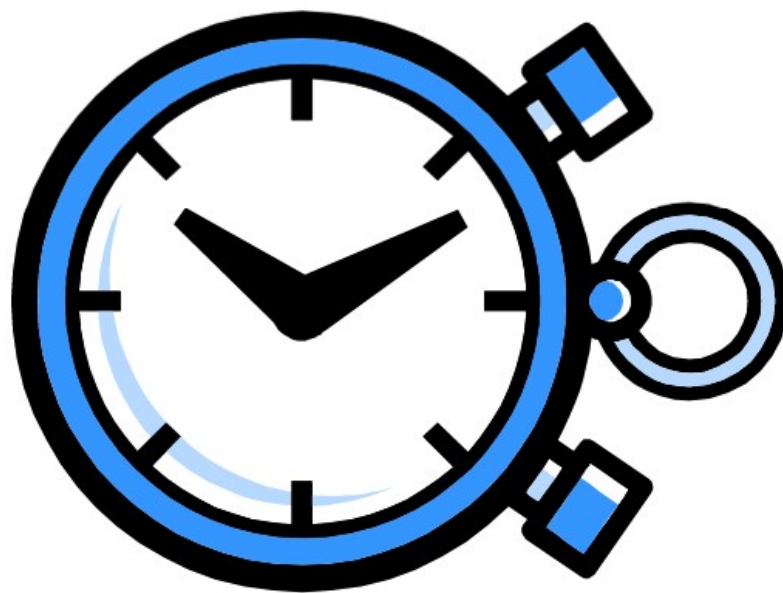


Υλοποίηση Διαδικτυακού τόπου Εκμάθησης της γλώσσας προγραμματισμού Python, με τη χρήση αυτής

Συντάκτες
Πάσσας Νικόλαος
Κεφαλιανός Ιωάννης

Υπεύθυνος Καθηγητής
Σωμαράς Χρήστος



Υλοποίηση Διαδικτυακού τόπου Εκμάθησης της γλώσσας προγραμματισμού Python, με τη χρήση αυτής

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή

Λάρισα .../...../2009

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

- 1.
- 2.
- 3.

Η εν λόγω πτυχιακή παρουσιάζει λεπτομερώς τα βήματα που πρέπει να ακολουθηθούν για την υλοποίηση ενός Διαδικτυακού τόπου Εκμάθησης, με ενσωματωμένο ένα πλήρη σύστημα διαχείρισης περιεχομένου στηριζόμενο στην γλώσσα προγραμματισμού Python. Οι απαιτήσεις που πρέπει να έχει ένα τέτοιο σύστημα εντοπίζονται και μια σε βάθος ανάλυση πραγματοποιείται έως ότου οι απαιτήσεις ενός τέτοιου συστήματος ολοκληρωθούν.

Οι βασικές λειτουργίες αυτής της εφαρμογής είναι :

1. Δυνατότητα λειτουργίας της εφαρμογής μεταξύ διακομιστή – χρήστη
2. Δυνατότητα εισαγωγής του χρήστη στο σύστημα μέσω ασφαλών ελέγχων
3. Δυνατότητα διαφοροποίησης των χρηστών και των προνομίων τους

Δυνατότητα προσθήκης, αλλαγής και διαγραφής των πληροφοριών σε άμεσο χρόν

Πίνακας Περιεχομένων

1. Εισαγωγή.....	6
1.1 Αντικειμενικός σκοπός.....	7
1.2 Αναφορά των κεφαλαίων.....	8
2. Έρευνα	9
2.1 Παρόμοια εφαρμογή (Wikipedia)	9
2.1.1 Προδιαγραφές μιας ηλεκτρονικής εγκυκλοπαίδειας.....	9
2.1.2 Πλεονεκτήματα μιας Βίκι εγκυκλοπαίδειας.....	10
2.2 Γλώσσες Προγραμματισμού για την ανάπτυξη και υλοποίηση παρόμοιων εφαρμογών.....	11
2.2.1 Perl.....	11
2.2.2 Ruby.....	12
2.2.3 Python.....	13
2.3 Σύγκριση Γλωσσών Προγραμματισμού.....	13
2.3.1 Πώς Δουλεύει η PYTHON.....	15
2.4 Γλώσσα Προγραμματισμού της Εφαρμογής.....	15
2.4.1 Πλεονεκτήματα της γλώσσας προγραμματισμού PYTHON.....	16
2.5 Βασικές Εντολές Python.....	19
2.5.1 Η εντολή if.....	19
2.5.2 Η εντολή while.....	20
2.5.3 Η εντολή for.....	22
2.5.4 Η εντολή Break.....	22
2.5.5 Τοπικές μεταβλητές (Local variables)	23
2.5.6 Χρήση της εντολής global.....	24
2.5.7 Η εντολή return.....	25
2.6 Επικουρικά – Επιπλέον Εργαλεία	26
2.6.1 mySQL.....	26
2.6.2 Τεχνικές πληροφορίες και χαρακτηριστικά της MySQL	26
2.6.3 Τεχνικές πληροφορίες και χαρακτηριστικά της MySQL.....	27
2.6.4 Η MySQLdb.....	29
2.6.5 Το Λειτουργικό Σύστημα CentOS.....	30
2.6.6 Ο Apache server	30
2.7 Χρήση της Python από οργανισμούς – εταιρίες – εφαρμογές.....	31
3. Απαιτήσεις Συστήματος - Ανάλυση.....	33
3.1 Απαιτήσεις λειτουργίας.....	33
3.2 Επιπρόσθετες απαιτήσεις της λειτουργίας.....	34
3.3 Συνεργασία διακομιστή χρήστη	34
3.4 Εισαγωγή στο σύστημα μέσω προσωπικών πληροφοριών.....	34
3.5 Διαχωρισμός των προνομίων των εισαχθέντων χρηστών	34
3.6 Επιπρόσθετες απαιτήσεις της λειτουργίας (δυνατότητες διαχειριστή).....	35
4. Υλοποίηση του Συστήματος.....	36
4.1 Ανάλυση των λειτουργιών και βασικός κώδικας.....	36
4.1.1 Λειτουργίες Εφαρμογής	38

4.1.2 Λειτουργία Admin	39
4.1.3 Λειτουργία Χρήστη (user)	47
5. Δοκιμή	49

Πίνακας Εικόνων

Εικ. 2.1 Το σήμα του Apache HTTP Server	12
Εικ 2.2 Μερίδιο χρήσης διακομιστών HTTP πρωτοκόλλου	12
Εικ. 2.3 MySql	27
Εικ. 2.4 MySql Screenshot	29
Εικ 2.5 CentOS	30
Εικ 2.6 Το σήμα του Apache HTTP Server	30
Εικ 2.7 Μερίδιο χρήσης διακομιστών HTTP πρωτοκόλλου	31
Εικ. 5.1 URL	49
Εικ. 5.2 Αρχική Σελίδα	50
Εικ. 5.3 Κουμπί Εισόδου	50
Εικ 5.3 Login Διαδικασία	51
Εικ 5.4 Πίνακας Ελέγχου Admin	51
Εικ 5.5 Προσθήκη Κατηγοριών 1	52
Εικ 5.6 Προσθήκη Κατηγοριών 2	52
Εικ 5.7 Προσθήκη – Επεξεργασία Κειμένων	53

Κεφάλαιο 1

1. Εισαγωγή

Η αναγκαιότητα χρήσης του Internet στις μέρες μας είναι προφανής, ακόμα και για ανθρώπους που δεν έχουν άμεση ή έμμεση σχέση με το αντικείμενο των ηλεκτρονικών υπολογιστών. «Ο πλούτος της παρεχόμενης πληροφορίας, η αμεσότητα ενημέρωσης, οι προσφερόμενες υπηρεσίες και τα εργαλεία που συγκροτούν την έννοια που αποκαλείται συνολικά διαδίκτυο, προκαλεί δέος. Πολλοί μάλιστα μιλούν για "τεχνολογικό αποκλεισμό" σε περίπτωση που κάποιος δεν προσπαθήσει να προσαρμοστεί και να ενσωματώσει τη νέα αυτή πραγματικότητα στο τρόπο ζωής και εργασίας του. Το Internet έχει αρχίσει να ενηλικιώνεται και πλέον πρέπει να βελτιωθεί με τέτοιο τρόπο, ώστε να προάγει μια ευρύτερη ουμανιστική αντίληψη της χρήσης του υπολογιστή, με κέντρο τον άνθρωπο και μέσω του υπολογιστή, το οποίο έχει σαν πάγιο στόχο να εξυπηρετήσει αρτιότερα, ταχύτερα, πιο ευχάριστα και πιο πλήρως τις ανάγκες του.» [1]

Με την εισαγωγή του υπερκειμένου το 1990, το Internet απέκτησε το σημαντικότερο «συστατικό» του: Το World Wide Web (WWW). Με τη διάδοσή του, έδωσε νέα διάσταση στις διαδικασίες πρόσβασης πληροφοριών. Το πλήθος τους είναι πλέον τεράστιο και καλύπτουν κάθε θέμα. Πολλές πληροφορίες δεν είναι πλέον σε μορφή απλού κειμένου, αλλά περιέχουν ήχο, εικόνα, κίνηση και πολλά ακόμα χαρακτηριστικά στοιχεία των σύγχρονων τεχνολογιών. Μέσω του WWW, ένας άνθρωπος μπορεί να έχει πρόσβαση στις πληροφορίες που υπάρχουν σε οποιονδήποτε υπολογιστή του Διαδικτύου μέσω ενός φιλικού και αποδεκτού τρόπου, με αποτέλεσμα, η περιήγηση κάποιου στο Internet να γίνει εύκολη και ταυτόχρονα ευχάριστη.

Η χρήση της Java οδήγησε στην αύξηση των εφαρμογών διαδικτύου. Αυτό είχε ως αποτέλεσμα την ανάπτυξη πολλών εργαλείων για την ευκολότερη δημιουργία διαδικτυακών εφαρμογών, όπως, η JavaScript, η VBScript, η PHP, η C#, η ASP, καθώς και πολλές ακόμη, πιο πρόσφατες όπως η Ruby κι η Python. Σημαντικό ρόλο επίσης, στη συνεχή εξέλιξη κατέχουν και άλλες εφαρμογές, οι οποίες ωστόσο δεν

είναι σχεδιασμένες αποκλειστικά για το Internet. Τέτοιες είναι, εφαρμογές βάσεων δεδομένων, όπως η Oracle, η MySQL και ο Microsoft SQL Server (και όχι μόνο).

Όλα τα παραπάνω εργαλεία μαζί, οδηγούν στη έξαρση που υπάρχει στη σημερινή εποχή σε διαφορετικούς και σημαντικούς τομείς. Έτσι τα τελευταία χρόνια έχει παρουσιαστεί σημαντική πρόοδος σε τομείς όπως είναι το ηλεκτρονικό εμπόριο (e-commerce), τα ηλεκτρονικά βιβλία – εγκυκλοπαίδειες, η τηλεϊατρική, η τηλεεκπαίδευση (e-learning), η τηλεδιάσκεψη, η δημοσίευση αγγελιών μέσω του Internet, καθώς και σε πολλούς ακόμη.

Οι εφαρμογές που παρουσιάστηκαν παραπάνω δημιουργήθηκαν από τον άνθρωπο για τον άνθρωπο. Η αναφορά που προηγήθηκε σε κάποιες από αυτές, οδηγεί στο συμπέρασμα, πως το Internet σήμερα δεν είναι μόνο εργαλείο ψυχαγωγίας ή επικοινωνίας, αλλά ταυτόχρονα ένα εργαλείο που εξυπηρετεί τον άνθρωπο σε διάφορους σημαντικούς τομείς της ζωής του όπως η ενημέρωση, η εκπαίδευση και η διευκόλυνσή του σε πολλούς τομείς της ζωής του.

Ο βασικός σκοπός της πτυχιακής εργασίας είναι να δημιουργηθεί μία εφαρμογή η οποία θα έχει την δυνατότητα συνεργασίας μεταξύ διακομιστή και χρήστη (server model). Αυτό σημαίνει ότι η εν λόγω εφαρμογή θα είναι λειτουργική διαδικτυακά. Εφαρμογές που λειτουργούν διαδικτυακά πρέπει να έχουν τη δυνατότητα να δέχονται έναν μεγάλο αριθμό επισκεπτών. Η εφαρμογή αυτή θα πρέπει να δίνει την δυνατότητα στους χρήστες να διαχωρίζονται σε απλούς επισκέπτες, ενεργούς χρήστες ή απόλυτους διαχειριστές μέσω προσωπικών κωδικών για ασφαλή εισαγωγή στο σύστημα.

1.1 Αντικειμενικός σκοπός

Για την υλοποίηση του διαδικτυακού αυτού συστήματος, τα στάδια τα οποία θα πρέπει να επιτευχθούν ώστε να επιτευχθεί ο στόχος είναι τα εξής :

- Επιλογή της γλώσσας προγραμματισμού που θα χρησιμοποιηθεί για την δημιουργία του διαδικτυακού τόπου
- Καταγραφή και ανάλυση των απαιτήσεων για μια πιο σταθερή και ολοκληρωμένη εφαρμογή.
- Σχεδίαση και οπτική μορφοποίηση της εφαρμογής

- Υλοποίηση της εφαρμογής με βάση τα παραπάνω
- Δοκιμή και έλεγχος των λειτουργιών για την υλοποίηση αυτού.

1.2 Αναφορά των κεφαλαίων

Η παρούσα αναφορά αρχικά αναλύει τις τεχνολογίες που «εμπλέκονται» για την δημιουργία ενός τέτοιου συστήματος, παρουσιάζει τις προδιαγραφές που πρέπει να έχει μια τέτοια εφαρμογή και αναλύει τους προαναφερθέν αντικειμενικούς σκοπούς έναν προς έναν.

Γίνεται μια λεπτομερής έρευνα παρόμοιων εφαρμογών και εργαλείων που χρειάζεται για την υλοποίηση αυτών (Κεφάλαιο 2). Στο τρίτο κεφάλαιο θα καταγραφούν οι απαιτήσεις του συστήματος, όπου βάση αυτών θα γίνει η ανάλυση της εφαρμογής. Το σημαντικότερο κεφάλαιο είναι αυτό της υλοποίησης, όπου γίνεται αναφορά στα στάδια υλοποίησης της εφαρμογής (Κεφάλαιο 4), ενώ στο τελευταίο κεφάλαιο (Κεφάλαιο 5), παρουσιάζονται δοκιμές που έχουν γίνει για την πιστοποίηση της ομαλής λειτουργίας του συστήματος

Κεφάλαιο 2

2. Έρευνα

Η ανάλυση παρόμοιων εφαρμογών είναι απαραίτητη στην αρχή κάθε έργου για την κατανόηση των αντικειμενικών σκοπών. Βοηθάει στην κατανόηση των βαθύτερων προβλημάτων της υλοποίησης του έργου και αποφυγής των εμποδίων που πιθανότατα να προκύψουν κατά την υλοποίηση.

Αυτό το κεφάλαιο αναφέρεται σε εφαρμογές που σαν βάση έχουν την υλοποίηση διαδικτυακού τόπου εκμάθησης με σύστημα διαχείρισης περιεχομένων.

2.1 Παρόμοια εφαρμογή (Wikipedia)

Η γνωστότερη ηλεκτρονική εγκυκλοπαίδεια στον κόσμο είναι η Wikipedia, που πρόσφατα μεταφράστηκε και κυκλοφόρησε και στα ελληνικά με τον τίτλο Βικιπαίδεια.

Η Βικιπαίδεια είναι μία εγκυκλοπαίδεια ελεύθερου περιεχομένου, η οποία γράφεται από χρήστες που συμμετέχουν εθελοντικά σε αυτήν την προσπάθεια. Ελεύθερο περιεχόμενο σημαίνει ότι επιτρέπεται η χρήση, η αντιγραφή και η τροποποίηση των κειμένων από οποιονδήποτε και για οποιονδήποτε σκοπό.

