample data θα χαθείτε...
Πατάμε next

Στην έβδομη και τελευταία σελίδα μας ζητάει να σβήσουμε το φάκελο installation από το directory που βρίσκονται τα αρχεία μας. Πάμε λοιπόν στο C:\xampp\htdocs\mycoolsite και σβήνουμε το φάκελο.

Είμαστε έτοιμοι! Πατάμε το κουμπάκι admin και μας βγάζει σε μια οθόνη για να κάνουμε log in σαν admin. Εισάγουμε τα στοιχεία μας... και πατάμε Log In

Στο επόμενο κεφάλαιο θα δούμε τις δυνατότητες και τις λειτουργίες που μας προσφέρει το Joomla.

1.5 Ο ΔΙΚΤΥΑΚΟΣ ΠΕΡΙΗΓΗΤΗΣ MOZILLA FIREFOX 3.6

Το περιβάλλον στο οποίο επιλέξαμε να δουλέψουμε είναι ο περιηγητής Mozilla Firefox. Ο **Mozilla Firefox** είναι ένας ελεύθερος και ανοιχτού κώδικα φυλλομετρητής (*browser*) του παγκοσμίου ιστού. Προήλθε από το Application Suite της Mozilla και η ανάπτυξή του εξακολουθεί να γίνεται κατά μεγάλο ποσοστό από την Mozilla Corporation, ενώ συνεισφέρουν και μεμονωμένοι χρήστες σε μικρότερο βαθμό. Ο Firefox κατείχε το 25% της καταγεγραμμένης χρήσης φυλλομετρητών Ιστού για τον Νοέμβριο του 2009, κατατάσσοντας τον στην δεύτερη θέση των πιο δημοφιλών φυλλομετρητών παγκοσμίως, μετά τον internet explorer.

Για την απεικόνιση των ιστοσελίδων, ο Firefox χρησιμοποιεί την μηχανή διάταξης Gecko, η οποία εφαρμόζει τα περισσότερα από τα σημερινά πρότυπα του Παγκόσμιου Ιστού αλλά και επιπλέον πρότυπα που θα ισχύουν στον μέλλον.

Στις λειτουργίες του Firefox περιλαμβάνονται φραγή αυτόκλητα αναδυόμενων παραθύρων, περιήγηση με καρτέλες, ορθογραφικός έλεγχος, επιμέρους εύρεση, ενεργοί σελιδοδείκτες, διαχείριση των μεταφορτώσεων, ιδιωτική περιήγηση και ένα ενσωματωμένο πεδίο αναζήτησης με δυνατότητα επιλογής της επιθυμητής μηχανής αναζήτησης. Περαιτέρω λειτουργίες ενεργοποιούνται μέσω πρόσθετων

που αναπτύχθηκαν από τρίτους. Τα πιο δημοφιλή από τα πρόσθετα είναι το NoScript που απενεργοποιεί τα JavaScripts, ο ενσωματωμένος στην γραμμή κατάστασης αναπαραγωγής πολυμέσων FoxyTunes, το Adblock Plus που κάνει φραγή διαφημίσεων, το StumbleUpon, το DownThemAll! και η γραμμή εργαλείων Web Developer.

Ο Firefox λειτουργεί σε αρκετές εκδόσεις των Microsoft Windows, στο Mac Os X, στο GNU/Linux, και σε πολλά λειτουργικά συστήματα που προήλθαν από το Unix. Η τρέχουσα έκδοσή, 3.6, κυκλοφόρησε στις 21 Ιανουαρίου 2010. Ο πηγαίος κώδικας του Firefox είναι ελεύθερο λογισμικό και διακινείται ελεύθερα υπό την τριπλή αδειοδότηση GNU GPL/GNU LGPL/MPL.

1.6 ADOBE PHOTOSHOP CS3

Τις εικόνες που χρησιμοποιήσαμε τις επεξεργαστήκαμε πρώτα με το adobe photoshop cs3. Το παραπάνω πρόγραμμα επεξεργασίας εικόνας είναι το καλύτερο στο είδος του. Χρησιμοποιήθηκε κυρίως για την μεταβολή του μεγέθους των εικόνων καθώς και για την πρόσθεση κειμένου πάνω στην εικόνα του κεντρικού banner.

Το **Adobe Photoshop**, ή απλά **Photoshop**, είναι ένα πρόγραμμα επεξεργασίας γραφικών που αναπτύχθηκε και κυκλοφόρησε από την Adobe Systems. Αυτή τη στιγμή αποτελεί ηγέτη της αγοράς (*market leader*) των προγραμμάτων επεξεργασίας εικόνων, και είναι το προϊόν - σήμα κατατεθέν της Adobe Systems. Χαρακτηρίζεται ως "απαραίτητο εργαλείο για τους επαγγελματίες γραφίστες" και θεωρείται πως προώθησε τις αγορές των Macintosh, και στη συνέχεια των Windows.

Η 12η, και πιο πρόσφατη έκδοση του Adobe Photoshop, είναι η Adobe Photoshop CS5, που κυκλοφόρησε τον Απρίλιο του 2010.

Το 1987 ο Τόμας Κνολ, ένας φοιτητής του Πανεπιστημίου του Μίσιγκαν, ανέπτυξε ένα πρόγραμμα που εμφάνιζε εικόνες grayscale σε μονοχρωματικό περιβάλλον. Αυτό το πρόγραμμα, το οποίο ονόμασε Display, τράβηξε την προσοχή του αδερφού του Τζον Κνολ, ο οποίος πρότεινε στον Τόμας να αναπτύξει ένα πλήρες πρόγραμμα επεξεργασίας εικόνας. Ο Τόμας έκανε διάλειμμα έξι μηνών από τις σπουδές του το 1988 και, σε συνεργασία με τον αδερφό του, ανέπτυξε το πρόγραμμα, το οποίο ονόμασαν ImagePro. Αργότερο το ίδιο έτος, ο Τόμας μετονόμασε

το πρόγραμμα του σε Photoshop και έπειτα από συμφωνία με την κατασκευάστρια εταιρία σαρωτών Barneyscan, το πρόγραμμα διανεμήθηκε μαζί με μερικούς σαρωτές. Συνολικά 200 αντίγραφα του προγράμματος διανεμήθηκαν.

Εν τω μεταξύ, ο Τζον ταξίδεψε στο Σίλικον Βάλει και παρουσίασε το πρόγραμμα του στους μηχανικούς της Apple και στην Adobe. Και οι δύο παρουσιάσεις ήταν επιτυχείς, καθώς η Adobe αποφάσισε να αγοράσει την άδεια να διανείμει το πρόγραμμα τον Σεπτέμβριο του 1988. Η επόμενη έκδοση του προγράμματος, η Photoshop 1.0, κυκλοφόρησε το 1990 αποκλειστικά για συστήματα Macintosh και είχε μέγεθος 1.44 MB.

ΚΕΦΑΛΑΙΟ 2

Η Γλώσσα Προγραμματισμού PHP

2.1 Τι Είναι η PHP

Η PHP, όπου τα αρχικά σημαίνουν Hypertext PreProcessor, είναι μια γλώσσα συγγραφής σεναρίων (scripting language) που ενσωματώνεται μέσα

στον κώδικα της HTML και εκτελείται στην πλευρά του server (server-side scripting).

Ανταγωνιστικές της τεχνολογίας PHP είναι οι εξής γλώσσες προγραμματισμού : ASP (Active Server Pages) της εταιρείας Microsoft, CFML

(ColdFusion Markup Language) της εταιρείας Allaire και JSP (JavaServer Pages) της εταιρείας Sun.

Το μεγαλύτερο μέρος της σύνταξής της, η PHP το έχει δανειστεί από την C, την Java και την Perl και διαθέτει και μερικά δικά της μοναδικά χαρακτηριστικά. Ο σκοπός της γλώσσας είναι να δώσει τη δυνατότητα στους

web developers να δημιουργούν δυναμικά παραγόμενες ιστοσελίδες.

Ακολουθεί ένα εισαγωγικό παράδειγμα :

```
<html>
<head>
<title> Παράδειγμα </title>
</head>
<body>
<?php echo "Γεια σας, είμαι ένα script της PHP!"; ?>
</body>
</html>
```

Προσέξτε πόσο διαφέρει από ένα CGI script που γράφεται σ' άλλες γλώσσες, όπως η Perl ή η C, όπου αντί να γράψουμε ένα πρόγραμμα με πολλές εντολές για να δημιουργήσουμε κώδικα HTML, γράφουμε ένα HTML script με κάποιον ενσωματωμένο κώδικα για να κάνει κάτι, όπως στην συγκεκριμένη περίπτωση να εμφανίσει κάποιο κείμενο (μήνυμα). Ο κώδικας της PHP περικλείεται με ειδικά tags αρχής και τέλους για να μπορούμε να εισερχόμαστε και να εξερχόμαστε από το PHP mode.

Αυτό που ξεχωρίζει την PHP από μια γλώσσα όπως η JavaScript, η οποία εκτελείται στην πλευρά του χρήστη (client-side), είναι ότι ο κώδικάς της εκτελείται στον server. Αν είχαμε σ' έναν server ένα script παρόμοιο με το παραπάνω, ο χρήστης (client) θα λάμβανε το αποτέλεσμα της εκτέλεσης αυτού του script, χωρίς να είναι σε θέση να γνωρίζει ποιος μπορεί να είναι ο αρχικός κώδικας.

Μπορούμε ακόμη να ρυθμίσουμε (configure) τον web server ώστε να επεξεργάζεται όλα τα HTML αρχεία με την PHP και τότε δεν θα υπάρχει πράγματι κανένας τρόπος να μάθουν οι χρήστες τον κώδικά μας.

2.2Τι Μπορεί να Κάνει η PHP

Στο πιο βασικό επίπεδο, η PHP μπορεί να κάνει ό,τι και τα άλλα προγράμματα της τεχνολογίας CGI, όπως επεξεργασία των δεδομένων μιας φόρμας, δημιουργία δυναμικού περιεχομένου ιστοσελίδων ή αποστολή και λήψη cookies.

Ίσως το δυνατότερο και πιο σημαντικό χαρακτηριστικό της PHP είναι η υποστήριξη που παρέχει σε μια ευρεία γκάμα από βάσεις δεδομένων. Έτσι, το

να δημιουργήσουμε μια ιστοσελίδα που να παρέχει υποστήριξη σε βάσεις δεδομένων είναι απίστευτα απλό. Υποστηρίζει τις εξής βάσεις δεδομένων :

Adabas D dBase Empress FilePro Informix InterBase mSQL
MySQL Oracle PostgreSQL Solid Sybase Velocis Unix dbm

Η PHP παρέχει επίσης υποστήριξη για συνομιλία μ' άλλες υπηρεσίες, χρησιμοποιώντας πρωτόκολλα όπως τα IMAP, SNMP, NNTP, POP3 ή και το HTTP.

Η ιδέα για την δημιουργία της PHP ελήφθη το φθινόπωρο του 1994 από τον Rasmus Lerdorf. Οι πρώτες ανεπίσημες εκδόσεις (versions) της PHP χρησιμοποιήθηκαν στην αρχική του σελίδα (home page) για να μπορεί να παρακολουθεί αυτούς που έμπαιναν στην σελίδα. Η πρώτη έκδοση που δόθηκε

για χρήση στο κοινό ήταν διαθέσιμη στις αρχές του 1995 με το όνομα Personal Home Page Tools.

Αποτελείτο από μια πολύ απλοϊκή μηχανή ανάλυσης (parser engine) η οποία καταλάβαινε λίγες μόνο ειδικές μακροεντολές (macros) και έναν αριθμό

από utilities που βρίσκονταν σε κοινή χρήση στις home pages εκείνη την εποχή. Ένα guestbook, ένας μετρητής (counter) και κάποιο άλλο υλικό. Ο αναλυτής (parser) ξαναγράφηκε στα μέσα του 1995 και ονομάστηκε PHP/FI

Version 2.

Το όνομα FI προέρχεται από ένα άλλο πακέτο που είχε γράψει ο Rasmus και το οποίο διερμήνευε (interpreted) τα δεδομένα από φόρμες της

HTML. Συνδύασε τα εργαλεία scripts της Personal Home Page με τον Form

Interpreter και πρόσθεσε υποστήριξη για mSQL. Έτσι γεννήθηκε η PHP/FI, η

οποία αναπτύχθηκε αλματωδώς και διάφοροι χρήστες άρχισαν να συνεισφέρουν κώδικα σ' αυτήν.

Υπολογίζεται ότι μέχρι τα τέλη του 1996, η PHP/FI χρησιμοποιείτο σε τουλάχιστον 15.000 web sites σ' όλον τον κόσμο και στα μέσα του 1997 αυτός

ο αριθμός είχε ξεπεράσει τις 50.000. Στα μέσα του 1997 είχαμε επίσης μια

αλλαγή στην ανάπτυξη της PHP.

Σταμάτησε να αποτελεί το αγαπημένο αντικείμενο του Rasmus και έγινε ο στόχος μιας πιο καλά οργανωμένης ομαδικής εργασίας. Ο αναλυτής (parser)

ξαναγράφηκε από την αρχή από τους Zeev Suraski και Andi Gutmans και αυτός ο νέος parser αποτέλεσε τη βάση για την PHP Version 3. Ένα μεγάλο

μέρος του utility code μεταφέρθηκε από την PHP/FI στην PHP3 και ένα μεγάλο μέρος του ξαναγράφηκε από την αρχή.

Σήμερα, η PHP/FI και η PHP3 έρχονται μ' έναν αριθμό εμπορικών προϊόντων όπως ο web server C2 StrongHold και το RedHat Linux.

Σύμφωνα

με μια συντηρητική εκτίμηση, η PHP χρησιμοποιείται από περισσότερα από

150.000 sites σ' όλον τον κόσμο.

Υπάρχουν τέσσερις τρόποι για να μπορέσουμε να ξεφύγουμε από την HTML και να μπούμε στην μέθοδο συγγραφής κώδικα της PHP (PHP code

mode) :

1ος τρόπος

```
<? echo ("Είναι η απλούστερη, μια εντολή επεξεργασίας SGML \n"); ?>
```

2ος τρόπος

```
<?php echo("Αν θέλουμε να εξυπηρετήσουμε XML έγγραφα \n"); ?>
```

3ος τρόπος

```
<script language="php">
```

```
echo ("Σε μερικούς editors, όπως ο FrontPage, δεν αρέσουν οι εντολές επεξεργασίας");
```

```
</script>
```

4ος τρόπος

```
<% echo ("Μπορούμε να χρησιμοποιήσουμε και tags με στυλ ASP"); %>
```

```
<%= $variable; # Είναι μια συντόμευση για το "<%echo .." %>
```

Ο πρώτος τρόπος είναι διαθέσιμος μόνο αν έχουμε ενεργοποιήσει τα σύντομα (short) tags. Αυτό μπορεί να γίνει με τη συνάρτηση short_tags(), ενεργοποιώντας το short_open_tag configuration setting στο αρχείο config της

PHP ή μεταγλωττίζοντας την PHP με την επιλογή --enable-short-tags option.

Ο τέταρτος τρόπος είναι διαθέσιμος μόνο αν έχουν ενεργοποιηθεί τα tags με στυλ ASP με το asp_tags configuration setting. Η υποστήριξη για τα

ASP-style tags προστέθηκε στην έκδοση 3.0.4.

2.3 Τερματισμός Εντολών

Οι εντολές στην PHP τερματίζονται με τον ίδιο τρόπο όπως στην C και την Perl, δηλ. μ' έναν χαρακτήρα ; (semicolon). Μπορούμε, όμως, να

δηλώσουμε το τέλος μιας εντολής και με το tag κλεισίματος (closing tag) ?>.

Έτσι, τα παρακάτω είναι ισοδύναμα :

```
<?php
echo "This is a test";
?>
και
<?php echo "This is a test" ?>
```

Σχόλια (Comments)

Η PHP χρησιμοποιεί τον ίδιο τρόπο σχολιασμού όπως η C, η C++ και το Unix shell. Για παράδειγμα :

```
<?php
echo "Αυτή είναι μια δοκιμή"; // Σχόλιο μίας γραμμής της C++
/* Αυτό είναι ένα σχόλιο (comment) της C σε πολλές γραμμές
και αυτή είναι μια άλλη γραμμή σχολίου */
echo "Αυτή είναι άλλη μια δοκιμή";
echo "Μια τελική δοκιμή"; # Σχόλιο της shell
?>
```

Τα σχόλια μίας γραμμής σχολιάζουν μέχρι το τέλος της γραμμής ή το τρέχον μπλοκ του PHP κώδικα, ανάλογα με το ποιο εμφανίζεται πρώτο.

```
<h1> Αυτό είναι ένα <?# echo "απλό";?> παράδειγμα. </h1>
```

```
<p> Το header θα εμφανίσει το 'Αυτό είναι ένα παράδειγμα.' </p>
```

Πρέπει να είμαστε προσεκτικοί για να μην φωλιάζουμε (nest) τα σχόλια τύπου C.

```
<?php
/*
echo "Αυτή είναι μια δοκιμή";
/* Αυτό το σχόλιο θα δημιουργήσει πρόβλημα */
*/
?>
```

2.4 Οι Τύποι Δεδομένων της PHP

Η PHP υποστηρίζει τους εξής τύπους δεδομένων :

- array
- floating-point numbers
- integer
- object
- string

Ο τύπος δεδομένων μιας μεταβλητής δεν ορίζεται συνήθως από τον προγραμματιστή αλλά αποφασίζεται την ώρα εκτέλεσης (runtime) από την

PHP ανάλογα με το περιβάλλον (context) στο οποίο χρησιμοποιείται η

μεταβλητή. Αν θέλουμε να κάνουμε μια μεταβλητή να μετατραπεί σ' έναν συγκεκριμένο τύπο, μπορούμε είτε να μετατρέψουμε (cast) τη μεταβλητή ή να χρησιμοποιήσουμε τη συνάρτηση `settype()` σ' αυτή. Πρέπει να έχουμε υπόψη μας ότι μια μεταβλητή μπορεί να συμπεριφερθεί διαφορετικά σε συγκεκριμένες καταστάσεις, ανάλογα με το τι τύπο δεδομένων έχει εκείνη την στιγμή.

Οι Ακέραιοι (Integers)

Οι ακέραιοι (integers) μπορούν να καθορισθούν χρησιμοποιώντας μια από τις εξής συντάξεις :

`$a = 1234;` # δεκαδικός αριθμός

`$a = -123;` # αρνητικός αριθμός

`$a = 0123;` # οκταδικός αριθμός (ισοδύναμος με τον δεκαδικό 83)

`$a = 0x12;` # δεκαεξαδικός αριθμός (ισοδύναμος με τον δεκαδ. 18)

Οι Αριθμοί Κινητοί Υποδιαστολής

Οι αριθμοί κινητής υποδιαστολής (floating point numbers ή doubles), μπορούν να καθορισθούν χρησιμοποιώντας μια από τις εξής συντάξεις :

`$a = 1.234;`

`$a = 1.2e3;`

Τα Αλφαριθμητικά (Strings)

Τα αλφαριθμητικά (strings) μπορούν να καθορισθούν χρησιμοποιώντας ένα από δύο σύνολα οριοθετών (delimiters). Αν το string περικλείεται από

διπλά εισαγωγικά (double-quotes, `"`), οι μεταβλητές μέσα στο string θα επεκταθούν.

Όπως ισχύει στην C και την Perl, ο χαρακτήρας backslash (`\`) μπορεί να χρησιμοποιηθεί για να καθορίσουμε τους ειδικούς χαρακτήρες :

Ειδικός Χαρακτήρας Νόημα

`\n` Νέα γραμμή (newline)

`\r` carriage

`\t` Οριζόντιο tab (στηλοθέτης)

`\\` Χαρακτήρας backslash

`\$` Σύμβολο του δολαρίου

`\"` Διπλά εισαγωγικά

`\[0-7]{1,3}` Η σειρά των χαρακτήρων που ταιριάζει με την κανονική έκφραση ένας χαρακτήρας του 8δικού συστήματος (octal notation)

`\x[0-9A-Fa-f]{1,2}` Η σειρά των χαρακτήρων που ταιριάζει με την κανονική έκφραση ένας χαρακτήρας του 16δικού συστήματος (hexadecimal notation)

Ο δεύτερος τρόπος για να οριοθετήσουμε (delimit) ένα string

χρησιμοποιεί τον χαρακτήρα μονού εισαγωγικού (single-quote, '). Όταν ένα

string περικλείεται από μονά εισαγωγικά, οι μόνοι ειδικοί χαρακτήρες (escapes) που γίνονται αντιληπτοί είναι οι \\ και \'. Οι μεταβλητές δεν επεκτείνονται (αναλύονται) μέσα σ' ένα string που περικλείεται από μονά εισαγωγικά.

Ένας άλλος τρόπος για να οριοθετήσουμε strings είναι να χρησιμοποιήσουμε τη σύνταξη heredoc syntax (">>>"). Θα πρέπει να υπάρχει

ένα αναγνωριστικό (identifier) μετά από το >>>, μετά το string και μετά το ίδιο

αναγνωριστικό για να κλείσει. Ακολουθεί ένα παράδειγμα.

```
$str = >>>EOD
```

Παράδειγμα ενός string

που εκτείνεται σε πολλές γραμμές

χρησιμοποιώντας τη σύνταξη heredoc syntax.

```
EOD;
```

Τα strings μπορούν να ενωθούν (concatenated) με τον τελεστή '.' (dot), ενώ ο τελεστής της πρόσθεσης '+' δεν μπορεί να κάνει συνένωση.

Μπορούμε να έχουμε πρόσβαση στους χαρακτήρες των strings αντιμετωπίζοντας το string σαν έναν πίνακα χαρακτήρων, χρησιμοποιώντας

μια σύνταξη που θυμίζει C. Ακολουθούν παραδείγματα.

```
<?php
```

```
/* Εκχώρηση ενός string */
```

```
$str = "This is a string";
```

```
/* Προσθήκη σ' ένα string */
```

```
$str = $str . " with some more text";
```

```
/* Ένας άλλος τρόπος προσθήκης */
```

```
$str .= " and a newline at the end.\n";
```

```
/* Αυτό το string θα τελειώνει ως '<p> Number : 9 </p>' */
```

```
$num = 9;
```

```
$str = "<p> Number : $num </p>";
```

```
/* Και αυτό ως '<p> Number : $num </p>' */
```

```
$num = 9;
```

```
$str = '<p> Number : $num </p>';
```

```
/* Διαβάζουμε τον πρώτο χαρακτήρα ενός string */
```

```
$str = 'This is a test.';
```

```
$first = $str[0];
```

```
/* Διαβάζουμε τον τελευταίο χαρακτήρα ενός string */
```

```
$str = 'This is still a test.';
```

```
$last = $str[strlen($str)-1];
```

?>

Μετατροπή Strings

Όταν ένα string αποτιμάται σαν μια αριθμητική τιμή, η προκύπτουσα τιμή και ο τύπος δεδομένων καθορίζονται ως εξής : Το string θα αποτιμηθεί σε

τύπο δεδομένων double αν περιέχει έναν από τους χαρακτήρες '.', 'e' ή 'E', αλλιώς θα αποτιμηθεί σαν ακέραιος (integer).

Η τιμή δίνεται από το αρχικό τμήμα του string. Αν το string ξεκινά με έγκυρα αριθμητικά δεδομένα, αυτή θα είναι και η τιμή που θα χρησιμοποιηθεί.

Διαφορετικά, η τιμή του θα είναι 0.

Έγκυρα αριθμητικά δεδομένα σημαίνει ένα προαιρετικό πρόσημο, ακολουθούμενο από ένα ή περισσότερα ψηφία, και ίσως μια υποδιαστολή,

ακολουθούμενα από έναν προαιρετικό εκθέτη. Ο εκθέτης είναι ένα 'e' ή 'E'

ακολουθούμενος από ένα ή περισσότερα ψηφία.

Όταν η πρώτη έκφραση είναι ένα string, ο τύπος δεδομένων της μεταβλητής θα εξαρτηθεί από τη δεύτερη έκφραση.

```
$foo = 1 + "10.5"; // $foo is double (11.5)
```

```
$foo = 1 + "-1.3e3"; // $foo is double (-1299)
```

```
$foo = 1 + "bob-1.3e3"; // $foo is integer (1)
```

```
$foo = 1 + "bob3"; // $foo is integer (1)
```

```
$foo = 1 + "10 Small Pigs"; // $foo is integer (11)
```

```
$foo = 1 + "10 Little Piggies"; // $foo is integer (11)
```

```
$foo = "10.0 pigs " + 1; // $foo is integer (11)
```

```
$foo = "10.0 pigs " + 1.0; // $foo is double (11)
```

Αν θέλουμε να δοκιμάσουμε κάποια από τις παραπάνω εκφράσεις, μπορούμε να την αντιγράψουμε και να εισάγουμε την παρακάτω γραμμή για

να δούμε τι γίνεται :

```
echo "\$foo==\$foo; type is " . gettype($foo) . "<br>\n";
```

2.5 Πίνακες Μίας Διάστασης (Single Dimension Arrays)

Οι πίνακες (arrays) ενεργούν και σαν πίνακες hash (associative arrays) και σαν δεικτοδοτούμενοι πίνακες (indexed arrays) ή διανύσματα (vectors).

Η PHP υποστηρίζει και τους scalar και τους associative πίνακες. Στην πραγματικότητα, δεν υπάρχει καμία διαφορά ανάμεσά τους. Μπορούμε να

δημιουργήσουμε έναν πίνακα με τις συναρτήσεις list() ή array() ή μπορούμε

να ορίσουμε την τιμή κάθε στοιχείου του πίνακα, ως εξής :

```
$a[0] = "abc";
```

```
$a[1] = "def";
```

```
$b["foo"] = 13;
```

Μπορούμε επίσης να δημιουργήσουμε έναν πίνακα προσθέτοντας απλά τιμές στον πίνακα. Όταν εκχωρούμε μια τιμή σε μια μεταβλητή πίνακα χρησιμοποιώντας κενές αγκύλες, η τιμή θα προστεθεί στο τέλος του πίνακα.

```
$a[] = "hello"; // $a[2] == "hello"
```

```
$a[] = "world"; // $a[3] == "world"
```

Μπορούμε να ταξινομήσουμε τους πίνακες με τις συναρτήσεις `asort()`, `arsort()`, `ksort()`, `rsort()`, `sort()`, `uasort()`, `usort()` και `uksort()`, ανάλογα με τον

τύπο της ταξινόμησης που θέλουμε να κάνουμε.

Μπορούμε να μετρήσουμε τον αριθμό των στοιχείων ενός πίνακα με τη συνάρτηση `count()` και μπορούμε να διασχίσουμε έναν πίνακα με τις συναρτήσεις `next()` και `prev()` ή με τη συνάρτηση `each()`.

Πίνακες Πολλών Διαστάσεων (Multi-Dimension Arrays)

Οι πίνακες πολλών διαστάσεων είναι αρκετά απλοί. Για κάθε διάσταση (dimension) του πίνακα, προσθέτουμε μια τιμή [key]. Ακολουθούν παραδείγματα.

```
$a[1] = $f; # Πίνακες μίας διάστασης
```

```
$a["foo"] = $f;
```

```
$a[1][0] = $f; # Πίνακες δύο διαστάσεων
```

```
$a["foo"][2] = $f; # Μπορούμε να αναμείξουμε αριθμητικούς
```

```
$a[3]["bar"] = $f; # και associative δείκτες (indices)
```

```
$a["foo"][4]["bar"][0] = $f; # Πίνακας τεσσάρων διαστάσεων
```

Στην PHP3 δεν είναι δυνατό να αναφερθούμε σε πίνακες πολλών διαστάσεων απευθείας μέσα από strings. Για παράδειγμα, η επόμενη εντολή

δεν θα έχει το επιθυμητό αποτέλεσμα :

```
$a[3]['bar'] = 'Bob';
```

```
echo "Αυτό δεν θα δουλέψει : $a[3][bar]";
```

Στην PHP3, το παραπάνω θα δημιουργήσει την έξοδο Αυτό δεν θα δουλέψει : Array[bar]. Ο τελεστής ένωσης string, όμως, μπορεί να χρησιμοποιηθεί για να το ξεπεράσουμε αυτό :

```
$a[3]['bar'] = 'Bob';
```

```
echo "Αυτό θα δουλέψει : " . $a[3][bar];
```

Στην PHP4, όμως, μπορούμε να παρακάμψουμε αυτό το πρόβλημα αν περικλείσουμε την αναφορά στον πίνακα (μέσα στο string) με άγκιστρα, ως

εξής :

```
$a[3]['bar'] = 'Bob';  
echo "Αυτό θα δουλέψει : {$a[3][bar]}";  
Μπορούμε να καταχωρήσουμε στοιχεία σε πίνακες με πολλούς τρόπους.  
Τα δύο επόμενα παραδείγματα γεμίζουν έναν πίνακα μίας διάστασης με  
τα ίδια
```

στοιχεία :

```
# Παράδειγμα 1 :
```

```
$a["color"] = "red";  
$a["taste"] = "sweet";  
$a["shape"] = "round";  
$a["name"] = "apple";  
$a[3] = 4;
```

```
# Παράδειγμα 2 :
```

```
$a = array(  
    "color" => "red",  
    "taste" => "sweet",  
    "shape" => "round",  
    "name" => "apple",  
    3 => 4  
);
```

Μπορούμε να φωλιάσουμε (nest) τη συνάρτηση array() για πίνακες πολλαπλών διαστάσεων :

```
<?
```

```
$a = array(  
    "apple" => array(  
        "color" => "red",  
        "taste" => "sweet",  
        "shape" => "round"  
    ),  
    "orange" => array(  
        "color" => "orange",  
        "taste" => "tart",  
        "shape" => "round"  
    ),  
    "banana" => array(  
        "color" => "yellow",  
        "taste" => "paste-y",  
        "shape" => "banana-shaped"  
    )  
);  
echo $a["apple"]["taste"]; # Θα δώσει το αποτέλεσμα "sweet"  
?>
```

2.6 Τα Αντικείμενα (Objects)

Για να αρχικοποιήσουμε (initialize) ένα αντικείμενο (object), χρησιμοποιούμε την εντολή `new` για να δημιουργήσουμε μια μεταβλητή από το αντικείμενο.

```
class foo {  
function do_foo () {  
echo "Doing foo."  
}  
}  
$bar = new foo;  
$bar->do_foo();
```

Κόλπα με τους Τύπους Δεδομένων (Type Juggling)

Η PHP δεν υποστηρίζει τον σαφή (explicit) ορισμό τύπων δεδομένων στις δηλώσεις μεταβλητών και αυτό γιατί ο τύπος μιας μεταβλητής καθορίζεται από το περιβάλλον (context) στο οποίο χρησιμοποιείται αυτή η μεταβλητή.

Αυτό σημαίνει ότι αν εκχωρήσουμε μια τιμή `string` σε μια μεταβλητή `var`, η `var` θα γίνει ένα `string` και αν αργότερα εκχωρήσουμε μια ακέραια τιμή στην `var`, αυτή θα γίνει ακέραια (integer).

Ένα παράδειγμα της αυτόματης μετατροπής τύπου στην PHP είναι ο τελεστής πρόσθεσης `+`. Αν κάποιος από τους τελεστέους (operands) είναι μια τιμή `double`, τότε όλοι οι τελεστέοι εκτιμούνται σαν τύπου `doubles` και το αποτέλεσμα θα είναι τύπου `double`.

Αλλιώς, οι τελεστέοι θα θεωρηθούν ότι είναι ακέραιοι (integers) και το αποτέλεσμα θα είναι επίσης `integer`. Αυτό βέβαια δεν αλλάζει τους τύπους δεδομένων των ίδιων των τελεστέων και η μόνη αλλαγή είναι στο πώς εκτιμούνται οι τελεστέοι.

```
$foo = "0"; // $foo is string (ASCII 48)  
$foo++; // $foo is the string "1" (ASCII 49)  
$foo += 1; // $foo is now an integer (2)  
$foo = $foo + 1.3; // $foo is now a double (3.3)  
$foo = 5 + "10 Little Piggies"; // $foo is integer (15)  
$foo = 5 + "10 Small Pigs"; // $foo is integer (15)
```

Αν θέλουμε να δοκιμάσουμε κάποια από τις παραπάνω εκφράσεις, μπορούμε να την αντιγράψουμε και να εισάγουμε την παρακάτω γραμμή για να δούμε τι γίνεται :

```
echo "\$foo==\$foo; type is " . gettype( $foo ) . "<br>\n";
```

Η Μετατροπή Τύπων (Type Casting)

Η μετατροπή τύπων (type casting) στην PHP εργάζεται όπως και στην C : το όνομα του επιθυμητού τύπου γράφεται μέσα σε παρενθέσεις πριν από τη

μεταβλητή η οποία θα μετατραπεί.

```
$foo = 10; // Το $foo είναι τύπου integer
```

```
$bar = (double) $foo; // Το $bar είναι τύπου double
```

Οι μετατροπές (casts) που επιτρέπονται είναι οι εξής :

(int), (integer) - cast to integer

(real), (double), (float) - cast to double

(string) - cast to string

(array) - cast to array

(object) - cast to object

Μπορεί να υπάρχουν κενά και tabs μέσα στις παρενθέσεις :

```
$foo = (int) $bar;
```

```
$foo = ( int ) $bar;
```

Όταν κάνουμε μετατροπή από μια μεταβλητή scalar ή string σ' έναν πίνακα (array), η μεταβλητή θα γίνει το πρώτο στοιχείο του πίνακα :

```
$var = 'ciao';
```

```
$arr = (array) $var;
```

```
echo $arr[0]; // εμφανίζει 'ciao'
```

Όταν κάνουμε μετατροπή από μια μεταβλητή scalar ή string σ' ένα αντικείμενο (object), η μεταβλητή θα γίνει μια ιδιότητα (attribute) του αντικειμένου και το όνομα (name) της ιδιότητας θα είναι τύπου scalar :

```
$var = 'ciao';
```

```
$obj = (object) $var;
```

```
echo $obj->scalar; // εμφανίζει 'ciao'
```

2.7 Οι Μεταβλητές (Variables)

Οι μεταβλητές (variables) στην PHP παριστάνονται από το σύμβολο \$

ακολουθούμενο από το όνομα της μεταβλητής. Τα ονόματα των μεταβλητών

ξεχωρίζουν τα πεζά από τα κεφαλαία γράμματα (case-sensitive).

```
$var = "Bob";
```

```
$Var = "Joe";
```

```
echo "$var, $Var"; // εμφανίζει "Bob, Joe"
```

Στην PHP3, οι μεταβλητές πάντα εκχωρούνται με τιμή (by value), δηλαδή όταν εκχωρούμε μια έκφραση σε μια μεταβλητή, η τιμή της αρχικής

έκφρασης αντιγράφεται στη μεταβλητή προορισμού. Αυτό σημαίνει, για παράδειγμα, ότι αφού έχουμε εκχωρήσει την τιμή μιας μεταβλητής σε μια

άλλη, η αλλαγή σε μια απ' αυτές τις μεταβλητές δεν θα επηρεάσει την άλλη.

Η PHP4 προσφέρει και έναν άλλον τρόπο για να εκχωρήσουμε τιμές σε μεταβλητές : με αναφορά (by reference), δηλ. η νέα μεταβλητή αναφέρεται

(references) ή αποτελεί ένα ψευδώνυμο (alias) ή δείχνει (points) στην αρχική μεταβλητή.

Οι αλλαγές στη νέα μεταβλητή επηρεάζουν και την αρχική και το αντίστροφο. Για να κάνουμε εκχώρηση με αναφορά, τοποθετούμε το σύμβολο

& (ampersand) πριν από την αρχική μεταβλητή. Για παράδειγμα, ο επόμενος

κώδικας εμφανίζει δύο φορές το μήνυμα 'My name is Bob' :

```
<?php
```

```
$foo = 'Bob'; // Εκχώρηση της τιμής 'Bob' στην
```

```
$foo
```

```
$bar = &$foo; // Αναφορά στην $foo μέσω της
```

```
$bar.
```

```
$bar = "My name is $bar"; // Η $bar αλλάζει
```

```
echo $foo; // Η $foo αλλάζει επίσης
```

```
echo $bar;
```

```
?>
```

Κάτι σημαντικό που πρέπει να σημειώσουμε είναι ότι μόνο ονοματισμένες μεταβλητές (named variables) μπορούν να εκχωρηθούν με αναφορά (by reference).

```
<?php
```

```
$foo = 25;
```

```
$bar = &$foo; // Έγκυρη εκχώρηση
```

```
$bar = &(24 * 7); // Μη έγκυρη εκχώρηση
```

```
function test() {
```

```
return 25;
```

```
}
```

```
$bar = &test(); // Μη έγκυρη
```

```
?>
```

Οι Προκαθορισμένες Μεταβλητές

Η PHP παρέχει έναν μεγάλο αριθμό από προκαθορισμένες μεταβλητές (predefined variables) σ' οποιοδήποτε script εκτελεί. Όμως, πολλές απ' αυτές

τις μεταβλητές δεν μπορούν να τεκμηριωθούν πλήρως (documented) γιατί

εξαρτώνται από τον server στον οποίο εκτελούνται, την έκδοση (version) και

την ρύθμιση (setup) του server καθώς και από άλλους παράγοντες. Παρ' όλα αυτά, θα δούμε μια λίστα από προκαθορισμένες μεταβλητές που είναι διαθέσιμες σε μια εγκατάσταση της PHP 3 που εκτελείται σαν ένα

module σε μια εγκατάσταση του Apache 1.3.6.

Οι Μεταβλητές Apache

Αυτές οι μεταβλητές δημιουργούνται από τον Apache webserver. Αν χρησιμοποιούμε έναν άλλον webserver, ίσως να λείπουν μερικές ή να υπάρχουν άλλες που δεν εμφανίζονται εδώ. Ένας μεγάλος αριθμός των μεταβλητών που θα δούμε υπάρχουν στην προδιαγραφή CGI 1.1.

GATEWAY_INTERFACE

Ποια αναθεώρηση (revision) της προδιαγραφής CGI χρησιμοποιεί ο server, όπως π.χ. CGI/1.1.

SERVER_NAME

Το όνομα του server host στον οποίο εκτελείται το τρέχον script.

Μπορεί να είναι και το όνομα ενός εικονικού (virtual) host.

SERVER_SOFTWARE

Το string αναγνώρισης του server (server identification string), το οποίο δίνεται στις επικεφαλίδες (headers) όταν ο server απαντάει σε αιτήσεις (requests).

SERVER_PROTOCOL

Το όνομα (name) και η αναθεώρηση (revision) του πρωτοκόλλου πληροφοριών (information protocol) μέσω του οποίου ζητήθηκε η σελίδα, δηλ.

HTTP/1.0.

REQUEST_METHOD

Ποια μέθοδος αίτησης (request method) χρησιμοποιήθηκε για να έχουμε πρόσβαση στη σελίδα, δηλ. GET, HEAD, POST, PUT.

QUERY_STRING

Το query string, αν υπάρχει, μέσω του οποίου έχουμε πρόσβαση στη σελίδα.

DOCUMENT_ROOT

Το document root directory στο οποίο εκτελείται το τρέχον script, όπως ορίζεται στο αρχείο σύνθεσης (configuration file) του server.

HTTP_ACCEPT

Τα περιεχόμενα του Accept : η επικεφαλίδα (header) της τρέχουσας αίτησης (request), αν υπάρχει κάποια.

HTTP_ACCEPT_CHARSET

Τα περιεχόμενα του Accept-Charset : η επικεφαλίδα (header) της τρέχουσας αίτησης (request), αν υπάρχει κάποια. Παράδειγμα : iso-8859-1,*,utf-8'.

HTTP_ENCODING

Τα περιεχόμενα του Accept-Encoding : η επικεφαλίδα (header) της

τρέχουσας αίτησης (request), αν υπάρχει κάποια. Παράδειγμα : gzip.

HTTP_ACCEPT_LANGUAGE

Τα περιεχόμενα του Accept-Language : η επικεφαλίδα (header) της τρέχουσας αίτησης (request), αν υπάρχει κάποια. Παράδειγμα : en.

HTTP_CONNECTION

Τα περιεχόμενα του Connection : η επικεφαλίδα (header) της τρέχουσας αίτησης (request), αν υπάρχει κάποια. Παράδειγμα : Keep-Alive.

HTTP_HOST

Τα περιεχόμενα του Host : η επικεφαλίδα (header) της τρέχουσας αίτησης (request), αν υπάρχει κάποια.

HTTP_REFERER

Η διεύθυνση της σελίδας, αν υπάρχει, η οποία έφερε τον φυλλομετρητή στην τρέχουσα σελίδα.

HTTP_USER_AGENT

Τα περιεχόμενα του User_Agent : η επικεφαλίδα (header) της τρέχουσας αίτησης (request), αν υπάρχει κάποια. Αυτό το string φανερώνει το

λογισμικό του φυλλομετρητή που χρησιμοποιείται για να δούμε την τρέχουσα

σελίδα, όπως π.χ. Mozilla/4.5 [en] (X11; U; Linux 2.2.9 i586). Μπορούμε να

χρησιμοποιήσουμε αυτήν την τιμή με τη συνάρτηση get_browser() για να προσαρμόσουμε τη λειτουργικότητα της σελίδας μας σύμφωνα με τις δυνατότητες του φυλλομετρητή του χρήστη.

REMOTE_ADDR

Η IP διεύθυνση από την οποία βλέπει ο χρήστης την τρέχουσα σελίδα.

REMOTE_PORT

Η θύρα (port) που χρησιμοποιείται στο μηχάνημα του χρήστη για να επικοινωνήσει με τον web server.

SCRIPT_FILENAME

Η απόλυτη διαδρομή (pathname) του τρέχοντος εκτελούμενου script.

SERVER_ADMIN

Η τιμή που δίνεται στην οδηγία (directive) SERVER_ADMIN (για τον Apache) στο αρχείο σύνθεσης (configuration file) του web server.

SERVER_PORT

Η θύρα (port) στο μηχάνημα του server που χρησιμοποιείται από τον web server για επικοινωνία. Στις προκαθορισμένες ρυθμίσεις, έχει την τιμή 80,

ενώ αν χρησιμοποιούμε το SSL, για παράδειγμα, θα αλλάξει σ' ο,τιδήποτε

είναι η δική μας ορισμένη ασφαλής θύρα HTTP.

SERVER_SIGNATURE

Ένα string που περιέχει την έκδοση (version) του server και το virtual

host name, τα οποία προστίθενται στις παραγόμενες από τον server σελίδες, αν είναι ενεργοποιημένες.

SCRIPT_NAME

Περιέχει τη διαδρομή (path) του τρέχοντος script. Είναι χρήσιμη για σελίδες που πρέπει να δείχνουν στον εαυτό τους.

REQUEST_URI

Το URI που δόθηκε για να μπορούμε να έχουμε πρόσβαση στη σελίδα, όπως π.χ. /index.html.

Οι Μεταβλητές της PHP

Αυτές οι μεταβλητές δημιουργούνται από την ίδια την PHP.

argv

Είναι ένας πίνακας (array) από τα ορίσματα (arguments) που μεταβιβάζονται στο script. Όταν το script εκτελείται από τη γραμμή εντολών, αυτό μας δίνει μια πρόσβαση στις παραμέτρους της γραμμής εντολών, κάτι που θυμίζει την C. Όταν καλείται μέσω της μεθόδου GET, η μεταβλητή αυτή θα περιέχει το query string.

argc

Περιέχει τον αριθμό των παραμέτρων της γραμμής εντολών που μεταβιβάζονται στο script, αν αυτό εκτελείται από τη γραμμή εντολών βέβαια.

PHP_SELF

Το όνομα αρχείου (filename) του τρέχοντα εκτελούμενου script. Αν το PHP εκτελείται σαν ένας επεξεργαστής από τη γραμμή εντολών, αυτή η μεταβλητή δεν είναι διαθέσιμη.

HTTP_COOKIE_VARS

Ένας associative πίνακας (array) από μεταβλητές που μεταβιβάζονται στο τρέχον script μέσω HTTP cookies.

HTTP_GET_VARS

Ένας associative πίνακας (array) από μεταβλητές που μεταβιβάζονται στο τρέχον script μέσω της μεθόδου HTTP GET.

HTTP_POST_VARS

Ένας associative πίνακας (array) από μεταβλητές που μεταβιβάζονται στο τρέχον script μέσω της μεθόδου HTTP POST.

Η Εμβέλεια των Μεταβλητών

Η εμβέλεια (scope) μιας μεταβλητής είναι το περιβάλλον (context) μέσα στο οποίο ορίζεται. Οι περισσότερες από τις PHP μεταβλητές έχουν μία μόνο περιοχή εμβέλειας. Για παράδειγμα :

```
$a = 1;
```

```
include "b.inc";
```

Εδώ, η μεταβλητή \$a θα είναι διαθέσιμη μέσα στο συμπεριλαμβανόμενο (included) b.inc script. Όμως, στις οριζόμενες από τον προγραμματιστή συναρτήσεις (user-defined functions) υπάρχει μια τοπική εμβέλεια. Μια μεταβλητή που χρησιμοποιείται μέσα σε μια συνάρτηση είναι εξ ορισμού περιορισμένη στην τοπική εμβέλεια αυτής της συνάρτησης. Για παράδειγμα :

```
$a = 1; /* καθολική εμβέλεια */
```

```
Function Test () {
```

```
echo $a; /* αναφέρεται στην τοπική μεταβλητή */
```

```
}
```

```
Test ();
```

Αυτό το script δεν θα εμφανίσει κάποια έξοδο επειδή η εντολή echo αναφέρεται σε μια τοπική μεταβλητή (local variable) \$a, η οποία δεν έχει αποκτήσει τιμή μέσα στην εμβέλειά της. Αυτό είναι διαφορετικό από τη γλώσσα C στο ότι οι καθολικές μεταβλητές (global variables) της C είναι αυτόματα διαθέσιμες στις συναρτήσεις εκτός κι αν επικαλύπτονται σαφώς από μια τοπική δήλωση.

Αυτό μπορεί να προκαλέσει προβλήματα στο ότι μπορεί κάποιος άθελά του να αλλάξει μια καθολική μεταβλητή. Στην PHP οι καθολικές μεταβλητές

πρέπει να δηλωθούν σαν global μέσα σε μια συνάρτηση αν πρόκειται να τις

χρησιμοποιήσουμε μέσα σ' αυτή τη συνάρτηση. Ακολουθεί ένα

παράδειγμα :

```
$a = 1;
```

```
$b = 2;
```

```
Function Sum () {
```

```
global $a, $b;
```

```
$b = $a + $b;
```

```
}
```

```
Sum ();
```

```
echo $b;
```

Το παραπάνω script θα εμφανίσει το 3. Δηλώνοντας τις \$a και \$b σαν global μέσα στη συνάρτηση, όλες οι αναφορές και στις δύο μεταβλητές θα

αφορούν τις καθολικές τιμές. Δεν υπάρχει κάποιος περιορισμός στον αριθμό

των καθολικών μεταβλητών που μπορεί να χειριστεί μια συνάρτηση.

Ένας δεύτερος τρόπος για να έχουμε πρόσβαση σε μεταβλητές σε

καθολική εμβέλεια είναι να χρησιμοποιήσουμε τον ειδικό πίνακα που ορίζεται στην PHP με όνομα \$GLOBALS. Έτσι, το προηγούμενο παράδειγμα θα μπορεί να ξαναγραφεί ως εξής :

```
$a = 1;
$b = 2;
Function Sum () {
    $GLOBALS["b"] = $GLOBALS["a"] + $GLOBALS["b"];
}
Sum ();
echo $b;
```

Ο πίνακας \$GLOBALS είναι ένας associative πίνακας με το όνομα της καθολικής μεταβλητής να αποτελεί το key και τα περιεχόμενα αυτής της μεταβλητής να αποτελούν την τιμή του στοιχείου του πίνακα.

Ένα άλλο σημαντικό χαρακτηριστικό είναι οι στατικές μεταβλητές. Μια στατική μεταβλητή (static variable) υπάρχει μόνο στην τοπική εμβέλεια μιας

συνάρτησης αλλά δεν χάνει την τιμή της όταν η εκτέλεση του προγράμματος

εγκαταλείπει τη συνάρτηση. Ας δούμε το επόμενο παράδειγμα :

```
Function Test () {
    $a = 0;
    echo $a;
    $a++;
}
```

Αυτή η συνάρτηση δεν είναι και τόσο χρήσιμη εφόσον κάθε φορά που καλείται καταχωρεί στο \$a το 0 και εκτυπώνει το "0". Η εντολή \$a++, η οποία

αυξάνει την τιμή της μεταβλητής, δεν κάνει τίποτα εφόσον μόλις τελειώνει η

συνάρτηση, εξαφανίζεται η μεταβλητή \$a.

Για να δημιουργήσουμε μια χρήσιμη συνάρτηση μέτρησης η οποία δεν θα χάνει τον έλεγχο της τρέχουσας μέτρησης, θα πρέπει να δηλώσουμε τη

μεταβλητή \$a σαν static :

```
Function Test () {
    static $a = 0;
    echo $a;
    $a++;
}
```

Τώρα, κάθε φορά που καλείται η συνάρτηση Test(), θα εκτυπώνει την τιμή της \$a και θα την αυξάνει.

Οι στατικές μεταβλητές παρέχουν επίσης έναν τρόπο για να ασχοληθούμε τις αναδρομικές συναρτήσεις. Μια αναδρομική συνάρτηση (recursive function) είναι αυτή που καλεί τον εαυτό της. Η επόμενη συνάρτηση

μετράει αναδρομικά έως το 10 και χρησιμοποιεί τη στατική μεταβλητή \$count

για να ξέρει πότε να σταματήσει :

```
Function Test () {  
static $count = 0;  
$count++;  
echo $count;  
if ($count < 10) {  
Test ();  
}  
$count- -;  
}
```

Μεταβλητές Μεταβλητές

Μερικές φορές είναι βολικό να μπορούμε να έχουμε μεταβλητά ονόματα μεταβλητών (variable variable names), δηλ. ένα όνομα μεταβλητής το

οποίο μπορεί να ορισθεί και να χρησιμοποιηθεί δυναμικά. Όπως γνωρίζουμε,

μια κανονική μεταβλητή ορίζεται με μια εντολή σαν την εξής :

```
$a = "hello";
```

Μια μεταβλητή μεταβλητή αποκτά την τιμή μιας μεταβλητής και την αντιμετωπίζει σαν το όνομα μιας μεταβλητής. Στο παραπάνω παράδειγμα, το

hello, μπορεί να χρησιμοποιηθεί σαν το όνομα μιας μεταβλητής

χρησιμοποιώντας δύο σύμβολα \$, ως εξής :

```
$$a = "world";
```

Σ' αυτό το σημείο έχουμε ορίσει δύο μεταβλητές και τις έχουμε αποθηκεύσει στο συμβολικό δένδρο της PHP : η \$a με περιεχόμενο "hello" και

η \$hello με περιεχόμενο "world". Συνεπώς, η επόμενη εντολή :

```
echo "$a ${$a}";
```

παράγει την ίδια ακριβώς έξοδο με την :

```
echo "$a $hello";
```

Δηλαδή και οι δύο παράγουν το : hello world.

Για να μπορέσουμε να χρησιμοποιήσουμε μεταβλητές μεταβλητές με πίνακες (arrays), θα πρέπει να λύσουμε ένα πρόβλημα ασάφειας. Δηλαδή, αν

γράψουμε \$\$a[1], τότε ο αναλυτής (parser) θα πρέπει να γνωρίζει αν

σκοπεύαμε να χρησιμοποιήσουμε το `$a[1]` σαν μια μεταβλητή ή αν θέλαμε να είναι το `$$a` η μεταβλητή και μετά το `[1]` ο δείκτης (index) απ' αυτή τη μεταβλητή. Η σύνταξη για να επιλύσουμε αυτήν την αμφιβολία είναι : `$$a[1]` για την πρώτη περίπτωση και `$$a}[1]` για τη δεύτερη.

Μεταβλητές Εκτός της PHP

Όταν υποβάλλεται μια φόρμα σ' ένα PHP script, όλες οι μεταβλητές αυτής της φόρμας γίνονται αυτόματα διαθέσιμες στο script από την PHP.

Για

παράδειγμα, ας δούμε την εξής φόρμα :

```
<form action="foo.php3" method="post">  
Όνομα : <input type="text" name="name"><br>  
<input type="submit">  
</form>
```

Όταν υποβληθεί η φόρμα, η PHP θα δημιουργήσει μια μεταβλητή με όνομα `$name`, η οποία θα περιέχει την τιμή που καταχωρήθηκε στο πεδίο `name`

της φόρμας.

Η PHP καταλαβαίνει επίσης τους πίνακες στο περιβάλλον των μεταβλητών φόρμας αλλά μόνο σε μία διάσταση. Μπορούμε, για παράδειγμα,

να ομαδοποιήσουμε σχετικές μεταβλητές μαζί ή να χρησιμοποιήσουμε αυτό το

χαρακτηριστικό για να ανακτήσουμε τιμές από μια λίστα πολλαπλής επιλογής

(multiple select input) :

```
<form action="array.php" method="post">  
Όνομα : <input type="text" name="personal[name]"><br>  
Email : <input type="text" name="personal[email]"><br>  
Μπύρα : <br>  
<select multiple name="beer[]">  
<option value="Fix"> Fix  
<option value="Heineken"> Heineken  
<option value="Pilsen"> Pilsen  
</select>  
<input type="submit">  
</form>
```

Αν το χαρακτηριστικό `track_vars` της PHP είναι ενεργοποιημένο, είτε με τη ρύθμιση σύνθεσης (configuration setting) `track_vars` ή με την οδηγία

(directive) `<?php_track_vars?>`, τότε οι μεταβλητές που υποβάλλονται με τις

μεθόδους POST ή GET θα βρίσκονται επίσης στους global associative πίνακες

(arrays) \$HTTP_POST_VARS και \$HTTP_GET_VARS.

Οι Μεταβλητές Image Submit

Όταν υποβάλλουμε μια φόρμα, μπορούμε να χρησιμοποιήσουμε μια εικόνα (image) αντί για το στάνταρτ πλήκτρο submit, μ' ένα tag σαν το :

```
<input type=image src="image.gif" name="sub">
```

Όταν ο χρήστης κάνει κλικ κάπου πάνω στην εικόνα, η φόρμα θα σταλεί στον server με δύο επιπλέον μεταβλητές, τις sub_x και sub_y, οι οποίες περιέχουν τις συντεταγμένες (coordinates) του κλικ που έκανε ο χρήστης μέσα

στην εικόνα.

Τα HTTP Cookies

Η PHP υποστηρίζει τα HTTP cookies όπως ορίζεται από τις προδιαγραφές της Netscape. Τα cookies είναι ένας μηχανισμός για να αποθηκεύονται δεδομένα στον απομακρυσμένο φυλλομετρητή και έτσι να

μπορούμε να παρακολουθούμε ή να αναγνωρίζουμε τους χρήστες.

Μπορούμε

να ορίσουμε τα cookies με τη συνάρτηση SetCookie().

Τα cookies αποτελούν μέρος του HTTP header, έτσι η συνάρτηση SetCookie() πρέπει να κληθεί πριν σταλεί κάποια έξοδος στον φυλλομετρητή.

Αυτός είναι ο ίδιος περιορισμός που ισχύει και για τη συνάρτηση Header(). Τα

cookies που στέλνονται σε μας από τον client θα μετατραπούν αυτόματα σε

μια μεταβλητή της PHP όπως συμβαίνει με τα δεδομένα των μεθόδων GET και

POST.

Αν θελήσουμε να εκχωρήσουμε πολλαπλές τιμές σ' ένα μόνο cookie, απλά προσθέτουμε τα σύμβολα [] στο όνομα του cookie, ως εξής :

```
SetCookie ("MyCookie[]", "Testing", time()+3600);
```

Ένα cookie θα αντικαταστήσει ένα ήδη υπάρχον με το ίδιο όνομα στον φυλλομετρητή μας εκτός κι αν διαφέρουν η διαδρομή (path) ή το domain.

Έτσι, για μια εφαρμογή shopping cart μπορεί να θέλουμε να έχουμε έναν μετρητή (counter) και να το μεταβιβάσουμε αυτό ως εξής :

```
$Count++;
```

```
SetCookie ("Count", $Count, time()+3600);
```

```
SetCookie ("Cart[$Count]", $item, time()+3600);
```

Οι Μεταβλητές Περιβάλλοντος (Environment)

Η PHP κάνει αυτόματα διαθέσιμες τις μεταβλητές περιβάλλοντος (environment) σαν κανονικές PHP μεταβλητές.

echo \$HOME;

/* Δείχνει τη μεταβλητή environment HOME, αν έχει ορισθεί */

Εφόσον οι πληροφορίες που έρχονται μέσω των μηχανισμών GET, POST και Cookie δημιουργούν επίσης αυτόματα μεταβλητής PHP, είναι μερικές φορές καλύτερο να διαβάσουμε ρητά μια μεταβλητή από το environment για να είμαστε σίγουροι ότι λαμβάνουμε τη σωστή έκδοση. Η συνάρτηση getenv() μπορεί να χρησιμοποιηθεί γι' αυτόν τον σκοπό. Μπορούμε επίσης να ορίσουμε μια μεταβλητή environment με τη συνάρτηση putenv().

Οι Τελείες (Dots) στις Εισερχόμενες

Μεταβλητές

Τυπικά, η PHP δεν αλλάζει τα ονόματα των μεταβλητών όταν αυτά μεταβιβάζονται σ' ένα script. Όμως, θα πρέπει να σημειωθεί ότι η τελεία (dot)

δεν αποτελεί έναν έγκυρο χαρακτήρα σ' ένα όνομα μεταβλητής της PHP, ως εξής :

\$varname.ext; /* μη έγκυρο όνομα μεταβλητής */

Τώρα, αυτό που βλέπει ο αναλυτής (parser) είναι μια μεταβλητή με όνομα \$varname ακολουθούμενη από τον τελεστή ένωσης string και από το ext

που δεν ταιριάζει με κάποια γνωστή λέξη κλειδί.

Γι' αυτόν τον λόγο, είναι σημαντικό να σημειώσουμε ότι η PHP αντικαθιστά αυτόματα τις τελείες στα εισερχόμενα ονόματα μεταβλητών με

τον χαρακτήρα _ (underscore).

Καθορισμός των Τύπων Μεταβλητών

Επειδή η PHP καθορίζει τους τύπους των μεταβλητών και τους μετατρέπει όπως χρειάζεται, δεν είναι πάντα σίγουρο τι τύπο δεδομένων έχει

μια δεδομένη μεταβλητή σε κάποια δεδομένη χρονική στιγμή.

Η PHP περιέχει αρκετές συναρτήσεις που μπορούμε να χρησιμοποιήσουμε για να βρούμε τον τύπο δεδομένων μιας μεταβλητής. Αυτές

είναι οι gettype(), is_long(), is_double(), is_string(), is_array() και is_object().

2.8 Οι Σταθερές (Constants)

Η PHP ορίζει αρκετές σταθερές (constants) και παρέχει έναν μηχανισμό για να ορίσουμε περισσότερες κατά την ώρα εκτέλεσης (run-time). Οι σταθερές είναι σαν τις μεταβλητές, εκτός από το ότι πρέπει να ορισθούν με τη

συνάρτηση define() και ότι δεν μπορούν να ξαναορισθούν αργότερα σε μια άλλη τιμή.

Οι προκαθορισμένες σταθερές, οι οποίες είναι πάντα διαθέσιμες, είναι οι εξής :

__FILE__

Το όνομα του αρχείου script που αναλύεται (parsed) αυτή τη στιγμή. Αν χρησιμοποιείται μέσα σ' ένα αρχείο το οποίο έχει συμπεριληφθεί (included),

τότε δίνεται το όνομα του αρχείου που έχει συμπεριληφθεί και όχι το όνομα

του πατρικού αρχείου (parent file).

__LINE__

Ο αριθμός της γραμμής μέσα στο τρέχον αρχείο script που αναλύεται (parsed). Αν χρησιμοποιείται μέσα σ' ένα αρχείο το οποίο έχει συμπεριληφθεί

(included), τότε δίνεται η θέση μέσα στο αρχείο που έχει συμπεριληφθεί.

PHP_VERSION

Η αναπαράσταση σε string της έκδοσης (version) του PHP parser που χρησιμοποιείται εκείνη τη στιγμή, όπως π.χ. '3.0.8-dev'.

PHP_OS

Το όνομα του λειτουργικού συστήματος στο οποίο εκτελείται ο PHP parser, όπως π.χ. 'Linux'.

TRUE

Μια τιμή true.

FALSE

Μια τιμή false.

E_ERROR

Δηλώνει ένα λάθος (error) διαφορετικό από ένα parsing error από το οποίο είναι αδύνατη η ανάκαμψη (recovery).

E_WARNING

Δηλώνει μια συνθήκη όπου η PHP γνωρίζει ότι κάτι είναι λάθος αλλά συνεχίζει έτσι κι αλλιώς. Ένα παράδειγμα μπορεί να είναι μια μη έγκυρη κανονική έκφραση στην ereg().

E_PARSE

Ο parser σταμάτησε λόγω μη έγκυρης σύνταξης στο αρχείο script και η ανάκαμψη (recovery) είναι αδύνατη.

E_NOTICE

Κάτι συνέβη που μπορεί να είναι ένα λάθος (error) ή όχι. Η εκτέλεση συνεχίζεται. Τέτοια παραδείγματα μπορεί να είναι η χρήση ενός string χωρίς

εισαγωγικά σαν hash index ή η χρήση μιας μεταβλητής που δεν έχει ορισθεί

ακόμη.

Οι σταθερές του τύπου `E_*` χρησιμοποιούνται συνήθως με τη συνάρτηση `error_reporting()` για να ορίσουμε το επίπεδο αναφοράς λάθους (error reporting level). Μπορούμε να ορίσουμε επιπλέον μεταβλητές με τη συνάρτηση `define()`.

Πρέπει να έχουμε υπόψη μας ότι αυτές είναι μεταβλητές και όχι μακροεντολές της C (C-style macros) και συνεπώς μόνο έγκυρα αριθμητικά δεδομένα (scalar data) μπορούν να παριστάνονται από μια μεταβλητή. Ακολουθεί ένα παράδειγμα με ορισμό σταθεράς.

```
<?php
define("CONSTANT", "Hello world.");
echo CONSTANT; // εμφανίζει το "Hello world."
?>
```

Ακολουθεί ένα παράδειγμα με χρήση των προκαθορισμένων σταθερών `__FILE__` και `__LINE__`.

```
<?php
function report_error($file, $line, $message) {
echo "An error occurred in $file on line $line : $message.";
}
report_error(__FILE__, __LINE__, "Something went wrong!");
?>
```

2.9 Οι Εκφράσεις (Expressions)

Οι εκφράσεις (expressions) είναι από τους σημαντικότερους θεμέλιους λίθους της PHP. Στην PHP, σχεδόν ο,τιδήποτε γράφουμε αποτελεί μια έκφραση. Σαν έναν πολύ απλοϊκό αλλά ακριβή ορισμό για μια έκφραση μπορούμε να πούμε ότι «είναι ο,τιδήποτε έχει μια τιμή».

Οι βασικότερες μορφές εκφράσεων είναι οι σταθερές (constants) και οι μεταβλητές (variables). Για παράδειγμα, όταν γράφουμε `$a = 5`, εκχωρούμε το

5 στο `$a`. Το 5 είναι μια έκφραση (expression) με την τιμή 5 και σ' αυτήν την

περίπτωση το 5 είναι μια ακέραια σταθερά.