Αυτός ο ιστοχώρος λειτουργεί σαν ένα Βίκι, δηλαδή, μπορεί οποιοσδήποτε να επεξεργαστεί οποιοδήποτε άρθρο απλά πατώντας στο κουμπί "Επεξεργασία", σύνδεση που υπάρχει σε όλα τα άρθρα της Βικιπαίδεια εκτός από ένα μικρό αριθμό προστατευμένων σελίδων.

Η υλοποίηση του Διαδικτυακού τόπου εκμάθησης της γλώσσας Python προσπαθεί να προσεγγίσει αυτήν την Βίκι εφαρμογή, και να «αντιγράψει» όσο δυνατόν γίνεται αποτελεσματικότερα τις λειτουργίες αυτής. Το Βίκι απαρτίζεται από ένα πλήθος λειτουργιών και υπηρεσιών, μερικές από τις οποίες θα είναι κοινές και για τα τρία επίπεδα χρηστών και κάποιες θα αναφέρονται ξεχωριστά σε κάθε επίπεδο.

2.1 .1 Προδιαγραφές μιας ηλεκτρονικής εγκυκλοπαίδειας

Τα βασικότερα χαρακτηριστικά μιας τέτοια ηλεκτρονικής εγκυκλοπαίδειας για κάθε επίπεδο χρήστη είναι τα εξής:

- Διαχείριση αποκλειστικά μέσω διαδικτύου. Δεν χρειάζεται εγκατάσταση κάποιου επιπλέον προγράμματος στον υπολογιστή.
- Δυνατότητα προσθήκης, επεξεργασίας και διαγραφής των πληροφοριών σε άμεσο χρόνο από τους χρήστες της εφαρμογής.
- Δυνατότητα προσθήκης εικόνων για κάθε μία σελίδα με πληροφορίες.
- Δυνατότητα αναζήτησης ή παράθεση περιεχομένων των άρθρων για ευκολότερη εύρεση των πληροφοριών.
- Αναλυτική προβολή όλων των στοιχείων της αγγελίας.
- Προβολή στοιχείων λογαριασμού των μελών.
- Εύκολη διαχείριση μελών. Εμφάνιση πληροφοριών και στατιστικών στοιχείων των μελών.
- Δυνατότητα δημιουργίας αντιγράφων ασφαλείας και άμεση επαναφορά του σε ιδιαίτερη περίπτωση
- Εισαγωγή και αποσύνδεση από το σύστημα.

2.1.2 Πλεονεκτήματα μιας Βίκι εγκυκλοπαίδειας

Η λειτουργία του διαδικτύου ως εργαλείου εύρεσης πληροφοριών μάς δίνει μια μεγάλη ποικιλία νέων δυνατοτήτων στο χώρο της αναζήτησης αγγελιών και στον τρόπο λειτουργίας τέτοιων δικτυακών τόπων. Τα βασικά πλεονεκτήματα της χρήσης μιας ηλεκτρονικής εγκυκλοπαίδειας είναι :

1. Παγκόσμια πρόσβαση.
2. Προσφορά απεριόριστων πληροφοριών με ελάχιστο κόστος.
3. Δυνατότητα άμεσης ανταπόκρισης.
4. Χαμηλό κόστος παροχής πληροφορίας.
5. Χαμηλό κόστος ενημέρωσης.
6. Χαμηλό κόστος δημιουργίας ενός δικτυακού τόπου.

7. Χαμηλό κόστος για την παροχή αλληλεπίδρασης με τον πελάτη.

Όπως φαίνεται από τα παραπάνω, η ολοένα αυξανόμενη υποδομή στο χώρο του Διαδικτύου σε συνδυασμό με τη συνεχή δημιουργία νέων εύχρηστων λειτουργιών, κάνει τους ιστοχώρους ηλεκτρονικών εγκυκλοπαιδειών πιο προσιτούς στον κόσμο, με αποτέλεσμα όλο και περισσότεροι να είναι σε θέση να βρουν, να επεξεργαστούν ή και να προσθέσουν πληροφορίες σε μια ηλεκτρονική εγκυκλοπαίδεια που τους ενδιαφέρει.

Στο όχι και τόσο μακρινό μέλλον το διαδίκτυο αναμένεται να εισβάλλει ακόμη πιο δυναμικά στη ζωή των περισσότερων ανθρώπων, αναιρώντας τις αποστάσεις. Πλησιάζει μια νέα εποχή, με τους ηλεκτρονικούς υπολογιστές να αναλαμβάνουν εν μέρει, το ρόλο του καθηγητή ή του μαθητή, και τους χρήστες του Internet να αποκομίζουν καθημερινά πολλά και σημαντικά οφέλη από τη νέα πραγματικότητα.

2.2 Γλώσσες Προγραμματισμού για την ανάπτυξη και υλοποίηση παρόμοιων εφαρμογών

2.2.1 Perl

Ξεκινώντας με την Perl μπορούμε να πούμε ότι είναι εξαιρετικά ισχυρή, δημοφιλής, ανοικτού κώδικα γλώσσα προγραμματισμού. Τα προγράμματα γραμμένα σε Perl είναι πολύ εύκολα όταν είναι μικρά. Η γλώσσα αυτή είναι εξαιρετική σε μικρά, απλά, λειτουργικά προγράμματα. Ωστόσο, γρήγορα γίνεται βαρετή στα μεγαλύτερα προγράμματα.

Συγκρινόμενα με την Perl, τα προγράμματα σε Python είναι σίγουρα πιο απλά, πιο καθαρά, πιο εύκολα στο γράψιμο, άρα και πιο κατανοητά και εύκολα στη συντήρηση. Πρέπει να σημειώσουμε ότι η Perl έχει υποστεί τόσα πολλά hacks και αλλαγές, που μοιάζει σαν να είναι η ίδια ένα τεράστιο hack. Δυστυχώς, η επερχόμενη Perl (Perl 6) δε φαίνεται να προσφέρει καθόλου βελτιώσεις σ' αυτό τον τομέα.

Το μοναδικό και πολύ σημαντικό πλεονέκτημα της Perl, είναι η τεράστια βιβλιοθήκη [CPAN](#) (Comprehensive Perl Archive Network). Όπως αναφέρει και το όνομα, αυτή είναι μια μεγάλη συλλογή από συναρμολογούμενα κομμάτια της Perl και είναι πραγματικά δύσκολο να κατανοήσει κανείς το μέγεθος και το βάθος της. Παρόλα αυτά τα κομμάτια αυτά είναι πολύ ισχυρά και μπορούν να κάνουν το οτιδήποτε σε έναν υπολογιστή. Ένας

από τους λόγους για τους οποίους η Perl έχει περισσότερες βιβλιοθήκες από την Python είναι γιατί είναι παλαιότερη γλώσσα από ότι η Python. Παρόλα αυτά φαίνεται να αλλάζουν τα δεδομένα με το ολοένα με τον καιρό και η Python αποκτά όλο και περισσότερο έδαφος.

2.2.2 Ruby

Η Ruby είναι μια ακόμα δημοφιλής, ανοικτού κώδικα, γλώσσα προγραμματισμού. Η εκμάθηση της γλώσσας αυτής κρύβει πολλές δυσκολίες και παγίδες στη συγγραφή της. Για ανθρώπους που πρόκειται να χρησιμοποιήσουν ή Python ή Ruby, θα ήταν καλύτερο να χρησιμοποιήσουν την Python, καθαρά από την άποψη της ευκολίας εκμάθησης

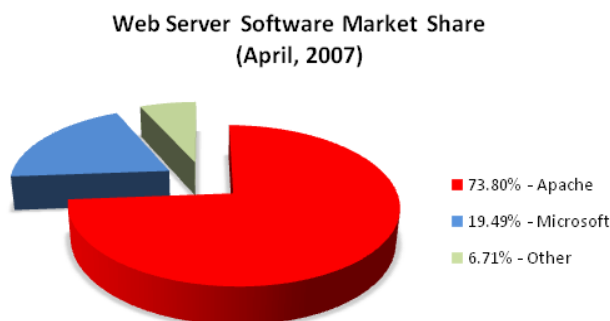
2.3 Ο Apache server

APACHE
HTTP SERVER



Εικ. 2.1 Το σήμα του Apache HTTP Server

Όποτε επισκέπτεστε έναν [ιστότοπο](#) ο πλοηγός σας (browser) επικοινωνεί με έναν διακομιστή [HTTP](#). Ο Apache είναι ένας από τους δημοφιλέστερους, εν μέρει γιατί λειτουργεί σε διάφορες πλατφόρμες σαν τα [Windows](#), το [Linux](#), το [Unix](#), και το [Mac OS X](#). Διατηρείται τώρα από μια κοινότητα ανοικτού κώδικα με επιτήρηση από το Ίδρυμα Λογισμικού Apache (Apache Software Foundation).



Εικ 2.2 Μερίδιο χρήσης διακομιστών HTTP πρωτοκόλλου

Ο **Apache HTTP Server**, στον οποίο συνήθως αναφερόμαστε ως Apache, είναι ένας διακομιστής που έχει ξεχωριστή σημασία στην πρόομη ανάπτυξη του Διαδικτύου (World Wide Web) και το 2009 έγινε ο πρώτος διακομιστής που ξεπέρασε τον αριθμό των 100 εκατομμυρίων ιστοσελίδων που τον χρησιμοποιούν. Από το 1996 είναι ο πλέον δημοφιλής διακομιστής του πρωτοκόλλου HTTP στο WWW. Η εκμάθηση της γλώσσας αυτής κρύβει πολλές δυσκολίες και παγίδες στη συγγραφή της. Για ανθρώπους που πρόκειται να χρησιμοποιήσουν ή Python ή Ruby, θα ήταν καλύτερο να χρησιμοποιήσουν την Python, καθαρά από την άποψη της ευκολίας εκμάθησης.

2.2.3 Python

Τέλος, η Python είναι προσανατολισμένη στην συγγραφή μικρότερων και συμπυκνωμένων εφαρμογών. Έτσι τα προγράμματα σε Python είναι αισθητά μικρότερα από τα αντίστοιχά τους σε γλώσσες όπως οι C/C++ ή Java. Αυτό αποτελεί ακόμη ένα πλεονέκτημα της Python σε σχέση με τις προαναφερθείσες γλώσσες. Επίσης η γλώσσα αυτή ανήκει στις very-high languages που τις δίνει μεγάλο πεδίο εφαρμογής σε σχέση με την απλή σύνταξη της. Γλώσσες προγραμματισμού όπως η Java και C/C++ είναι χαμηλότερου επιπέδου και πολλές φορές έχουν να κάνουν με το hardware.

2.3 Σύγκριση Γλωσσών Προγραμματισμού

Είναι παράδοση όποτε μαθαίνουμε μία καινούργια γλώσσα προγραμματισμού, το πρώτο πρόγραμμα που γράφουμε και τρέχουμε να είναι το πρόγραμμα «Hello, World!». Δηλαδή το μόνο που κάνει είναι να λέει «Hello World!» όταν το τρέξουμε.

Hello World στην C++

```
#include <iostream.h>
```

```
Void main()
```

```
{
```

```
    cout << "Hello World!" << endl;
```

```
    return 0;
```

```
}
```

Hello World στην C#

```
using System;

class HelloWorld

{   public static int Main(String[] args)

    {

        Console.WriteLine("Hello, World!");

        return 0;

    }

}
```

Hello World στην Java

```
class HelloWorld {

    static public void main( String args[] ) {

        System.out.println( "Hello World!" );

    }

}
```

Hello World στην Python

```
#!/usr/bin/python

#name: helloworld.py

print('Hello World!')
```

Αν τρέξουμε αυτό το πρόγραμμα ανοίγοντας ένα τερματικό στο Linux ή γραμμή εντολών στο DOS και εισάγοντας την εντολή `helloworld.py`. Σημείωση : πριν δώσουμε την εντολή για την εκτέλεση του προγράμματος `helloworld.py` θα πρέπει να έχετε μεταβεί στο φάκελο όπου είναι αποθηκευμένο το αρχείο `helloworld.py`.

2.3.1 Πώς Δουλεύει η PYTHON

Στις δύο πρώτες γραμμές του προγράμματος έχουμε τα σχόλια. Οτιδήποτε δεξιά του συμβόλου # είναι σχόλια. Τα σχόλια χρησιμοποιούνται κυρίως ως σημειώσεις για τον αναγνώστη του προγράμματος. Η Python δε λαμβάνει υπ' όψη της τα σχόλια, εκτός από την ειδική περίπτωση της πρώτης γραμμής όπως εδώ. Αυτή ονομάζεται γραμμή **shebang** επειδή πρώτοι δύο χαρακτήρες του αρχείου πηγαίου κώδικα είναι #! ακολουθούμενοι από την τοποθεσία ενός προγράμματος, αυτό πληροφορεί το Linux σύστημα ότι αυτό το πρόγραμμα θα πρέπει να εκτελεστεί με αυτό το διερμηνευτή όταν θα έρθει η ώρα της εκτέλεσής του. Σημειώστε ότι μπορείτε πάντα να τρέξετε το πρόγραμμα σε οποιαδήποτε πλατφόρμα καθορίζοντας το interpreter απ' ευθείας στη γραμμή εντολών όπως για παράδειγμα με την εντολή `helloworld.py`

2.4 Γλώσσα Προγραμματισμού της Εφαρμογής

Η Python είναι μια από τις σπάνιες γλώσσες που είναι απλές και ισχυρές. Στην ανάπτυξη ενός λογισμικού πρέπει να συγκεντρωθούμε περισσότερο στη λύση ενός προβλήματος παρά στο συντακτικό ή τη δομή της γλώσσας στην οποία προγραμματίζουμε.

Η Python είναι μια εύκολη στην εκμάθηση και ισχυρή γλώσσα προγραμματισμού. Έχει αποδοτικές δομές δεδομένων (data structures) υψηλού επιπέδου και μια απλή αλλά αποτελεσματική προσέγγιση στον αντικειμενοστρεφή προγραμματισμό (object oriented). Η απλή σύνταξη της Python και οι δυναμικοί τύποι της, μαζί με τη λειτουργία της ως διερμηνευόμενη (αντί μεταγλωττιζόμενης) γλώσσας, την καθιστούν την ιδανική γλώσσα για δημιουργία σεναρίων εντολών και για ταχεία ανάπτυξη εφαρμογών σε πολλούς τομείς και στις περισσότερες πλατφόρμες.

2.4.1 Πλεονεκτήματα της γλώσσας προγραμματισμού PYTHON

Η γλώσσα Python, όπως ειπώθηκε παραπάνω, είναι μια γλώσσα υψηλού επιπέδου, πολυχρηστική που έχει εφαρμογές σε πολλές διαφορετικές κατηγορίες προβλημάτων και απαιτήσεων.

Η γλώσσα περιέχει μια τεράστια βιβλιοθήκη που καλύπτει περιοχές επεξεργασίας κειμένου (συνηθισμένες εκφράσεις, Unicode, υπολογισμός διαφορών μεταξύ αρχείων), πρωτοκόλλων Διαδικτύου (HTTP, FTP, SMTP, XML-RPC, POP, IMAP, CGI programming), λογισμικού (unit testing, logging, profiling, parsing Python code), και πλατφόρμες λειτουργικών συστημάτων (κλήσεις συστήματος, συστημάτων αρχείων, TCP/IP sockets). Στο Παράρτημα μπορείτε να βρείτε τον κατάλογο με την βασική βιβλιοθήκη της Python, τους τύπους δεδομένων που υποστηρίζει, τα build-in modules (γραμμένα σε C) αλλά και κάποιες βασικές διεργασίες – λύσεις που είναι αναγκαίες σε κάθε προγραμματιστή. Μπορείτε να βρείτε επίσης και άλλα και έναν πίνακα περιεχομένων με τα τελευταία έτοιμα πακέτα που προσφέρονται στην Python. (συνολικά 6609, μέχρι τις 21/5/2009)

Φυσικά, δεν πρέπει να ξεχνάμε το βασικό πλεονέκτημα της Python σε σχέση με άλλες γλώσσες, την απλότητα δηλαδή και την ευκολία που παρουσιάζει στην δομή της και την ανάγνωση της. Πιο εμπειριστατωμένα η γλώσσα PYTHON είναι :

1. Απλή

Η Python είναι μια απλή και μινιμαλιστική γλώσσα. Το διάβασμα ενός καλού προγράμματος σε Python είναι σαν το διάβασμα των Αγγλικών, αλλά πολύ αυστηρών Αγγλικών! Αυτή η ομοιότητα της Python με ψευδο-κώδικα είναι ένα από τα πιο ισχυρά σημεία της. Σας επιτρέπει να συγκεντρώνεστε στη λύση του προβλήματος αντί στην ίδια τη γλώσσα.

2. Εύκολη στην εκμάθηση

Όπως θα δείτε, είναι εξαιρετικά απλό να ξεκινήσετε με την Python. Η Python έχει μια ασυνήθιστα απλή σύνταξη, όπως έχει ήδη αναφερθεί.

3. Ελεύθερη και Ανοικτού Κώδικα

Η Python είναι ένα παράδειγμα ΕΛΛΑΚ (Ελεύθερο Λογισμικό και Λογισμικό Ανοικτού Κώδικα). Με απλά λόγια, μπορείτε να διανείμετε αντίγραφα αυτού του λογισμικού, να διαβάσετε τον πηγαίο κώδικά του, να κάνετε αλλαγές σ' αυτό και να χρησιμοποιήσετε κομμάτια του σε νέα ελεύθερα προγράμματα. Το ΕΛΛΑΚ βασίζεται στην ιδέα μιας κοινότητας που μοιράζεται τη γνώση. Αυτός είναι ένας από τους λόγους για τους οποίους η Python είναι τόσο καλή -δημιουργήθηκε και βελτιώνεται συνεχώς από μια κοινότητα που το μόνο που θέλει είναι μια καλύτερη Python.

4. Γλώσσα υψηλού επιπέδου

Όταν γράφετε προγράμματα στην Python, δε χρειάζεται ποτέ να νοιάζεστε για τις χαμηλού επιπέδου λεπτομέρειες όπως η διαχείριση της μνήμης που χρησιμοποιείται από τα προγράμματά σας, κ.λπ.

5. Φορητή

Λόγω του ανοικτού της κώδικα, η Python έχει υλοποιηθεί (δηλαδή αλλάχθηκε για να λειτουργεί) σε πολλές πλατφόρμες. Όλα τα Python προγράμματά σας μπορούν να δουλέψουν σε οποιαδήποτε από αυτές τις πλατφόρμες χωρίς να χρειάζονται καθόλου αλλαγές αν είστε αρκετά προσεκτικοί ώστε να αποφύγετε να χρησιμοποιήσετε χαρακτηριστικά που εξαρτώνται από κάθε σύστημα.

Μπορείτε να χρησιμοποιήσετε την Python στο Linux, στα Windows, στο FreeBSD, σε Macintosh, στο Solaris, στο OS/2, στην Amiga, στο AROS, στο AS/400, στο BeOS, στο OS/390, στο z/OS, στο Palm OS, στο QNX, στο VMS, στο Psion, στο Acorn RISC OS, στο VxWorks, σε PlayStation, στο Sharp Zaurus, στα Windows CE ακόμα και σε PocketPC !

6. Διερμηνεύσιμη

Ένα πρόγραμμα που γράφεται σε μια μεταγλωττιζόμενη γλώσσα όπως η C ή η C++ μετατρέπεται από την πηγαία γλώσσα, για παράδειγμα τη C ή τη C++ σε μια γλώσσα που μιλάει ο υπολογιστής σας (δυναδικός κώδικας δηλαδή 0 και 1) χρησιμοποιώντας ένα μεταγλωττιστή με διάφορες σημαίες και επιλογές. Όταν τρέχετε το πρόγραμμα, ο συνδέτης αντιγράφει το πρόγραμμα στη μνήμη και αρχίζει να το τρέχει.

Η Python, από την άλλη, δε χρειάζεται μεταγλώττιση σε δυαδικό αρχείο. Απλά τρέχετε το πρόγραμμα απ' ευθείας από τον πηγαίο κώδικα. Εσωτερικά, η Python μετατρέπει τον

πηγαίο κώδικα σε μια ενδιάμεση μορφή που ονομάζεται bytecode και μετά το μεταφράζει στη γλώσσα του υπολογιστή και μετά το τρέχει. Όλο αυτό, στην πραγματικότητα κάνει τη χρήση της Python πολύ πιο εύκολη αφού δε χρειάζεται να ανησυχείτε για τη μεταγλώττιση του προγράμματος, τη σύνδεση με τις κατάλληλες βιβλιοθήκες, κ.λπ, κ.λπ. Αυτό επίσης κάνει τα προγράμματα της Python εξαιρετικά φορητά, αφού μπορείτε απλά να αντιγράψετε το πρόγραμμα Python που φτιάξατε σε έναν άλλο υπολογιστή και να δουλέψει έτσι απλά!

7. Αντικειμενοστραφής

Η Python υποστηρίζει τόσο το διαδικασιοστρεφή προγραμματισμό (procedure-oriented) όσο και τον αντικειμενοστραφή προγραμματισμό (object-oriented). Στο διαδικασιοστρεφή προγραμματισμό, το πρόγραμμα δομείται πάνω σε διαδικασίες ή συναρτήσεις οι οποίες δεν είναι τίποτε άλλο από επαναχρησιμοποιήσιμα κομμάτια από προγράμματα. Στις αντικειμενοστραφείς γλώσσες, τα προγράμματα δομούνται πάνω σε αντικείμενα τα οποία συνδυάζουν δεδομένα και λειτουργικότητα. Η Python έχει έναν πολύ ισχυρό αλλά πολύ απλό τρόπο για αντικειμενοστραφή προγραμματισμό, ειδικά όταν συγκρίνεται με μεγάλες γλώσσες όπως η C++ ή η Java.

8. Ελεγκτάσιμη

Αν χρειάζεστε ένα κρίσιμο κομμάτι κώδικα να τρέχει πολύ γρήγορα ή αν πρέπει να έχετε ένα κομμάτι ενός αλγόριθμου που να μην είναι ανοικτό, τότε μπορείτε να προγραμματίσετε εκείνο το κομμάτι σε C ή C++ και μετά να το χρησιμοποιείτε από το Python πρόγραμμά σας.

9. Ενσωματώσιμη

Μπορείτε να ενσωματώσετε την Python μέσα στα προγράμματα σε C/C++ για να τους δώσετε δυνατότητες 'scripting' για τους χρήστες σας.

10. Εκτεταμένες βιβλιοθήκες

Η Πρότυπη βιβλιοθήκη της Python είναι πραγματικά τεράστια. Μπορεί να σας βοηθήσει να κάνετε διάφορα πράγματα σχετικά με κανονικές εκφράσεις, δημιουργία τεκμηρίωσης,

δοκιμές μονάδων, νημάτωση, βάσεις δεδομένων, περιηγητές ιστού, CGI, FTP, email, XML, XML-RPC, HTML, αρχεία WAV, κρυπτογράφηση, γραφικές διεπαφές χρήστη (GUI -graphical user interfaces), Tk, και άλλα πράγματα που εξαρτώνται από το σύστημα. Θυμηθείτε ότι όλα αυτά είναι διαθέσιμα όποτε είναι εγκατεστημένη η Python. Αυτό ονομάζεται φιλοσοφία 'Batteries Included' της Python.

Επιπλέον από την πρότυπη βιβλιοθήκη, υπάρχουν διάφορες άλλες βιβλιοθήκες υψηλής ποιότητας όπως η [wxPython](#), η [Twisted](#), η [Python Imaging Library](#) και πολλές άλλες.

Η Python είναι πραγματικά μια συναρπαστική και ισχυρότατη γλώσσα, έχει το σωστό συνδυασμό απόδοσης και χαρακτηριστικών που κάνουν τη δημιουργία προγραμμάτων διασκέδαση.

2.5 Βασικές Εντολές Python

2.5.1 Η εντολή if

Η εντολή `if` χρησιμοποιείται για να ελέγχει μια συνθήκη και εάν ισχύει ή δεν ισχύει. Εάν ισχύει τότε εκτελείται ένα σύνολο εντολών το λεγόμενο `if-block`. Σε διαφορετική περίπτωση (`else`) γίνεται επεξεργασία ενός άλλου συνόλου εντολών το λεγόμενο `else-block`. Πρέπει να αναφέρουμε ότι η χρήση του όρου `else` είναι προαιρετική. Παρακάτω ακολουθεί παράδειγμα της εντολής `if` (`if.py`)

```
#!/usr/bin/python
```

```
# name: if.py
```

```
number = 28
```

```
guess = int(input('Input an integer number: '))
```

```
if guess == number:
```

```
    print('Congratulations, you guess right.') # το block ξεκινάει εδώ
```

```
    print('(But you didn't win anything!') # το block τελειώνει εδώ
```

```
elif guess < number:
    print('No, it is bigger.') # Neo block
else:
    print('No,it is smaller.')

print('End')
```

Η τελευταία αυτή εντολή εκτελείται πάντα μετά την εκτέλεση της εντολής if

Αυτό που χρησιμοποιήσαμε εδώ είναι ο όρος `elif` ο οποίος στην πραγματικότητα συνδυάζει δύο συσχετιζόμενες εντολές `if else-if else` σε μία εντολή `if-elif-else`. Αυτό κάνει το πρόγραμμα ευκολότερο και μειώνει τον αριθμό των εσοχών που απαιτούνται. Οι εντολές `elif` και `else` πρέπει επίσης να έχουν άνω και κάτω τελεία στο τέλος της λογικής γραμμής, ακολουθούμενες από τις αντίστοιχες πλήθος εντολών, φυσικά με κατάλληλες εσοχές. Επίσης μπορείτε να έχετε άλλη μία εντολή `if` εντός της του μπλόκ `if (if-block)` μιας εντολής `if`. Αυτό αποκαλείται φωλιασμένη `if`. Πρέπει να αναφέρουμε ότι τα τμήματα `elif` όπως και `else` είναι προαιρετικά. Η μικρότερη εντολή `if` που μπορεί να υπάρχει είναι:

```
if True:
    print('Yes, its true')
```

2.5.2 Η εντολή `while`

Η εντολή `while` επιτρέπει να εκτελείτε επανειλημμένα ένα σύνολο εντολών, όσο μια συνθήκη είναι σε ισχύ, δηλαδή παραμένει αληθής. Η εντολή `while` είναι ένα παράδειγμα αυτού που αποκαλείται εντολή βρόχου `loop statement`. Παρακάτω ακολουθεί παράδειγμα της εντολής `while`.

```
#!/usr/bin/python
# name: while.py

number = 28
run = True

while run:

    guess = int(input('Input integer number : '))

    if guess == number:

        print('Congratulations, you guess right.')
        run = False # αυτό κάνει τον βρόχο while να σταματήσει εδώ
    elif guess < number:

        print('No, it is bigger.')
    else:

        print('No, it is smaller.')

else:

    print('the loop statement while is ended.')

print('End')
```

Το `else` εκτελείται όταν η συνθήκη του βρόχου `while` αποκτά την τιμή `False`. Αυτό μπορεί να συμβεί ακόμα και κατά την πρώτη φορά που ελέγχεται η προϋπόθεση. Εάν υπάρχει ένας όρος `else` για ένα βρόχο `while`, εκτελείται πάντοτε, εκτός κι αν διακόψετε τον βρόχο με την εντολή `break`.

Οι τιμές `True` (αληθής) και `False` (ψευδής) ονομάζονται Μπουλείαν τύποι (`Boolean types`) και θεωρούμε ότι αντιστοιχούν στις τιμές 1 και 0 αντίστοιχα.

2.5.3 Η εντολή for

Η εντολή `for` είναι εντολή βρόχου, η οποία επαναλαμβάνεται σε μια ακολουθία αντικειμένων. Αυτό σημαίνει ότι εκτελείται κάθε αντικείμενο σε μια ακολουθία. Παρακάτω ακολουθεί παράδειγμα της εντολής `for`.

```
#!/usr/bin/python
#name: for.py
for i in range(1, 5):
    print(i)
else:
    print('The for loop is ended!')
```

Εδώ παρέχουμε δύο αριθμούς και η `range` επιστρέφει μια ακολουθία αριθμών, ξεκινώντας από τον πρώτο αριθμό έως τον αριθμό μετά το κόμμα. Για παράδειγμα, η `range(1, 5)` δίνει την ακολουθία `[1, 2, 3, 4]`. Εξ ορισμού, η `range` έχει ως αριθμό αύξησης το 1.

2.5.4 Η εντολή Break

Η εντολή `break` χρησιμοποιείται για τη διακοπή μιας εντολής βρόχου. Μπορούμε να χρησιμοποιήσουμε την εντολή αυτή και να διακόψουμε την εκτέλεση της εντολής βρόχου ακόμη κι αν η προϋπόθεση του βρόχου δεν έχει γίνει `False` ή δεν έχει επαναληφθεί-ολοκληρωθεί σ' όλη την ακολουθία των αντικειμένων.

```
#!/usr/bin/python
#name: break.py

while True:
    a = (input('Input something:'))
```

```
if a == 'quit':  
    break  
  
    print('The length of the string is', len(a))  
print('Done')
```

Σε αυτό το κομμάτι κώδικα παίρνουμε τα δεδομένα που εισάχθηκαν του χρήστη με την μέθοδο της επανάληψης και εκτυπώνουμε το μήκος κάθε εισαγωγής, δηλαδή το σύνολο χαρακτήρων. Έχουμε ορίσει έναν όρο για την διακοπή του προγράμματος, ελέγχοντας αν τα εισαχθέντα δεδομένα του χρήστη είναι 'quit'. Σταματάμε το πρόγραμμα διακόπτοντας τον βρόχο και τερματίζεται το πρόγραμμα.

2.5.5 Τοπικές μεταβλητές (Local variables)

Όταν δηλώνουμε μεταβλητές μέσα σε μία συνάρτηση, αυτές δεν έχουν καμία σχέση με άλλες μεταβλητές που έχουν την ίδια ονομασία και χρησιμοποιούνται έξω από αυτήν τη συνάρτηση, δηλαδή τα ονόματα των μεταβλητών χρησιμοποιούνται μόνο τοπικά στη συνάρτηση. Αυτό ονομάζεται εμβέλεια - scope των μεταβλητών. Όλες οι μεταβλητές έχουν την εμβέλεια του τμήματος όπου έχουν δηλωθεί, αρχίζοντας από το σημείο στο οποίο ορίζεται το όνομα. Παράδειγμα:

```
#!/usr/bin/python  
  
#name: local.py  
  
a = 7  
  
def func(a):  
    print('The char a is', a)  
    a = 5  
    print('Change the local char a to', a)
```

```
func(a)

print('The char a is still', a)
```

Αποτέλεσμα:

```
$ python local.py

The char a is 7

Change the local char a to 5

The char a is still 7
```

Στη συνάρτηση του προγράμματος ,την πρώτη φορά που χρησιμοποιούμε την τιμή με το όνομα `a`, η Python χρησιμοποιεί την τιμή της παραμέτρου που έχει δηλωθεί στη συνάρτηση. Έπειτα βάζουμε την τιμή 5 στο `a`. Η ονομασία `a` είναι μόνο τοπικά στη συνάρτησή. Έτσι όταν αλλάζουμε την τιμή του `a` στη συνάρτηση, το `a` που ορίστηκε στο κύριο τμήμα του προγράμματος δεν επηρεάζεται. Με το `print` τυπώνουμε και βλέπουμε ότι όντως δε επηρεάστηκε η μεταβλητή.