Μετά απ' αυτήν την εκχώρηση, θα αναμέναμε η τιμή της `$a` να είναι ίση με 5, έτσι αν γράψουμε `$b = $a`, θα περιμέναμε να συμπεριφερθεί σαν να είχαμε γράψει `$b = 5`. Μ' άλλα λόγια, το `$a` είναι μια έκφραση (expression) με

την τιμή 5 επίσης.

Λίγο περισσότερα σύνθετα παραδείγματα για τις εκφράσεις είναι οι συναρτήσεις (functions), όπως για παράδειγμα η ακόλουθη συνάρτηση :

```
function foo () {  
return 5;  
}
```

Οι συναρτήσεις είναι εκφράσεις με την τιμή της τιμής επιστροφής τους (return value). Έτσι, εφόσον η συνάρτηση foo() επιστρέφει το 5, η τιμή της

έκφρασης foo() είναι 5. Συνήθως, βέβαια οι συναρτήσεις δεν επιστρέφουν

απλά μια στατική τιμή αλλά κάνουν και υπολογισμούς.

Η PHP υποστηρίζει τους εξής τρεις τύπους τιμών scalar : ακέραιες τιμές (integer), τιμές κινητής υποδιαστολής (floating point) και αλφαριθμητικές τιμές

(string) values. Οι τιμές scalar είναι τιμές που δεν μπορούμε να διασπάσουμε

σε μικρότερα κομμάτια, σ' αντίθεση με τους πίνακες (arrays), για παράδειγμα.

Η PHP υποστηρίζει επίσης δύο σύνθετους (composite, non-scalar) τύπους : τους πίνακες (arrays) και τα αντικείμενα (objects). Ο καθένας απ'

αυτούς τους τύπους τιμών μπορεί να εκχωρηθεί σε μεταβλητές ή να επιστραφεί

από συναρτήσεις.

Η PHP αντιμετωπίζει τις εκφράσεις με τον ίδιο τρόπο που τις αντιμετωπίζουν πολλές από τις υπόλοιπες γλώσσες. Η PHP είναι μια προσανατολισμένη σε εκφράσεις (expression-oriented) γλώσσα, με την έννοια

ότι σχεδόν ο,τιδήποτε αποτελεί μια έκφραση.

Στο παράδειγμα που είδαμε προηγουμένως, το \$a = 5 είναι μια έκφραση που έχει την τιμή 5. Έτσι, το να γράψουμε κάτι σαν το \$b = (\$a = 5)

είναι το ίδιο με το να γράψουμε \$a = 5; \$b = 5;. Εφόσον οι εκχωρήσεις γίνονται από δεξιά προς τα αριστερά, μπορούμε να γράψουμε επίσης \$b = \$a = 5.

Ένα άλλο καλό παράδειγμα είναι οι pre- και post- increment and decrement. Οι χρήστες της PHP/FI 2 και πολλών άλλων γλωσσών ίσως να

είναι εξοικειωμένοι με την γραφή variable++ και variable--. Πρόκειται για τους

τελεστές αύξησης (increment) και μείωσης (decrement).

Στην PHP/FI 2, η εντολή \$a++ δεν έχει καμία τιμή καθώς δεν αποτελεί έκφραση και έτσι δεν μπορούμε να την εκχωρήσουμε ή να την

χρησιμοποιήσουμε κατά κάποιον τρόπο. Η PHP βελτιώνει τις δυνατότητες για

αύξηση/μείωση (increment/decrement) περιλαμβάνοντας κι αυτές τις εκφράσεις επίσης, όπως στην C.

Το pre-increment, που γράφεται ως ++\$variable είναι ίσο με την τιμή της μεταβλητής αυξημένης κατά ένα, δηλ. η PHP αυξάνει την τιμή της μεταβλητής πριν χρησιμοποιήσει την τιμή της.

Το post-increment, που γράφεται ως \$variable++ είναι ίσο με την αρχική τιμή της \$variable, πριν αυτή αυξηθεί, δηλ. η PHP αυξάνει την μεταβλητή αφού έχει χρησιμοποιήσει την τιμή της.

Ένας πολύ κοινός τύπος εκφράσεων είναι οι εκφράσεις σύγκρισης (comparison expressions), οι οποίες αποτιμούνται σε 0 ή 1, που σημαίνουν

FALSE ή TRUE αντίστοιχα.

Η PHP υποστηρίζει τους τελεστές σύγκρισης >, >=, ==, !=, < και <=.

Αυτές οι εκφράσεις χρησιμοποιούνται συνήθως μέσα σε εκτελέσεις υπό συνθήκη (conditional execution), όπως είναι οι εντολές if.

Το τελευταίο παράδειγμα εκφράσεων που θα δούμε είναι οι συνδυασμένες εκφράσεις τελεστή και εκχώρησης. Γνωρίζουμε ήδη ότι αν θέλουμε να αυξήσουμε το \$a κατά 1, μπορούμε να γράψουμε απλά \$a++ ή

++\$a.

Αλλά, αν θέλουμε να προσθέσουμε περισσότερο από το 1, όπως για παράδειγμα το 3; Θα μπορούσαμε να γράψουμε το \$a++ πολλές φορές, αλλά

αυτό δεν είναι προφανώς κάτι αποδοτικό.

Η πρόσθεση του 3 στην τρέχουσα τιμή του \$a μπορεί να γραφεί σαν \$a += 3, που σημαίνει «πάρε την τιμή της \$a, πρόσθεσε το 3 σ' αυτήν και εκχώρησε το αποτέλεσμα πίσω ξανά στην \$a». Υπάρχουν και οι τελεστές \$a -=

5, δηλ. αφαίρεση του 5 από την τιμή της \$a, \$b *= 7, δηλ.

πολλαπλασιασμός

της τιμής της \$b με το 7 κ.ά.

Υπάρχει μια ακόμα περίεργη έκφραση, ο τριαδικός τελεστής υπό συνθήκη (ternary conditional operator) :

\$first ? \$second : \$third

Αν η τιμή της πρώτης υποέκφρασης είναι true, δηλ. όχι μηδενική, τότε αποτιμάται η δεύτερη υποέκφραση και αυτό είναι και το αποτέλεσμα της έκφρασης υπό συνθήκη (conditional expression). Αλλιώς, αποτιμάται η τρίτη

υποέκφραση και αυτή είναι η τιμή.

Το ακόλουθο παράδειγμα αναφέρεται στα παραπάνω :

```
function double($i) {
```

```

return $i*2;
}
$b = $a = 5; /* εκχωρείται η τιμή 5 στις μεταβλητές $a και $b */
$c = $a++;
/* post-increment, εκχωρείται η αρχική τιμή της $a (5) στην $c */
$e = $d = ++$b;
/* pre-increment, εκχωρείται η αυξημένη τιμή της $b (6) στις $d και $e
και σ' αυτό το σημείο, οι $d και $e είναι ίσες με 6 */
$f = double($d++);
/* εκχωρείται η διπλή τιμή της $d πριν από την αύξησή της, δηλ. 2*6 =
12 στην $f */
$g = double(++$e);
/* εκχωρείται η διπλή τιμή της $e μετά από την αύξησή της, δηλ. 2*7 =
14 στην $g */
$h = $g += 10;
/* πρώτα, η $g αυξάνεται κατά 10 και τελειώνει με την τιμή 24, η τιμή
της εκχώρησης (24) εκχωρείται μετά στην $h και η $h τελειώνει με την
τιμή 24 επίσης */

```

Η PHP δεν διαθέτει έναν λογικό τύπο δεδομένων (boolean) και η αληθής τιμή των εκφράσεων στην PHP υπολογίζεται με παρόμοιο τρόπο με

την Perl. Αυτό σημαίνει ότι μια μη μηδενική αριθμητική τιμή είναι TRUE και το 0 είναι FALSE.

Οι αρνητικές τιμές δεν είναι ίσες με μηδέν και έτσι θεωρούνται TRUE. Το άδειο string και το string "0" είναι FALSE και όλα τα άλλα strings είναι

TRUE. Με τις μη scalar τιμές (πίνακες και αντικείμενα), αν μια τιμή δεν περιέχει κάποιο στοιχείο θεωρείται FALSE, αλλιώς θεωρείται TRUE.

2.10 Οι Αριθμητικοί Τελεστές

Οι αριθμητικοί τελεστές (arithmetic operators) της PHP είναι οι εξής :

Παράδειγμα Όνομα Αποτέλεσμα

$\$a + \b Πρόσθεση Άθροισμα των $\$a$ και $\$b$

$\$a - \b Αφαίρεση Διαφορά των $\$a$ και $\$b$

$\$a * \b Πολλαπλασιασμός Γινόμενο των $\$a$ και $\$b$

$\$a / \b Διαίρεση Πηλίκο των $\$a$ και $\$b$

$\$a \% \b Ακέραιο υπόλοιπο

(modulus)

Ακέραιο υπόλοιπο του $\$a$ διαιρούμενο με το $\$b$

Οι Τελεστές Εκχώρησης

Ο βασικός τελεστής εκχώρησης (assignment operator) είναι το =.

Σημαίνει ότι ο αριστερός τελεστής γίνεται ίσος με την τιμή της έκφρασης που

υπάρχει στα δεξιά.

Η τιμή μιας έκφρασης εκχώρησης είναι η τιμή που εκχωρείται, δηλ. η τιμή της έκφρασης $\$a = 3$ είναι το 3. Αυτό μας δίνει τη δυνατότητα να κάνουμε

μερικά έξυπνα κόλπα :

```
 $\$a = (\$b = 4) + 5;$  // το  $\$a$  γίνεται ίσο με 9 και το  $\$b$  με 4
```

Εκτός από τον βασικό τελεστή εκχώρησης, υπάρχουν «συνδυασμένοι τελεστές» γι' όλους τους δυαδικούς αριθμητικούς και αλφαριθμητικούς τελεστές οι οποίοι μας δίνουν τη δυνατότητα να χρησιμοποιήσουμε μια τιμή σε

μια έκφραση και μετά να ορίσουμε την τιμή της με το αποτέλεσμα αυτής της

έκφρασης. Για παράδειγμα :

```
 $\$a = 3;$ 
```

```
 $\$a += 5;$  // κάνει το  $\$a$  ίσο με 8 σαν  $\$a = \$a + 5;$ 
```

```
 $\$b = "Hello ";$ 
```

```
 $\$b .= "There!";$ 
```

```
// κάνει το  $\$b$  ίσο με "Hello There!" σαν  $\$b = \$b . "There!";$ 
```

Η παραπάνω εκχώρηση αντιγράφει την αρχική μεταβλητή στην καινούργια, που αποκαλείται εκχώρηση με τιμή (assignment by value) και έτσι

οι αλλαγές που θα συμβούν στη μια απ' αυτές δεν θα επηρεάσουν και την άλλη.

Η PHP4 υποστηρίζει την εκχώρηση με αναφορά (assignment by reference), χρησιμοποιώντας την σύνταξη $\$var = \&\$othervar;$, κάτι που δεν

ισχύει στην PHP3. Η εκχώρηση με αναφορά σημαίνει ότι και οι δύο μεταβλητές δείχνουν στα ίδια δεδομένα και τίποτα δεν αντιγράφεται.

Οι Τελεστές Δυαδικών Πράξεων

Οι τελεστές δυαδικών πράξεων (bitwise operators) μάς δίνουν τη δυνατότητα να αλλάξουμε την τιμή συγκεκριμένων δυαδικών ψηφίων (bits)

μέσα σ' έναν ακέραιο.

Παράδειγμα Όνομα Αποτέλεσμα

$\$a \& \b And Επιστρέφει 1 αν τα αντίστοιχα bits είναι 1 και στην $\$a$ και στην $\$a | \b Or Επιστρέφει 1 αν ένα από τα αντίστοιχα bits είναι ίσα με 1 στην $\$a$ $\$a \hat{=} \b Xor Επιστρέφει 1 αν ένα από τα αντίστοιχα bits είναι ίσα με 1 στην $\$a$ ταυτόχρονα

$\sim \$a$ Not Επιστρέφει 1 αν το αντίστοιχο bit του $\$a$ είναι 0, αλλιώς επιστρέφει $\$a \ll \b Shift left Μετακινεί τα bits του $\$a$ κατά $\$b$ βήματα προς τα αριστερά, πολλαπλασιασμός επί 2

`$a >> $b` Shift right Μετακινεί τα bits του `$a` κατά `$b` βήματα προς τα δεξιά, όπου το 2

Οι Τελεστές Σύγκρισης

Οι τελεστές σύγκρισης (comparison operators) μάς δίνουν τη δυνατότητα να συγκρίνουμε δύο τιμές.

Παράδειγμα Όνομα Αποτέλεσμα

`$a == $b` Ίσο True αν το `$a` είναι ίσο με το `$b`

`$a === $b` Ακριβώς ίδιο True αν τα `$a` είναι ίσο με `$b` και είναι του ίδιου `$a != $b` Όχι ίσο True αν το `$a` δεν είναι ίσο με το `$b`

`$a < $b` Μικρότερο από True αν το `$a` είναι μικρότερο από το `$b`

`$a > $b` Μεγαλύτερο από True αν το `$a` είναι μεγαλύτερο από το `$b`

`$a <= $b` Μικρότερο από ή ίσο με True αν το `$a` είναι μικρότερο ή ίσο από το `$b`

`$a >= $b` Μεγαλύτερο από ή ίσο με True αν το `$a` είναι μεγαλύτερο ή ίσο από το `$b`

Ένας άλλος τελεστής υπό συνθήκη (conditional operator) είναι ο `?:` ή τριαδικός (ternary) τελεστής, ο οποίος λειτουργεί όπως στην C και σ' άλλες

γλώσσες, ως εξής :

`(expr1) ? (expr2) : (expr3);`

Η παραπάνω έκφραση αποτιμάται στην `expr2` αν η `expr1` έχει αποτιμηθεί σε true και στην `expr3` αν η `expr1` έχει αποτιμηθεί σε false.

Οι Τελεστές Εκτέλεσης

Η PHP υποστηρίζει έναν τελεστή εκτέλεσης (execution operator), τον backticks (```). Η PHP θα προσπαθήσει να εκτελέσει τα περιεχόμενα των backticks σαν μια εντολή shell. Η έξοδος μπορεί να ανατεθεί σε μια μεταβλητή.

```
$output = `ls -al`;
```

```
echo "<pre>$output</pre>";
```

Μπορούμε να δούμε επίσης και τις συναρτήσεις `system()`, `passthru()`, `exec()`, `popen()` και `escapeshellcmd()`.

Οι Τελεστές Αύξησης/Μείωσης

Η PHP υποστηρίζει τους τελεστές αύξησης και μείωσης που θυμίζουν την C (C-style pre- and post-increment and decrement operators).

Παράδειγμα Όνομα Αποτέλεσμα

`++$a` Pre-increment Αυξάνει το `$a` κατά ένα και μετά το επιστρέφει

`$a++` Post-increment Επιστρέφει το `$a` και μετά το αυξάνει κατά ένα

`--$a` Pre-decrement Μειώνει το `$a` κατά ένα και μετά το επιστρέφει

`$a--` Post-decrement Επιστρέφει το `$a` και μετά το μειώνει κατά ένα

Ακολουθεί ένα απλό παράδειγμα script :

```
<?php
```

```
echo "<h3> Postincrement </h3>";
```

```
$a = 5;
```

```

echo "Πρέπει να είναι 5 : " . $a++ . "<br>\n";
echo "Πρέπει να είναι 6 : " . $a . "<br>\n";
echo "<h3> Preincrement </h3>";
$a = 5;
echo "Πρέπει να είναι 6 : " . ++$a . "<br>\n";
echo "Πρέπει να είναι 6 : " . $a . "<br>\n";
echo "<h3> Postdecrement </h3>";
$a = 5;
echo "Πρέπει να είναι 5 : " . $a-- . "<br>\n";
echo "Πρέπει να είναι 4 : " . $a . "<br>\n";
echo "<h3> Predecrement </h3>";
$a = 5;
echo "Πρέπει να είναι 4 : " . --$a . "<br>\n";
echo "Πρέπει να είναι 4 : " . $a . "<br>\n";
?>

```

Οι Λογικοί Τελεστές

Οι λογικοί τελεστές (logical operators) της PHP είναι οι εξής :

Παράδειγμα Όνομα Αποτέλεσμα

`$a and $b And True` αν και το `$a` και το `$b` είναι true

`$a or $b Or True` αν ένα από τα `$a` ή `$b` είναι true

`$a xor $b Xor True` αν ένα από τα `$a` ή `$b` είναι true αλλά όχι και τα δύο

`!$a Not True` αν το `$a` δεν είναι true

`$a && $b And True` αν και το `$a` και το `$b` είναι true

`$a || $b Or True` αν ένα από τα `$a` ή `$b` είναι true

Ο λόγος που υπάρχουν δύο διαφορετικές παραλλαγές των τελεστών `and` και `or` είναι ότι λειτουργούν με διαφορετικές προτεραιότητες.

Οι Τελεστές των Αλφαριθμητικών (Strings)

Υπάρχουν δύο τελεστές για τα αλφαριθμητικά (strings). Ο πρώτος είναι ο τελεστής συνένωσης (concatenation operator), '.', ο οποίος επιστρέφει την

ένωση του δεξιού και του αριστερού του ορίσματος. Ο δεύτερος είναι ο τελεστής εκχώρησης συνένωσης (concatenating assignment operator), '.=',

```
$a = "Hello ";
```

```
$b = $a . "World!"; // το $b περιέχει το "Hello World!"
```

```
$a = "Hello ";
```

```
$a .= "World!"; // το $a περιέχει το "Hello World!"
```

2.11 Οι Δομές Ελέγχου (Control Structures)

Ένα script της PHP αποτελείται από μια σειρά εντολών, όπου μια εντολή μπορεί να είναι μια εκχώρηση, μια κλήση συνάρτησης, ένας βρόχος,

μια εντολή υπό συνθήκη ή ακόμη και μια εντολή που δεν κάνει τίποτα (μια κενή, empty, εντολή). Οι εντολές τελειώνουν συνήθως με τον χαρακτήρα ; (semicolon).

Επιπλέον, οι εντολές μπορούν να ομαδοποιηθούν σε μια εντολή-ομάδα (statement-group) αν περικλείσουμε μια ομάδα εντολών με άγκιστρα { και }.

Μια εντολή-ομάδα αποτελεί και η ίδια μια εντολή.

Η Εντολή If

Η σύνταξη της εντολής if στην PHP είναι παρόμοια μ' αυτήν της C :

if (έκφραση)

... εντολή ...

Αν η έκφραση αποτιμηθεί σε TRUE, η PHP θα εκτελέσει την εντολή, ενώ αν αποτιμηθεί σε FALSE, θα την αγνοήσει.

Το επόμενο παράδειγμα θα εμφανίσει ένα μήνυμα αν το \$a είναι μεγαλύτερο από το \$b :

```
if ($a > $b)
```

```
print "Το a είναι μεγαλύτερο από το b";
```

Μπορούμε να ομαδοποιήσουμε περισσότερες από μία εντολές, ως εξής :

```
if ($a > $b) {
```

```
print "Το a είναι μεγαλύτερο από το b";
```

```
$b = $a;
```

```
}
```

Η Εντολή Else

Η εντολή else επεκτείνει μια εντολή if για να εκτελέσει μια εντολή στην περίπτωση που η έκφραση στην εντολή if αποτιμηθεί σε FALSE. Για παράδειγμα, ο ακόλουθος κώδικας εμφανίζει ένα ανάλογο μήνυμα :

```
if ($a > $b) {
```

```
print "Το a είναι μεγαλύτερο από το b";
```

```
} else {
```

```
print "Το a ΔΕΝ είναι μεγαλύτερο από το b";
```

```
}
```

Η Εντολή Elseif

Η εντολή elseif είναι ένας συνδυασμός των εντολών if και else.

Επεκτείνει μια εντολή if για να εκτελέσει μια διαφορετική εντολή στην περίπτωση που η έκφραση της εντολής if αποτιμηθεί σε FALSE, αλλά θα εκτελέσει αυτήν την εναλλακτική έκφραση μόνο αν η συνθήκη έκφρασης της

elseif αποτιμηθεί σε TRUE.

Για παράδειγμα, ο επόμενος κώδικας ελέγχει τρεις περιπτώσεις και θα εμφανίσει ένα ανάλογο μήνυμα αν το a είναι μεγαλύτερο, ίσο ή μικρότερο από

```

το b :
if ($a > $b) {
print "Το a είναι μεγαλύτερο από το b";
} elseif ($a == $b) {
print "Το a είναι ίσο με το b";
} else {
print "Το a είναι μικρότερο από το b";
}

```

Ένας Άλλος Τρόπος Σύνταξης των Δομών Ελέγχου

Η PHP προσφέρει μια εναλλακτική σύνταξη για μερικές από τις δομές ελέγχου της (control structures) και πιο συγκεκριμένα τις if, while, for και switch.

Η βασική μορφή της εναλλακτικής σύνταξης είναι να αλλάξει την αγκύλη ανοίγματος στον χαρακτήρα : και την αγκύλη κλεισίματος στο endif;, endwhile;, endfor; ή endswitch; αντίστοιχα.

```
<?php if ($a==5): ?>
```

```
Το a είναι ίσο με 5
```

```
<?php endif; ?>
```

Στο παραπάνω παράδειγμα, ένα μπλοκ της HTML βρίσκεται μέσα σε μια εντολή if που είναι γραμμένη με την εναλλακτική σύνταξη. Το μπλοκ της

HTML θα εμφανισθεί μόνο αν το \$a είναι ίσο με 5.

Η εναλλακτική σύνταξη ισχύει και για τις εντολές else και elseif επίσης.

Ακολουθεί μια δομή if με elseif και else στην εναλλακτική σύνταξη :

```
if ($a == 5):
```

```
print "Το a είναι ίσο με 5";
```

```
print "...";
```

```
elseif ($a == 6):
```

```
print "Το a είναι ίσο με 6";
```

```
print "!!!";
```

```
else:
```

```
print "Το a δεν είναι ούτε 5 ούτε 6";
```

```
endif;
```

Η Εντολή While

Οι βρόχοι while αποτελούν τον απλούστερο τύπο βρόχου στην PHP και συμπεριφέρονται όπως ακριβώς οι αντίστοιχοι βρόχοι στην C. Η βασική μορφή μιας εντολής while είναι η εξής :

while (έκφραση) ... εντολή ...

Η εντολή while λέει στην PHP να εκτελεί συνέχεια την ή τις εντολές για

όσο διάστημα η έκφραση της while αποτιμάται σε TRUE. Η τιμή της έκφρασης

ελέγχεται κάθε φορά στην αρχή του βρόχου.

Όπως και με την εντολή if, μπορούμε να ομαδοποιήσουμε πολλές εντολές μέσα στον ίδιο βρόχο while χρησιμοποιώντας τα { και } ή την εναλλακτική σύνταξη :

while (έκφραση): ... εντολή ... **endwhile**;

Τα δύο επόμενα παραδείγματα είναι ολόιδια και εμφανίζουν και τα δύο τους αριθμούς 1 έως 10 :

```
$i = 1;
```

```
while ($i <= 10) {  
  print $i++;  
}
```

```
$i = 1;
```

```
while ($i <= 10):
```

```
  print $i;
```

```
  $i++;
```

```
endwhile;
```

Η Εντολή Do .. While

Οι βρόχοι do .. while είναι πολύ παρόμοιοι με τους βρόχους while, εκτός από το ότι η έκφραση ελέγχεται στο τέλος κάθε επανάληψης και όχι

στην αρχή. Η βασική διαφορά τους από τους βρόχους while είναι ότι η πρώτη

επανάληψη ενός βρόχου do .. while θα εκτελεσθεί σίγουρα τουλάχιστον μία

φορά.

Υπάρχει μία μόνο σύνταξη για τον βρόχο do .. while :

```
$i = 0;
```

```
do {
```

```
  print $i;
```

```
} while ($i>0);
```

Ο παραπάνω βρόχος θα εκτελεσθεί μία φορά ακριβώς, εφόσον μετά από την πρώτη επανάληψη, όταν ελέγχεται η έκφραση, αποτιμάται σε FALSE και

έτσι τερματίζεται η εκτέλεση του βρόχου.

Μπορούμε να χρησιμοποιήσουμε την εντολή break για να

σταματήσουμε την εκτέλεση ενός βρόχου στη μέση του κώδικα :

```
do {
```

```
  if ($i < 5) {
```

```
    print "Το i δεν είναι αρκετά μεγάλο";
```

```
    break;
```

```
  }
```

```

$i *= $factor;
if ($i < $minimum_limit) {
break;
}
print "Το i είναι OK";
... process I ...
} while(0);

```

Η Εντολή For

Οι βρόχοι for είναι οι πιο πολύπλοκοι βρόχοι στην PHP.

Συμπεριφέρονται όπως οι αντίστοιχοί τους στην C και η σύνταξη ενός βρόχου

for είναι η εξής :

for (έκφραση1; έκφραση2; έκφραση3) ... εντολή ...

Η πρώτη έκφραση (έκφραση1) αποτιμάται (εκτελείται) μία φορά, χωρίς να υπάρχει κάποια συνθήκη, στην αρχή του βρόχου.

Στην αρχή της κάθε επανάληψης αποτιμάται η έκφραση2 και αν αποτιμηθεί σε TRUE, ο βρόχος συνεχίζεται και εκτελούνται οι περιεχόμενες

εντολές. Αν αποτιμηθεί σε FALSE, σταματάει η εκτέλεση του βρόχου.

Στο

τέλος της κάθε επανάληψης αποτιμάται (εκτελείται) η έκφραση3.

Και οι τρεις εκφράσεις μπορούν να είναι κενές (empty). Αν είναι κενή η έκφραση2, αυτό σημαίνει ότι ο βρόχος θα εκτελείται ασταμάτητα. Αυτό είναι

χρήσιμο όταν θέλουμε να βγούμε από τον βρόχο χρησιμοποιώντας μια εντολή

break.

Και τα τέσσερα επόμενα παραδείγματα εμφανίζουν τους αριθμούς από 1 έως 10 :

/ παράδειγμα 1 */*

```

for ($i = 1; $i <= 10; $i++) {
print $i;
}

```

/ παράδειγμα 2 */*

```

for ($i = 1;;$i++) {
if ($i > 10) {
break;
}
print $i;
}

```

/ παράδειγμα 3 */*

```

$i = 1;
for (;) {

```

```

if ($i > 10) {
break;
}
print $i;
$i++;
}

```

/ παράδειγμα 4 */*

```

for ($i = 1; $i <= 10; print $i, $i++) ;

```

Η PHP υποστηρίζει επίσης και την εναλλακτική σύνταξη για τους βρόχους for :

for (έκφραση1; έκφραση2; έκφραση3): εντολή; ...; **endfor**;

Ενώ άλλες γλώσσες χρησιμοποιούν την εντολή foreach για να διασχίσουν έναν πίνακα (array) ή ένα hash, η PHP χρησιμοποιεί την εντολή

while και τις συναρτήσεις list() και each() γι' αυτόν τον σκοπό.

Η Εντολή Break

Με την εντολή break μπορούμε να εξέλθουμε από μια δομή ελέγχου χωρίς να περιμένουμε να ικανοποιηθεί η συνθήκη εξόδου του βρόχου.

```

$i = 0;
while ($i < 10) {
if ($arr[$i] == "stop") {
break;
}
$i++;
}

```

Η Εντολή Continue

Η εντολή continue χρησιμοποιείται σε δομές βρόχου για να συνεχίσει την εκτέλεση του προγράμματος από την αρχή του βρόχου και να αγνοήσει

έτσι τις υπόλοιπες εντολές μέχρι το τέλος του βρόχου.

```

while (list($key, $value) = each($arr)) {
if ($key % 2) { // αγνοεί τους άρτιους αριθμούς
continue;
}
do_something_odd ($value);
}

```

Η Εντολή Switch

Η εντολή switch είναι παρόμοια με μια σειρά εντολών if στην ίδια έκφραση. Υπάρχουν πολλές περιπτώσεις όπου θέλουμε να συγκρίνουμε την

ίδια μεταβλητή ή έκφραση με πολλές διαφορετικές τιμές και να εκτελέσουμε

ένα διαφορετικό κομμάτι κώδικα ανάλογα με την τιμή της μεταβλητής.

Τα δύο επόμενα παραδείγματα είναι δύο διαφορετικοί τρόποι για να γράψουμε το ίδιο πράγμα, όπου το ένα χρησιμοποιεί μια σειρά από εντολές if

και το άλλο χρησιμοποιεί την εντολή switch :

```
if ($i == 0) {  
    print "Το i είναι ίσο με 0";  
}  
if ($i == 1) {  
    print "Το i είναι ίσο με 1";  
}  
if ($i == 2) {  
    print "Το i είναι ίσο με 2";  
}  
switch ($i) {  
    case 0:  
        print "Το i είναι ίσο με 0";  
        break;  
    case 1:  
        print "Το i είναι ίσο με 1";  
        break;  
    case 2:  
        print "Το i είναι ίσο με 2";  
        break;  
}
```

Μόλις η εντολή switch βρει μια εντολή case με μια τιμή που να ταιριάζει με την τιμή της έκφρασης της switch, η PHP αρχίζει να εκτελεί τις

εντολές. Η PHP συνεχίζει να εκτελεί τις εντολές μέχρι το τέλος του μπλοκ της

switch ή μόλις συναντήσει μια εντολή break.

Αν δεν συμπεριλάβουμε μια εντολή break στο τέλος μιας εντολής case, η PHP θα συνεχίσει να εκτελεί τις εντολές και από τις επόμενες εντολές case,

όπως για παράδειγμα :

```
switch ($i) {  
    case 0:  
        print "Το i είναι ίσο με 0";  
    case 1:  
        print "Το i είναι ίσο με 1";  
    case 2:  
        print "Το i είναι ίσο με 2";  
}
```

Εδώ, αν το \$i είναι ίσο με 0, η PHP θα εκτελέσει όλες τις εντολές print,

αν το \$i είναι ίσο με 1, η PHP θα εκτελέσει τις δύο τελευταίες εντολές
print

και μόνο αν το \$i είναι ίσο 2, θα έχουμε την αναμενόμενη συμπεριφορά
και θα

εκτελεσθεί μόνο η τελευταία εντολή print.

Μια εντολή case μπορεί να είναι κενή και έτσι ο έλεγχος να
μεταβιβασθεί στην επόμενη εντολή case.

```
switch ($i) {  
case 0:  
case 1:  
case 2:  
print "Το i είναι μικρότερο από 3 αλλά όχι αρνητικό";  
break;  
case 3:  
print "Το i είναι ίσο με 3";  
}
```

Μια ειδική περίπτωση case αποτελεί το default case, το οποίο ταιριάζει
μ' ο,τιδήποτε δεν ταίριαξε από τις άλλες cases. Για παράδειγμα :

```
switch ($i) {  
case 0:  
print "Το i είναι ίσο με 0";  
break;  
case 1:  
print "Το i είναι ίσο με 1";  
break;  
case 2:  
print "Το i είναι ίσο με 2";  
break;  
default:  
print "Το i δεν είναι ίσο με 0, 1 ή 2";  
}
```

Η εναλλακτική σύνταξη των δομών ελέγχου υποστηρίζεται και στην
εντολή switch.

```
switch ($i):  
case 0:  
print "Το i είναι ίσο με 0";  
break;  
case 1:  
print "Το i είναι ίσο με 1";  
break;  
case 2:  
print "Το i είναι ίσο με 2";  
break;
```

```
default:  
print "Το i δεν είναι ίσο με 0, 1 ή 2";  
endswitch;
```

Η Συνάρτηση require()

Η εντολή `require()` αντικαθιστά τον εαυτό της μ' ένα συγκεκριμένο αρχείο, όπως ακριβώς δουλεύει δηλαδή η εντολή `#include` στην C. Κάτι σημαντικό που πρέπει να έχουμε υπόψη μας είναι ότι όταν ένα αρχείο χρησιμοποιείται σε μια από τις συναρτήσεις `include()` ή `require()`,

η ανάλυση (parsing) ξεφεύγει από τον έλεγχο της PHP, πηγαίνει στον έλεγχο της HTML στην αρχή του αρχείου και επανέρχεται στον έλεγχο της PHP ξανά στο τέλος.

Γι' αυτόν τον λόγο, ο κώδικας που υπάρχει μέσα στο αρχείο και ο οποίος πρέπει να εκτελεσθεί σαν κώδικας της PHP πρέπει να περικλείεται με

έγκυρα tags αρχής και τέλους της PHP.

Η `require()` δεν είναι ουσιαστικά μια συνάρτηση της PHP και αποτελεί περισσότερο μια δομή της γλώσσας. Υπόκειται σε μερικούς διαφορετικούς

ρόλους απ' ό,τι οι συναρτήσεις. Για παράδειγμα, η `require()` δεν υπόκειται σε

δομές ελέγχου (control structures) και ακόμη, δεν επιστρέφει κάποια τιμή.

Σ' αντίθεση με την `include()`, η `require()` θα διαβασθεί πάντα στο αρχείο (target file), ακόμη κι αν η γραμμή στην οποία βρίσκεται δεν εκτελείται ποτέ.

Αν θέλουμε να συμπεριλάβουμε ένα αρχείο υπό συνθήκη, πρέπει να χρησιμοποιήσουμε τη συνάρτηση `include()`.

Η εντολή υπό συνθήκη δεν θα επηρεάσει την `require()`. Όμως, αν η γραμμή στην οποία βρίσκεται η `require()` δεν εκτελεσθεί, δεν θα εκτελεσθεί

ούτε ο κώδικας που υπάρχει στο αρχείο (target file).

Παρόμοια, οι δομές βρόχου δεν επηρεάζουν τη συμπεριφορά της `require()`. Αν και ο κώδικας που περιέχεται στο αρχείο (target file) υπόκειται

ακόμα στον βρόχο, η ίδια η `require()` εκτελείται μία μόνο φορά.

Αυτό σημαίνει ότι δεν μπορούμε να τοποθετήσουμε μια εντολή `require()` μέσα σε μια δομή βρόχου και να αναμένουμε να συμπεριλάβει τα

περιεχόμενα ενός διαφορετικού αρχείου σε κάθε επανάληψη. Για να το κάνουμε αυτό, χρησιμοποιούμε μια εντολή `include()`.


```
require( 'header.inc' );
```

Πρέπει να έχουμε υπόψη μας ότι και η include() και η require() στην ουσία τραβούν τα περιεχόμενα του αρχείου (target file) στο ίδιο το καλών αρχείο script (calling script file) και δεν καλούν το target μέσω HTTP ή με κάτι παρόμοιο.

Έτσι, όποια μεταβλητή έχει ορισθεί στην εμβέλεια στην οποία λαμβάνει χώρα η συμπερίληψη θα είναι διαθέσιμη αυτόματα μέσα στο συμπεριλαμβανόμενο αρχείο, εφόσον έχει γίνει ουσιαστικά ένα μέρος του καλόντος αρχείου.

```
require( "file.inc?varone=1&vartwo=2" ); /* Δεν θα δουλέψει */
```

```
$varone = 1;
```

```
$vartwo = 2;
```

```
require( "file.inc" );
```

```
/* Οι $varone και $vartwo θα είναι διαθέσιμες στο file.inc */
```

Η Συνάρτηση include()

Η εντολή include() περιλαμβάνει και αποτιμά το καθορισμένο αρχείο.

Κάτι σημαντικό που πρέπει να γνωρίζουμε για το πώς λειτουργεί αυτό, είναι

ότι όταν ένα αρχείο γίνεται include() ή require(), η ανάλυση ξεφεύγει από τον

έλεγχο της PHP και πηγαίνει στον έλεγχο της HTML στην αρχή του αρχείου

(target file) και επανέρχεται ξανά στο τέλος.

Γι' αυτόν τον λόγο, ο κώδικας που υπάρχει μέσα στο αρχείο target και ο οποίος πρέπει να εκτελεσθεί σαν κώδικας της PHP πρέπει να περικλείεται με

έγκυρα tags αρχής και τέλους της PHP.

Αυτό συμβαίνει κάθε φορά που συναντάται η εντολή include(), έτσι μπορούμε να χρησιμοποιήσουμε μια εντολή include() μέσα σε μια δομή βρόχου για να συμπεριλάβουμε έναν αριθμό από διαφορετικά αρχεία.

```
$files = array( 'first.inc', 'second.inc', 'third.inc' );
```

```
for ( $i = 0; $i < count( $files ); $i++ ) {
```

```
include $files[ $i ];
```

```
}
```

Η include() διαφέρει από την require() στο ότι η εντολή include αποτιμάται κάθε φορά που συναντάται (και μόνο όταν εκτελείται), ενώ η εντολή require() αντικαθίσταται από το αρχείο required όταν συναντάται για

πρώτη φορά, αν τα περιεχόμενα του αρχείου θα αποτιμηθούν ή όχι. Για

παράδειγμα, αν βρίσκεται μέσα σε μια εντολή if η συνθήκη της οποίας έχει

αποτιμηθεί σε false.

Επειδή η include() αποτελεί μια ειδική δομή γλώσσας, πρέπει να την περικλείσουμε σ' ένα μπλοκ εντολών αν βρίσκεται μέσα σ' ένα μπλοκ

υπό

συνθήκη (conditional block).

```
/* Αυτό είναι ΛΑΘΟΣ και δεν θα δουλέψει όπως αναμένεται */
```

```
if ($condition)
```

```
include($file);
```

```
else
```

```
include($other);
```

```
/* Αυτό είναι ΣΩΣΤΟ */
```

```
if ($condition) {
```

```
include($file);
```

```
} else {
```

```
include($other);
```

```
}
```

Και στην PHP3 και στην PHP4, είναι δυνατό να εκτελέσουμε μια εντολή return μέσα σ' ένα αρχείο που έχει γίνει include(), ώστε να τερματίσουμε την επεξεργασία σ' αυτό το αρχείο και να επιστρέψουμε στο

script που το κάλεσε. Υπάρχουν, όμως, μερικές διαφορές στον τρόπο που δουλεύει αυτό.

Η πρώτη είναι ότι στην PHP3, το return δεν μπορεί να εμφανισθεί μέσα σ' ένα μπλοκ εκτός κι αν είναι ένα function block, όπου το return εφαρμόζεται

σ' εκείνη τη συνάρτηση και όχι σ' ολόκληρο το αρχείο. Στην PHP4, όμως, δεν

υπάρχει αυτός ο περιορισμός.

Επίσης, η PHP4 μάς δίνει τη δυνατότητα να επιστρέψουμε τιμές από τα αρχεία που έχουν γίνει include(). Μπορούμε να αντιμετωπίσουμε την τιμή μιας

κλήσης στην include() όπως και με μια κανονική συνάρτηση. Αυτό δημιουργεί

ένα parse error στην PHP3.

Ακολουθεί ένα παράδειγμα με την include() στην PHP3 και την PHP4. Υποθέτουμε ότι το αρχείο test.inc βρίσκεται στον ίδιο κατάλογο με το κύριο

αρχείο :

```
<?php
```

```
echo "Before the return <br>\n";
```

```
if ( 1 ) {
```

```
return 27;
}
echo "After the return <br>\n";
?>
```

Υποθέτουμε ότι το κύριο αρχείο (main.html) περιέχει τα εξής :

```
<?php
$retval = include( 'test.inc' );
echo "File returned : '$retval'<br>\n";
?>
```

Όταν κληθεί η main.html στην PHP3, θα δημιουργηθεί ένα λάθος ανάλυσης (parse error) στη γραμμή 2 και αυτό γιατί δεν μπορούμε να πάρουμε

την τιμή μιας συνάρτησης include() στην PHP3. Στην PHP4, όμως, το αποτέλεσμα θα είναι :

```
Before the return
File returned : '27'
```

Τώρα, ας υποθέσουμε ότι το main.html έχει τροποποιηθεί ώστε να περιέχει τα εξής :

```
<?php
include( 'test.inc' );
echo "Back in main.html<br>\n";
?>
```

Στην PHP4, η έξοδος θα είναι :

```
Before the return
Back in main.html
```

Όμως, η PHP3 θα δώσει την εξής έξοδο :

```
Before the return
27Back in main.html
```

```
Parse error: parse error in
/home/torben/public_html/phptest/main.html on line 5
```

Το παραπάνω parse error είναι ένα αποτέλεσμα του γεγονότος ότι η εντολή return περικλείεται σ' ένα non-function block μέσα στο αρχείο test.inc.

Αν μετακινηθεί η εντολή return έξω από το μπλοκ, η έξοδος θα είναι η εξής :

```
Before the return
27Back in main.html
```

2.12 Συναρτήσεις Οριζόμενες από τον Χρήστη

Μπορούμε να ορίσουμε μια συνάρτηση (function) χρησιμοποιώντας μια σύνταξη σαν την εξής :

```
function foo ($arg_1, $arg_2, ..., $arg_n) {
echo "Example function.\n";
```

```
return $retval;
}
```

Μέσα σε μια συνάρτηση μπορεί να υπάρχει ένας έγκυρος κώδικας της PHP, ακόμη κι άλλες συναρτήσεις και ορισμοί τάξεων (class definitions).

Στην

PHP3, οι συναρτήσεις πρέπει να ορισθούν πριν γίνει αναφορά σ' αυτές, ενώ

δεν απαιτείται κάτι τέτοιο στην PHP4.

Η PHP δεν υποστηρίζει την υπέρβαση των συναρτήσεων (function overloading), ούτε είναι δυνατό να καταργήσουμε τον ορισμό (undefine)

ή να

ορίσουμε ξανά (redefine) ήδη δηλωμένες συναρτήσεις.

Η PHP3 δεν υποστηρίζει μεταβλητό αριθμό ορισμάτων στις συναρτήσεις αν και υποστηρίζει τα προκαθορισμένα ορίσματα (default arguments). Η PHP4 υποστηρίζει και τα δύο.

Τα Ορίσματα των Συναρτήσεων

Μπορούμε να περάσουμε πληροφορίες σε συναρτήσεις μέσω της λίστας ορισμάτων (argument list), που είναι μια λίστα μεταβλητών ή και σταθερών

χωρισμένων με κόμματα.

Η PHP υποστηρίζει τη μεταβίβαση (πέραςμα) των ορισμάτων (arguments) με τιμή (by value), που είναι το προεπιλεγμένο, τη μεταβίβαση με

αναφορά (by reference) καθώς και τις προκαθορισμένες τιμές ορισμάτων (default argument values).

Οι λίστες ορισμάτων μεταβλητού μήκους (variable-length argument lists) υποστηρίζονται μόνο στην PHP4. Μπορούμε να πετύχουμε ένα παρόμοιο

αποτέλεσμα στην PHP3 μεταβιβάζοντας έναν πίνακα ορισμάτων σε μια συνάρτηση :

```
function takes_array($input) {
echo "$input[0] + $input[1] = ", $input[0]+$input[1];
}
```

Μεταβίβαση Ορισμάτων με Αναφορά

Εξ ορισμού, τα ορίσματα των συναρτήσεων μεταβιβάζονται με τιμή (by value), που σημαίνει ότι αν αλλάξουμε την τιμή του ορίσματος μέσα στη συνάρτηση, δεν αλλάζει και εκτός της συνάρτησης. Αν θέλουμε να μπορεί μια

συνάρτηση να τροποποιεί τα ορίσματά της, πρέπει να τα μεταβιβάσουμε με

αναφορά (by reference).

Αν θέλουμε ένα όρισμα σε μια συνάρτηση να μεταβιβάζεται πάντα με

αναφορά, πρέπει να προσθέσουμε το σύμβολο & (ampersand) πριν από το

όνομα του ορίσματος στον ορισμό της συνάρτησης :

```
function add_some_extra(&$string) {  
$string .= 'and something extra.';  
}
```

```
$str = 'This is a string, ';
```

```
add_some_extra($str);
```

```
echo $str; // εμφανίζει 'This is a string, and something extra.'
```

Αν θέλουμε να περάσουμε μια μεταβλητή με αναφορά σε μια συνάρτηση που δεν το κάνει αυτό εξ ορισμού, θα πρέπει να προσθέσουμε τον

χαρακτήρα & (ampersand) στο όνομα του ορίσματος κατά την κλήση της συνάρτησης :

```
function foo ($bar) {  
$bar .= ' and something extra.';  
}
```

```
$str = 'This is a string, ';
```

```
foo ($str);
```

```
echo $str; // εμφανίζει 'This is a string, '
```

```
foo (&$str);
```

```
echo $str; // εμφανίζει 'This is a string, and something extra.'
```

Προκαθορισμένες Τιμές Ορισμάτων

Μια συνάρτηση μπορεί να ορίσει προκαθορισμένες τιμές του στυλ της C++ για ορίσματα scalar, ως εξής :

```
function makecoffee ($type = "cappuccino") {  
return "Making a cup of $type.\n";  
}
```

```
echo makecoffee ();
```

```
echo makecoffee ("espresso");
```

Η έξοδος από τον παραπάνω κώδικα είναι η εξής :

```
Making a cup of cappuccino.
```

```
Making a cup of espresso.
```

Η προκαθορισμένη τιμή πρέπει να είναι μια σταθερή έκφραση και όχι, για παράδειγμα, μια μεταβλητή ή ένα μέλος μιας τάξης (class member).

Πρέπει να έχουμε υπόψη μας ότι όταν χρησιμοποιούμε προκαθορισμένα ορίσματα σε συναρτήσεις, αυτά θα πρέπει να βρίσκονται στα δεξιά των μη

προκαθορισμένων ορισμάτων, αλλιώς δεν θα έχουμε τα αναμενόμενα αποτελέσματα. Ακολουθεί ένα παράδειγμα :

```
function makeyogurt ($type = "acidophilus", $flavour) {  
return "Making a bowl of $type $flavour.\n";  
}
```

```
echo makeyogurt ("raspberry"); // δεν δουλεύει όπως περιμέναμε
```

Η έξοδος του παραπάνω παραδείγματος είναι η εξής :

```
Warning: Missing argument 2 in call to makeyogurt() in  
/usr/local/etc/httpd/htdocs/php3test/functest.html on line 41
```

```
Making a bowl of raspberrry.
```

Τώρα, συγκρίνετε το παραπάνω με το εξής :

```
function makeyogurt ($flavour, $type = "acidophilus") {  
return "Making a bowl of $type $flavour.\n";  
}
```

```
echo makeyogurt ("raspberry"); // δουλεύει όπως περιμέναμε
```

Η έξοδος αυτού του παραδείγματος είναι η εξής :

```
Making a bowl of acidophilus raspberrry.
```

Επιστρεφόμενες Τιμές Συναρτήσεων

Οι τιμές των συναρτήσεων επιστρέφονται με την προαιρετική εντολή

return. Όλοι οι τύποι δεδομένων μπορούν να επιστραφούν, ανάμεσά τους οι

λίστες (lists) και τα αντικείμενα (objects).

```
function square ($num) {  
return $num * $num;  
}
```

```
echo square (4); // εμφανίζει '16'
```

Δεν μπορούμε να επιστρέψουμε πολλαπλές τιμές από μια συνάρτηση, αλλά μπορούμε να πετύχουμε ένα παρόμοιο αποτέλεσμα αν επιστρέψουμε μια λίστα (list).

```
function small_numbers() {  
return array (0, 1, 2);  
}
```

```
list ($zero, $one, $two) = small_numbers();
```

2.13 Οι Μεταβλητές Συναρτήσεις

Η PHP υποστηρίζει την έννοια των μεταβλητών συναρτήσεων (variable functions). Αυτό σημαίνει ότι αν ένα όνομα μεταβλητής έχει παρενθέσεις, η

PHP θα αναζητήσει μια συνάρτηση που έχει το ίδιο όνομα με την τιμή της

μεταβλητής και θα προσπαθήσει να την εκτελέσει.

```
<?php
```

```
function foo() {  
echo "In foo(<br>\n";  
}
```

```
function bar( $arg = " ) {  
echo "In bar(); argument was '$arg'.<br>\n";
```

```

}
$func = 'foo';
$func();
$func = 'bar';
$func( 'test' );
?>

```

2.14 Τάξεις και Αντικείμενα

Μια τάξη (class) είναι μια συλλογή από μεταβλητές και από συναρτήσεις που εφαρμόζονται σ' αυτές τις μεταβλητές. Για να ορίσουμε μια

τάξη χρησιμοποιούμε την εξής σύνταξη :

```

<?php
class Cart {
var $items; // Τα items στο shopping cart
// Προσθέτουμε $num προϊόντα του $artnr στο cart
function add_item ($artnr, $num) {
$this->items[$artnr] += $num;
}
// Αφαιρούμε $num προϊόντα του $artnr από το cart
function remove_item ($artnr, $num) {
if ($this->items[$artnr] > $num) {
$this->items[$artnr] -= $num;
return true;
} else {
return false;
}
}
}
?>

```

Αυτό ορίζει μια τάξη με όνομα `Cart` η οποία αποτελείται από έναν πίνακα προϊόντων στο cart (καλάθι αγορών) και δύο συναρτήσεις για να μπορούμε να προσθέσουμε και να αφαιρέσουμε προϊόντα από το cart. Οι τάξεις είναι τύποι και μπορούμε να δημιουργήσουμε μια μεταβλητή ενός συγκεκριμένου τύπου με τον τελεστή `new`, ως εξής :

```

$cart = new Cart;
$cart->add_item("10", 1);

```

Ο παραπάνω κώδικας δημιουργεί ένα αντικείμενο με όνομα `$cart` από την τάξη `Cart`. Η συνάρτηση `add_item()` αυτού του αντικειμένου καλείται για

να προσθέσει ένα προϊόν με κωδικό αριθμό 10 στο cart.

Οι τάξεις μπορεί να είναι επεκτάσεις (extensions) άλλων τάξεων. Η προκύπτουσα τάξη έχει όλες τις μεταβλητές και τις συναρτήσεις της βασικής

τάξης και ό,τι προσθέσουμε εμείς. Αυτό μπορούμε να το κάνουμε με τη λέξη

κλειδί `extends`. Δεν υποστηρίζεται η πολλαπλή κληρονομικότητα (multiple inheritance).

```
class Named_Cart extends Cart {  
  var $owner;  
  function set_owner ($name) {  
    $this->owner = $name;  
  }  
}
```

Ο παραπάνω κώδικας ορίζει μια τάξη με όνομα `Named_Cart` που έχει όλες τις μεταβλητές και τις συναρτήσεις της τάξης `Cart` συν μια επιπλέον μεταβλητή με όνομα `$owner` και μια επιπλέον συνάρτηση με όνομα `set_owner()`. Μπορούμε τώρα να ορίσουμε και να μάθουμε τον ιδιοκτήτη (`owner`) ενός `cart`.

```
$ncart = new Named_Cart; // Δημιουργία ενός named  
cart  
$ncart->set_owner ("kris"); // Όνομα του ιδιοκτήτη του  
cart  
print $ncart->owner;
```

// εκτύπωση του ονόματος του ιδιοκτήτη του `cart`

```
$ncart->add_item ("10", 1);
```

// μια συνάρτηση που την έχει κληρονομήσει από το `cart`

Μέσα στις συναρτήσεις μιας τάξης, η μεταβλητή `$this` σημαίνει το ίδιο το αντικείμενο. Μπορούμε να χρησιμοποιήσουμε τη σύνταξη `$this->something`

για να έχουμε πρόσβαση σε μια οποιαδήποτε μεταβλητή ή συνάρτηση με όνομα `something` του τρέχοντος αντικειμένου.

Οι δημιουργοί (`constructors`) είναι συναρτήσεις σε μια τάξη που καλούνται αυτόματα όταν δημιουργούμε ένα νέο στιγμιότυπο (`instance`) μιας

τάξης. Μια συνάρτηση γίνεται δημιουργός (`constructor`) όταν έχει το ίδιο όνομα με την τάξη.

```
class Auto_Cart extends Cart {  
  function Auto_Cart () {  
    $this->add_item ("10", 1);  
  }  
}
```

Ο παραπάνω κώδικας δημιουργεί μια τάξη με όνομα `Auto_Cart` που είναι μια επέκταση της τάξης `Cart` συν έναν δημιουργό (`constructor`) ο οποίος

αρχικοποιεί το cart μ' ένα στοιχείο του προϊόντος που έχει κωδικό αριθμό 10

κάθε φορά που δημιουργείται ένα νέο Auto_Cart με τον τελεστή new. Οι δημιουργοί μπορούν επίσης να λάβουν ορίσματα και αυτά τα ορίσματα μπορεί

να είναι προαιρετικά.

```
class Constructor_Cart extends Cart {  
function Constructor_Cart ($item = "10", $num = 1) {  
$this->add_item ($item, $num);  
}  
}
```

```
// Shop the same old boring stuff.
```

```
$default_cart = new Constructor_Cart;
```

```
// Shop for real ...
```

```
$different_cart = new Constructor_Cart ("20", 17);
```

Δημιουργία Εικόνων Gif

Η PHP δεν περιορίζεται στο να δημιουργεί μόνο μια έξοδο κώδικα της HTML. Μπορεί επίσης να χρησιμοποιηθεί για να δημιουργήσει αρχεία εικόνων GIF ή και ροές (streams) εικόνων GIF. Θα χρειασθεί να μεταγλωττίσουμε την PHP με τη βιβλιοθήκη GD των συναρτήσεων εικόνας

για να μπορέσει να δουλέψει αυτό.

```
<?php  
Header("Content-type: image/gif");  
$string=implode($argv, " ");  
$im = imagecreatefromgif("images/button1.gif");  
$orange = ImageColorAllocate($im, 220, 210, 60);  
$px = (imagesx($im)-7.5*strlen($string))/2;  
ImageString($im, 3, $px, 9, $string, $orange);  
ImageGif($im);  
ImageDestroy($im);  
?>
```

Αυτό το παράδειγμα θα κληθεί από μια σελίδα μ' ένα tag σαν το εξής :

```

```

Το παραπάνω script button.php3 λαμβάνει μετά αυτό το string "text" και το τοποθετεί στην κορυφή μιας εικόνας βάσης που είναι η "images/button1.gif" και εξάγει την εικόνα που δημιουργείται.

Αυτό αποτελεί έναν πολύ βολικό τρόπο για να αποφύγουμε τη σχεδίαση νέων εικόνων για πλήκτρα εντολής κάθε φορά που θέλουμε να αλλάξουμε το

κείμενο ενός πλήκτρου. Με τη μέθοδο αυτή δημιουργούνται δυναμικά.

Επικύρωση (Authentication) του HTTP με την PHP

Η επικύρωση (authentication) του HTTP στην PHP είναι διαθέσιμη μόνο όταν εκτελείται σαν ένα Apache module και δεν είναι συνεπώς διαθέσιμη στο CGI. Σ' ένα script της PHP σ' ένα Apache module, μπορούμε να χρησιμοποιήσουμε τη συνάρτηση Header() για να στείλουμε ένα μήνυμα "Authentication Required" στον φυλλομετρητή του χρήστη (πελάτη) για να

εμφανίσει (pop up) ένα παράθυρο εισόδου Username/Password.

Αφού ο χρήστης έχει καταχωρήσει ένα username και ένα password, θα κληθεί ξανά το URL που περιέχει το PHP script με τις μεταβλητές \$PHP_AUTH_USER, \$PHP_AUTH_PW και \$PHP_AUTH_TYPE που περιέχουν αντίστοιχα το user name, το password και τον τύπο επικύρωσης

(authentication type).

Ένα παράδειγμα script το οποίο θα επέβαλε την επικύρωση του πελάτη (client authentication) σε μια σελίδα, είναι το εξής :

```
<?php
if(!isset($PHP_AUTH_USER)) {
Header("WWW-Authenticate : Basic realm=\"My Realm\"");
Header("HTTP/1.0 401 Unauthorized");
echo "Κείμενο που θα σταλεί αν πατηθεί το Cancel\n";
exit;
} else {
echo "Hello $PHP_AUTH_USER <P>";
echo "Καταχωρήσατε το $PHP_AUTH_PW για password <P>";
}
?>
```

Αντί να εκτυπώσουμε απλά τα \$PHP_AUTH_USER και \$PHP_AUTH_PW, θα θέλαμε πιθανώς να ελέγξουμε αν είναι έγκυρα τα username και password. Αυτό μπορεί να γίνει στέλνοντας ένα ερώτημα (query) σε μια βάση

δεδομένων ή αναζητώντας τον χρήστη σ' ένα αρχείο dbm.

Για να μπορέσουμε να εμποδίσουμε κάποιον από το να γράψει ένα script που να αποκαλύπτει το password μιας σελίδας που έχει επικυρωθεί (authenticated) μέσω ενός παραδοσιακού εξωτερικού μηχανισμού, οι μεταβλητές PHP_AUTH δεν θα ορισθούν αν είναι ενεργοποιημένη η εξωτερική επικύρωση (external authentication) γι' αυτή τη συγκεκριμένη σελίδα.

Σ' αυτήν την περίπτωση, η μεταβλητή \$REMOTE_USER μπορεί να χρησιμοποιηθεί για να αναγνωρισθεί ο εξωτερικά επικυρωμένος χρήστης. Παρ' όλα αυτά, όμως, τα παραπάνω δεν αποτρέπουν κάποιον που ελέγχει ένα μη επικυρωμένο (non-authenticated) URL από το να κλέβει passwords από επικυρωμένα URLs στον ίδιο server.

Ακολουθεί ένα παράδειγμα επικύρωσης (authentication) HTTP που επιβάλλει ένα νέο name/password.

```
<?php
function authenticate() {
Header( "WWW-authenticate: basic realm="Test Authentication
System");
Header( "HTTP/1.0 401 Unauthorized");
echo "Πρέπει να καταχωρήσετε ένα έγκυρο login ID και
password για να έχετε πρόσβαση σ' αυτά τα στοιχεία\n";
exit;
}
if(!isset($PHP_AUTH_USER) || ($SeenBefore == 1 &&
!strcmp($OldAuth, $PHP_AUTH_USER)) ) {
authenticate();
}
else {
echo "Welcome: $PHP_AUTH_USER<BR>";
echo "Old: $OldAuth";
echo "<FORM ACTION=\"$PHP_SELF\"
METHOD=POST>\n";
echo "<INPUT TYPE=HIDDEN NAME=\"SeenBefore\"
VALUE=\"1\">\n";
echo "<INPUT TYPE=HIDDEN NAME=\"OldAuth\"
VALUE=\"$PHP_AUTH_USER\">\n";
echo "<INPUT TYPE=Submit
VALUE=\"Re Authenticate\">\n";
echo "</FORM>\n";
}
?>
```

2.15 Τα Cookies

Η PHP υποστηρίζει καθαρά τα HTTP cookies, τα οποία είναι ένας μηχανισμός αποθήκευσης δεδομένων στον απομακρυσμένο φυλλομετρητή για να μπορούμε έτσι να παρακολουθούμε ή να αναγνωρίζουμε τους χρήστες.

Μπορούμε να ορίσουμε cookies με τη συνάρτηση setcookie().

Τα cookies αποτελούν μέρος της επικεφαλίδας (header) του HTTP και έτσι η συνάρτηση setcookie() πρέπει να κληθεί πριν σταλεί κάποια έξοδος στον

φυλλομετρητή. Αυτός είναι ο ίδιος περιορισμός που έχει και η συνάρτηση header().

Τα cookies που στέλνονται σε μας από τον πελάτη (client) μετατρέπονται αυτόματα σε μια μεταβλητή της PHP όπως ακριβώς συμβαίνει

με τις μεθόδους GET και POST. Αν θελήσουμε να εκχωρήσουμε πολλαπλές

τιμές σ' ένα μόνο cookie, προσθέτουμε τα [] στο όνομα του cookie.

Uploads με τη Μέθοδο POST

Η PHP μπορεί να λάβει uploads αρχείων από έναν συμβατό φυλλομετρητή με το RFC-1867, όπως είναι ο Netscape Navigator και ο Microsoft Internet Explorer. Με τη δυνατότητα αυτή μπορεί κάποιος να κάνει

upload και κείμενο (text) και δυαδικά αρχεία (binary files).

Με τις συναρτήσεις για επικύρωση της PHP και χειρισμό αρχείων, έχουμε πλήρη έλεγχο για το ποιος έχει το δικαίωμα να κάνει upload και το τι

πρέπει να γίνει με το αρχείο αφού έχει γίνει upload.

Μπορούμε να δημιουργήσουμε μια φόρμα ειδικά για upload αρχείων, ως εξής :

```
<FORM ENCTYPE="multipart/form-data" ACTION="_URL_" METHOD=POST>
```

```
<INPUT TYPE="hidden" name="MAX_FILE_SIZE" value="1000">
```

```
Send this file : <INPUT NAME="userfile" TYPE="file">
```

```
<INPUT TYPE="submit" VALUE="Send File">
```

```
</FORM>
```

Το _URL_ πρέπει να δείχνει σ' ένα αρχείο της PHP, ενώ το κρυμμένο πεδίο MAX_FILE_SIZE πρέπει να προηγείται από το πεδίο file input και η τιμή

του είναι το μέγιστο αποδεκτό μέγεθος αρχείου (maximum filesize) σε bytes.

Σ' αυτό το αρχείο προορισμού, ορίζονται οι ακόλουθες μεταβλητές για ένα επιτυχημένο upload :

\$userfile, είναι το προσωρινό όνομα αρχείου με το οποίο αποθηκεύθηκε το αρχείο που έγινε upload στο μηχάνημα του server.

\$userfile_name, είναι το αρχικό όνομα του αρχείου στο σύστημα του αποστολέα.

\$userfile_size, είναι το μέγεθος του αρχείου που γίνεται upload σε bytes.

\$userfile_type, είναι ο τύπος mime του αρχείου, όπως image/gif.

Πρέπει να έχουμε υπόψη μας ότι το τμήμα \$userfile των παραπάνω μεταβλητών είναι το ίδιο με το όνομα του πεδίου INPUT που έχει τύπο TYPE=file στη φόρμα upload, όπως userfile στο παραπάνω παράδειγμα. Τα αρχεία εξ ορισμού αποθηκεύονται στον προκαθορισμένο προσωρινό κατάλογο του server (temporary directory). Αυτό μπορεί να αλλάξει ορίζοντας

τη μεταβλητή περιβάλλοντος TMPDIR στο περιβάλλον στο οποίο εκτελείται η

PHP. Αν την ορίσουμε χρησιμοποιώντας τη συνάρτηση putenv() μέσα από ένα

script της PHP δεν θα δουλέψει.

Το script της PHP το οποίο λαμβάνει το αρχείο που γίνεται upload θα εφαρμόσει όποια λογική είναι απαραίτητη για να καθορίσει το τι πρέπει να

γίνει με το αρχείο που γίνεται upload. Μπορούμε, για παράδειγμα, να χρησιμοποιήσουμε τη μεταβλητή \$file_size για να απορρίψουμε όλα τα αρχεία

που είναι είτε πάρα πολύ μικρά ή πάρα πολύ μεγάλα.

Μπορούμε να χρησιμοποιήσουμε τη μεταβλητή \$file_type για να απορρίψουμε όλα τα αρχεία που δεν ταιριάζουν με κάποια συγκεκριμένα κριτήρια. Θα πρέπει είτε να διαγράψουμε το αρχείο από τον προσωρινό κατάλογο ή να το μετακινήσουμε κάπου αλλού ή να το μετονομάσουμε.