2.5.6 Χρήση της εντολής `global`

Εάν θέλουμε να τοποθετήσουμε μια τιμή σε ένα όνομα που ορίζεται στο κορυφαίο επίπεδο του προγράμματος, τότε πρέπει να δηλώσουμε στην Python ότι το όνομα δεν είναι τοπικό αλλά ολικό - `global`. Αυτό γίνεται με τη χρήση της εντολής `global`. Είναι αδύνατον να εκχωρήσουμε μια τιμή μέσα σε μια μεταβλητή που ορίζουμε εκτός μιας συνάρτησης χωρίς την εντολή `global`. Παράδειγμα:

```
#!/usr/bin/python

#name: global.py

x = 7

def func():

    global x

    print('The char x is', x)
```



```
x = 5

print('Change the global char x to', x)

func()

print('The value x is', x)
```

Αποτέλεσμα:

```
$ python global.py

The char x is 7

Change the global char x to 5

The value x is 5
```

2.5.7 Η εντολή return

Η εντολή return χρησιμοποιείται για να έχουμε επιστροφή από μια συνάρτηση, δηλαδή να βγούμε από τη συνάρτηση. Παράδειγμα:

```
#!/usr/bin/python

#name: return.py

def maximum(a, b):

    if a > b:

        return a

    elif a == b:

        return 'The numbers are equal'

    else:

        return b
```

```
print(maximum(1, 5))
```

Αποτέλεσμα:

```
$ python return.py
```

```
5
```

Η συνάρτηση `maximum` επιστρέφει το μέγιστο των παραμέτρων. Χρησιμοποιεί μια απλή εντολή `if else` για να βρει τη μεγαλύτερη τιμή και μετά επιστρέφει την τιμή αυτή.

2.6 Επικουρικά – Επιπλέον Εργαλεία

Παραπάνω είδαμε και αναλύθηκε η δυναμική που έχει η γλώσσα προγραμματισμού PYTHON. Για την δημιουργία όμως μιας τέτοιας εφαρμογής απαιτείται συνδυασμός εργαλείων και τεχνολογιών που αναλύονται στις παρακάτω ενότητες.

2.6.1 mySQL

Η MySQL είναι μια σχεσιακή βάση διαχείρισης δεδομένων. Τρέχει ως διακομιστής παρέχοντας πρόσβαση σε πολλαπλούς χρήστες σε έναν αριθμό βάσεων δεδομένων. Ο κώδικας της εφαρμογής αυτής είναι διαθέσιμος υπό τις προϋποθέσεις της GNU General Public License, όπως επίσης και σε μια ποικιλία ιδιοκτησιακών συμφωνιών. Η MySQL ανήκει και χρηματοδοτείται από την Σουηδική εταιρία MySQL A, η οποία υπάγεται στην Sun Microsystems.

2.6.2 Τεχνικές πληροφορίες και χαρακτηριστικά της MySQL

Η MySQL είναι γραμμένη στις γλώσσες C και C++. Μπορεί να συνεργαστεί μαζί με πολλές διαφορετικές πλατφόρμες συστημάτων όπως AIX, BSDi, FreeBSD, HP-UX, i5/OS, Linux, Mac OS, Microsoft Windows και γενικά σε όλα τα λειτουργικά. Οι βιβλιοθήκες που χρειάζονται για πρόσβαση στις βάσεις δεδομένων της MySQL

είναι διαθέσιμες σε όλες τις γνωστές γλώσσες προγραμματισμού με χαρακτηριστικό APIs. Επιπροσθέτως, η πλατφόρμα ODBC που αποκαλείται ως MyODBC, προσθέτει ακόμα περισσότερες γλώσσες προγραμματισμού σε αυτές που υποστηρίζουν μια βάση δεδομένων MySQL, όπως την ASP ή την ColdFusion. Ο MySQL server και άλλες βιβλιοθήκες έχουν υλοποιηθεί κυρίως με ANSI C ή ANSI C++.



Εικ. 2.3 MySql

Η MySQL εγγυάται την ασφαλή διαχείριση των δεδομένων, μέσω ενός συστήματος απόδοσης προνομίων και κωδικών σε όλους τους χρήστες. Οι κωδικοί (passwords) παραμένουν ασφαλείς, λόγω της απόκρυψης που γίνεται κατά τη σύνδεση στο server, ενώ γίνεται και επιπλέον επαλήθευση των χρηστών από τον ίδιο τον εξυπηρέτη.

Η MySQL, αποτελεί το πιο διαδεδομένο σύστημα βάσης δεδομένων με ενσωματωμένη SQL, το οποίο κυκλοφορεί με τη μορφή ανοιχτού πηγαίου κώδικα (open source). Αυτό σημαίνει πως οποιοσδήποτε μπορεί να δει και να τροποποιήσει τον κώδικα του – DataBase – Connectivity) συνδέσεων. Επιπρόσθετα, παρέχει και τοπικό χαρακτήρα, έχοντας τη δυνατότητα εμφάνισης μηνυμάτων λαθών σε πολλές γλώσσες και τον καθορισμό του συνόλου χαρακτήρων (character set) που προτιμά ο χρήστης.

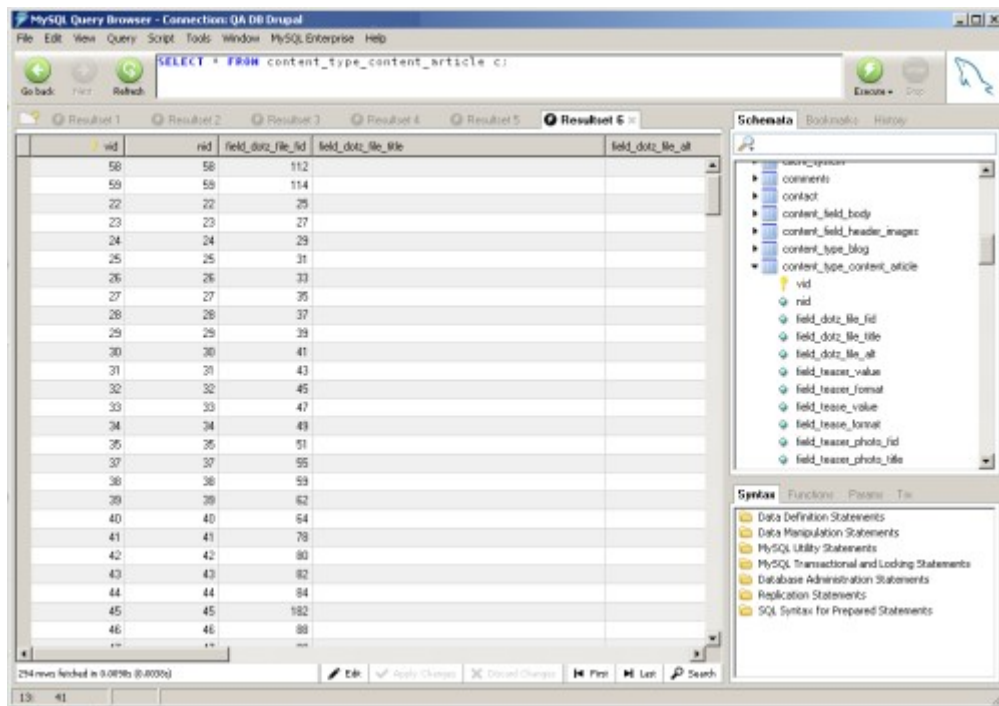
2.6.3 Τεχνικές πληροφορίες και χαρακτηριστικά της MySQL

Στην πραγματικότητα, είναι δύσκολο να πει κανείς για πιο λόγο προτιμάται το ένα ή το άλλο σύστημα βάσης δεδομένων. Παρόλ' αυτά, κάποιες μικρές λεπτομέρειες και διαφορές, καθορίζουν τελικά ποιο θα χρησιμοποιηθεί για την ανάπτυξη web εφαρμογών. Σε τελικό στάδιο, η απόφαση για το ποιο λογισμικό θα χρησιμοποιηθεί εξαρτάται από τις απαιτήσεις του έργου που πρόκειται να υλοποιηθεί.

Η απάντηση στο παραπάνω ερώτημα είναι αδύνατον να δοθεί μονολεκτικά ή μέσα σε μια φράση. Υπάρχουν πολλοί λόγοι που συγκλίνουν στην άποψη υπέρ της MySQL, μερικοί από τους οποίους είναι και οι εξής :

- ✓ **Λειτουργικό σύστημα** : Η MySQL λειτουργεί σε όλα τα γνωστά λειτουργικά συστήματα που κυκλοφορούν σήμερα (Windows, Linux, Unix, κ.λ.π.). Αντίθετα, άλλα συστήματα βάσεων δεδομένων όπως ο SQL Server, εφόσον είναι προϊόν της Microsoft, λειτουργεί μόνο σε περιβάλλον Windows, γεγονός που περιορίζει το πεδίο εφαρμογής του.
- ✓ **Απόδοση** : Η MySQL ηγείται σε αυτό τον τομέα, κυρίως λόγω της μορφής με την οποία αποθηκεύονται οι πίνακες. Αυτή η μορφή, με όνομα MyISAM, έχει πολύ μικρές απαιτήσεις σε επεξεργαστική ισχύ, μνήμη και χωρητικότητα.
- ✓ **Διάθεση πηγαίου κώδικα** : Όπως ήδη έχει αναφερθεί, ο πηγαίος κώδικας της MySQL είναι διαθέσιμος σε οποιονδήποτε θέλει να το μελετήσει ή να τον τροποποιήσει.
- ✓ **Κόστος άδειας χρήσης** : Είναι γεγονός πως η MySQL διανέμεται σε δυο εκδόσεις, όπου η μια είναι χωρίς κανένα κόστος και παρέχεται στο Internet, ενώ η δεύτερη, αν και επί πληρωμή, δεν έχει υπερβολικό κόστος. Αντίθετα, το κόστος της άδειας χρήσης άλλων συστημάτων βάσεων δεδομένων, είναι ένα σημαντικό στοιχείο που πρέπει να ληφθεί υπόψη.
- ✓ **Πρόσβαση από πολλούς χρήστες** : Η MySQL, με ευκολία μπορεί να διαχειρίζεται ταυτόχρονα πολλούς χρήστες, αφού από την αρχή είχε σχεδιαστεί να είναι ένα πολυχρηστικό (multi – user) σύστημα.
- ✓ **Διαχείριση μεγάλων βάσεων δεδομένων** : Όπως έχει ήδη αναφερθεί, η MySQL μπορεί να διαχειριστεί τεράστιες βάσεις δεδομένων, με εκατομμύρια εγγραφές και χιλιάδες πίνακες. Στην Access, αυτό δεν είναι δυνατόν, δηλαδή δεν μπορεί να γίνει διαχείριση βάσεων δεδομένων μεγάλου μεγέθους.
- ✓ **Ασφάλεια** : Η ασφάλεια σε άλλα συστήματα δεν αποτελεί δυνατό σημείο και αυτό οφείλεται στο ότι οι πίνακες, εφόσον αποθηκεύονται σε τοπικούς δίσκους, μπορούν να ανοιχτούν και να διαχειριστούν με απευθείας πρόσβαση στον υπολογιστή που βρίσκονται. Επιπλέον τη διαχείριση της ασφάλειας σε MySQL, την αναλαμβάνει ο ίδιος ο MySQL Server. Έτσι, οποιοσδήποτε

προσπαθήσει να έχει πρόσβαση στα δεδομένα χωρίς τα σωστά στοιχεία (όνομα χρήστη και κωδικό πρόσβασης), θα αποτυγχάνει διαρκώς.



Εικ. 2.4 MySql Screenshot

2.6.4 Η MySQLdb

Η MySQLdb είναι ένας ελαφρύς κατασκευαστής που χρησιμοποιεί η Python, δημιουργημένος με βάση την MySQL. Η λειτουργία του έχει σκοπό την συμβατότητα της Python DB API με την MySQL. Στην πραγματικότητα, μια μεγάλη ποσότητα κώδικα που υλοποιεί την API (Application Programming Interface – Διεπαφή Προγραμματισμού Εφαρμογών, η επαφή δηλαδή που γίνεται μεταξύ ενός Λειτουργικού Συστήματος, μιας βιβλιοθήκης ή μιας εφαρμογής με ένα άλλο πρόγραμμα για ανταλλαγή δεδομένων) είναι γραμμένος σε mysql για καλύτερη απόδοση.

2.6.5 Το Λειτουργικό Σύστημα CentOS



Εικ 2.5 CentOS

Το CentOS (Community enterprise OS) είναι μία διανομή του λειτουργικού συστήματος Linux που στηρίζεται σε μια άλλη διανομή, το Red Hat. Αναπτύχθηκε και ανήκει στον Lance Davis, διατίθεται δωρεάν, υποστηρίζεται από την κοινότητα ανοικτού λογισμικού και ειδικεύεται στο να παρέχει μια ελεύθερη πλατφόρμα διαχείρισης δικτύου σε επιχειρήσεις, προσπαθώντας την ίδια στιγμή να παραμείνει 100% συμβατή με την διανομή Red Hat.

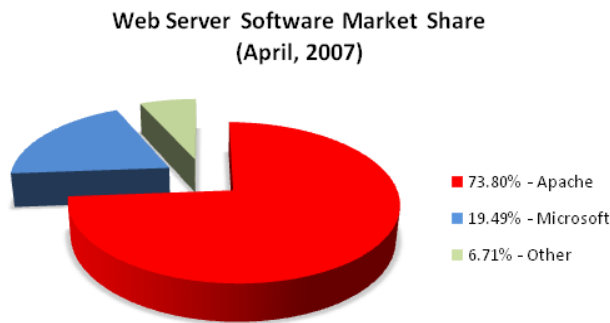
2.6.6 Ο Apache server

APACHE
HTTP SERVER



Εικ 2.6 Το σήμα του Apache HTTP Server

Όποτε επισκέπτεστε έναν ιστότοπο ο πλοηγός σας (browser) επικοινωνεί με έναν διακομιστή HTTP. Ο Apache είναι ένας από τους δημοφιλέστερους, εν μέρει γιατί λειτουργεί σε διάφορες πλατφόρμες σαν τα Windows, το Linux, το Unix, και το Mac OS X. Διατηρείται τώρα από μια κοινότητα ανοικτού κώδικα με επιτήρηση από το Ίδρυμα Λογισμικού Apache (Apache Software Foundation).



Εικ 2.7 Μερίδιο χρήσης διακομιστών HTTP πρωτοκόλλου

Ο **Apache HTTP Server**, στον οποίο συνήθως αναφερόμαστε ως Apache, είναι ένας διακομιστής που έχει ξεχωριστή σημασία στην πρόοδη ανάπτυξη του Διαδικτύου (World Wide Web) και το 2009 έγινε ο πρώτος διακομιστής που ξεπέρασε τον αριθμό των 100 εκατομμυρίων ιστοσελίδων που τον χρησιμοποιούν. Από το 1996 είναι ο πλέον δημοφιλής διακομιστής του πρωτοκόλλου HTTP στο WWW.

2.7 Χρήση της Python από οργανισμούς – εταιρίες – εφαρμογές

Η Python χρησιμοποιείται ευρέως από οργανισμούς-εταιρίες-εφαρμογές όπως:

- την πασίγνωστη μηχανή αναζήτησης Google που τον τελευταίο καιρό επεκτείνεται ταχύτατα και στις εφαρμογές λογισμικού
- το Yahoo για την υπηρεσία χαρτών που προσφέρει
- την NASA και πολλές άλλες εταιρίες με ειδικευση στις φυσικές επιστήμες και την πρόβλεψη του καιρού
- το YouTube – ιστοσελίδα φιλοξενίας μικρών βίντεο
- το BitTorrent client
- σε πάρα πολλά λογισμικά ασφαλείας και διαχείρισης οικονομικής ειδικευσης όπως η ABN - Amro Bank
- την Maya και την Blender – εφαρμογές ανοικτού λογισμικού με ειδικευση στον σχεδιασμό 3D γραφικών ή το Paint Shop Pro – λογισμικό επεξεργασίας γραφικών για Windows
- το ArcGIS – λογισμικό που αποτελεί ένα σύστημα κάθε είδους γεωγραφικών πληροφοριών

- πολλές πλατφόρμες παιχνιδιών όπως το Civilization IV και το Battlefield 2
- πολλά πανεπιστημια ή βιβλιοθήκες σε λογισμικό που βοηθάει στην οργάνωσή τους, πχ. το University of California ή η ψηφιακή βιβλιοθήκη της Νέας Ζηλανδίας
- η CIA για την κατασκευή της ιστοσελίδας της
- το Anaconda – λογισμικό εγκατάστασης των εκδόσεων του Linux Red Hat και Fedora
- το Portage – λογισμικό διαχείρισης στην έκδοση Gentoo του Linux, με χρήσεις και σε FreeBSD, Mac OS X, Solaris, ενώ γενικά η χρήση του πλέον είναι επιβεβλημένη σε κάθε έκδοση του Linux ή του Mac OS X.

Κεφάλαιο 3

3. Απαιτήσεις Συστήματος - Ανάλυση

Προδιαγράφοντας τις απαιτήσεις του συστήματος είναι μία πολύ βασική εργασία η οποία έπεται της έρευνας και προηγείται της υλοποίησης. Αν οι απαιτήσεις του συστήματος είναι ολοκληρωμένες και σωστές τότε η ανάπτυξη αυτού είναι ευκολότερη και γρηγορότερη.

Οι απαιτήσεις του συστήματος μπορούν να εντοπιστούν εάν σαν στόχο έχουμε να προκαταλάβουμε τη λειτουργία του συστήματος. Δια μέσου αυτής της ενέργειας και με τη βοήθεια της έρευνας (Κεφάλαιο 2) πολλά προβλήματα και εμπόδια μπορούν να αποφευχθούν.

Όπως προαναφέραμε έλλειψη της κατανόησης των απαιτήσεων του συστήματος μπορούν να οδηγήσουν σε αποτυχία του έργου και σε χάσιμο αναπτυξιακού χρόνου.

Αυτό το κεφάλαιο εντοπίζει και αναλύει τις απαιτήσεις του παρόντος έργου.

3.1 Απαιτήσεις Λειτουργίας

Οι απαιτήσεις του συστήματος που αναφέρονται επιγραμματικά παρακάτω θα αναλυθούν σε βάθος μέσω αυτού του κεφαλαίου, οι οποίες είναι:

- Χρήση της εφαρμογής μέσω του διαδικτύου με την συνεργασία διακομιστή – χρήστη.
- Εισαγωγή στο σύστημα μέσω προσωπικών πληροφοριών (κωδικούς εισαγωγής).
- Διαχωρισμός των προνομίων των εισαχθέντων χρηστών στο σύστημα.
- Προσθαφαίρεση χρηστών
- Δυνατότητα επεξεργασίας της εφαρμογής από τους χρήστες, όπως εισαγωγή και μετατροπή και διαγραφή κειμένων

3.2 Επιπρόσθετες απαιτήσεις της λειτουργίας

- Δυνατότητα αλλαγής της θεματολογίας
- Δυνατότητα επιπρόσθετων λειτουργιών (Plug-ins)
- Δυνατότητα δημιουργίας αντιγράφων ασφαλείας (Backup) και επαναφοράς αυτού του αντιγράφου (Restore).

3.3 Συνεργασία διακομιστή χρήση

Η εφαρμογή πρέπει να έχει την δυνατότητα να είναι προσβάσιμη μέσω διαδικτύου. Αυτό επιτυγχάνεται με το να αποθηκευτεί η εφαρμογή σε έναν διακομιστή - Server. Αυτός ο διακομιστής θα έχει εγκατεστημένο το απαραίτητο λειτουργικό το οποίο θα στηρίζει την γλώσσα προγραμματισμού σε συνεργασία με την βάση δεδομένων που θα χρησιμοποιηθούν για την υλοποίηση της εφαρμογής. Όλα τα παραπάνω έχουν την προϋπόθεση ότι ο χρήστης που θα συνδέεται με την εφαρμογή δεν θα χρειάζεται κάποιο ειδικό προ-εγκατεστημένο πρόγραμμα ή λειτουργικό για να έχει πρόσβαση σε αυτήν.

3.4 Εισαγωγή στο σύστημα μέσω προσωπικών πληροφοριών

Όταν ένας χρήστης συνδέεται στην εφαρμογή, θα έχει τη δυνατότητα περιήγησης και ανάγνωσης των πληροφοριών – άρθρων της εφαρμογής. Επίσης μέσω προσωπικών κωδικών για κάθε χρήστη θα του δίνεται η δυνατότητα να εισέρχεται στο σύστημα και να διαφοροποιεί την εφαρμογή αυτή.

3.5 Διαχωρισμός των προνομίων των εισαχθέντων χρηστών

Ο διαχωρισμός προνομίων των κάθε χρηστών είναι βασική προϋπόθεση για την ομαλή λειτουργία της εφαρμογής. Οι χρήστες θα χωρίζονται σε τρεις κατηγορίες ανάλογα με τα προνόμια τους. Η πρώτη ομάδα είναι η ομάδα απλού χρήστη – αναγνώστη τα προνόμια του οποίου περιορίζονται μόνο στην ανάγνωση των κειμένων – άρθρων.

Η δεύτερη ομάδα η οποία μπορεί να ονομαστεί και ως ενεργό μέλος δίνει τη δυνατότητα στο μέλος αυτό να ανήκει σε μία ονομαστική ομάδα η οποία θα έχει ως προνόμιο, πέρα της ανάγνωσης των άρθρων – κειμένων, τη δημιουργία υποκειμένων και την τροποποίηση αυτών. Επίσης θα μπορεί να τροποποιήσει άρθρα που έχουν ήδη δημιουργηθεί από άλλους χρήστες ή και διαχειριστές.

Οι διαχειριστές θα είναι η τρίτη ομάδα χρηστών που πέρα των προνομίων που θα έχουν οι δύο παραπάνω ομάδες, θα έχουν επίσης μια επιπλέον πληθώρα δυνατοτήτων.

3.6 Επιπρόσθετες απαιτήσεις της λειτουργίας (δυνατότητες διαχειριστή)

Όπως προαναφέραμε ο διαχειριστής ο οποίος θα έχει τη δυνατότητα να δημιουργεί άλλους χρήστες ή διαχειριστές, θα μπορεί επίσης να δημιουργεί και ομάδες χρηστών και να ορίζει τα δικαιώματα αυτών.

Οι επιπρόσθετες λειτουργίες του διαχειριστή θα πρέπει να του δίνουν τη δυνατότητα να διαχειρίζεται, πέρα από τους χρήστες, την καθαυτή εφαρμογή. Ποιο συγκεκριμένα αλλαγές στον τρόπο εμφάνισης της εφαρμογής (template), διαχείριση της συνδεσιμότητας με τον διακομιστή, όπως και τη δυνατότητα δημιουργίας αντιγράφου(Backup) και επαναφοράς του αντιγράφου αυτού (restore), είναι κάποιες επιπρόσθετες απαιτήσεις οι οποίες είναι απαραίτητες για την καλύτερη και πιο ολοκληρωμένη λειτουργία της εφαρμογής αυτής.

Κεφάλαιο 4

4. Υλοποίηση του Συστήματος

Μετά από μια ενδελεχή έρευνα που καταγράφηκε παραπάνω, τα βασικά προγράμματα που θα χρησιμοποιηθούν ώστε να υλοποιηθεί ο διαδικτυακός αυτός τόπος θα είναι η Python 2.5 σε συνεργασία με την MySQLdb της Python. Το λειτουργικό σύστημα που θα χρησιμοποιηθεί θα είναι το CentOS 5.0, όπου πάνω του θα είναι στημένος ο Apache server. Οι λόγοι που επιλέξαμε τα εργαλεία αυτά καθώς και αυτό το λειτουργικό σύστημα ήταν κυρίως ότι:

- ✓ Όλα τα παραπάνω είναι ανοιχτού κώδικα.
- ✓ Συνεργάζονται μεταξύ τους με απόλυτη επιτυχία.
- ✓ Η συνδυασμένη τους χρήση, επιτρέπει τη δημιουργία δυναμικών διαδικτυακών εφαρμογών με υψηλά κριτήρια αξιοπιστίας, απόδοσης, ταχύτητας και ασφάλειας.

Παρακάτω θα αναλυθεί η λειτουργία και η υλοποίηση του συστήματος με κάποια βασικά σημεία του κώδικα

4.1 Ανάλυση των λειτουργιών και βασικός κώδικας

Έχοντας ξεκινήσει τον server και αφού έχουμε κάνει την ανάλογη συνδεσμολογία που περιγράψαμε παραπάνω, αρχίζει να εκτελείται η διαδικτυακή μας εφαρμογή.

Βασικό κομμάτι για το ξεκίνημα, είναι η σύνδεση μέσω συγκεκριμένης θύρας (PORT) της εφαρμογής με τον server, που επιτυγχάνεται με το αρχείο server.py μέσω της συνάρτησης `startServer()`

```

def startServer():

    wsgi_app = setupWSGIApp()

    #wsgi_app = validate.validator(wsgi_app)

    #logging.basicConfig(level=logging.DEBUG)

    amiweb.startServer(wsgi_app, general_config.PORT)

```

Έχοντας πλέον συνδεθεί με τον server, η σύνδεση της Βάσης Δεδομένων με την εφαρμογή είναι το επόμενο βήμα. Η συνάρτηση **configureDatabase()** εκτελείται, κάνοντας τους ανάλογους ελέγχους ταυτοποίησης (κωδικοί εισόδου)

```

db_pool = None

def configureDatabase():

    global db_pool

    pool = amidb.ConnectionPool

    if getattr(general_config, 'USE_POOL', None) == False:

        pool = amidb.FreshPool

        print "Database connection pooling is disabled."

    dbinfo = amidb.AmiDBInfo(general_config.DB_USER,

                             general_config.DB_PASSWORD,

                             general_config.DB_DATABASE,

                             general_config.DB_HOST,

                             general_config.TABLE_PREFIX)

    db_pool = pool(dbinfo)

```

Η παραπάνω συνάρτηση, συνεργάζεται άμεσα με την `general_config.py` το οποίο είναι υπεύθυνο για την παραμετροποίηση της σύνδεσης της εφαρμογής με τον server

```
PORT = 14002

USE_CACHING = False

BUILD_CACHE_ON_START = False

TITLE_PREFIX = "Python Pasas Nikos - "

BASE_URL = 'http://localhost:14002'

CURRENT_TEMPLATE.BASE_URL = BASE_URL

COOKIE_NAME = "Python"

CHECK_LOGIN = True

ADMIN_USERNAME = "admin"

ADMIN_PASSWORD = "admin"

START_PAGE = "Main"

TABLE_PREFIX = 'sk_'

DB_USER = ""

DB_PASSWORD = ""

DB_HOST = 'localhost'

DB_DATABASE = 'TestDB'
```

Έχοντας κάνει όλους τους απαραίτητους ελέγχους, η σελίδα είναι ορατή στον εκάστοτε χρήστη.

4.1.1 Λειτουργίες Εφαρμογής

Η εφαρμογή χωρίζεται σε τρεις λειτουργίες. Στην λειτουργία admin, όπου είναι ο διαχειριστής του συστήματος, στην λειτουργία user όπου έχει εν μέρει διαχείριση του συστήματος και λειτουργία guest όπου δεν έχει κανένα προνόμιο διαχείρισης – μόνο περιήγηση στις σελίδες της εφαρμογής.

4.1.2 Λειτουργία Admin

Η λειτουργία admin ξεκινάει με το αρχείο login.tpl το οποίο κάνει τους απαραίτητους ελέγχους κωδικών πρόσβασης μέσω της βάσης δεδομένων.

```
function doLogin() {  
    AJS.setHTML(AJS.\$('error'), "");  
  
    Indicator.append(AJS.\$('login_indicator'));  
    var item = {username: AJS.\$('username').value, 'password':  
AJS.\$('password').value};  
    var d = getRequest('users/checkLoginInfo');  
    d.addCallback(checkLoginCallback);  
    d.sendReq(item);  
    return false;  
}
```

Με την εισαγωγή των σωστών κωδικών ο διαχειριστής του συστήματος έχει πλέον την δυνατότητα, μέσω του συστήματος διαχείρισης δεδομένων (CMS) να :

- προσθέσει Χρήστες μέσω του αρχείου UserManager_index.tpl

Ο παρακάτω κώδικας είναι υπεύθυνος στο να εξάγει τα υπάρχον group της βάσης, να δημιουργεί και να τα εμφανίζει σ ένα drop down κατάλογο

Δημιουργεί την drop down λίστα επιλογής

```
function getGroupDropBox(user_id, current_user_group) {  
    var no_group_option = OPTION({'value': 'no_group'}, "No group");  
    if(current_user_group == 0){  
        no_group_option.selected = "selected";  
    }  
  
    var gb_select = SELECT({'style': 'margin-left: 10px;'}, no_group_option);  
    gb_select.onchange = function () {
```

```

        var group_id = gb_select.value;

        var d = getRequest('admin/GroupManager/changeUserGroup');

        d.sendReq({'user_id': user_id, 'group_id': group_id});

    };

    var g_opt = OPTION({'value': $group.id}, "$group.name");

    if(current_user_group == $group.id) {

        g_opt.selected = "selected";

    }

    gb_select.appendChild(g_opt);

#end for

return gb_select;

}

```

Δίνει δικαιώματα admin ή χρήστη στη νέα ομάδα

```

function getTypeDropBox(user_id, user_type) {

    var ut_select = SELECT({'style': 'margin-left: 10px;'});

    var user = OPTION({'value': 'user'}, "User");

    var admin = OPTION({'value': 'admin'}, "Admin");

    if(user_type == "admin")

        admin.selected = true;

    else

        user.selected = true;

    ut_select.appendChild(user);

    ut_select.appendChild(admin);

    ut_select.onchange = function () {

        var type = ut_select.value;

        var d = getRequest('admin/UserManager/changeUserType');
    }
}

```



```

        d.sendReq({'user_id': user_id, 'type': type});
    };

    return ut_select;
}

```

- προσθέσει Ομάδες μέσω του αρχείου GroupManager_index.tpl

Προσθέτει ένα καινούριο group

```

function onSubmit() {
    var name = AJS.\\$('f_name').value;
    if(name != "") {
        var d = loadJSONDoc('admin/GroupManager/add');
        var add_elm = function(item_json) {
            list_commander.add(item_json);
            list_commander.render();
        };
        d.addCallbacks(add_elm);
        d.sendReq({'name': name});
    }
    else {
        alert("Πρέπει να δώσετε κάποιο όνομα.");
    }
}

```

- διαχειριστεί όλα τα μενού και τα υπομενού της εν λόγω ιστοσελίδας μέσω του αρχείου SiteManager_index.tpl, UploadManager_index.tpl & menu_edit.tpl