Uploading Πολλών Αρχείων

Μπορούμε να κάνουμε upload πολλά αρχεία ταυτόχρονα και να έχουμε τις πληροφορίες αυτόματα οργανωμένες σε πίνακες (arrays). Για να γίνει αυτό,

πρέπει να χρησιμοποιήσουμε την ίδια σύνταξη υποβολής πίνακα στη φόρμα

της HTML όπως κάνουμε με τις λίστες επιλογής και τα πλαίσια ελέγχου :

```
<form action="file-upload.html" method="post"
enctype="multipart/form-data">
```

```
Send these files : <br>
```

```
<input name="userfile[]" type="file"><br>
```

```
<input name="userfile[]" type="file"><br>
```

```
<input type="submit" value="Send files">
```

```
</form>
```

Όταν υποβάλλεται η παραπάνω φόρμα, θα σχηματισθούν οι πίνακες \$userfile, \$userfile_name και \$userfile_size. Ο καθένας απ' αυτούς θα είναι

ένας πίνακας με αριθμητικούς δείκτες.

Για παράδειγμα, ας υποθέσουμε ότι υποβάλλονται τα αρχεία

/home/test/review.html και /home/test/xwp.out. Σ' αυτήν την περίπτωση, το

`$userfile_name[0]` θα περιέχει την τιμή `review.html` και το `$userfile_name[1]` θα περιέχει την τιμή `xwp.out`. Παρόμοια, το `$userfile_size[0]` θα περιέχει το μέγεθος αρχείου του `review.html` κοκ.

Υποστήριξη για τη Μέθοδο PUT

Η PHP παρέχει υποστήριξη για τη μέθοδο PUT του HTTP που χρησιμοποιείται από πελάτες (clients) όπως ο Netscape Composer και ο W3C

Amaya. Οι αιτήσεις (requests) της PUT είναι πολύ απλούστερες από το upload

ενός αρχείου και μοιάζουν ως εξής :

PUT /path/filename.html HTTP/1.1

Αυτό σημαίνει συνήθως ότι ο απομακρυσμένος πελάτης θα ήθελε να αποθηκεύσει τα περιεχόμενα ως εξής : `/path/filename.html`. Δεν αποτελεί προφανώς μια καλή ιδέα για το Apache ή την PHP να επιτρέπουν στον οποιονδήποτε να επικαλύπτει αρχεία στο δικό μας web tree.

Έτσι, για να χειριστούμε μια τέτοια αίτηση (request) πρέπει πρώτα να πούμε στον web server ότι θέλουμε ένα συγκεκριμένο script της PHP να χειριστεί την αίτηση. Στο Apache το κάνουμε αυτό με την οδηγία (directive)

Script, που μπορεί να τοποθετηθεί οπουδήποτε στο αρχείο σύνθεσης (configuration file) του Apache :

Script PUT /put.php3

Το παραπάνω λέει στο Apache να στείλει όλες τις αιτήσεις PUT για τα URIs που ταιριάζουν στο περιεχόμενο με το script `put.php3`. Αυτό προϋποθέτει, φυσικά, ότι έχουμε ενεργοποιήσει την PHP για την επέκταση

`.php3` και ότι είναι ενεργή η PHP.

Μέσα στο αρχείο `put.php3`, θα γράψουμε κάτι σαν το εξής :

```
<? copy($PHP_UPLOADED_FILE_NAME,  
$DOCUMENT_ROOT.$REQUEST_URI); ?>
```

Το παραπάνω θα αντιγράψει το αρχείο στην τοποθεσία που ζητείται από τον απομακρυσμένο πελάτη. Θα θέλουμε πιθανώς να κάνουμε κάποιους ελέγχους ή/και να επικυρώσουμε τον χρήστη πριν να λάβει χώρα αυτή η αντιγραφή του αρχείου.

Το μόνο τρυκ εδώ είναι ότι όταν η PHP βλέπει μια αίτηση με την μέθοδο PUT, αποθηκεύει το αρχείο που γίνεται upload σ' ένα προσωρινό αρχείο όπως ακριβώς εκείνα που χειρίζονται από την μέθοδο POST.

Όταν τελειώσει η αίτηση (request), διαγράφεται αυτό το προσωρινό αρχείο. Έτσι, το script της PHP που θα γράψουμε για την μέθοδο PUT πρέπει

να αντιγράψει κάπου αλλού αυτό το αρχείο.

Το όνομα αυτού του προσωρινού αρχείου βρίσκεται στην μεταβλητή \$PHP_PUT_FILENAME και μπορούμε να βρούμε το προτεινόμενο όνομα για το αρχείο προορισμού στην \$REQUEST_URI, που είναι αυτό που καθόρισε ο απομακρυσμένος χρήστης, αλλά δεν είμαστε υποχρεωμένοι να το χρησιμοποιήσουμε. Θα μπορούσαμε, για παράδειγμα, να αντιγράψουμε όλα τα αρχεία που γίνονται upload σ' έναν ειδικό κατάλογο uploads.

Χρήση Απομακρυσμένων Αρχείων

Μπορούμε να ανοίξουμε ένα αρχείο σ' έναν απομακρυσμένο web server, αν αναλύσουμε (parse) την έξοδο για τα δεδομένα που θέλουμε και μετά να χρησιμοποιήσουμε αυτά τα δεδομένα σ' ένα ερώτημα μιας βάσης δεδομένων (database query) ή να τα εξάγουμε στο δικό μας website. Το επόμενο παράδειγμα εμφανίζει τον τίτλο (title) μιας απομακρυσμένης σελίδας.

```
<?php
$file = fopen("http://www.php.net/", "r");
if (!$file) {
echo "<p>Unable to open remote file.\n";
exit;
}
while (!feof($file)) {
$line = fgets($file, 1024);
/* This only works if the title and its tags are on one line. */
if (eregi("<title>(.*?)</title>", $line, $out)) {
$title = $out[1];
break;
}
}
fclose($file);
?>
```

Το επόμενο παράδειγμα αποθηκεύει δεδομένα σ' έναν απομακρυσμένο server.

```
<?php
$file = fopen("ftp://ftp.php.net/incoming/outputfile", "w");
if (!$file) {
echo "<p>Unable to open remote file for writing.\n";
exit;
}
/* Εδώ καταχωρούμε τα δεδομένα */
```

```
fputs($file, "$HTTP_USER_AGENT\n");  
fclose($file);  
?>__
```

ΚΕΦΑΛΑΙΟ 3

Ο Κώδικας HTML

3.1 Εισαγωγή

Στο κεφάλαιο αυτό θα επιχειρήσουμε να κάνουμε μία εισαγωγή στον κώδικα html. Αρκετοί ίσως να θεωρούσαν το εγχείρημα αυτό άσκοπο καθώς ο κώδικας δεν πολυχρησιμοποιείται. Αυτό συμβαίνει διότι υπάρχουν αρκετά προγράμματα που προσφέρουν ιδιαίτερα εύχρηστο και απλό περιβάλλον για την κατασκευή επαγγελματικών και κομψών ιστοσελίδων με πολλές δυνατότητες σε ελάχιστο χρόνο και χωρίς να απαιτούν από τον χρήστη ειδικές γνώσεις. Επιπλέον η γλώσσα html τείνει να εξαφανιστεί καθώς υπάρχουν αρκετές άλλες περισσότερο εξελιγμένες και με μεγαλύτερες δυνατότητες όπως η XML, η sHTML κ.α. Ωστόσο η άποψη αυτή (ότι είναι άσκοπη η γνώση της html) είναι αβάσιμη καθώς αρκετές φορές θα χρειαστεί κάποιος να επέμβει στον κώδικα τόσο για αλλαγές ή ενέργειες τις οποίες το πρόγραμμα που χρησιμοποιούμε δεν μπόρεσε να εκτελέσει όσο και για διορθώσεις σφαλμάτων που μπορεί να οφείλονται σε πολλούς λόγους όπως π.χ. ασυμβατότητα προγράμματος κατασκευής ιστοσελίδων και παροχέα φιλοξενίας ιστοσελίδας. Στις περιπτώσεις αυτές αλλά και σε πολλές άλλες η γνώση html όχι μόνο μας διευκολύνει αλλά δύναται και να είναι η μόνη λύση για την κατασκευή μίας ιστοσελίδας.

3.2 Βασικές Γνώσεις HTML

Πριν ξεκινήσουμε την εισαγωγή στον κώδικα αυτό σκόπιμο θεωρούμε να τονίσουμε ότι η html δεν είναι γλώσσα προγραμματισμού. Δηλαδή μέσα από την html δεν μπορούμε να κατασκευάσουμε προγράμματα (ακόμα και η αλληλεπίδραση χρήστη-διακομιστή πολλές φορές καθίσταται δύσκολη), αλλά μόνο να δώσουμε στον περιηγητή (web browser) τις οδηγίες για το τι θα εμφανίζει στο χρήστη που θα προσπελάσει μία ιστοσελίδα.

Για να γράψουμε κώδικα html δεν χρειαζόμαστε κάποιο πρόγραμμα που να μεταφράζει σε γλώσσα υπολογιστή ή μέσα από το οποίο να γίνεται η όλη διαδικασία κατασκευής ιστοσελίδων. Έτσι ένας απλός επεξεργαστής

κειμένου (text editor) όπως το Σημειωματάριο (Notepad) των Windows ή το Kate των Linux είναι αρκετός για να φτιάξουμε τις ιστοσελίδες μας.

Το αρχείο μας θα πρέπει να έχει κατάληξη .htm ή .html για να θεωρείται ιστοσελίδα από το λειτουργικό μας σύστημα και να ανοίγει με το κατάλληλο πρόγραμμα περιήγησης (π.χ. Internet Explorer, Mozilla, Firefox, Netscape, κ.α.). Πρέπει επίσης να αναφέρουμε ότι τα προγράμματα περιήγησης (όπως τα παραπάνω) έχουν ενσωματωμένες τις οδηγίες για την "μετάφραση" των εντολών html που περιέχονται στο αρχείο μας. Δύναται επίσης να υπάρχουν διαφορές από web browser σε web browser που οφείλονται σε διαφορετικό "στυλ μετάφρασης".

3.3 Παράδειγμα Δημιουργίας Ιστοσελίδας

Εφόσον έχουμε ανοίξει έναν text editor (όπως το Σημειωματάριο ή Notepad των Windows, που μπορεί να βρεθεί στο Έναρξη > Προγράμματα > Βοηθήματα ή Έναρξη > Εκτέλεση > "notepad" και [OK] ή όπως το TextEdit στα Mac που βρίσκεται στα Applications) είμαστε έτοιμοι να γράψουμε κώδικα html. Η βασική εντολή που πρέπει γράψουμε προκειμένου το πρόγραμμα περιήγησης να "αναγνωρίσει" τον κώδικα είναι <html>. Γράφουμε html μέσα σε <> τα οποία επεξηγούν στο πρόγραμμα περιήγησης ότι το περιεχόμενό τους συνιστά εντολή. Κάθε εντολή του κώδικα html πρέπει και να κλείνει (σχεδόν το 95% των εντολών, υπάρχουν και μερικές όπως θα δούμε που δεν κλείνουν). Με το κλείσιμο εννοούμε ότι παύει να ισχύει η ιδιότητα που έχουμε θέσει. Έτσι στο τέλος του κειμένου θα πρέπει να κλείσουμε την εντολή <html> για να κλείσουμε και το αρχείο μας. Όταν βάλουμε εντολή για έντονα γράμματα (Bold) θα πρέπει να την κλείσουμε εκεί που επιθυμούμε να σταματήσουν να εμφανίζονται έντονα γράμματα. Κάθε εντολή κλεισίματος είναι παρόμοια με την εντολή ανοίγματος μόνο που περιέχει τον χαρακτήρα / αμέσως μετά το <. Δηλαδή η εντολή κλεισίματος της <html> είναι: </html>.

Έτσι μέχρι στιγμής έχουμε το εξής (καλό είναι μετά από κάθε εντολή να αλλάζουμε γραμμή):

```
<html>  
</html>
```

Αυτή είναι η πρώτη μας ιστοσελίδα που βέβαια δεν περιέχει απολύτως τίποτα. Αν την αποθηκεύσουμε ως paradeigma.html και την ανοίξουμε με έναν περιηγητή ιστοσελίδων θα δούμε ένα γκρι παράθυρο χωρίς να εμφανίζει τίποτα.

Για να γράψουμε κάτι, πληκτρολογούμε ανάμεσα στην εντολή ανοίγματος και κλεισίματος το κείμενο που επιθυμούμε, π.χ.:

```
<html>
Αυτό          είναι          ένα          παράδειγμα
</html>
```

Κατά την περιήγηση αυτής της σελίδας θα εμφανιστεί μόνο το κείμενο: Αυτό είναι ένα παράδειγμα

Στο σημείο αυτό πολύ σημαντικό είναι να διευκρινίσουμε το εξής: κατά την ανάγνωση του κώδικα από το πρόγραμμα η οποιαδήποτε μορφοποίηση θέτουμε στον κειμενογράφο ακυρώνεται. Έτσι αν πατήσουμε στο παραπάνω παράδειγμα ENTER ανάμεσα στο "είναι" και στο "ένα" ενώ στον επεξεργαστή κειμένου θα εμφανιστεί η αλλαγή γραμμής (ή παραγράφου) στην ιστοσελίδα μας δεν θα εμφανιστεί. Για να αλλάξουμε συνεπώς παράγραφο θα πρέπει να βάλουμε την ειδική εντολή <P> η οποία δεν κλείνει. Ομοίως αν στον επεξεργαστή μας έχουμε κείμενο υπογραμμισμένο δεν θα εμφανιστεί η υπογράμμιση στην ιστοσελίδα (γι' αυτό άλλωστε ένας απλός κειμενογράφος χωρίς πολλές δυνατότητες είναι αρκετός και μάλιστα προτιμητέος).

3.4 Βασικές Εντολές

Στο σημείο αυτό θα περάσουμε στις πρώτες βασικές εντολές που θα πρέπει να γνωρίζουμε για να έχουμε την επιθυμητή μορφοποίηση στο κείμενο μας.

 - Έντονη γραφή (bold)
<U> - Υπογραμμισμένα (Underlined)

<I> - Πλάγια γραφή (Italics)

<CENTER> - Κεντρική στοίχιση
<LEFT> - Αριστερή στοίχιση
<RIGHT> - Δεξιά στοίχιση

Οι παραπάνω εντολές κλείνουν (, </U>, </I>, </CENTER> </LEFT>, </RIGHT>)

<P> - Παράγραφος

 - Συνέχιση γραφής στην αμέσως αποκάτω γραμμή
<HR> - Η εντολή αυτή χαράσσει μία οριζόντια γραμμή στην σελίδα μας (χρησιμοποιείται κυρίως όταν αλλάζουμε θεματική ενότητα)

Οι τρεις παραπάνω εντολές δεν κλείνουν

3.5 Υπερσυνδέσεις - Links

Το ξεχωριστό με τις ιστοσελίδες που τις κάνει να διαφοροποιούνται από το απλό κείμενο είναι ότι παρέχουν υπερσυνδέσεις, δηλαδή μέσα από μία ιστοσελίδα μπορεί κανείς να παραπεμφθεί σε πλήθος άλλων (εξωτερικών ή εσωτερικών) ιστοσελίδων. Το χαρακτηριστικό αυτό είναι ένα από τα κυριότερα που έχουν ωθήσει στην τωρινή ανάπτυξη του διαδικτύου.

Ο κώδικας για την δημιουργία υπερσύνδεσης είναι: `ΜΠΛΑ-ΜΠΛΑ`. Στον παραπάνω κώδικα το πρώτο μέρος είναι `` δηλαδή στο ΜΠΛΑ-ΜΠΛΑ γράφουμε ότι θέλουμε να εμφανίζεται στην ιστοσελίδα μας και τέλος το `` είναι η εντολή κλεισίματος του link μας.

Εικόνες

Προκειμένου η ιστοσελίδα μας να είναι ελκυστική δεν θα πρέπει να αρκούμαστε στο απλό κείμενο μας. Θα πρέπει για να μην γίνεται βαρετή η ανάγνωση να προστίθενται εικόνες που θα κάνουν την ανάγνωση πιο ευχάριστη και την ιστοσελίδα μας περισσότερο κομψή και ευπαρουσίαστη.

Η Εντολή για την προσθήκη εικόνας είναι: ``. Το `εκείπουείναιηεικόναμας` δηλώνει την τοποθεσία στην οποία θα ανατρέχει ο περιηγητής για να αντλεί την εικόνα και να την εμφανίζει στην ιστοσελίδα μας. Έτσι αν κάποιος έχει ανεβάσει την ιστοσελίδα του σε έναν παροχέα αυτός του προσφέρει χώρο για τις ιστοσελίδες και τα υπόλοιπα αρχεία του. Αν η διεύθυνση όπου βρίσκεται η ιστοσελίδα μας είναι για παράδειγμα `www.tositemoy.gr` τότε η εικόνα μπορεί να βρίσκεται στον φάκελο: `www.tositemoy.gr/eikones`.

Πολλές φορές κατασκευάζουμε τις ιστοσελίδες μας σε έναν τοπικό φάκελο στον σκληρό μας ή σε άλλο αποθηκευτικό μέσο. Κατασκευάζουμε στον φάκελο με την ιστοσελίδα μας έναν υποφάκελο

με το όνομα π.χ. images (για να το θυμόμαστε ευκολότερα) όπου αποθηκεύουμε τις εικόνες που εμφανίζονται στην ιστοσελίδα μας. Αργότερα όταν κατασκευάσουμε όλο το site ανεβάζουμε όλα τα αρχεία (όχι μόνο τις ιστοσελίδες *.html αλλά και τους φακέλους με τις εικόνες, τους ήχους, τα βίντεο, κ.λ.π.). Στην αμέσως προηγούμενη περίπτωση (η οποία είναι και η συνηθέστερη) η εντολή θα είναι κάπως έτσι: . Συνιστάται η χρήση όχι ολόκληρης της διαδρομής του δίσκου (π.χ. c:/istoselida/images/image1.gif) αλλά της συντομημένης, καθώς κατά το ανέβασμα θα πρέπει να την αλλάξουμε. Η συντομημένη μορφή μας εξασφαλίζει ότι η εικόνα μας θα λειτουργεί τόσο στην περιήγηση τοπικά όσο και στο ίντερνετ. Τέλος θα πρέπει να αναφέρουμε όπως είναι προφανές ότι η εντολή αυτή δεν κλείνει.

3.6 Προχωρημένες εντολές

Στην ενότητα αυτή θα εξετάσουμε ορισμένες περισσότερο προχωρημένες εντολές της html. Αν και με αυτές που ήδη έχουμε δει μέχρι τώρα επαρκούν για την κατασκευή μίας ιστοσελίδας οι επόμενες θα μας βοηθήσουν στο να βελτιώσουμε αλλά και να εμπλουτίσουμε τις σελίδες μας.

Δομή σελίδας

<HEAD> Η εντολή αυτή μπαίνει στην αρχή της σελίδας ώστε να περικλείει εντολές που περιέχουν πληροφορίες προς τον browser αλλά δεν είναι μέρος της σελίδας. Παράδειγμα: <HEAD> ... </HEAD>

<BODY> Η εντολή αυτή δηλώνει στο πρόγραμμα που χρησιμοποιούμε για την περιήγηση ότι ο παρακάτω κώδικας και τα δεδομένα που γράφουμε ανήκουν στο κυρίως μέρος της ιστοσελίδας. Πρακτικά μετά την εντολή <HTML> και την HEAD με ότι περικλείει, μπαίνει η <BODY> ενώ κλείνει συνήθως αμέσως πριν από την </HTML> (η εντολή κλεισίματος όπως αναμέναμε είναι: </BODY>).

<TITLE> Η εντολή αυτή δηλώνει στον web browser τι τίτλο επιθυμούμε να έχει η ιστοσελίδα. Ο τίτλος εμφανίζεται στην επάνω λωρίδα του παραθύρου του web browser (εκεί όπου εμφανίζεται ο τίτλος της εφαρμογής / προγράμματος που χρησιμοποιούμε). Η TITLE είναι απαραίτητο να μπει μέσα στην HEAD. Παράδειγμα: <HEAD> <TITLE>My Homepage</TITLE> </HEAD>

Πίνακες

Πολλές φορές σε μία ιστοσελίδα θα πρέπει να απεικονίσουμε δεδομένα σε μορφή πίνακα για καλύτερη ταξινόμηση ή για ευκολότερη παρουσίαση. Πέρα από την απεικόνιση δεδομένων όμως οι πίνακες μπορεί να χρησιμοποιηθούν και για κομψότερη εμφάνιση της ιστοσελίδας μας. Παρακάτω παρουσιάζονται οι εντολές για τη δημιουργία πίνακα.

Η εντολή για να δώσουμε στον περιηγητή να καταλάβει ότι σκοπεύουμε να απεικονίσουμε τα επόμενα δεδομένα σε πίνακα είναι: <TABLE>. Η εντολή φυσικά κλείνει (</TABLE>) μόλις τελειώσουμε με τον πίνακα. Ωστόσο μόνο η εντολή table δεν εξηγεί στο πρόγραμμα πόσες στήλες και γραμμές θέλουμε να συμπεριλάβουμε στον πίνακα μας. Οι εντολές για την δημιουργία γραμμών είναι <TR> και για τη δημιουργία στηλών <TD>. Πρέπει πάντα στον κώδικα να ξεκινάμε με τις γραμμές και μετά με τις στήλες. Το επόμενο παράδειγμα θα μας εξηγήσει αναλυτικά πως φτιάχνουμε έναν πίνακα σε html:

```
<TABLE>
<TR>
<TD>1</TD>                                <TD>2</TD>
</TR>
<TR>
<TD>3</TD>                                <TD>4</TD>
</TR>
</TABLE>
```

Ο πίνακας του παραδείγματος εξηγεί στον περιηγητή ιστοσελίδων να κατασκευάσει έναν πίνακα με τη πρώτη σειρά/γραμμή (<TR>) να περιέχει δύο στήλες (<TD>): στην πρώτη να γράφει τα δεδομένα "1" και στην δεύτερη τα δεδομένα "2". Ο πίνακας όμως συνεχίζει με δεύτερη γραμμή (<TR>) η οποία επίσης έχει δύο στήλες (<TD>) με τα δεδομένα "3" και "4" αντίστοιχα. Παρατηρούμε επίσης ότι μόλις τελειώσουν τα δεδομένα σε μία στήλη μας, την κλείνουμε με την εντολή </TD> για να αρχίσουμε καινούργια στήλη (<TD>) -εφόσον επιθυμούμε-. Ομοίως μόλις τελειώσουμε με την πρώτη γραμμή θα πρέπει να την κλείσουμε με το αντίστοιχο κλείσιμο της <TR> δηλαδή την: </TR>. Σημείωση: Τα κενά δεν παίζουν κανένα ρόλο στην απεικόνιση της σελίδας απλώς διευκολύνουν την ανάγνωση από τον χρήστη.

Επικεφαλίδες

<H#> Η εντολή αυτή δηλώνει τις επικεφαλίδες των διάφορων μερών του κειμένου μας π.χ. τα κεφάλαια. Μπορεί να δηλώσει μέχρι 6 κεφάλαια-υποκεφάλαια ξεκινώντας από το H1 έως το H6. π.χ.:

```
<H1>Πρώτο κεφάλαιο</H1>
Το κείμενο του πρώτου κεφαλαίου είναι.....
Για να δηλώσουμε ότι ένα μέρος του κειμένου είναι υπομέρος ενός
άλλου ορίζουμε στην αρχή του επικεφαλίδα επιπέδου +1 από το μητρικό
του, π.χ. για ένα υποκεφάλαιο του παραπάνω θα γράφαμε:
<H1>Πρώτο κεφάλαιο</H1>
Το κείμενο του πρώτου κεφαλαίου είναι.....
.
```

```
<H2>Υποκεφάλαιο</H2>
```

```
Κείμενο υποκεφαλαίου....
Οποτεδήποτε θέλουμε μπορούμε να ορίσουμε αρχή ενός κεφαλαίου
οποιοδήποτε επιπέδου. Συνήθως στους browser οι τίτλοι των κεφαλαίων
εμφανίζονται με μεγαλύτερη γραμματοσειρά, αλλά αυτή η εντολή δεν θα
πρέπει να χρησιμοποιείται απλώς για λόγους εμφάνισης αλλά μόνο για να
σηματιστεί η δομή του κειμένου.
```

Λίστες

Ορισμένες φορές χρειάζεται να εμφανίσουμε κάποια στοιχεία σε λίστες, με ταξινόμηση ή όχι.

Αταξινόμητες λίστες

Για να δημιουργήσουμε λίστα χωρίς ταξινόμηση βάζουμε αρχικά την εντολή με την οποία δηλώνουμε ότι τα επόμενα θα είναι δεδομένα αταξινόμητης λίστας. Έπειτα για κάθε δεδομένο ή δεδομένα που θέλουμε να τοποθετήσουμε βάζουμε την εντολή . Στο τέλος του κάθε στοιχείου κλείνουμε και προχωράμε σε νέο στοιχείο της λίστας. Συνήθως το κάθε στοιχείο μιας αταξινόμητης λίστας εμφανίζεται μετά από κάποιο σύμβολο όπως η κουκκίδα. Παράδειγμα:

```
Οι λόγοι που οδήγησαν στο φαινόμενο είναι :
<UL>
<LI>Λόγος α α</LI>
<LI>Λόγος β β</LI>
</UL>
```

Αυτό έχει ως αποτέλεσμα:

Οι λόγοι που οδήγησαν στο φαινόμενο είναι:

- Λόγος α
- Λόγος β

Ταξινομημένη λίστα

Η ταξινομημένη λίστα είναι παρόμοια με την αταξινομητη. Αντί για την αρχική εντολή βάζουμε . Το κάθε στοιχείο μιας ταξινομημένης λίστας εμφανίζεται με αύξουσα αρίθμηση ξεκινώντας από το 1. Παράδειγμα:

```
Οι λόγοι που οδήγησαν στο φαινόμενο είναι :  
<OL>  
<LI>Λόγος α</LI> α</LI>  
<LI>Λόγος β</LI> β</LI>  
</OL>
```

Αυτό έχει ως αποτέλεσμα:

Οι λόγοι που οδήγησαν στο φαινόμενο είναι:

1. Λόγος α
2. Λόγος β

Χρώμα Φόντου

Για να κάνουμε την ιστοσελίδα μας περισσότερο ευπαρουσίαστη πολλές φορές βάζουμε χρώμα για φόντο. Η εντολή είναι: <body bgcolor="#####">. Η εντολή δεν κλείνει όπως είναι λογικό. Οι χαρακτήρες ##### υποδηλώνουν ότι πρέπει να αντικατασταθούν από τα κατάλληλους χαρακτήρες που υποδηλώνουν το χρώμα όμως ο πρώτος χαρακτήρας διατηρείται σταθερός. Το χρώμα στην html βγαίνει βάσει του τρόπου που απεικονίζεται το χρώμα σε συσκευές όπως η οθόνη, δηλαδή RGB (red-green-blue). Η οθόνη του υπολογιστή (όπως άλλωστε και η τηλεόραση αλλά και οι υπόλοιπες παρόμοιες συσκευές) συνθέτουν το χρώμα από τα τρία βασικά χρώματα: κόκκινο-πράσινο-μπλε. Οι χαρακτήρες ##### αναφέρονται σε αυτά χρώματα τα Τα δύο πρώτα # αναφέρονται στο κόκκινο, τα επόμενα δύο # στο πράσινο και τέλος τα τελευταία δύο # στο μπλε (όπως και η σειρά RGB). Οι τιμές που παίρνουν είναι σύμφωνες με το δεκαεξαδικό σύστημα δηλαδή από 0 έως F (0 είναι η μικρότερη τιμή και F η μεγαλύτερη). Η σειρά είναι 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F. Αν θέλουμε να έχουμε π.χ. άσπρο χρώμα θα πρέπει να βάλουμε τις τιμές "FFFFFF" (αφού το άσπρο αποτελεί την σύνθεση όλων των χρωμάτων). Αντίστροφα για το μαύρο "000000".Εναλλακτικά σε τελευταίες εκδόσεις της html μπορούμε να βάλουμε τα ονόματα των χρωμάτων στα αγγλικά (π.χ. green, red, κ.λ.π.) αντί για τα ορίσματα στο δεκαεξαδικό σύστημα, ωστόσο αυτό δεν μας

δίνει πολλές επιλογές όπως για παράδειγμα όλες τις αποχρώσεις του πράσινου ή του κόκκινου κ.λ.π.

Χρώμα Γραμματοσειράς

Ομοίως μπορούμε να αλλάζουμε και τα χρώματα της γραμματοσειράς μας για κομψότερες ιστοσελίδες. Η εντολή είναι `` και κλείνει με απλό ``. Εδώ ισχύουν ακριβώς ότι και παραπάνω για το φόντο όσον αφορά το ορίσματα για τα #####.

Εικόνα για φόντο

Εναλλακτικά αντί για χρώμα, μπορούμε να βάλουμε εικόνα για φόντο. Η εντολή είναι: `<body background="eikona.gif">`. Η εντολή όπως είναι λογικό δεν κλείνει.

Το προσθετικό BORDER

Το προσθετικό BORDER είναι ένα τμήμα που μπορεί να προστεθεί σε ορισμένες εντολές που εμφανίζουν πλαίσια. Συγκεκριμένα μπορούμε να χρησιμοποιήσουμε το BORDER στους πίνακες, στη δημιουργία πλαισίων και όταν βάζουμε ως απεικόνιση συνδέσμου εικόνα αντί για κείμενο (π.χ. ``). Στο παράδειγμα αυτό μόλις κάνουμε κλικ στην εικόνα: eikona.gif θα μεταφερθούμε στην σελίδα: selida.html). Το BORDER υποδηλώνει το μέγεθος του πλαισίου. Έτσι όταν έχουμε `BORDER="0"` δεν εμφανίζεται το πλαίσιο. Όσο μεγαλύτερα νούμερα βάζουμε τόσο μεγαλύτερο (χονδρότερο) είναι και το πλαίσιο. Το προσθετικό μπαίνει στο τέλος του κειμένου της εντολής πριν βάλουμε τον χαρακτήρα `>`. (π.χ. `<TABLE BORDER="0">` ή ``)

Το προσθετικό ALT

Το προσθετικό ALT προστίθεται και αυτό σε ορισμένες εντολές και υποδηλώνει τον εναλλακτικό τρόπο εμφάνισης. Η κύρια χρήση του ALT είναι στις εικόνες. π.χ. ``. Ορισμένοι παλαιοί web browser δεν είχαν την δυνατότητα εμφάνισης εικόνων (τόρα βέβαια όλοι έχουν αυτήν την δυνατότητα). Το προσθετικό αυτό υποδήλωνε στο περιηγητή που δεν εμφάνιζε εικόνες να εμφανίσει το κείμενο που περιείχε το προσθετικό. Στο προηγούμενο παράδειγμα θα εμφανιζόταν στη θέση της εικόνας το κείμενο "Εικόνα 1" (χωρίς τα εισαγωγικά). Ακόμα όμως και σήμερα χρησιμοποιείται το προσθετικό αυτό. Αν το χρησιμοποιήσουμε σε μία ιστοσελίδα μας, μέχρι να φορτωθούν οι εικόνες θα εμφανίζονται τα ορίσματα των προσθετικών ALT των εικόνων αυτών (κάτι τέτοιο θα είναι ιδιαίτερα ορατό αν χρησιμοποιούμε γραφικά με μεγάλο μέγεθος τα οποία χρειάζονται κάποιο χρόνο για να φορτωθούν). Τυπωμένο κείμενο

<TT> Η παραπάνω εντολή (με κλείσιμο </TT>) αλλάζει την προεπιλεγμένη γραμματοσειρά μας σε γράμματα γραφομηχανής. Δηλαδή στο παράδειγμα <TT>Αυτά είναι γράμματα γραφομηχανής</TT> οι λέξεις: "Αυτά είναι γράμματα γραφομηχανής" θα φαίνονται σαν γράμματα γραφομηχανής και θα ξεχωρίζουν από το υπόλοιπο κείμενο.