Προσθέτει ένα καινούριο εσωτερικό ή εξωτερικό μενού

Εσωτερικό

```
insertInternalButtons: function(action_bar) {  
    action_bar.addButtons('page',  
this.lc.btnEdit('Edit item', 'menu_edit/update', 'static_core/images/edit.png', {'name':  
'Name'}),  
    this.btn_up,  
    this.btn_down,  
    //Delete  
    this.btn_delete  
);
```

Εξωτερικό

```
insertExternalButtons: function(action_bar) {  
    action_bar.addButtons('external',  
this.lc.btnEdit('Edit item', 'menu_edit/updateExternal',  
'static_core/images/edit.png', {'name': 'Name', 'url': 'URL'}),  
    this.btn_up,  
    this.btn_down,  
    this.btn_delete  
);
```

- διαχειριστεί βασικές λειτουργίες του διαχειριστή μέσω των SiteManager_index.tpl & UploadManager_index.tpl

```
function justDelete(elm, id) {  
    GB_P.refresh_content = true;  
    var d = getRequest('admin/SiteManager/deletePage');  
    d.sendReq({'id': id});  
}
```

```
function deleteVisually(elm, child_count) {  
  
    var conty = AJS.\$('conty');  
  
    var children = getElementsByTagAndClassName("tr", null,  
conty);  
  
    for(var i = 0; i < child_count; i++) {  
  
        removeElement(children[elm.rowIndex + 1 + i]);  
  
    }  
  
    removeElement(elm);  
  
}
```

```
function deletePage(elm, id, child_count) {  
  
    if(id == 1) {  
  
        alert("You can't delete the main page!");  
  
        return;  
  
    }  
  
    else {  
  
        if(confirm("Are you sure that you want to remove this page (ALL  
under pages will ALSO be deleted!)?")) {  
  
            justDelete(elm, id);  
  
            deleteVisually(elm, child_count);  
  
        }  
  
    }  
  
}
```

- διαχειρίζεται το προκαθορισμένο template της επιλογής του μέσω του αρχείου TemplateManager_index.tpl

```
function changeTemplate(elm) {
    var d = AJS.getRequest('admin/TemplateManager/changeTemplate');
    d.addCallback(function() { window.location.reload(); } );
    d.sendReq({'id': AJS.serializeJSON(elm.json.id)});
    GB_P.GB_CURRENT.reload_on_close = true;
}

action_bar.addButtons('default',
    new ActionButton({'text': 'Change to',
        'image': 'static_core/admin/images/restart.png',
        'on_click_fn': changeTemplate})
);
```

- να προσθέσει ανάλογο plugin μέσω του αρχείου PluginManager_index.tpl

Τα ήδη προεγκαταστημένα Plugin εμφανίζονται μέσω του παρακάτω κώδικα

```
var SyntaxShower = {
    switchState: function(elm) {
        var new_elm;
        if(elm.state == "on") {
            AJS.removeElement(elm.tr_elm);
            new_elm = SyntaxShower._getElm("off");
        }
        else {
            var td = AJS.TD({'colSpan': 2, 'class': 'syntax'});
```

```

AJS.ACN(td, SyntaxShower._formatSyntax(elm));

var new_tr = AJS.TR(td);

AJS.insertAfter(new_tr, elm.parentNode.parentNode);

new_elm = SyntaxShower._getElm("on");

new_elm.tr_elm = new_tr;

}

new_elm.syntax = elm.syntax;

AJS.swapDOM(elm, new_elm);

},

```

Αλλαγές στον κώδικα μπορούν να γίνουν μέσω αυτής της λειτουργίας

```

insert: function(syntax) {

    var state = "off";

    var elm = SyntaxShower._getElm(state);

    elm.syntax = syntax;

    AJS.DI(elm);

},

```

- να δημιουργήσει αντίγραφα ασφαλείας μέσω του αρχείου BackupManager_index.tpl

```

function onSubmit() {

    Indicator.append(AJS.\$('create_snapshot'));

    list_commander.action_bar.setDisable(true);

    var name = AJS.\$('f_snapshotname').value;

    if(name != "") {

        //Set order

```

```
var order = 1;

var l_elms = getElementsByTagAndClassName("tr", "lc_item");

if(l_elms.length > 0) {

    //The last items order

    var ord = l_elms[0].json.order;

    if(ord)

        order = parseInt(ord) + 1;

}

var d = loadJSONDoc('admin/BackupManager/createSnapshot');

var add_elm = function(item_json) {

    list_commander.action_bar.setDisable(false);

    Indicator.remove(AJS.\$('create_snapshot'));

    list_commander.add(item_json, true);

    list_commander.render();

};

d.addCallbacks(add_elm);

d.sendReq({'name': name, 'order': order});

}

else {

    alert("The name may not be empty");

}

}
```



```
<option value="group-//-$group"
selected="selected">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;$group</option>
#else
<option value="group-//-$group">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;$group</option>
#end if
#end for
#end if
```

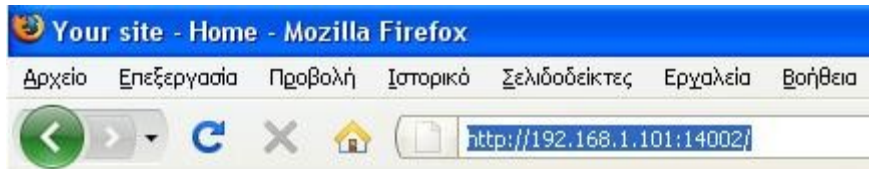

Κεφάλαιο 5

5.Δοκιμή

Η δοκιμή είναι το τελικό αλλά βασικό στάδιο της υλοποίησης ενός έργου που έχει σαν σκοπό την πιστοποίηση της ομαλής και ολοκληρωμένης λειτουργίας αυτού. Για το λόγο αυτό εφαρμόζουμε μια σειρά δοκιμών που θα πιστοποιήσουν την εγκυρότητα του συστήματος και θα αποκαλύψουν τυχόν λάθη και ατέλειες αυτού.

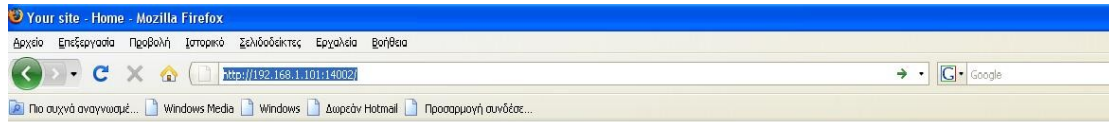
Παρακάτω γίνεται μία αναφορά στα στάδια που ακολουθεί ένας χρήστης ώστε να εισέλθει στο σύστημα έχοντας πάντα τα ανάλογα δικαιώματα, ενεργώντας συγκεκριμένα ως διαχειριστής, επεξεργάζεται το σύστημα και τέλος αποσυνδέεται.

Έχοντας ξεκινήσει τον διακομιστή – server και συνδέοντας το τερματικό μας με αυτόν, ανοίγουμε έναν φυλλομετρητή (browser) και πληκτρολογούμε τη διεύθυνση <http://192.168.1.101:14002/>



Εικ. 5.1 URL

Η σελίδα που εμφανίζεται είναι η αρχική της εφαρμογής και είναι ορατή από όλους τους χρήστες με προνόμια επισκέπτη.



Python CMS



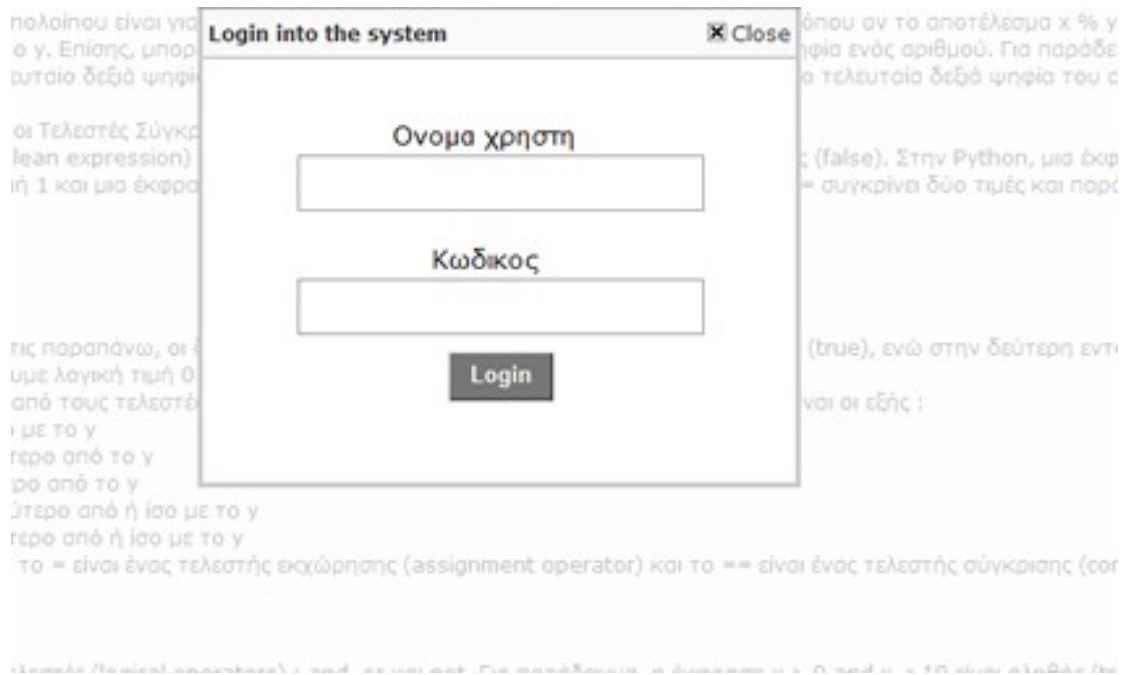
Εικ. 5.2 Αρχική Σελίδα

Για να εισέλθει ο εγγεγραμμένος χρήστης στο σύστημα , κάνει κλικ στο τέλος της σελίδας στο link είσοδος.



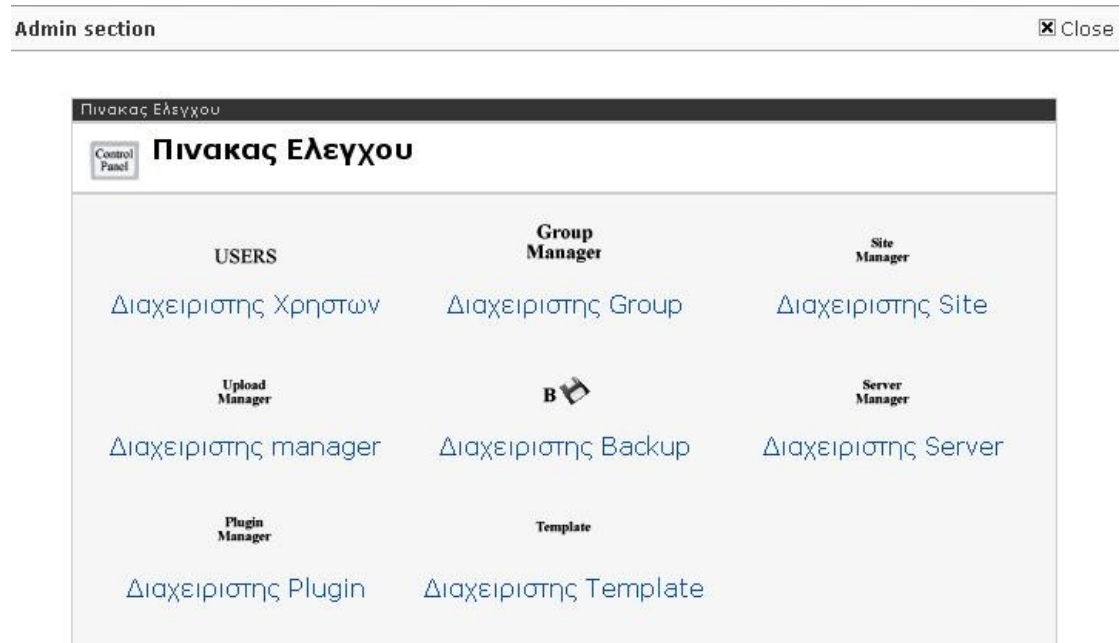
Εικ. 5.3 Κουμπί Εισόδου

Βάζοντας τους έγκυρους κωδικούς ασφαλείας, έχοντας τα προνόμια του διαχειριστή μπορεί να εισέλθει στο σύστημα διαχείρισης της εφαρμογής.



Εικ 5.3 Login Διαδικασία

Η συνολική διαχείριση του συστήματος γίνεται από τον παρακάτω πίνακα ελέγχου.



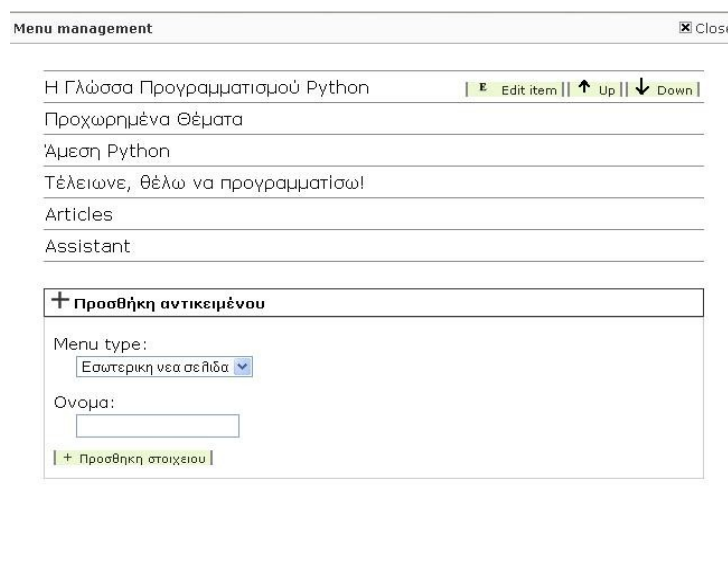
Εικ 5.4 Πίνακας Ελέγχου Admin

Η διαχείριση κειμένων άρθρων και κατηγοριών γίνονται μέσω της κεντρικής σελίδας διαχειριστή.



Εικ 5.5 Προσθήκη Κατηγοριών 1

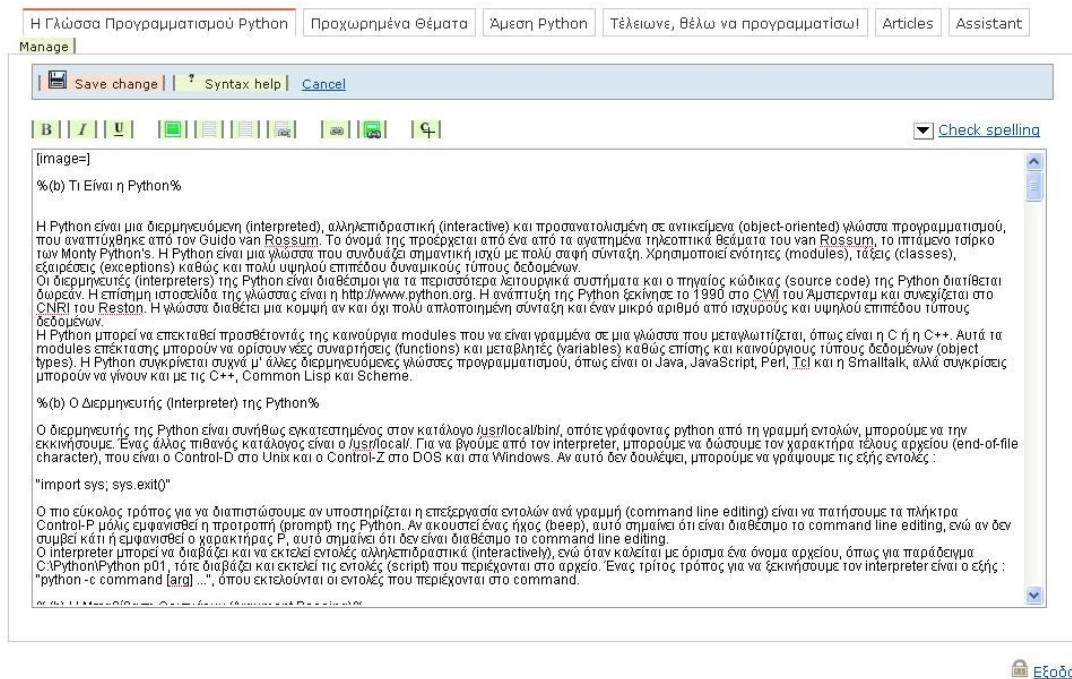
Η διαχείριση κατηγοριών που είναι προνόμιο του διαχειριστή και μόνο γίνεται από το κουμπί επάνω δεξιά από την κεντρική σελίδα διαχειριστή.



Εικ 5.6 Προσθήκη Κατηγοριών 2

Όσον αφορά την προσθήκη και επεξεργασία κειμένων αλλά και υποκατηγοριών γίνεται από την παρακάτω σελίδα.

Python CMS



The screenshot shows the Python CMS interface. At the top, there are navigation tabs: "Η Γλώσσα Προγραμματισμού Python", "Προχωρημένα Θέματα", "Άσκηση Python", "Τέλειωσε, Θέλω να προγραμματίσω!", "Articles", and "Assistant". Below these is a "Manage" section with a "Save change" button, "Syntax help", and "Cancel". The main content area is a text editor with a toolbar containing bold, italic, underline, list, link, unlink, and other icons. The text in the editor is as follows:

[image=]

%(b) Τι Είναι η Python%

Η Python είναι μια διερμηνεύσιμη (interpreted), αλληλεπιδραστική (interactive) και προσανατολισμένη σε αντικείμενα (object-oriented) γλώσσα προγραμματισμού, που αναπτύχθηκε από τον Guido van Rossum. Το όνομά της προέρχεται από ένα από τα αγαπημένα τηλεοπτικά θέματα του van Rossum, το ιταλικό τσίρκο των Monty Python's. Η Python είναι μια γλώσσα που συνδυάζει σημαντική ισχύ με πολύ σαφή σύνταξη. Χρησιμοποιεί ενότητες (modules), τάξεις (classes), εξαιρέσεις (exceptions) καθώς και πολύ υψηλού επιπέδου δυναμικούς τύπους δεδομένων.

Οι διερμηνευτές (interpreters) της Python είναι διαθέσιμοι για τα περισσότερα λειτουργικά συστήματα και ο πηγαίος κώδικας (source code) της Python διατίθεται δωρεάν. Η επίσημη ιστοσελίδα της γλώσσας είναι η <http://www.python.org>. Η ανάπτυξη της Python ξεκίνησε το 1990 στο CNYI του Άμστερνταμ και συνεχίζεται στο CNRI του Reston. Η γλώσσα διαθέτει μια κομψή αν και όχι πολύ απλοποιημένη σύνταξη και έναν μικρό αριθμό από ισχυρούς και υψηλού επιπέδου τύπους δεδομένων.

Η Python μπορεί να επεκταθεί προσθέτοντας της καινούργια modules που να είναι γραμμένα σε μια γλώσσα που μεταγλωττίζεται, όπως είναι η C ή η C++. Αυτά τα modules επέκταση μπορούν να ορίσουν νέες συναρτήσεις (functions) και μεταβλητές (variables) καθώς επίσης και καινούργιους τύπους δεδομένων (object types). Η Python συγκρίνεται συχνά με άλλες διερμηνεύσιμες γλώσσες προγραμματισμού, όπως είναι οι Java, JavaScript, Perl, Tcl και η Smalltalk, αλλά συγκρίσεις μπορούν να γίνουν και με τις C++, Common Lisp και Scheme.

%(b) Ο Διερμηνευτής (Interpreter) της Python%

Ο διερμηνευτής της Python είναι συνήθως εγκατεστημένος στον κατάλογο `/usr/local/bin/`, οπότε γράφοντας `python` από τη γραμμή εντολών, μπορούμε να την εκκινήσουμε. Ένας άλλος πιθανός κατάλογος είναι ο `/usr/local/`. Για να βρούμε από τον interpreter, μπορούμε να δώσουμε τον χαρακτήρα τέλους αρχείου (end-of-file character), που είναι ο Control-D στο Unix και ο Control-Z στο DOS και στα Windows. Αν αυτό δεν δουλέψει, μπορούμε να γράψουμε τις εξής εντολές :

```
"import sys; sys.exit()"
```

Ο πιο εύκολος τρόπος για να διαπιστώσουμε αν υποστηρίζεται η επεξεργασία εντολών ανά γραμμή (command line editing) είναι να πατήσουμε τα πλήκτρα Control-P μόλις εμφανισθεί η προτροπή (prompt) της Python. Αν ακουστεί ένας ήχος (beep), αυτό σημαίνει ότι είναι διαθέσιμο το command line editing, ενώ αν δεν συμβεί κάτι ή εμφανισθεί ο χαρακτήρας P, αυτό σημαίνει ότι δεν είναι διαθέσιμο το command line editing.

Ο interpreter μπορεί να διαβάζει και να εκτελεί εντολές αλληλεπιδραστικά (interactively), ενώ όταν καλείται με όρισμα ένα όνομα αρχείου, όπως για παράδειγμα C:\Python\Python\python r01, τότε διαβάζει και εκτελεί τις εντολές (script) που περιέχονται στο αρχείο. Ένας τρίτος τρόπος για να ξεκινήσουμε τον interpreter είναι ο εξής : `python -c command [arg] ...`, όπου εκτελούνται οι εντολές που περιέχονται στο command.

%(b) Η Μεταβλητή Συστήματος PYTHONPATH

[Εξοδος](#)

Εικ 5.7 Προσθήκη – Επεξεργασία Κειμένων

Τέλος για να βγούμε από το σύστημα κάνουμε κλικ στο "Εξοδος".

Παράρτημα

Βασική Βιβλιοθήκη Python

- [1. Introduction](#)
- [2. Built-in Functions](#)
- [3. Non-essential Built-in Functions](#)
- [4. Built-in Constants](#)
 - [4.1. Constants added by the `site` module](#)
- [5. Built-in Objects](#)
- [6. Built-in Types](#)
 - [6.1. Truth Value Testing](#)
 - [6.2. Boolean Operations — `and`, `or`, `not`](#)
 - [6.3. Comparisons](#)
 - [6.4. Numeric Types — `int`, `float`, `long`, `complex`](#)
 - [6.5. Iterator Types](#)
 - [6.6. Sequence Types — `str`, `unicode`, `list`, `tuple`, `buffer`, `xrange`](#)
 - [6.7. Set Types — `set`, `frozenset`](#)
 - [6.8. Mapping Types — `dict`](#)
 - [6.9. File Objects](#)
 - [6.10. Context Manager Types](#)
 - [6.11. Other Built-in Types](#)
 - [6.12. Special Attributes](#)
- [7. Built-in Exceptions](#)
- [8. String Services](#)
 - [8.1. `string` — Common string operations](#)
 - [8.2. `re` — Regular expression operations](#)
 - [8.3. `struct` — Interpret strings as packed binary data](#)

- [8.4. **difflib** — Helpers for computing deltas](#)
- [8.5. **StringIO** — Read and write strings as files](#)
- [8.6. **cStringIO** — Faster version of **StringIO**](#)
- [8.7. **textwrap** — Text wrapping and filling](#)
- [8.8. **codecs** — Codec registry and base classes](#)
- [8.9. **unicodedata** — Unicode Database](#)
- [8.10. **stringprep** — Internet String Preparation](#)
- [8.11. **fpformat** — Floating point conversions](#)
- [9. Data Types](#)
 - [9.1. **datetime** — Basic date and time types](#)
 - [9.2. **calendar** — General calendar-related functions](#)
 - [9.3. **collections** — High-performance container datatypes](#)
 - [9.4. **heapq** — Heap queue algorithm](#)
 - [9.5. **bisect** — Array bisection algorithm](#)
 - [9.6. **array** — Efficient arrays of numeric values](#)
 - [9.7. **sets** — Unordered collections of unique elements](#)
 - [9.8. **sched** — Event scheduler](#)
 - [9.9. **mutex** — Mutual exclusion support](#)
 - [9.10. **queue** — A synchronized queue class](#)
 - [9.11. **weakref** — Weak references](#)
 - [9.12. **UserDict** — Class wrapper for dictionary objects](#)
 - [9.13. **UserList** — Class wrapper for list objects](#)
 - [9.14. **UserString** — Class wrapper for string objects](#)
 - [9.15. **types** — Names for built-in types](#)
 - [9.16. **new** — Creation of runtime internal objects](#)
 - [9.17. **copy** — Shallow and deep copy operations](#)
 - [9.18. **pprint** — Data pretty printer](#)
 - [9.19. **repr** — Alternate **repr\(\)** implementation](#)
- [10. Numeric and Mathematical Modules](#)
 - [10.1. **numbers** — Numeric abstract base classes](#)
 - [10.2. **math** — Mathematical functions](#)

- [10.3. **cmath** — Mathematical functions for complex numbers](#)
- [10.4. **decimal** — Decimal fixed point and floating point arithmetic](#)
- [10.5. **fractions** — Rational numbers](#)
- [10.6. **random** — Generate pseudo-random numbers](#)
- [10.7. **itertools** — Functions creating iterators for efficient looping](#)
- [10.8. **functools** — Higher order functions and operations on callable objects](#)
- [10.9. **operator** — Standard operators as functions](#)
- [11. File and Directory Access](#)
 - [11.1. **os.path** — Common pathname manipulations](#)
 - [11.2. **fileinput** — Iterate over lines from multiple input streams](#)
 - [11.3. **stat** — Interpreting **stat\(\)** results](#)
 - [11.4. **statvfs** — Constants used with **os.statvfs\(\)**](#)
 - [11.5. **filecmp** — File and Directory Comparisons](#)
 - [11.6. **tempfile** — Generate temporary files and directories](#)
 - [11.7. **glob** — Unix style pathname pattern expansion](#)
 - [11.8. **fnmatch** — Unix filename pattern matching](#)
 - [11.9. **linecache** — Random access to text lines](#)
 - [11.10. **shutil** — High-level file operations](#)
 - [11.11. **dircache** — Cached directory listings](#)
 - [11.12. **macpath** — Mac OS 9 path manipulation functions](#)
- [12. Data Persistence](#)
 - [12.1. **pickle** — Python object serialization](#)
 - [12.2. **cPickle** — A faster **pickle**](#)
 - [12.3. **copy_reg** — Register **pickle** support functions](#)
 - [12.4. **shelve** — Python object persistence](#)
 - [12.5. **marshal** — Internal Python object serialization](#)
 - [12.6. **anydbm** — Generic access to DBM-style databases](#)
 - [12.7. **whichdb** — Guess which DBM module created a database](#)
 - [12.8. **dbm** — Simple “database” interface](#)
 - [12.9. **gdbm** — GNU’s reinterpretation of dbm](#)

- [12.10. **dbhash** — DBM-style interface to the BSD database library](#)
- [12.11. **bsddb** — Interface to Berkeley DB library](#)
- [12.12. **dumbdbm** — Portable DBM implementation](#)
- [12.13. **sqlite3** — DB-API 2.0 interface for SQLite databases](#)
- [13. Data Compression and Archiving](#)
 - [13.1. **zlib** — Compression compatible with **gzip**](#)
 - [13.2. **gzip** — Support for **gzip** files](#)
 - [13.3. **bz2** — Compression compatible with **bzip2**](#)
 - [13.4. **zipfile** — Work with ZIP archives](#)
 - [13.5. **tarfile** — Read and write tar archive files](#)
- [14. File Formats](#)
 - [14.1. **csv** — CSV File Reading and Writing](#)
 - [14.2. **ConfigParser** — Configuration file parser](#)
 - [14.3. **robotparser** — Parser for robots.txt](#)
 - [14.4. **netrc** — netrc file processing](#)
 - [14.5. **xdrlib** — Encode and decode XDR data](#)
 - [14.6. **plistlib** — Generate and parse Mac OS X **.plist** files](#)
- [15. Cryptographic Services](#)
 - [15.1. **hashlib** — Secure hashes and message digests](#)
 - [15.2. **hmac** — Keyed-Hashing for Message Authentication](#)
 - [15.3. **md5** — MD5 message digest algorithm](#)
 - [15.4. **sha** — SHA-1 message digest algorithm](#)
- [16. Generic Operating System Services](#)
 - [16.1. **os** — Miscellaneous operating system interfaces](#)
 - [16.2. **io** — Core tools for working with streams](#)
 - [16.3. **time** — Time access and conversions](#)
 - [16.4. **optparse** — More powerful command line option parser](#)
 - [16.5. **getopt** — Parser for command line options](#)
 - [16.6. **logging** — Logging facility for Python](#)
 - [16.7. **getpass** — Portable password input](#)
 - [16.8. **curses** — Terminal handling for character-cell displays](#)

- [16.9. **curses.textpad** — Text input widget for curses programs](#)
- [16.10. **curses.wrapper** — Terminal handler for curses programs](#)
- [16.11. **curses.ascii** — Utilities for ASCII characters](#)
- [16.12. **curses.panel** — A panel stack extension for curses.](#)
- [16.13. **platform** — Access to underlying platform’s identifying data.](#)
- [16.14. **errno** — Standard errno system symbols](#)
- [16.15. **ctypes** — A foreign function library for Python.](#)
- [17. Optional Operating System Services](#)
 - [17.1. **select** — Waiting for I/O completion](#)
 - [17.2. **threading** — Higher-level threading interface](#)
 - [17.3. **thread** — Multiple threads of control](#)
 - [17.4. **dummy_threading** — Drop-in replacement for the **threading** module](#)
 - [17.5. **dummy_thread** — Drop-in replacement for the **thread** module](#)
 - [17.6. **multiprocessing** — Process-based “threading” interface](#)
 - [17.7. **mmap** — Memory-mapped file support](#)
 - [17.8. **readline** — GNU readline interface](#)
 - [17.9. **rlcompleter** — Completion function for GNU readline](#)
- [18. Interprocess Communication and Networking](#)
 - [18.1. **subprocess** — Subprocess management](#)
 - [18.2. **socket** — Low-level networking interface](#)
 - [18.3. **ssl** — SSL wrapper for socket objects](#)
 - [18.4. **signal** — Set handlers for asynchronous events](#)
 - [18.5. **popen2** — Subprocesses with accessible I/O streams](#)
 - [18.6. **asyncore** — Asynchronous socket handler](#)
 - [18.7. **asynchat** — Asynchronous socket command/response handler](#)
- [19. Internet Data Handling](#)
 - [19.1. **email** — An email and MIME handling package](#)
 - [19.2. **json** — JSON encoder and decoder](#)
 - [19.3. **mailcap** — Mailcap file handling](#)
 - [19.4. **mailbox** — Manipulate mailboxes in various formats](#)