Πλαίσια

Η κατασκευή πλαισίων αποτελεί μία δύσκολη και χρονοβόρα σχετικά διαδικασία. Θα προσπαθήσουμε συνεπώς να την αναλύσουμε με τον ευκολότερο δυνατό και κατανοητό τρόπο. Το σημείο κλειδί εδώ είναι η δημιουργία μίας σελίδας "υποδομής" η οποία δεν θα περιέχει τίποτα για εμφάνιση εκτός από κώδικα. Στη σελίδα αυτή θα ορίσουμε πως ο περιηγητής μας θα εμφανίσει τα πλαίσια, που ακριβώς θα εμφανιστούν αυτά και τι σελίδες θα περιέχουν. Η πιο συνηθισμένη κατασκευή με πλαίσια περιλαμβάνει τον διαχωρισμό της οθόνης σε δύο πλαίσια. Ένα με το μενού - περιεχόμενα του ιστότοπου (το οποίο είναι συνήθως αριστερά) και ένα όπου εμφανίζονται οι επιμέρους ιστοσελίδες (συνήθως είναι το δεξί πλαίσιο και καταλαμβάνει το μεγαλύτερο μέρος της οθόνης).

3.6 Η σελίδα υποδομής

Θα πρέπει πάντα προκειμένου να εμφανιστούν τα πλαίσια να ορίσουμε την σελίδα υποδομής μας ως κεντρική σελίδα (π.χ. index.html). Την σελίδα αυτήν θα πρέπει να διαβάζει ο web browser για να εμφανίζει τα περιεχόμενα κάθε πλαισίου. Αν προσπαθήσουμε να ανοίξουμε απευθείας μία άλλη σελίδα του ιστότοπου μας, αυτή θα ανοίξει αλλά δεν θα εμφανιστούν τα άλλα πλαίσια. Θα παρουσιάσουμε την κατασκευή της ιστοσελίδας αυτής με ένα παράδειγμα:

```
<HTML>
<TITLE>Ο           ΙΣΤΟΤΟΠΟΣ           μου</TITLE>
<FRAMESET
                                COLS="180 , * ">
<FRAME          SRC="MENU .HTML "      NAME="ARISTERO ">
<FRAME          SRC="MAIN .HTML "      NAME="DEKSIO ">
</FRAMESET>
</HTML>
```

τη σελίδα αυτή δίνουμε στο πρόγραμμα μας την εντολή να κατασκευάσει δύο πλαίσια σε στήλες - σειρές (<FRAMESET COLS ...). Αν βάζαμε ROWS (γραμμές) αντί για "COLS" θα φτιάχναμε οριζόντια πλαίσια (φυσικά μπορούμε να φτιάξουμε ταυτόχρονα και οριζόντια και κάθετα). Στην ίδια εντολή γράφουμε μετά το σύμβολο: = τα εξής: (180,*). Αυτό

σημαίνει ότι το πρώτο πλαίσιο θα έχει πλάτος 180 pixels και το * υποδηλώνει ότι το υπόλοιπο της οθόνης θα αποτελεί το άλλο πλαίσιο. Θα μπορούσαμε να βάλουμε και ποσοστά επί τοις εκατό (π.χ. 20%, 80%) δηλαδή το πρώτο πλαίσιο 20% και το υπόλοιπο 80% στο δεύτερο (το άθροισμα φυσικά θα πρέπει να συγκεντρώνει το 100%). Η σειρά με την οποία κατασκευάζονται τα πλαίσια είναι από αριστερά προς δεξιά και από πάνω προς τα κάτω (δηλαδή τα 180 pixels στο παράδειγμα μας αναφέρονται στο αριστερό πλαίσιο και το υπόλοιπο τμήμα στο αριστερό).

Στις επόμενες σειρές (`<FRAME SRC="MENU.HTML" NAME="ARISTERO">` και `<FRAME SRC="MAIN.HTML" NAME="DEKSIO">`) δηλώνουμε τα ονόματα των πλαισίων μας με τη σειρά που τα έχουμε κατασκευάσει από την προηγούμενη εντολή και τις σελίδες που θα διαβάσει ο web browser μας για κάθε πλαίσιο από αυτά. Έτσι το πρώτο πλαίσιο με μέγεθος 180 pixel το ονομάζουμε ARISTERO και το δεύτερο DEKSIO. Τα ονόματα που θα διαβάσει το πρόγραμμα μας είναι `menu.html` για το αριστερό πλαίσιο και `main.html` για το δεξί.

Στο τέλος πριν κλείσουμε την εντολή html κλείνουμε την FRAMESET.

Παρατήρηση: Η μόνη εντολή που θα πρέπει (όχι βέβαια απαραίτητα) να περιέχει μία σελίδα υποδομής εκτός από αυτές για το καθορισμό των πλαισίων είναι η TITLE. Εξάλλου όλες οι άλλες εντολές δεν έχουν νόημα: αυτό που ο χρήστης θα δει είναι τα περιεχόμενα των πλαισίων. Αν θέλουμε επίσης να μην εμφανίζονται διαχωριστικές ανάγλυφες γραμμές ανάμεσα στα πλαίσια, μπορούμε στην εντολή `<FRAMESET COLS="180 , *">` να προσθέσουμε το τμήμα: `BORDER="0"` (δηλαδή θα γίνει κάπως έτσι: `<FRAMESET COLS="180 , *" BORDER="0">`).

Δημιουργία Συνδέσμων

Στις υπόλοιπες σελίδες όταν τοποθετούμε έναν σύνδεσμο (link) γράφουμε και το πλαίσιο προορισμού. Έτσι στην εντολή `keimeno` Θα προσθέσουμε το κομμάτι `TARGET="onoma_plaisiou"`. Η εντολή θα διαμορφωθεί δηλαδή έτσι: `keimeno`.

Στο προηγούμενο παράδειγμα μας η σελίδα `menu.html` μπορεί να περιέχει τις ιστοσελίδες του ιστότοπου μας. Αν για παράδειγμα μία ιστοσελίδα περιέχει φωτογραφίες μας θα πρέπει να συμπεριλάβουμε μεταξύ άλλων στο αρχείο `menu.html` τα εξής:

Δείτε τις φωτογραφίες μου -- Έτσι κάνοντας κλικ κάποιος στο κείμενο "Δείτε τις φωτογραφίες μου" στο δεξιό πλαίσιο θα ανοίξει η ιστοσελίδα "fotos.html".

Με αυτό τον τρόπο μπορούμε να έχουμε τα περιεχόμενα μας στο αριστερό πλαίσιο και με κλικ στους συνδέσμους που αυτό περιέχει, ο χρήστης θα μεταφέρεται στην αντίστοιχη σελίδα (η οποία θα εμφανίζεται στο δεξί πλαίσιο).

3.7 Οι Πληροφορίες της Ιστοσελίδας μας

Στην ενότητα αυτή παρουσιάζουμε ορισμένα βασικά ζητήματα που θα πρέπει ο ενδιαφερόμενος να γνωρίζει πριν κατασκευάσει μία ιστοσελίδα.

Εικόνες

Οι εικόνες που δημοσιεύονται στο διαδίκτυο πρέπει να έχουν μικρό σχετικά μέγεθος και αυτό για να κατεβαίνουν γρήγορα και εύκολα. Θα πρέπει να μην ξεχνάμε ότι ο επισκέπτης της ιστοσελίδας μας δεν έχει πάντα την υπομονή να περιμένει αρκετή ώρα για να κατέβει μία εικόνα. Ιδίως όταν οι σελίδες μας περιέχουν αρκετές εικόνες. Θα πρέπει λοιπόν να χρησιμοποιούμε ειδική κωδικοποίηση στις εικόνες μας. Συνήθως οι εικόνες με την υψηλότερη συμπίεση (και άρα το μικρότερο μέγεθος) είναι αυτές οι οποίες έχουν προέκταση gif ή jpg (εναλλακτικά: jpeg). Οι εικόνες αυτές συνήθως καταλαμβάνουν μικρό μέγεθος (ανάλογα βέβαια την ποιότητα που θα καθορίσουμε κατά την συμπίεση).

Για φωτογραφίες καλύτερα αποτελέσματα παρέχουν οι εικόνες jpg (ή jpeg), ενώ για γραφικά προτιμότερο είναι να χρησιμοποιούμε εικόνες gif οι οποίες μπορεί να καταλαμβάνουν ακόμα μικρότερο χώρο και συνεπώς να κατεβαίνουν ακόμα πιο γρήγορα σε σχέση με τις πρώτες. Ένα επίσης χαρακτηριστικό των εικόνων gif είναι ότι υποστηρίζουν και αδιαφάνεια (transparency). Αδιαφάνεια είναι όταν σε μία εικόνα ορισμένα εικονοστοιχεία (pixel) είναι διαφανή. Έτσι συνήθως αυτά τα pixel παίρνουν το χρώμα του φόντου της ιστοσελίδας μας. Έτσι αν θέλουμε να βάλουμε ως γραφικό ένα τρίγωνο ή ένα κύκλο για παράδειγμα, οι περιοχές έξω από το σχήμα οι οποίες δεν μπορούν να αποκοπούν καταστύουν την ιστοσελίδα μας λιγότερο κομψή. Θα πρέπει αυτές τις περιοχές γύρω από τον κύκλο ή το τρίγωνο (ή φυσικά από οποιοδήποτε άλλο σχήμα ή γραφικό) να γίνουν διαφανείς. Αν επιχειρούσαμε να

βάψουμε αυτές τις περιοχές με το χρώμα του φόντου της ιστοσελίδας μας εκτός από το μεγαλύτερο μέγεθος που θα αποκτούσε το γραφικό μας, θα καθιστούσε ασύμφορη την αλλαγή του χρώματος του φόντου μας, καθώς θα έπρεπε να ξαναλλάξουμε το χρώμα όλων των σχετικών εικόνων μας (κάτι ιδιαίτερα χρονοβόρο και ανιαρό). Τέλος άλλο ένα πολύ καλό χαρακτηριστικό που ενσωματώνουν οι εικόνες της μορφής gif είναι η δυνατότητα τους να είναι κινούμενες δηλαδή να εναλλάσσονται. Έτσι σε ένα γραφικό μπορούμε να βάλουμε δύο ή περισσότερες εικόνες που εναλλάσσονται με το πέρασμα του χρόνου. Ένα παράδειγμα είναι να βάλουμε για επικεφαλίδα της ιστοσελίδας μας για εικόνα, ένα μεγάλο χρωματιστό "Welcome" που θα αναβοσβήνει. Σχεδόν όλα τα προγράμματα επεξεργασίας εικόνας (ακόμα και τα πιο απλά) επιτρέπουν κωδικοποίηση αλλά και μετατροπή εικόνας από ένα τύπο αρχείου σε άλλο (π.χ. από bmp σε gif), έτσι δεν θα χρειαστεί να αποκτήσετε κάποιο εξειδικευμένο πρόγραμμα.

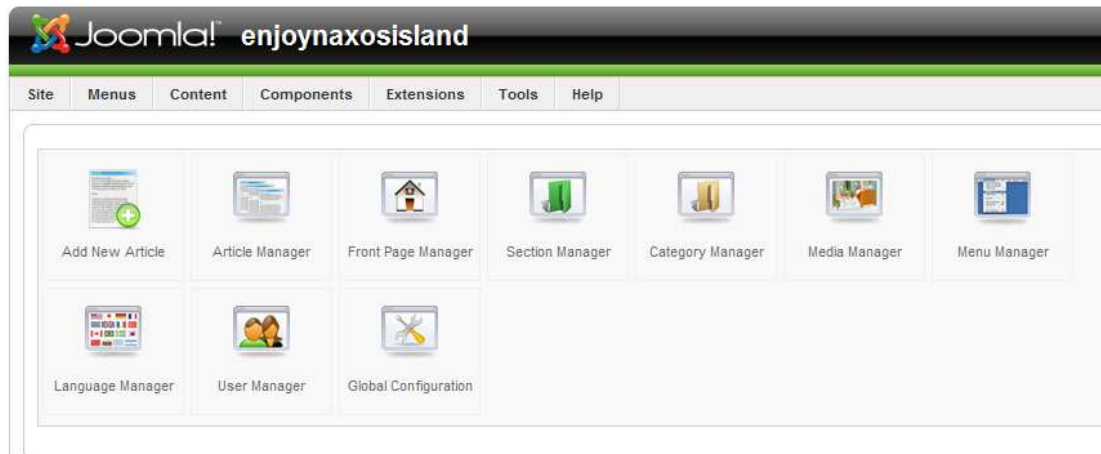
Τέλος θα πρέπει να τονίσουμε ότι τα γραφικά και οι εικόνες βελτιώνουν την αισθητική μίας ιστοσελίδας αλλά την καθιστούν αργή στο κατέβασμα. Η χρήση εικόνων σε ιστοσελίδες συνεπώς θα πρέπει να γίνεται με μέτρο και σύνεση. Η ανάρτηση πολλών και μεγάλων εικόνων σε μία ιστοσελίδα δεν συνίσταται σε καμία περίπτωση.

Κείμενο

Βασικός κανόνας που θα πρέπει να ακολουθείται πάντα κατά την κατασκευή ιστοσελίδων είναι η αποφυγή συνεχόμενης γραφής με κεφαλαία γράμματα. Τα κεφαλαία γράμματα δηλώνουν ότι κάποιος είναι "φωνακλάς" και "εκνευριστικός". Επιπλέον τα κεφαλαία γράμματα είναι κουραστικά για το μάτι. Ο κανόνας αυτός ισχύει φυσικά για μεγάλο κείμενο στα Κεφαλαία. Μπορούμε εννοείται να βάζουμε λέξεις ή ακόμα και μικρές φράσεις στα κεφαλαία όταν αυτό θεωρούμε ότι είναι απαραίτητο.

ΚΕΦΑΛΑΙΟ 4

4.1 ΤΟ JOOMLA (ΕΚΔΟΣΗ 1.5.8)

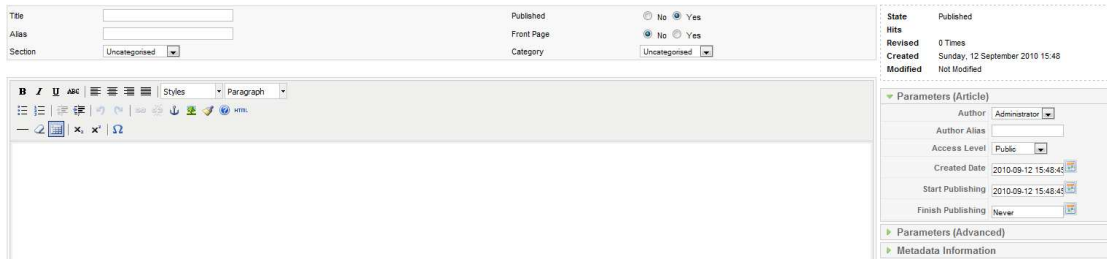


Εικόνα 1 Αρχική Σελίδα Joomla

Στην εικόνα 1 βλέπουμε την κεντρική σελίδα του Joomla απ' όπου αποκτούμε πρόσβαση εύκολα στα εργαλεία και τις δυνατότητες του cms μας. Στις παρακάτω ενότητες θα παρουσιάσουμε αυτά τα εργαλεία και τις δυνατότητες.

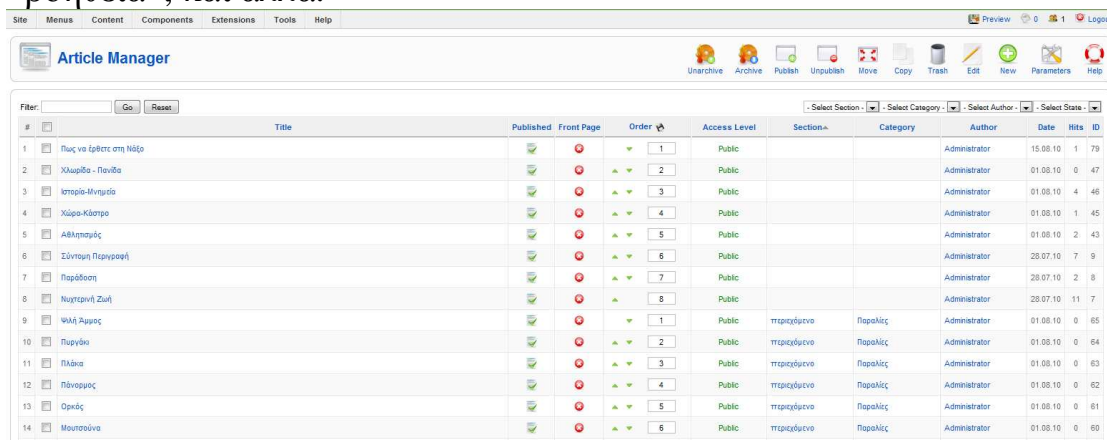
4.2 ΒΑΣΙΚΑ ΕΡΓΑΛΕΙΑ JOOMLA

Στην εικόνα 2 βλέπουμε την φόρμα δημιουργίας ενός άρθρου (article) που είναι και η βασική μορφή δεδομένων του cms μας.



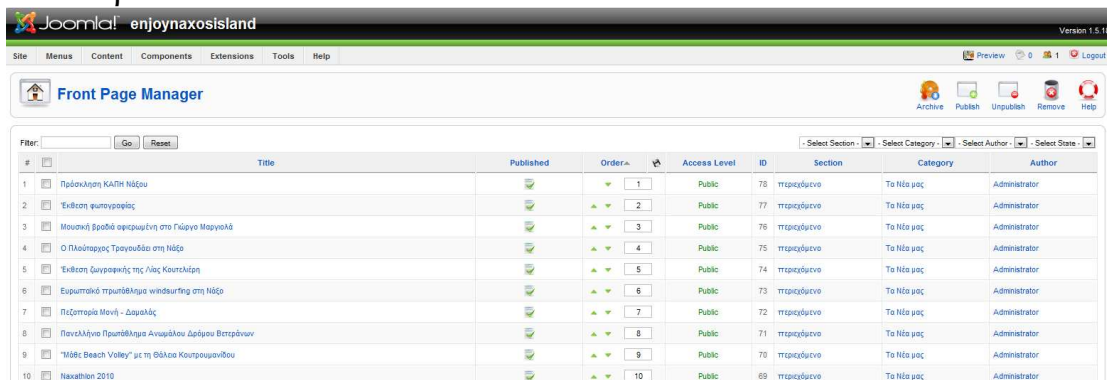
Εικόνα 2 δημιουργία άρθρου

Στην εικόνα 3 παρατηρούμε την διαχείριση άρθρων με τα εργαλεία που μας προσφέρει πάνω δεξιά, όπως «νέο», «διαγραφή», «αντιγραφή», «βοήθεια», και άλλα.



Εικόνα 3 Διαχείριση άρθρων

Εδώ βλέπουμε την διαχείριση της κεντρικής σελίδας της ιστοσελίδας μας όπου μπορούμε να διαλέξουμε ποια άρθρα θα εμφανίζονται και με ποιον τρόπο.

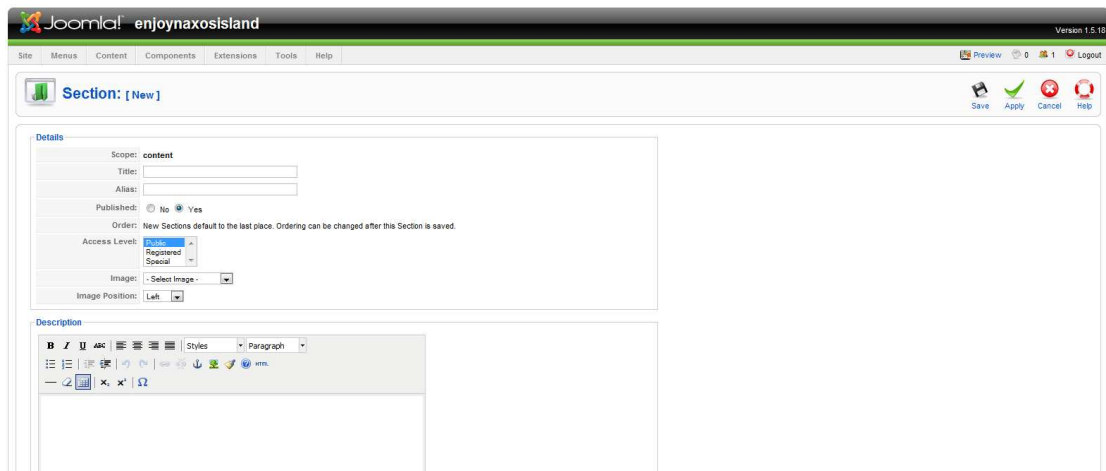


Εικόνα 4 Διαχείριση Αρχικής Σελίδας

Στις εικόνες 5 και 6 έχουμε την κεντρική σελίδα των υπερκατηγοριών καθώς την φόρμα δημιουργίας μιας νέας.



Εικόνα 5 Διαχείριση Υπερκατηγοριών

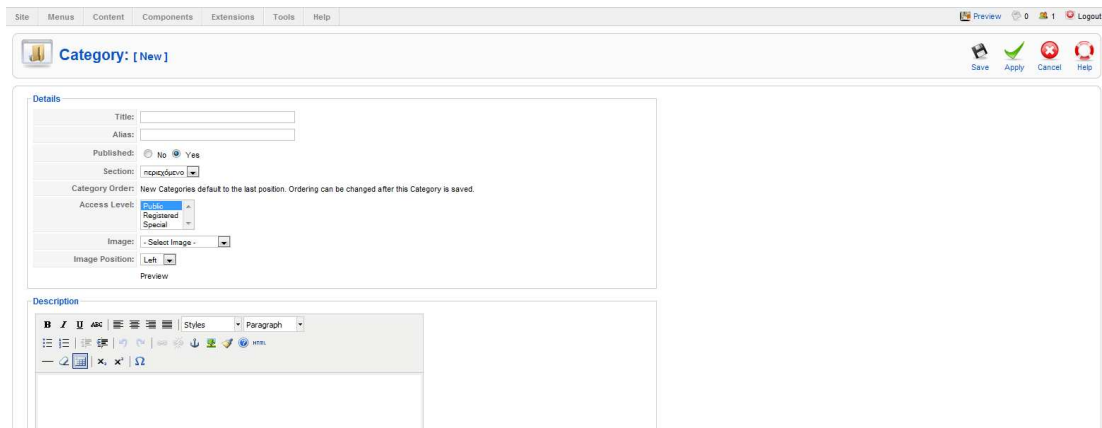


Εικόνα 6 Δημιουργία Υπερκατηγορίας

Στις παρακάτω δύο εικόνες έχουμε την διαχείριση των κατηγοριών καθώς και την σελίδα δημιουργίας νέας.

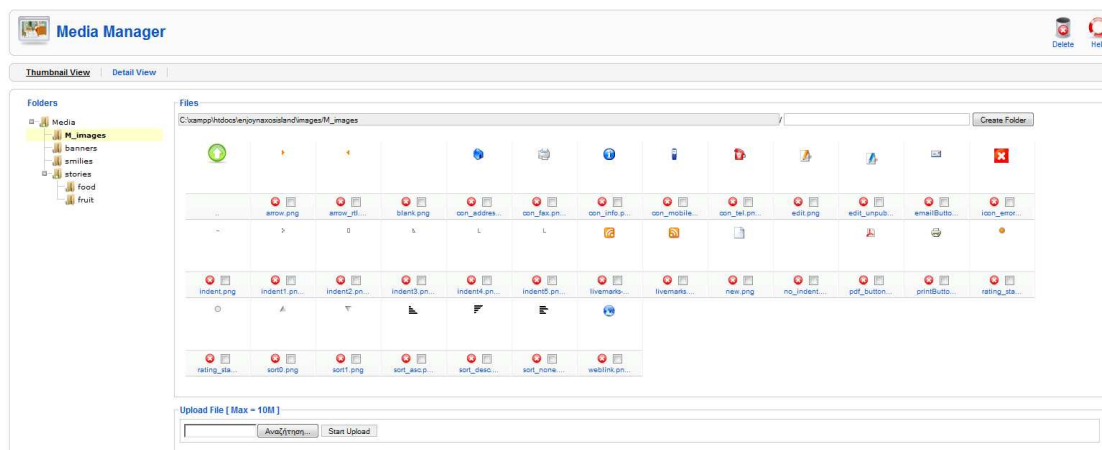


Εικόνα 7 Κατηγορίες

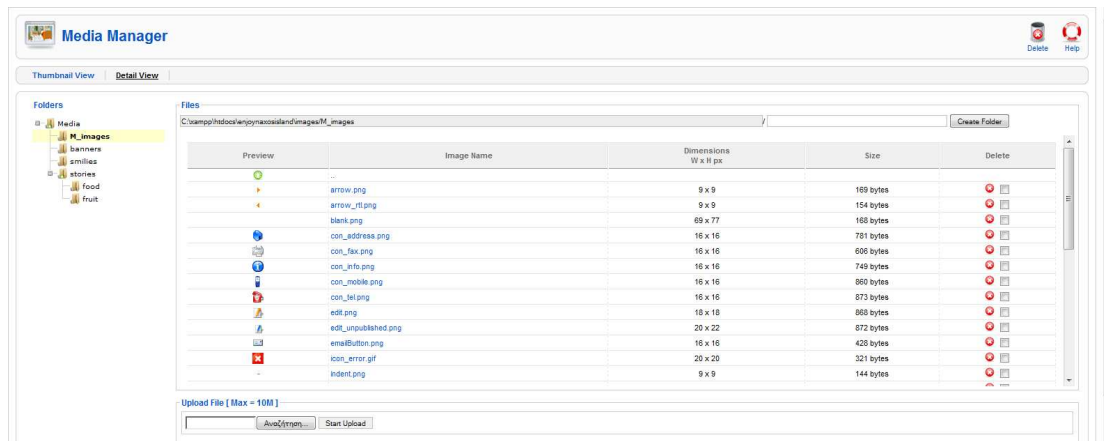


Εικόνα 8 Νέα Κατηγορία

Στις δύο εικόνες που ακολουθούν παρατηρούμε τους δύο τρόπους προβολής της διαχείρισης πολυμέσων του Joomla.

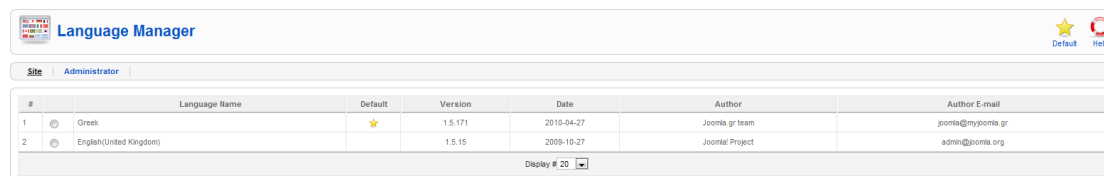


Εικόνα 9 thumbnail προβολή πολυμέσων



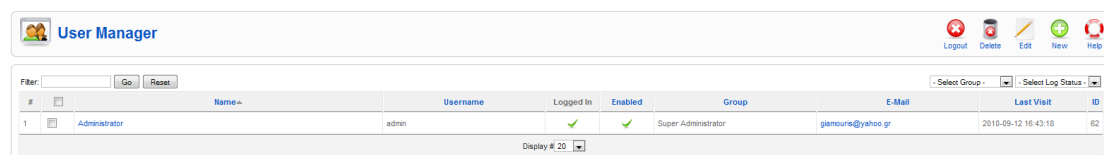
Εικόνα 10 λεπτομερής προβολή πολυμέσων

Στην επόμενη εικόνα μπορούμε να επιλέξουμε την γλώσσα που θα έχει η ιστοσελίδα μας καθώς και η διαχείριση αυτής απ' τις διαθέσιμες γλώσσες που έχουμε

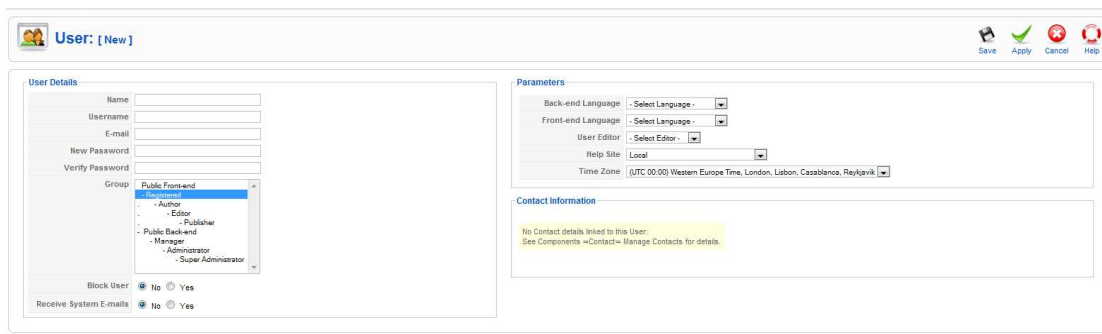


Εικόνα 11 διαχείριση γλώσσας

Σε αυτήν την σελίδα βλέπουμε την διαχείριση χρηστών καθώς και την σελίδα δημιουργίας νέου χρήστη και όλες τις παραμέτρους που μας προσφέρει το Joomla ως προς τις ιδιότητες και τα δικαιώματά τους.

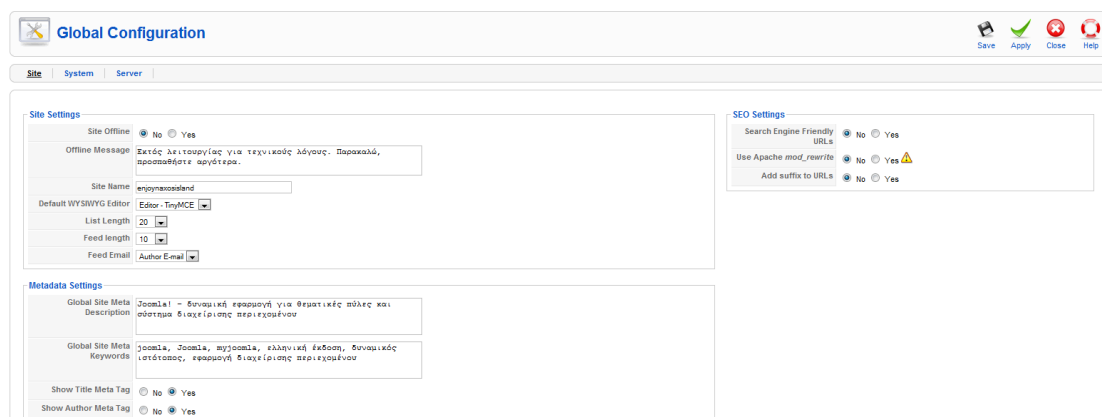


Εικόνα 12 Διαχείριση χρηστών



Εικόνα 13 Προσθήκη Νέου Χρήστη

Στις τρεις επόμενες εικόνες έχουμε τις γενικές ρυθμίσεις του Joomla και την κατηγοριοποίησή τους σε «ιστοσελίδας», «συστήματος» και «εξυπηρετητή».



Εικόνα 14 Ρυθμίσεις Ιστοσελίδας

Εικόνα 15 Ρυθμίσεις Συστήματος

Εικόνα 16 Ρυθμίσεις Εξυπηρητητή

4.3 JOOMLA ΚΑΙ ΔΥΝΑΤΟΤΗΤΕΣ

Σε αυτήν την ενότητα θα δούμε τα εργαλεία του Joomla με τα οποία διαμορφώσαμε την δομή και την μορφή του της ιστοσελίδας μας καθώς και ενεργοποιήσαμε και ρυθμίσαμε τις δυνατότητές της.

Στις εικόνες που ακολουθούν παρατηρούμε τα εργαλεία δημιουργίας διαμόρφωσης και διαχείρισης των menu που χρησιμοποιήσαμε.