- [19.5. **mhlib** — Access to MH mailboxes](#)
- [19.6. **mimetools** — Tools for parsing MIME messages](#)
- [19.7. **mimetypes** — Map filenames to MIME types](#)
- [19.8. **MimeWriter** — Generic MIME file writer](#)
- [19.9. **mimify** — MIME processing of mail messages](#)
- [19.10. **multifile** — Support for files containing distinct parts](#)
- [19.11. **rfc822** — Parse RFC 2822 mail headers](#)
- [19.12. **base64** — RFC 3548: Base16, Base32, Base64 Data Encodings](#)
- [19.13. **binhex** — Encode and decode binhex4 files](#)
- [19.14. **binascii** — Convert between binary and ASCII](#)
- [19.15. **quopri** — Encode and decode MIME quoted-printable data](#)
- [19.16. **uu** — Encode and decode uuencode files](#)
- [20. Structured Markup Processing Tools](#)
 - [20.1. **HTMLParser** — Simple HTML and XHTML parser](#)
 - [20.2. **sgmlib** — Simple SGML parser](#)
 - [20.3. **htmlib** — A parser for HTML documents](#)
 - [20.4. **htmlentitydefs** — Definitions of HTML general entities](#)
 - [20.5. **xml.parsers.expat** — Fast XML parsing using Expat](#)
 - [20.6. **xml.dom** — The Document Object Model API](#)
 - [20.7. **xml.dom.minidom** — Lightweight DOM implementation](#)
 - [20.8. **xml.dom.pulldom** — Support for building partial DOM trees](#)
 - [20.9. **xml.sax** — Support for SAX2 parsers](#)
 - [20.10. **xml.sax.handler** — Base classes for SAX handlers](#)
 - [20.11. **xml.sax.saxutils** — SAX Utilities](#)
 - [20.12. **xml.sax.xmlreader** — Interface for XML parsers](#)
 - [20.13. **xml.etree.ElementTree** — The ElementTree XML API](#)
- [21. Internet Protocols and Support](#)
 - [21.1. **webbrowser** — Convenient Web-browser controller](#)
 - [21.2. **cgi** — Common Gateway Interface support.](#)
 - [21.3. **cgitb** — Traceback manager for CGI scripts](#)
 - [21.4. **wsgiref** — WSGI Utilities and Reference Implementation](#)

- [21.5. urllib](#) — Open arbitrary resources by URL
- [21.6. urllib2](#) — extensible library for opening URLs
- [21.7. httplib](#) — HTTP protocol client
- [21.8. ftplib](#) — FTP protocol client
- [21.9. poplib](#) — POP3 protocol client
- [21.10. imaplib](#) — IMAP4 protocol client
- [21.11. nntplib](#) — NNTP protocol client
- [21.12. smtplib](#) — SMTP protocol client
- [21.13. smtpd](#) — SMTP Server
- [21.14. telnetlib](#) — Telnet client
- [21.15. uuid](#) — UUID objects according to RFC 4122
- [21.16. urlparse](#) — Parse URLs into components
- [21.17. SocketServer](#) — A framework for network servers
- [21.18. BaseHTTPServer](#) — Basic HTTP server
- [21.19. SimpleHTTPServer](#) — Simple HTTP request handler
- [21.20. CGIHTTPServer](#) — CGI-capable HTTP request handler
- [21.21. cookielib](#) — Cookie handling for HTTP clients
- [21.22. Cookie](#) — HTTP state management
- [21.23. xmlrpclib](#) — XML-RPC client access
- [21.24. SimpleXMLRPCServer](#) — Basic XML-RPC server
- [21.25. DocXMLRPCServer](#) — Self-documenting XML-RPC server
- [22. Multimedia Services](#)
 - [22.1. audioop](#) — Manipulate raw audio data
 - [22.2. imageop](#) — Manipulate raw image data
 - [22.3. aifc](#) — Read and write AIFF and AIFC files
 - [22.4. sunau](#) — Read and write Sun AU files
 - [22.5. wave](#) — Read and write WAV files
 - [22.6. chunk](#) — Read IFF chunked data
 - [22.7. colorsys](#) — Conversions between color systems
 - [22.8. imghdr](#) — Determine the type of an image
 - [22.9. sndhdr](#) — Determine type of sound file

- [22.10. **ossaudiodev** — Access to OSS-compatible audio devices](#)
- [23. Internationalization](#)
 - [23.1. **gettext** — Multilingual internationalization services](#)
 - [23.2. **locale** — Internationalization services](#)
- [24. Program Frameworks](#)
 - [24.1. **cmd** — Support for line-oriented command interpreters](#)
 - [24.2. **shlex** — Simple lexical analysis](#)
- [25. Graphical User Interfaces with Tk](#)
 - [25.1. **Tkinter** — Python interface to Tcl/Tk](#)
 - [25.2. **Tix** — Extension widgets for Tk](#)
 - [25.3. **ScrolledText** — Scrolled Text Widget](#)
 - [25.4. **turtle** — Turtle graphics for Tk](#)
 - [25.5. IDLE](#)
 - [25.6. Other Graphical User Interface Packages](#)
- [26. Development Tools](#)
 - [26.1. **pydoc** — Documentation generator and online help system](#)
 - [26.2. **doctest** — Test interactive Python examples](#)
 - [26.3. **unittest** — Unit testing framework](#)
 - [26.4. 2to3 - Automated Python 2 to 3 code translation](#)
 - [26.5. **test** — Regression tests package for Python](#)
 - [26.6. **test.test_support** — Utility functions for tests](#)
- [27. Debugging and Profiling](#)
 - [27.1. **bdb** — Debugger framework](#)
 - [27.2. **pdb** — The Python Debugger](#)
 - [27.3. Debugger Commands](#)
 - [27.4. The Python Profilers](#)
 - [27.5. **hotshot** — High performance logging profiler](#)
 - [27.6. **timeit** — Measure execution time of small code snippets](#)
 - [27.7. **trace** — Trace or track Python statement execution](#)
- [28. Python Runtime Services](#)
 - [28.1. **sys** — System-specific parameters and functions](#)

- [28.2. `__builtin__` — Built-in objects](#)
- [28.3. `future_builtins` — Python 3 builtins](#)
- [28.4. `__main__` — Top-level script environment](#)
- [28.5. `warnings` — Warning control](#)
- [28.6. `contextlib` — Utilities for `with`-statement contexts.](#)
- [28.7. `abc` — Abstract Base Classes](#)
- [28.8. `atexit` — Exit handlers](#)
- [28.9. `traceback` — Print or retrieve a stack traceback](#)
- [28.10. `__future__` — Future statement definitions](#)
- [28.11. `gc` — Garbage Collector interface](#)
- [28.12. `inspect` — Inspect live objects](#)
- [28.13. `site` — Site-specific configuration hook](#)
- [28.14. `user` — User-specific configuration hook](#)
- [28.15. `fpectl` — Floating point exception control](#)
- [29. Custom Python Interpreters](#)
 - [29.1. `code` — Interpreter base classes](#)
 - [29.2. `codeop` — Compile Python code](#)
- [30. Restricted Execution](#)
 - [30.1. `rexec` — Restricted execution framework](#)
 - [30.2. `Bastion` — Restricting access to objects](#)
- [31. Importing Modules](#)
 - [31.1. `imp` — Access the `import` internals](#)
 - [31.2. `importlib` — Import utilities](#)
 - [31.3. `zipimport` — Import modules from Zip archives](#)
 - [31.4. `pkgutil` — Package extension utility](#)
 - [31.5. `modulefinder` — Find modules used by a script](#)
 - [31.6. `runpy` — Locating and executing Python modules](#)
- [32. Python Language Services](#)
 - [32.1. `parser` — Access Python parse trees](#)
 - [32.2. Abstract Syntax Trees](#)
 - [32.3. `symtable` — Access to the compiler's symbol tables](#)

- [32.4. **symbol** — Constants used with Python parse trees](#)
- [32.5. **token** — Constants used with Python parse trees](#)
- [32.6. **keyword** — Testing for Python keywords](#)
- [32.7. **tokenize** — Tokenizer for Python source](#)
- [32.8. **tabnanny** — Detection of ambiguous indentation](#)
- [32.9. **pyclbr** — Python class browser support](#)
- [32.10. **py_compile** — Compile Python source files](#)
- [32.11. **compileall** — Byte-compile Python libraries](#)
- [32.12. **dis** — Disassembler for Python bytecode](#)
- [32.13. **pickletools** — Tools for pickle developers.](#)
- [32.14. **distutils** — Building and installing Python modules](#)
- [33. Python compiler package](#)
 - [33.1. The basic interface](#)
 - [33.2. Limitations](#)
 - [33.3. Python Abstract Syntax](#)
 - [33.4. Using Visitors to Walk ASTs](#)
 - [33.5. Bytecode Generation](#)
- [34. Miscellaneous Services](#)
 - [34.1. **formatter** — Generic output formatting](#)
- [35. MS Windows Specific Services](#)
 - [35.1. **msilib** — Read and write Microsoft Installer files](#)
 - [35.2. **msvcrt** – Useful routines from the MS VC++ runtime](#)
 - [35.3. **_winreg** – Windows registry access](#)
 - [35.4. **winsound** — Sound-playing interface for Windows](#)
- [36. Unix Specific Services](#)
 - [36.1. **posix** — The most common POSIX system calls](#)
 - [36.2. **pwd** — The password database](#)
 - [36.3. **spwd** — The shadow password database](#)
 - [36.4. **grp** — The group database](#)
 - [36.5. **crypt** — Function to check Unix passwords](#)
 - [36.6. **dl** — Call C functions in shared objects](#)

- [36.7. **termios** — POSIX style tty control](#)
- [36.8. **tty** — Terminal control functions](#)
- [36.9. **pty** — Pseudo-terminal utilities](#)
- [36.10. **fcntl** — The **fcntl\(\)** and **ioctl\(\)** system calls](#)
- [36.11. **pipes** — Interface to shell pipelines](#)
- [36.12. **posixfile** — File-like objects with locking support](#)
- [36.13. **resource** — Resource usage information](#)
- [36.14. **nis** — Interface to Sun's NIS \(Yellow Pages\)](#)
- [36.15. **syslog** — Unix syslog library routines](#)
- [36.16. **commands** — Utilities for running commands](#)
- [37. Mac OS X specific services](#)
 - [37.1. **ic** — Access to the Mac OS X Internet Config](#)
 - [37.2. **MacOS** — Access to Mac OS interpreter features](#)
 - [37.3. **macostools** — Convenience routines for file manipulation](#)
 - [37.4. **findertools** — The **finder**'s Apple Events interface](#)
 - [37.5. **EasyDialogs** — Basic Macintosh dialogs](#)
 - [37.6. **FrameWork** — Interactive application framework](#)
 - [37.7. **autoGIL** — Global Interpreter Lock handling in event loops](#)
 - [37.8. Mac OS Toolbox Modules](#)
 - [37.9. **ColorPicker** — Color selection dialog](#)
- [38. MacPython OSA Modules](#)
 - [38.1. **gensuitemodule** — Generate OSA stub packages](#)
 - [38.2. **aetools** — OSA client support](#)
 - [38.3. **aepack** — Conversion between Python variables and AppleEvent data containers](#)
 - [38.4. **aetypes** — AppleEvent objects](#)
 - [38.5. **MiniAEFrame** — Open Scripting Architecture server support](#)
- [39. SGI IRIX Specific Services](#)
 - [39.1. **al** — Audio functions on the SGI](#)
 - [39.2. **AL** — Constants used with the **al** module](#)
 - [39.3. **cd** — CD-ROM access on SGI systems](#)

- [39.4. **fl** — FORMS library for graphical user interfaces](#)
- [39.5. **FL** — Constants used with the **fl** module](#)
- [39.6. **flp** — Functions for loading stored FORMS designs](#)
- [39.7. **fm** — *Font Manager* interface](#)
- [39.8. **gl** — *Graphics Library* interface](#)
- [39.9. **DEVICE** — Constants used with the **gl** module](#)
- [39.10. **GL** — Constants used with the **gl** module](#)
- [39.11. **imgfile** — Support for SGI imglib files](#)
- [39.12. **jpeg** — Read and write JPEG files](#)
- [40. SunOS Specific Services](#)
 - [40.1. **sunaudiodev** — Access to Sun audio hardware](#)
 - [40.2. **SUNAUDIODEV** — Constants used with **sunaudiodev**](#)
- [41. Undocumented Modules](#)
 - [41.1. Miscellaneous useful utilities](#)
 - [41.2. Platform specific modules](#)
 - [41.3. Multimedia](#)
 - [41.4. Undocumented Mac OS modules](#)
 - [41.5. Obsolete](#)
 - [41.6. SGI-specific Extension modules](#)

Μπορείτε να βρείτε λεπτομέρειες στο <http://docs.python.org/library/>

Περιεχόμενα πρόσφατων έτοιμων πακέτων της Python

Updated	Package	Description
2009-05-22	python-gettext 1.0	Python Gettext implementation.
2009-05-22	Plone 3.3rc3	The Plone Content Management System
2009-05-22	belier 0.8	Ssh connection generation tool
2009-05-22	collective.types.citation 9.05dev-	Citation content type

	r86618	
2009-05-22	zope.mimetype 1.1.2	A simple package for working with MIME content types
2009-05-22	etm 266	event and task manager
2009-05-22	hurry.custom 0.5	A framework for allowing customizing templates
2009-05-22	pcp 0.1	Python Interface to SGI's Performance Co-Pilot client API
2009-05-22	WebTest 1.2	Helper to test WSGI applications
2009-05-22	flaker 0.2.4	Flaker.pl client
2009-05-22	bzs-svn 0.6.1	Support for Subversion branches in Bazaar
2009-05-22	ruffus 1.0.1	Light-weight Python Computational Pipeline Management
2009-05-22	pymongo 0.10.2	Python driver for MongoDB < http://www.mongodb.org >
2009-05-22	pygeoip 0.1	Pure Python GeoIP API
2009-05-22	motmot.FlyMovieFormat 0.5.4	support for .fmf (fly movie format) files
2009-05-22	zope.app.publisher 3.7.0	Implementations and means for configuration of Zope 3-style views and resources.
2009-05-22	zope.app.component 3.8.2	Local Zope Component Support
2009-05-22	plinkpipeline 1.3	
2009-05-22	tiddlyweb 0.9.35	An optionally headless, extensible RESTful datastore for TiddlyWiki
2009-05-22	experimental.opaquespeedup 1.0	Plone-specific optimization of looking up CMF's "opaque items"
2009-05-22	PYPOWER 0.1.1	Port of MATPOWER to Python
2009-05-22	z3ext.wizard 1.0.0	form wizard library
2009-05-22	z3ext.layoutform 1.4.0	UI Skin for z3c.form based on z3ext.layout
2009-05-22	repoze.what.plugins.redis 1.0rc2	The repoze.what Redis plugin
2009-05-22	bcryptor 1.1-1	Python wrapper for bcrypt
2009-05-22	collective.portlet.actions 1.0.1	A portlet that provides the links of an action category
2009-05-22	betahaus.emaillogin 0.2	Enables you to login using your emailaddress.

2009-05-22	Products.kupu 1.4.13	Kupu wysiwyg editor for Plone
2009-05-22	django-faces 0.1.10	
2009-05-22	cx_Oracle 5.0.2	Python interface to Oracle
2009-05-22	ltprotocol 0.2.0	Twisted-based client and server for protocols which begin with a length and type field
2009-05-21	xlutils 1.3.1	Utilities for working with Excel files that require both xlrd and xlwt
2009-05-21	mpgutils 0.8.3	
2009-05-21	collective.contacts 1.3.2	Address book content type for Plone
2009-05-21	grok 1.0a4	Grok: Now even cavemen can use Zope 3!
2009-05-21	pyhdf5io 0.7	Python module containing high-level hdf5 load and save functions.
2009-05-21	django-extensions 0.4.1	Extensions for Django
2009-05-21	PyISAPIe 1.1.0-rc1	1.1.0 Release Candidate: Python ISAPI extension for IIS with Django, WSGI, and Trac support
2009-05-21	PyQtShell 0.4.2	Pydee development environment and its PyQt4-based IDE tools: interactive Python shell, Python code editor, workspace (dict/list/string/array editor), doc viewer, history log, environment variables editor, ...
2009-05-21	membrete 0.1a2	Contact form app for Django

Περισσότερα μπορείτε να βρείτε στο <http://pypi.python.org/pypi>