#	Title	Type	Menu Item(s)	# Published	# Unpublished	# Trash	# Modules	ID
1	Main Menu	mainmenu		4	-	-	1	1
2	sidemenu	sidemenu		10	-	-	1	2

Display # 20

Εικόνα 17 Διαχείριση Menu

#	Name	ID	Menu	Type
Display # 20				

Joomla! is Free Software released under the GNU/GPL License.

Εικόνα 18 Κάδος Menu

#	Menu Item	Default	Published	Order	Access Level	Type	ItemID
1	Αρχική Σελίδα			1	Public	Articles » Front Page	1
2	Το Νέο μας			2	Public	Articles » Front Page	4
3	Διαγωνιά			3	Public	EstabAgent	6
4	Επικοινωνία			4	Public	Contacts » Contact	5

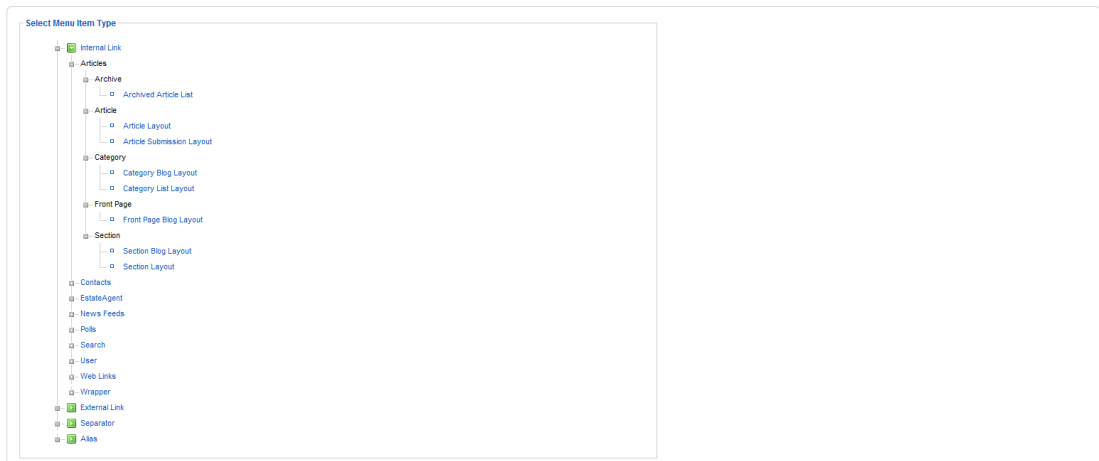
Display # 20

Εικόνα 19 Διαχείριση κεντρικού Menu

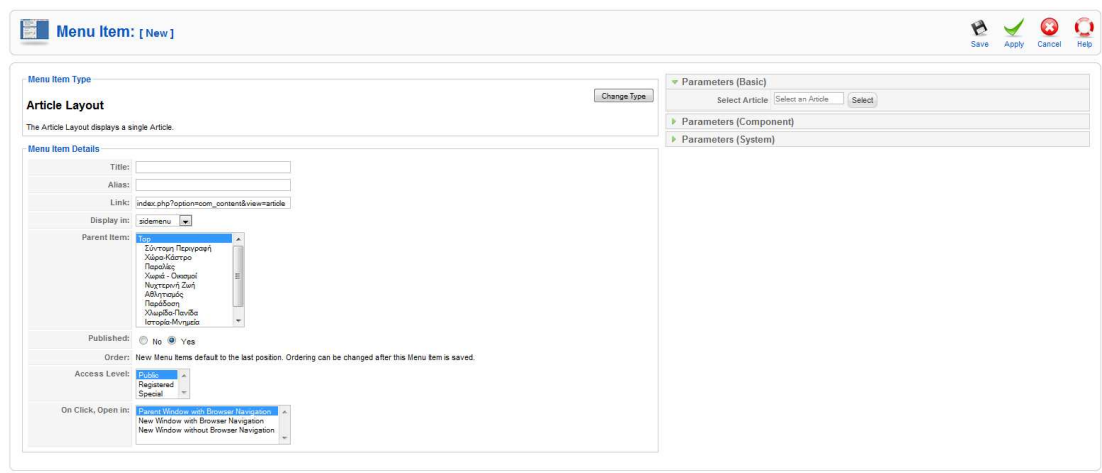
#	Menu Item	Default	Published	Order	Access Level	Type	ItemID
1	Σύντομη Περιγραφή			1	Public	Articles » Article	9
2	Χάρη-Κόστρο			2	Public	Articles » Article	10
3	Παραλίες			3	Public	Articles » Category	13
4	Χωριά - Οικισμοί			4	Public	Articles » Category	2
5	Ναυτιλία Ζωή			5	Public	Articles » Article	7
6	Αθλητισμός			6	Public	Articles » Article	3
7	Παράδοση			7	Public	Articles » Article	8
8	Χλωρίδα-Πανίδα			8	Public	Articles » Article	11
9	Ιστορία-Μνημεία			9	Public	Articles » Article	12
10	Πώς να έρθετε στη Νάξο			10	Public	Articles » Article	14

Display # 20

Εικόνα 20 Διαχείριση Πλευρικού Menu



Εικόνα 21 Προσθήκη Menu - Έπιλογή είδους



Εικόνα 22 Φόρμα Δημιουργίας Menu

Παρακάτω αναλύουμε ένα ιδιαίτερα χρήσιμο εργαλείο προβολής το Banner. Στις επόμενες εικόνες μπορούμε να δούμε την διαχείριση των banner καθώς και την δημιουργία νέου, την διαχείριση πελατών και την σύνδεσή τους με το αντίστοιχο banner καθώς και την διαχείριση των κατηγοριών των πελατών όπως και την δημιουργία νέων κατηγοριών για τους πελάτες με banner.

Banner Manager

[Publish](#)
[Unpublish](#)
[Copy](#)
[Delete](#)
[Edit](#)
[New](#)
[Parameters](#)
[Help](#)

Banners Clients Categories

Filter: [Go](#) [Reset](#)

#	Name	Client	Category	Published	Order	Sticky	Impressions	Clicks	Tags	ID
1	zastavel	zastavel	Συνεργός	✓	1	No	213 of Unlimited	2 - 0.94%		1
2	Naxos Tours	Naxos Tours	Συνεργός	✓	2	Yes	188 of Unlimited	0 - 0.00%		2
3	mikesbikes	mikesbikes	Συνεργός	✓	3	No	105 of Unlimited	2 - 1.29%		3
4	cineastra	cineastra	Συνεργός	✓	4	No	152 of Unlimited	0 - 0.00%		4
5	flivoskitecenter	flivoskitecenter	Συνεργός	✓	5	No	82 of Unlimited	0 - 0.00%		5
6	Flivosportclub	flivoskitecenter	Συνεργός	✓	6	No	58 of Unlimited	0 - 0.00%		6
7	aktalon	aktalon	Συνεργός	✓	7	No	129 of Unlimited	1 - 0.78%		7
8	butterfly	butterfly	Συνεργός	✓	8	Yes	0 of Unlimited	0 - 0.00%		8
9	choice	choice	Συνεργός	✓	9	No	125 of Unlimited	0 - 0.00%		9
10	emporio	emporio	Συνεργός	✓	10	No	124 of Unlimited	0 - 0.00%		10
11	enetiko	enetiko	Συνεργός	✓	11	No	121 of Unlimited	0 - 0.00%		11
12	escoba	escoba	Συνεργός	✓	12	No	120 of Unlimited	0 - 0.00%		12
13	flamingo	flamingo	Συνεργός	✓	13	No	119 of Unlimited	0 - 0.00%		13

Εικόνα 23 Διαχείριση banner

Details

Name:

Alias:

Show Banner: No Yes

Sticky: No Yes

Order:

Category:

Client Name:

Impressions Purchased: Unlimited

Click URL:

Clicks: 0 [Reset Clicks](#)

Custom Banner Code:

Description/Notes:

Banner Image Selector:

Width:

Height:

Banner Image:

Εικόνα 24 Δημιουργία banner

Banner Client Manager

[Delete](#)
[Edit](#)
[New](#)
[Help](#)

Banners Clients Categories

Filter: [Go](#) [Reset](#)

#	Client Name	Contact	# Banners	ID
1	aktalon	aktalon	1	6
2	butterfly	butterfly	1	7
3	choice	choice	1	8
4	cineastra	cineastra	1	4
5	emporio	emporio	1	9
6	enetiko	enetiko	1	10
7	escoba	escoba	1	11
8	flamingo	flamingo	1	12
9	flivoskitecenter	flivoskitecenter	2	5
10	traulino	traulino	1	13
11	mikesbikes	mikesbikes	1	3
12	Naxos Tours	Naxos Tours	1	2
13	zastavel	zastavel	1	1

Display # 20

Εικόνα 25 Διαχείριση banner Πελατών

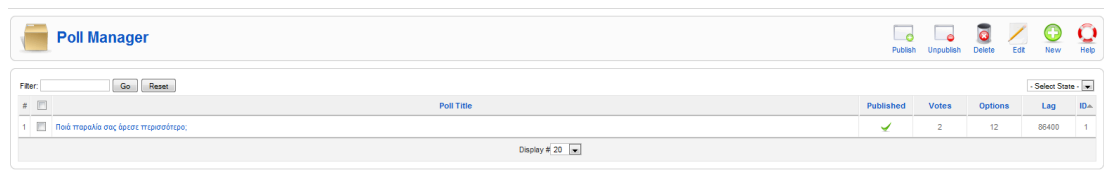
Εικόνα 26 Προσθήκη Πελάτη

#	Title	Published	Order	Access Level	ID
1	Ενδιάμετος	✓	1	Public	7

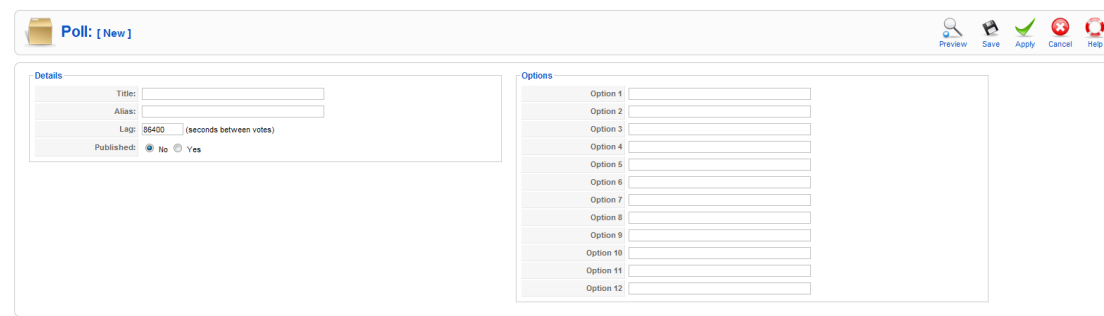
Εικόνα 27 Διαχείριση Κατηγοριών Πελατών

Εικόνα 28 Δημιουργία Κατηγοριών Πελατών

Άλλο ένα πολύ χρήσιμο εργαλείο είναι η ψηφοφορία (poll) του οποίου η διαχείριση και η δημιουργία φαίνονται παρακάτω.



Εικόνα 29 Διαχείριση Poll



Εικόνα 30 Δημιουργία Νέου Poll

Ένα πολύ βασικό εργαλείο είναι το Extension Manager. Εδώ βλέπουμε την δυνατότητα που μας προσφέρει να διαχειριστούμε ό,τι εγκαθηστούμε στην ιστοσελίδα μας και πιο συγκεκριμένα τους modules, τα plug-ins, τις γλώσσες και τα πρότυπα(templates).



Εικόνα 31 Εγκατάσταση components

#	Component	Enabled	Version	Date	Author	Compatibility
1	Banners	✓	1.5.0	April 2006	Joomla! Project	✓
2	EstateAgent	✓	1.5.0b1	Sep 2008	Estate Agent Improved team	✓
3	Newsfeeds	✓	1.5.0	April 2006	Joomla! Project	✓
4	Polls	✓	1.5.0	July 2004	Joomla! Project	✓
5	Webinars	✓	1.5.0	April 2006	Joomla! Project	✓
6	Content Page	✓	1.5.0	April 2006	Joomla! Project	✓
7	Cache Manager	✓	1.5.0	2006	Joomla! Project	✓
8	Configuration Manager	✓	1.5.0	2006	Joomla! Project	✓
9	Contact Items	✓	1.0.0	2006	Joomla! Project	✓
10	Control Panel	✓	1.5.0	2006	Joomla! Project	✓
11	Installation Manager	✓	1.5.0	2006	Joomla! Project	✓
12	Language Manager	✓	1.5.0	2006	Joomla! Project	✓
13	MailTo	✓	1.5.0	April 2006	Joomla! Project	✓
14	Mass Mail	✓	1.5.0	April 2006	Joomla! Project	✓
15	Media Manager	✓	1.5.0	April 2006	Joomla! Project	✓
16	Menus Manager	✓	1.5.0	2006	Joomla! Project	✓
17	Messaging	✓	1.5.0	2006	Joomla! Project	✓
18	Module Manager	✓	1.5.0	2006	Joomla! Project	✓
19	Plugin Manager	✓	1.5.0	2006	Joomla! Project	✓
20	Search	✓	1.5.0	2006	Joomla! Project	✓

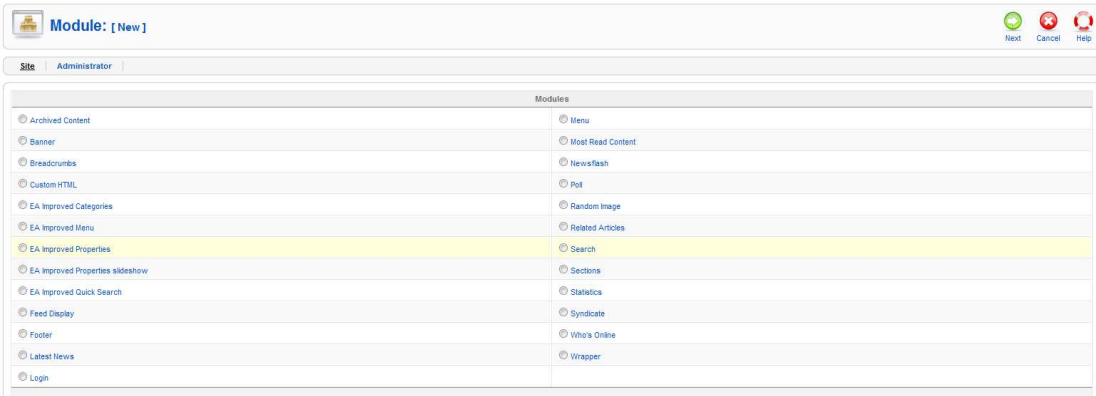
Εικόνα 32 Διαχείριση Components

#	Module	Client	Version	Date	Author	Compatibility
1	mod_banners	Site	1.5.0	July 2006	Joomla! Project	✓
2	mod_ea_categories	Site	1.5.x.1	Sep 2008	EA Improved team	✓
3	mod_ea_menu	Site	1.5.x.1	Mar 2009	EA Improved team	✓
4	mod_ea_properties	Site	1.5.x.1	Sep 2008	EA Improved team	✓
5	mod_ea_search	Site	1.5.x.2	Jan 2009	EA Improved team	✓
6	mod_ea_slideshow	Site	1.5.x.2	Sep 2008	EA Improved team	✓
7	mod_newsflash	Site	1.5.0	July 2006	Joomla! Project	✓
8	mod_poll	Site	1.5.0	July 2006	Joomla! Project	✓
9	mod_search	Site	1.0.0	July 2004	Joomla! Project	✓
10	mod_latest	Admin	1.0.0	July 2004	Joomla! Project	✓
11	mod_tagged	Admin	1.0.0	January 2005	Joomla! Project	✓
12	mod_menu	Admin	1.0.0	March 2006	Joomla! Project	✓
13	mod_popular	Admin	1.0.0	July 2004	Joomla! Project	✓
14	mod_stats	Admin	1.0.0	July 2004	Joomla! Project	✓
15	mod_status	Admin	1.5.0	Feb 2006	Joomla! Project	✓
16	mod_submenu	Admin	1.0.0	Feb 2006	Joomla! Project	✓
17	mod_tle	Admin	1.0.0	Nov 2005	Joomla! Project	✓
18	mod_mainmenu	Site	1.5.0	July 2006	Joomla! Project	✓

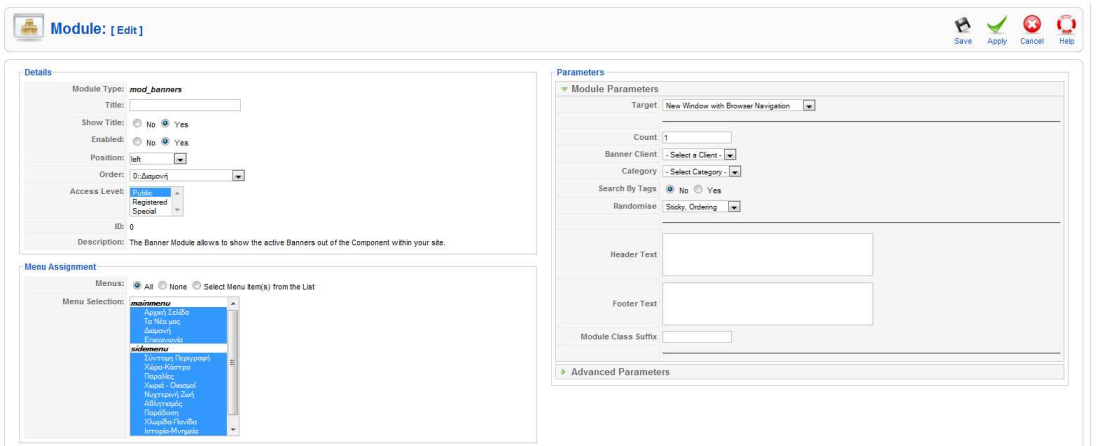
Εικόνα 33 Διαχείριση Modules

#	Module Name	Enabled	Order	Access Level	Position	Pages	Type	ID
1	Διοργάνη	✓	1	Public	left	All	mod_ea_categories	16
2	Η Νόζα	✓	2	Public	left	All	mod_mainmenu	24
3	EA Improved Menu	✗	3	Public	left	All	mod_ea_menu	17
4	Το πιο δημοφιλές	✓	3	Public	left	All	mod_ea_properties	18
5	Το Νέο μας	✗	4	Public	left	Varies	mod_newsflash	22
6	EA Improved Quick Search	✗	6	Public	left	All	mod_ea_search	20
7	flexvot1	✓	7	Public	left	All	mod_banners	29
8	aktion	✓	8	Public	left	All	mod_banners	30
9	choice	✓	9	Public	left	All	mod_banners	31
10	emporio	✓	10	Public	left	All	mod_banners	32
11	enetho	✓	11	Public	left	All	mod_banners	33
12	Αναζήτηση	✓	0	Public	right	All	mod_search	21
13	Οι πιο πρόσφατες καταχωρήσεις	✓	2	Public	right	All	mod_ea_slideshow	19
14	Ποιά παραλία σας άρεσε περισσότερο;	✓	3	Public	right	All	mod_poll	23
15	zastvavi	✓	4	Public	right	All	mod_banners	25
16	Naxos tours	✓	5	Public	right	All	mod_banners	26

Εικόνα 34 Διαχείριση Modules



Εικόνα 35 Δημιουργία Νέου Module πρώτο βήμα επιλογή είδους



Εικόνα 36 φόρμα δημιουργίας module

#	Plugin	Type	Version	Date	Author	Compatibility
1	Authentication - Gmail	authentication	1.5	February 2006	Joomla! Project	✓
2	Authentication - OpenID	authentication	1.5	February 2006	Joomla! Project	✓
3	Content - Code Highlighter (GeSHi)	content	1.5	November 2005	Joomla! Project	✓
4	Content - Email Cloaking	content	1.5	November 2005	Joomla! Project	✓
5	Content - Load Modules	content	1.5	November 2005	Joomla! Project	✓
6	Button - Image	editors-xtd	1.0.0	August 2004	Joomla! Project	✓
7	Button - Pagebreak	editors-xtd	1.5	August 2004	Joomla! Project	✓
8	Button - Readmore	editors-xtd	1.5	March 2006	Joomla! Project	✓
9	Search - Categories	search	1.5	November 2005	Joomla! Project	✓
10	Search - EstateAgent	search	1.0	Mar 2009	EA Improved team	✓
11	Search - Newsfeeds	search	1.5	November 2005	Joomla! Project	✓
12	Search - Sections	search	1.5	November 2005	Joomla! Project	✓
13	System - Debug	system	1.5	December 2006	Joomla! Project	✓
14	System - SEF	system	1.5	December 2007	Joomla! Project	✓
15	User - Joomla!	user	1.5	December 2006	Joomla! Project	✓
16	Authentication - Joomla!	authentication	1.5	November 2005	Joomla! Project	✓
17	Authentication - LDAP	authentication	1.5	November 2005	Joomla! Project	✓
18	Content - Page Navigation	content	1.5	January 2006	Joomla! Project	✓

Εικόνα 37 Διαχείριση Plug-ins

Plugin Manager Enable Disable Edit Help

Filter: Select Type: Select State:

#	Plugin Name	Enabled	Order	Access Level	Type	File	ID
1	Authentication - Joomla	✓	1	Public	authentication	joomla	1
2	Authentication - LDAP	✗	2	Public	authentication	ldap	2
3	Authentication - OpenID	✗	3	Public	authentication	openid	4
4	Authentication - Gmail	✗	4	Public	authentication	gmail	3
5	Content - Page Navigation	✓	2	Public	content	pagenavigation	17
6	Content - Rating	✓	4	Public	content	vote	13
7	Content - Email Cloaking	✓	5	Public	content	emailcloak	14
8	Content - Code Highlighter (GeSHi)	✗	5	Public	content	geshi	15
9	Content - Load Module	✓	6	Public	content	loadmodule	16
10	Content - Pagebreak	✓	10000	Public	content	pagebreak	12
11	Editor - No Editor	✓	0	Public	editors	none	18
12	Editor - TinyMCE	✓	0	Public	editors	tinymce	19
13	Editor - XStandard Like 2.0	✗	0	Public	editors	xstandard	20
14	Editor Button - Image	✓	0	Public	editors-xtd	image	21
15	Editor Button - Pagebreak	✓	0	Public	editors-xtd	pagebreak	22
16	Editor Button - Readmore	✓	0	Public	editors-xtd	readmore	23

Εικόνα 38 Διαχείριση Plug-ins

Extension Manager Uninstall

Install Components Modules Plugins Languages Templates

Only those Languages that may be uninstalled are accessible. The default Language may not be removed. All

#	Language	Client	Version	Date	Author
1	<input type="checkbox"/> Greek	Site	1.5.171	2010-04-27	Joomla.gr team
2	<input type="checkbox"/> English(UK)	Site	1.5.15	2009-10-27	Joomla! Project
3	<input type="checkbox"/> Greek	Admin	1.5.17	2010-04-27	Joomla.gr team
4	<input type="checkbox"/> English(UK)	Admin	1.5.15	2009-10-27	Joomla! Project

Display # 20

Εικόνα 39 Διαχείριση Προτύπων

Template Manager Default Edit Help

Site Administrator

#	Template Name	Default	Assigned	Version	Date	Author
1	beez			1.0.0	19 February 2007	Angie Radtke/Robert Deutz
2	JA_Purity			1.2.0	12/28/07	Joomla! Art.com
3	rhuuk_milkyway	★		1.0.2	11/20/06	Andy Miller

Display # 20

Εικόνα 40 Διαχείριση Προτύπων

Template: [Edit] Preview Edit HTML Edit CSS Save Apply Close Help

Details

Name: RRUUK Milkyway template

Description: MilkyWay is a fresh new template for Joomla!. The clean design of this template makes it very lightweight and fast.

Menu Assignment

Menus: Cannot assign default template.

Parameters

The parameter file templates/rhuuk_milkyway/params.ini is writable!

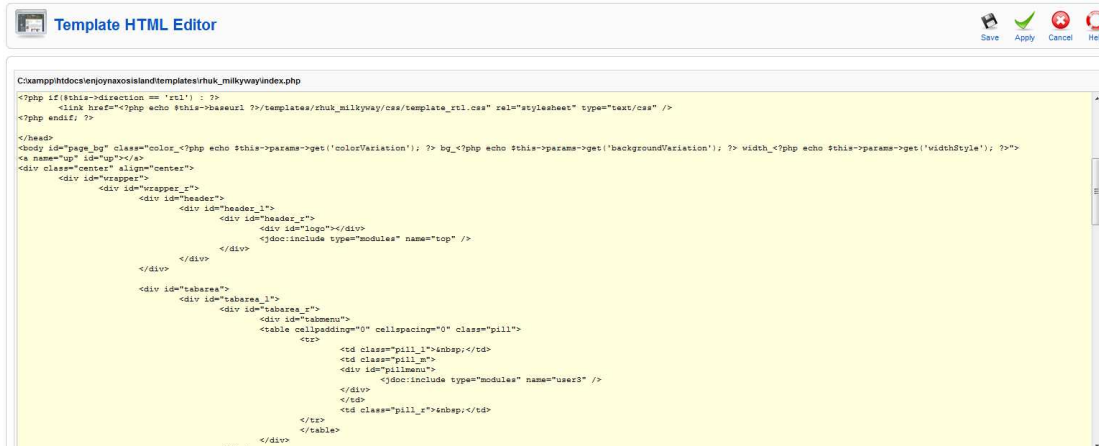
Color Variation:

Background Variation:

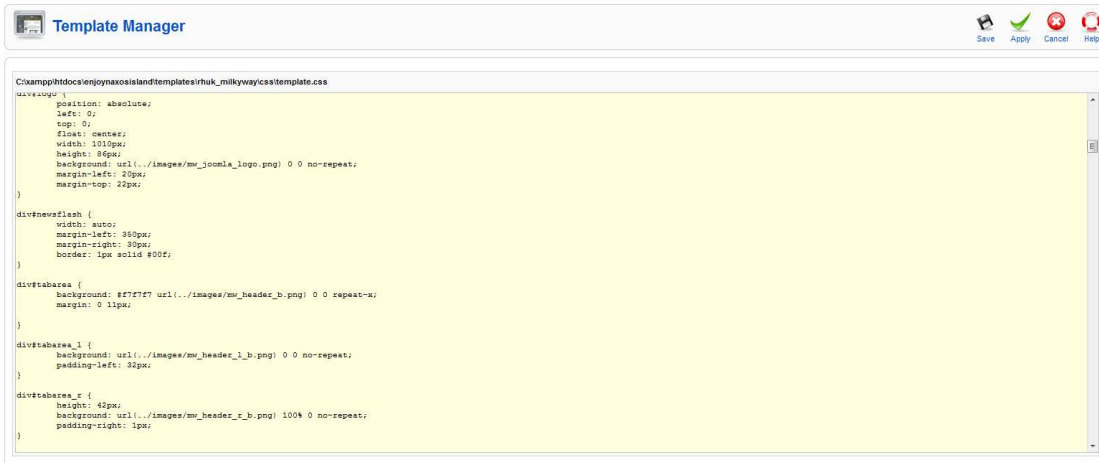
Template Width:

Joomla! is Free Software released under the GNU/GPL License.

Εικόνα 41 Κεντρική Σελίδα Επεξεργασίας Προτύπου



Εικόνα 42 Επεξεργασία html κώδικα προτύπου



Εικόνα 43 επεξεργασία css προτύπου



Εικόνα 44 Γλωσσών

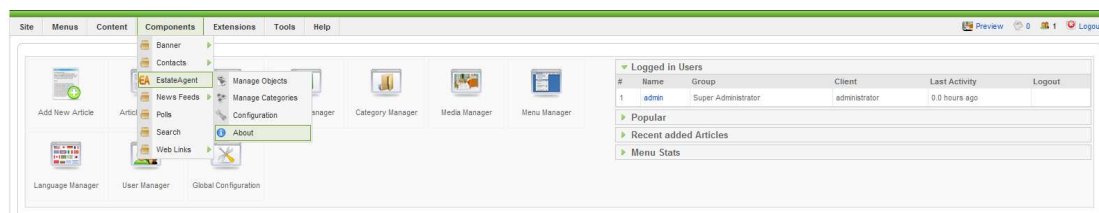
ΚΕΦΑΛΑΙΟ 5

5.1 TO PLUG-IN HOTELAGENT

Σε αυτό το κεφάλαιο θα δούμε το hotelagent το οποίο μας έδωσε την δυνατότητα να προσθέσουμε, να διαχειριστούμε και να προβάλουμε τους τύπους διαμονής τις Νάξου , προσφέροντας μας την δυνατότητα να εισάγουμε πολλές λεπτομέρειες και εικόνες για τα ξενοδοχεία και τα ενοικιαζόμενα δωμάτια που φιλοξενεί η ιστοσελίδα μας. Ένα πολύ σημαντικό χαρακτηριστικό του hotelagent είναι και η εμφάνιση της ακριβούς θέσης του εκάστοτε τύπου διαμονής μέσω της ενσωματωμένης εφαρμογής GoogleMaps. Ακολουθούν περισσότερες λεπτομέρειες.

5.2 HOTELAGENT ΚΑΙ ΡΥΘΜΙΣΕΙΣ

Στις επόμενες σελίδες θα δούμε την πληθώρα των ρυθμίσεων που μας προσφέρει το hoteagent και έτσι μας δίνει την δυνατότητα για ένα πολύ καλό αποτέλεσμα.



Εικόνα 45 Η πρόσβαση στο hotelagent

configuration.php is writeable

General Categories Properties Search Gallery FrontEnd MAP24.com Miscellaneous GoogleMap

EstateAgent Offline: No

Display errors and warnings: No

Offline Message: The εντομακοσλαλανδ is down for maintenance.
Please check back again soon.

Square Measure Unit: metro (m²)

Currency: € Decimal Digits: 0

Currency Format: 19.999,00

Currency sign before number: Yes

Display Copyright: No

Template: shadowbox

RSS Count: 10 (Zero to disable)

Estate Agent Improved v. 1.5.0 pro b1
© 2010 by EAImproved team

Εικόνα 46 HoteAgent γενικές ρυθμίσεις

Save Back

Manage Objects Manage Categories Configuration About

Estate Agent Improved

configuration.php is writeable

General Categories Properties Search Gallery FrontEnd MAP24.com Miscellaneous GoogleMap

Category Layout On Frontpage: Two Columns

Show Subcategories On Frontpage?: No

Show Categories Overview Recursive?: No

Show Category Links With Amount Of Containing Entries?: No

Estate Agent Improved v. 1.5.0 pro b1
© 2010 by EAImproved team

Εικόνα 47 HotelAgent ρυθμίσεις κατηγοριών τύπων διαμονής

Save Back

Manage Objects Manage Categories Configuration About

Estate Agent Improved

configuration.php is writeable

General Categories Properties Search Gallery FrontEnd MAP24.com Miscellaneous GoogleMap

Hot Offers Presentation: Yes

Amount Of Hot Offers On Frontpage?: 4

Position Of Hot Offers on first page: Bottom

Coloured Emphasis Of Hot Offers: #77913D

Show Complete Location?: Yes

Show properties Short information?: locate,district,locate,value (valid values are : town,district,locate,value,obj_id separate them with a comma ";")

Property Entries Per Page: 10

Show "Email To Friends"-Button: As Icon

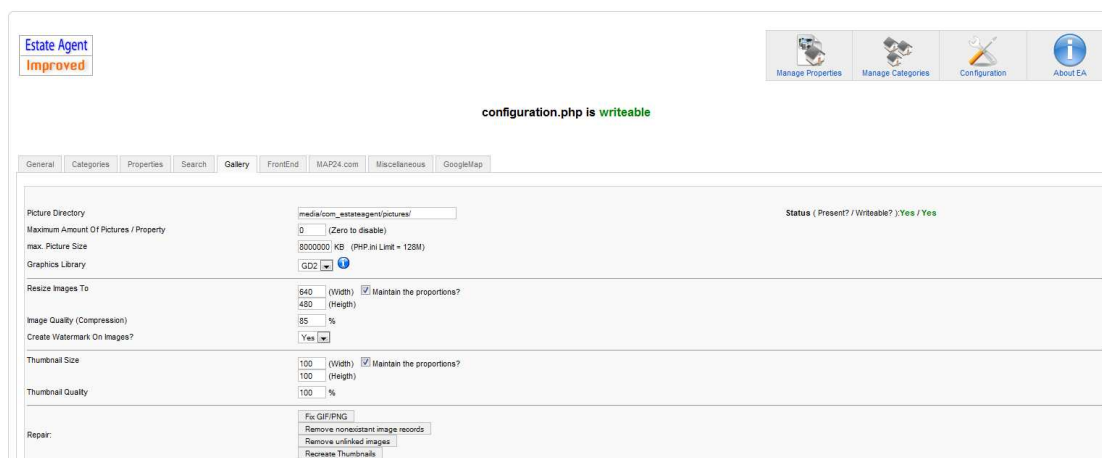
Show print button: As Icon

Enable Capcha Security images?: Yes

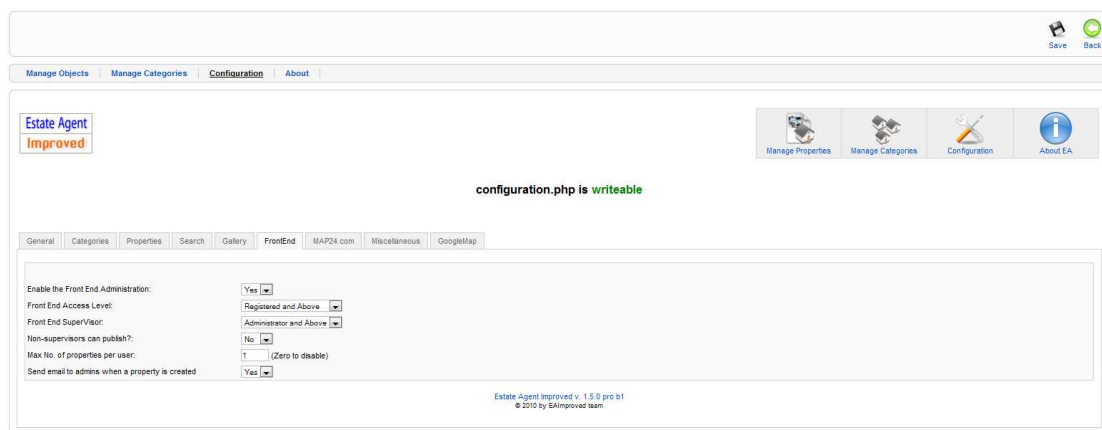
Εικόνα 48 HotelAgent ρυθμίσεις τύπων διαμονής



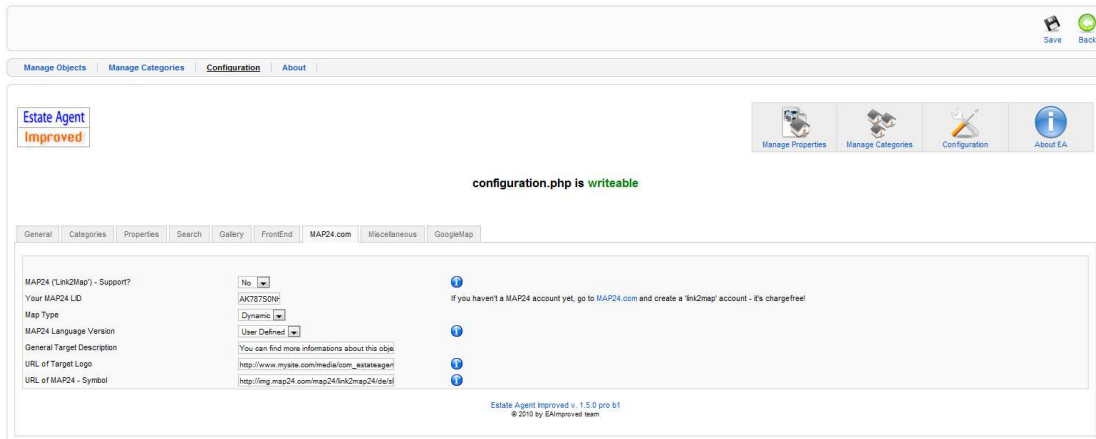
Εικόνα 49 HotelAgent ρυθμίσεις αναζήτησης



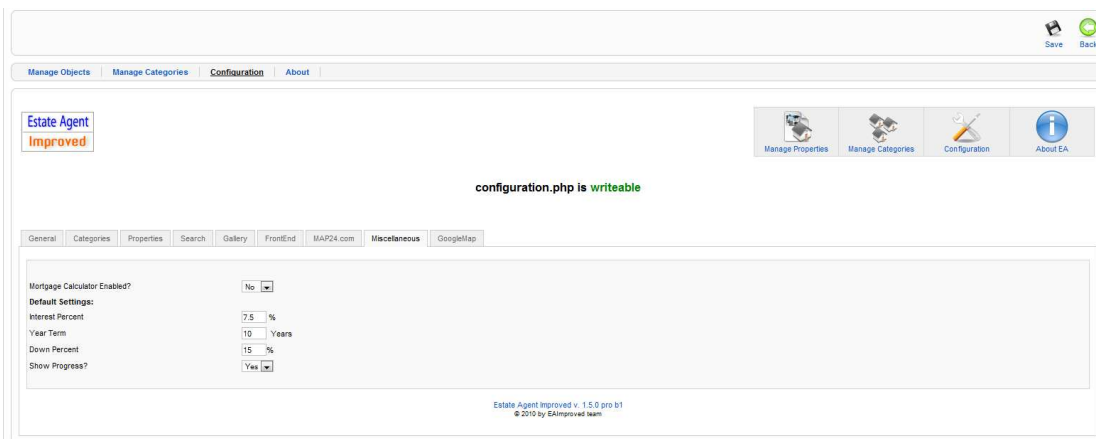
Εικόνα 50 HotelAgent ρυθμίσεις εκθέσεων εικόνων



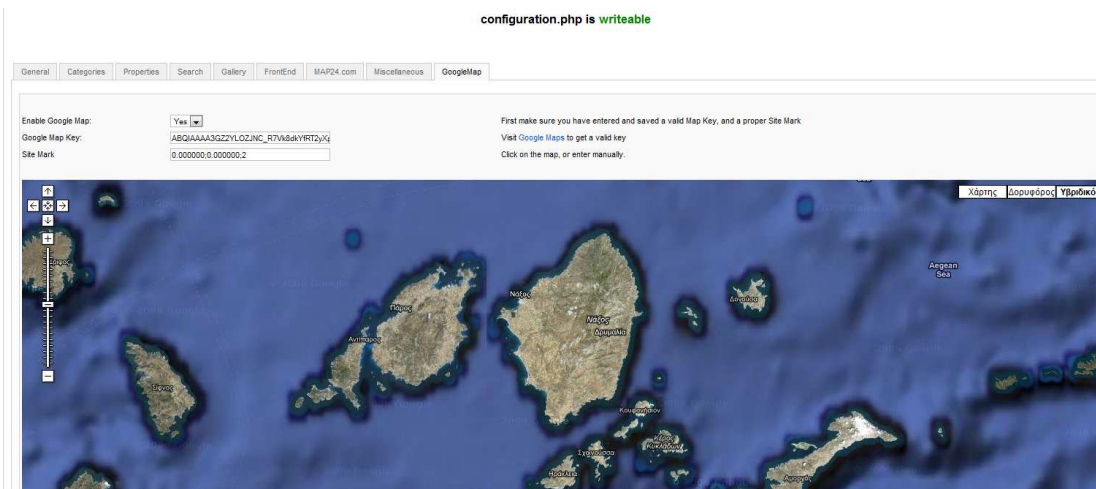
Εικόνα 51 HotelAgent frontend ρυθμίσεις



Εικόνα 52 HotelAgent ρυθμίσεις Map24.com



Εικόνα 53 HotelAgent διάφορες ρυθμίσεις



Εικόνα 54 HotelAgent ρυθμίσεις GoogleMaps

5.2 ΔΙΑΧΕΙΡΙΣΗ ΤΟΠΩΝ ΔΙΑΜΟΝΗΣ

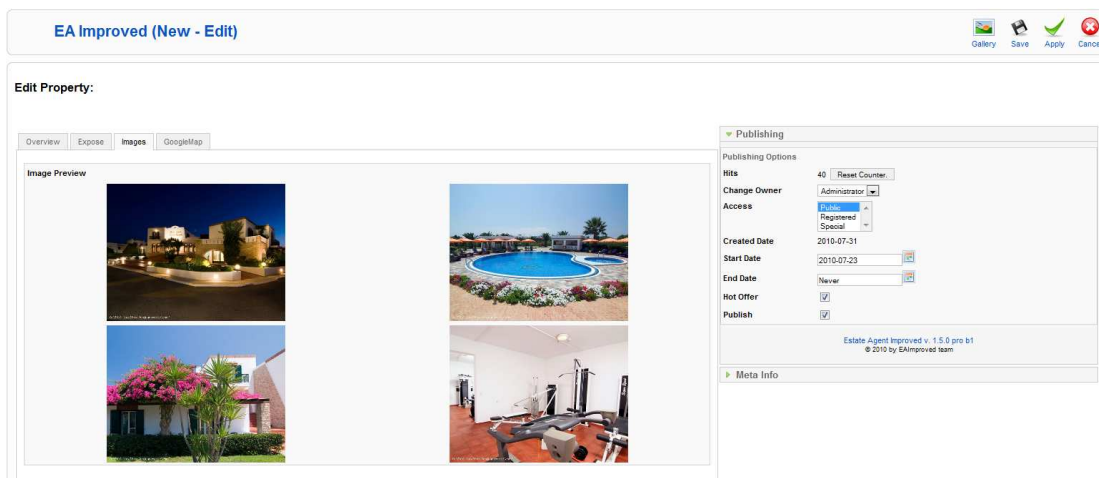
Στις επόμενες εικόνες θα δούμε την δημιουργία και διαχείριση τόπων διαμονής με λεπτομέρεια μέσω του screencapture.

#	Title	Description	Access	Entries	Status
1	Ενοδοσία	Η Ελλάδα αποτελεί έναν από τους δημοφιλέστερους τουριστικούς προορισμούς της Ευρώπης. Σε κάθε νησί της έχουμε επιλέξει για σας το καλύτερο (ενοδοχεία, θέρετρα ενοικιαζόμενα δωμάτια, studios, διακρίσματα, βίλλες... Ο κατάλογός με τα ενοδοχεία και τα ενοικιαζόμενα δωμάτια είναι πλαισιωμένος με φωτογραφίες και πανοραμικά κείμενα. Όλα τα ενοδοχεία, τα δωμάτια και οι βίλλες που έχουμε επιλέξει για σας έχουν το δικό τους διαδικτυακό τόπο (website).	Public	11	✓
2	Ενοικαζόμενο δωμάτιο	Ο κατάλογός με τα ενοικιαζόμενα δωμάτια είναι πλαισιωμένος με φωτογραφίες και πανοραμικά κείμενα. Όλα τα ενοδοχεία, τα δωμάτια και οι βίλλες που έχουμε επιλέξει για σας έχουν το δικό τους διαδικτυακό τόπο (website). Μπορείτε να μπείτε σε αυτό τα αλλε μέσα από τον διαδικτυακό τόπο της εφημερίδαςland.com με ένα μόνο κλικ, και να απορροφήσετε στη κορυφή στα δωμάτια των ενοικιαζόμενων χάρη στις πανοραμικές φωτογραφίες 360. Ετοιμαστέτε για μια ικανή ξένωση!	Public	11	✓
3	Rooms		Public	0	✗

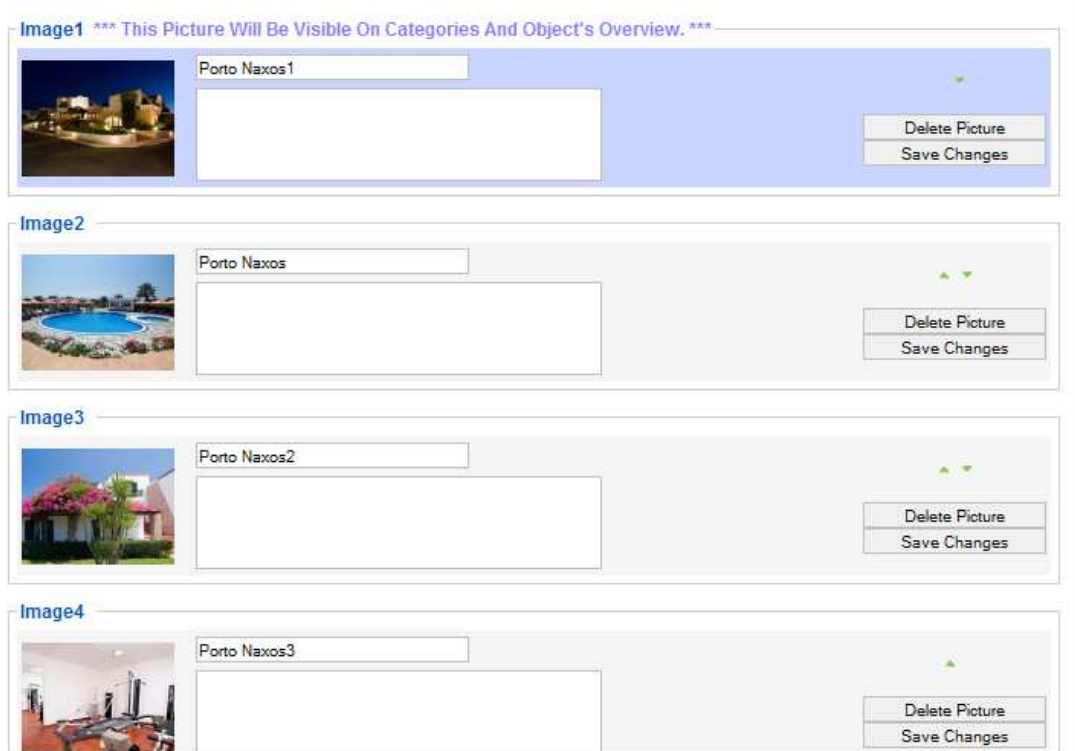
Εικόνα 55 Διαχείριση Κατηγοριών τόπων Διαμονής

Upper Category: Main Category
Order: 0 First
Category Name:
Description:
Access: Public
Start Date: 2010-08-13
End Date: Never
Publish: No

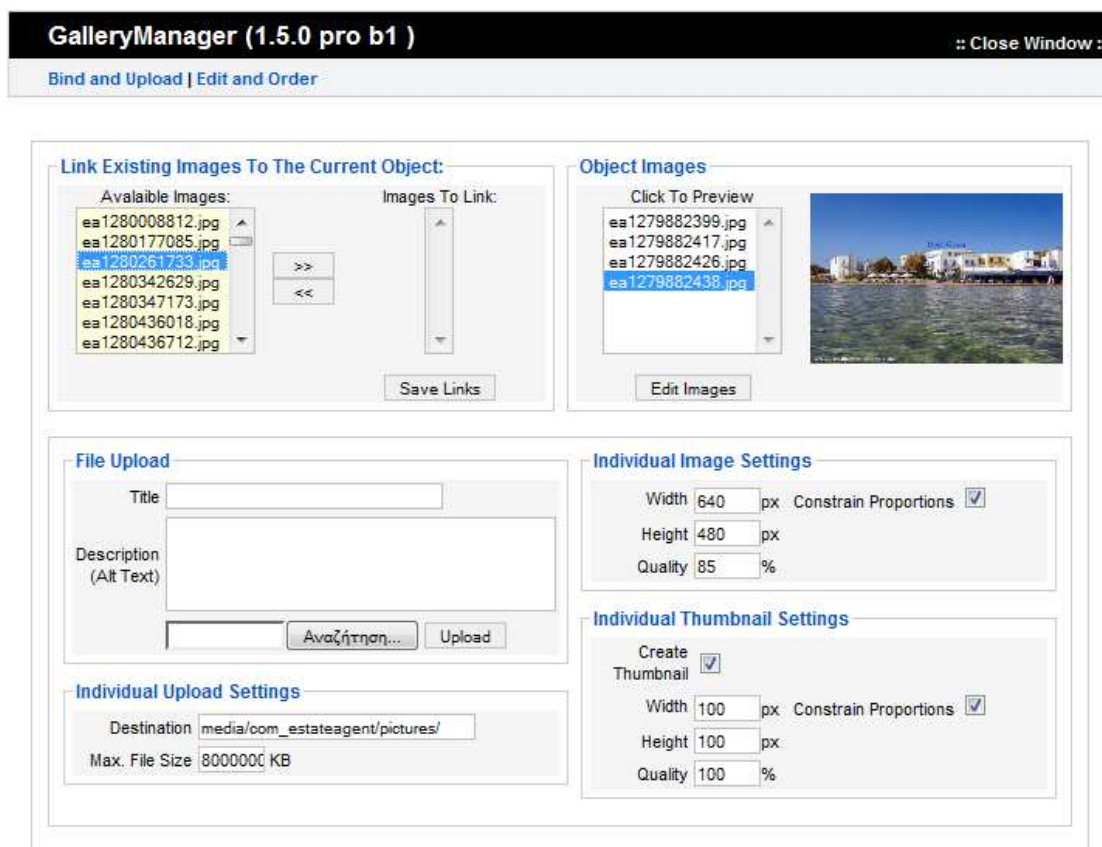
Εικόνα 56 Δημιουργία νέας κατηγορίας τόπου διαμονής



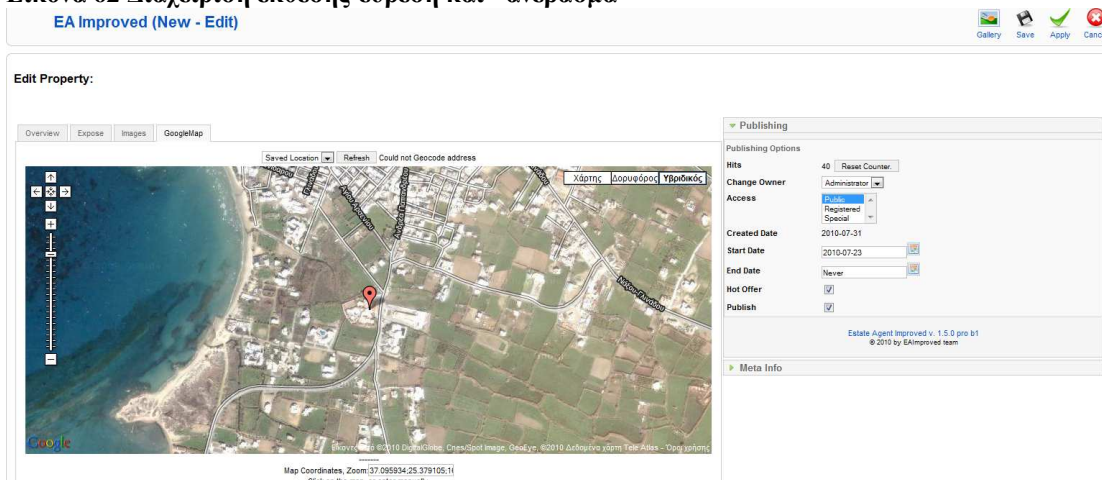
Εικόνα 60 Δημιουργία τύπου διαμονής- Έκθεση



Εικόνα 61 Διαχείριση έκθεσης- προσαρμογή



Εικόνα 62 Διαχείριση έκθεσης-εύρεση και "ανέβασμα"



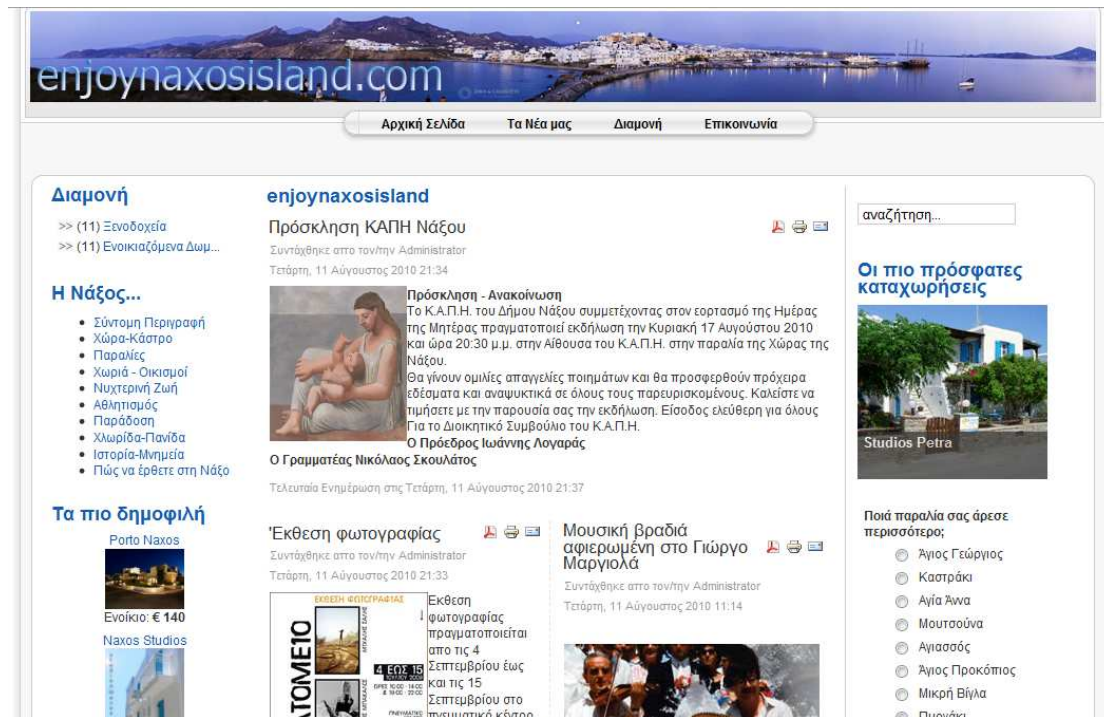
Εικόνα 63 Δημιουργία τόπου διαμονής-προσδιορισμός τοποθεσίας στο GoogleMaps

ΚΕΦΑΛΑΙΟ 6

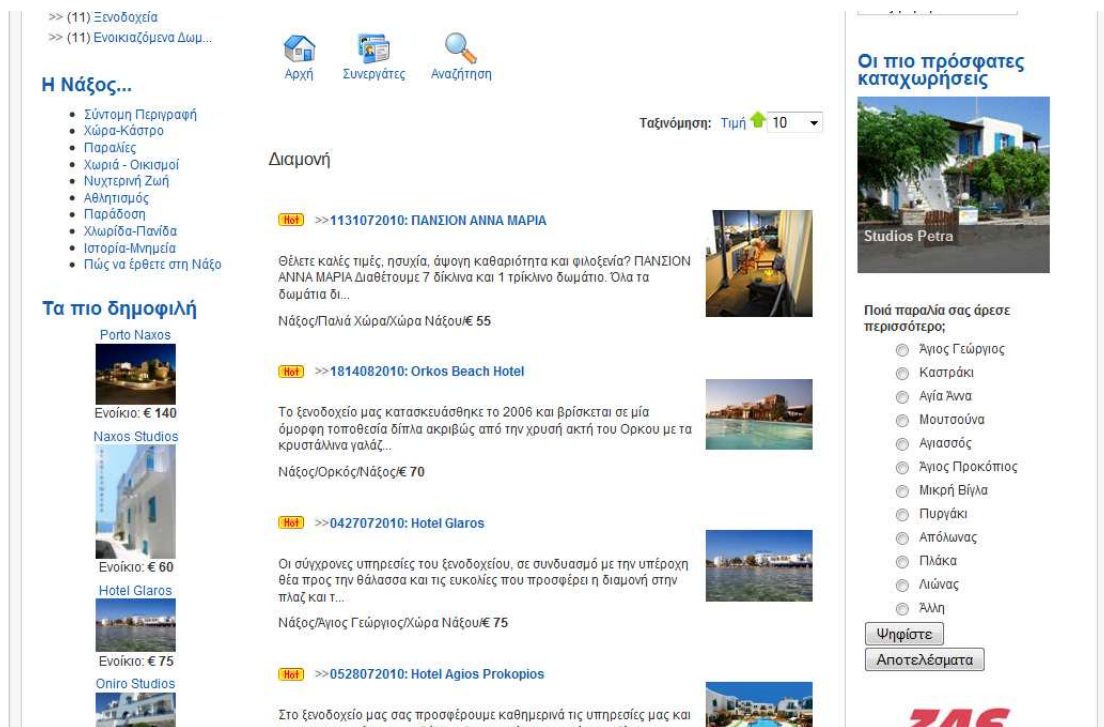
ΟΙ FRONTEND ΧΡΗΣΤΕΣ ΚΑΙ Η ΙΣΤΟΣΕΛΙΔΑ ΜΑΣ

Σε αυτό το κεφάλαιο θα δούμε τις δυνατότητες που παρέχει η ιστοσελίδα μας στον κάθε επισκέπτη(frontend χρήστη) που θα την επισκεφτεί και θα έχει την ευκαιρία να γνωρίσει πολλές πτυχές της νήσου όπως την ιστορία της, τις φυσικές της ομορφιές , την κουλτούρα αλλά και

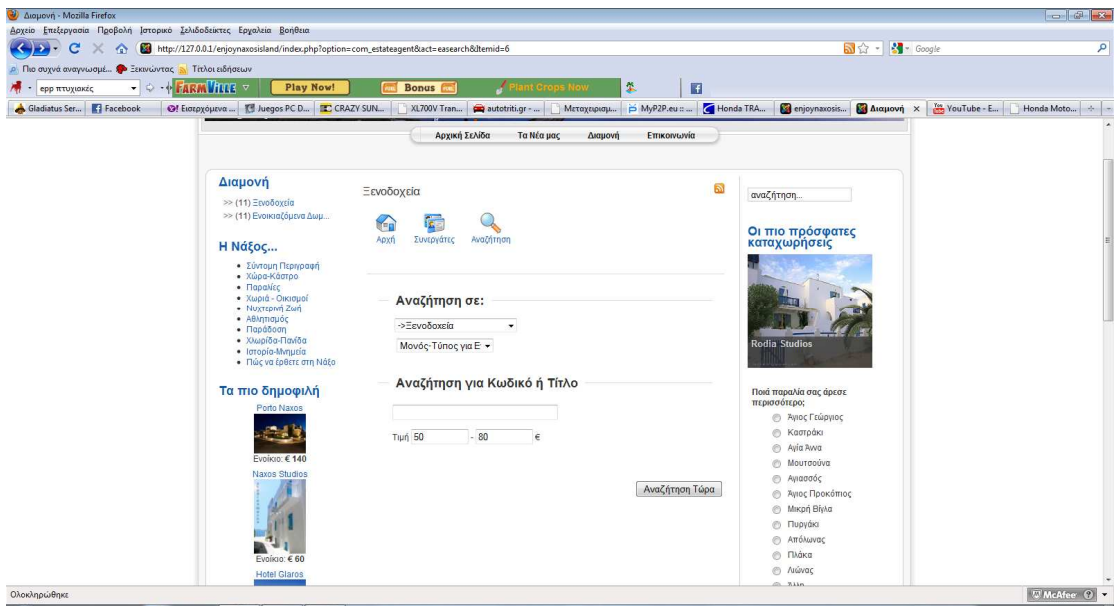
τα ξενοδοχεία και τα ενοικιαζόμενα δωμάτια όπου και μπορεί να κάνει κράτηση, αλλά και άλλες επιχειρήσεις που δραστηριοποιούνται στο νησί μέσω των banners .



Εικόνα 64 Κεντρική Σελίδα



Εικόνα 65 Εμφάνιση τύπων διαμονής



Εικόνα 66 Αναζήτηση με κριτήριο το όνομα ή την τιμή

- Σύνομη Περιγραφή
- Χώρα-Κάστρο
- Παραλίες
- Χωριά - Οικισμοί
- Νυχτερινή Ζωή
- Αθλητισμός
- Παράδοση
- Χλωρίδα-Πανίδα
- Ιστορία-Μνημεία
- Πώς να έρθετε στη Νάξο





Τα πιο δημοφιλή



Porto Naxos
Ενοίκιο: € 140

Naxos Studios
Ενοίκιο: € 60

Hotel Glaros
Ενοίκιο: € 75

ID: 1602082010 >> Plaza beach Hotel


Περιγραφή

82 δωμάτια με μπαλκόνι με θέα το απέραντο γαλάζιο της θάλασσάς, εξοπλισμένα με κάθε σύγχρονο μέσο: κλιματισμός, δορυφορική τηλεόραση, τηλέφωνο, Διαδίκτυο-σύνδεση, θέρμανση, ψυγείο, μουσική, σεσουάρ, χρηματοκιβώτιο, υπηρεσία καθαρισμού ρούχων, υπηρεσία δωματίου. Το ξενοδοχείο Plaza Beach αποτελείται από μονόκλινα, δίκλινα, τρικλινα, μίνι σουίτες, σουίτες καθώς και σουίτες για νεόνυμφους.

Οι μίνι-σουίτες και οι σουίτες είναι πολύ όμορφα δωμάτια με ένα καθιστικό (με ένα καναπέ που μπορεί να μετατραπεί και σε κρεβάτι) και μια κρεβατοκάμαρα ώστε να μπορούν να μείνουν έως και 4 άτομα.

Το μπάνιο στις σουίτες είναι εξοπλισμένο με ένα μικρό Jacuzzi.

Μερικές μίνι-σουίτες και σουίτες έχουν θέα σε κήπο, όχι θέα στη Θάλασσα.



Orkos Beach Hotel


Ποιά παραλία σας άρεσε περισσότερο;

- Άγιος Γεώργιος
- Καστράκι
- Αγία Άννα
- Μουτσούνα
- Αghασός
- Άγιος Προκόπιος
- Μικρή Βίγλα
- Πυργάκι
- Απόλλωνας
- Πλάκα
- Λιάνας
- Άλλη




Εικόνα 67 Παρουσίαση τύπου διαμονής 1από 3

Ενοίκιο: € 110

Naxos Kalimera Hotel



Ενοίκιο: € 110









Τόπος / Περιοχή

Η πανσιόν μας βρίσκεται στην παλιά Χώρα της Νάξου, κάτω από το κάστρο, με θέα το παραδοσιακό μοναστήρι του Αγίου Χρυσοστόμου.

Απέχουμε τρία λεπτά από το λιμάνι και το σταθμό των ΚΤΕΛ (υπάρχουν καθημερινά λεωφορεία προς όλα τα χωριά της Νάξου) και 10 λεπτά από το αεροδρόμιο και την αμμόθις παραλία του Αγίου Προκόπη.


Map



Εικόνα 68 Παρουσίαση τύπου διαμονής 2από 3

Clothes & Accessoires



Δήμος ή Περιοχή Παλιά Χώρα
Νομός Νάξος
Χώρα Ελλάδα

Διεύθυνση επικοινωνίας

Γιαμούρης Γεώργιος
 84300 Χώρα Νάξου
 Νάξος
 Ελλάδα

Το Τηλέφωνο σας: 2285024270
 giatouris@yahoo.gr

Κάντε Κράτηση

Όνομα *


Το Τηλέφωνο σας

Φαξ


E-Mail *

Θέμα Κωδικός ID:1131072010

Μήνυμα *

Κωδικός Ασφαλείας * 

Υποχρεωτικά



Εικόνα 69 Παρουσίαση τύπου διαμονής 3από 3

ΒΙΒΛΙΟΓΡΑΦΙΑ

-Barrie M. North, *Joomla!,A Users Guide second edition*

-Atkinson Leon, *Πλήρης οδηγός της PHP 5*

-www.naxos.gr

-www.inaxos.gr

-www.Joomla.org

-www.joomla.gr

-www.wikipedia.gr

-www.joomlashack.com

- www.w3schools.com