

ΥΠΟΥΡΓΕΙΟ ΕΘΝΙΚΗΣ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΓΕΝΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΕΚΠΑΙΔΕΥΣΗΣ ΕΝΗΛΙΚΩΝ
ΙΝΣΤΙΤΟΥΤΟ ΔΙΑΡΚΟΥΣ ΕΚΠΑΙΔΕΥΣΗΣ ΕΝΗΛΙΚΩΝ



ΤΕΧΝΟΛΟΓΙΕΣ ΠΛΗΡΟΦΟΡΙΚΗΣ- ΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΛΗΡΟΦΟΡΙΚΗ VI: ΔΥΝΑΜΙΚΕΣ ΕΦΑΡΜΟΓΕΣ ΠΑΓΚΟΣΜΙΟΥ ΙΣΤΟΥ

ΚΕΝΤΡΑ ΕΚΠΑΙΔΕΥΣΗΣ ΕΝΗΛΙΚΩΝ



ΥΠΟΥΡΓΕΙΟ ΕΘΝΙΚΗΣ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΕΑΕΚ



ΕΥΡΩΠΑΪΚΗ ΕΝΩΣΗ
ΣΥΧΡΗΜΑΤΟΔΟΤΗΣΗ
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



Η ΠΑΙΔΕΙΑ ΣΤΗΝ ΚΟΡΥΦΗ
Επιχειρησιακό Πρόγραμμα
Εκπαίδευσης και Αρχικής
Επαγγελματικής Κατάρτισης

Επιστημονική Ευθύνη	Κωνσταντίνος Σιασιάκος, Δρ Πληροφορικής
Συγγραφή	Παντελής Μπαλής

Το παρόν εκπαιδευτικό υλικό παράχθηκε στο πλαίσιο του Έργου «**Κέντρα Εκπαίδευσης Ενηλίκων II**», το οποίο εντάσσεται στο **Ε.Π.Ε.Α.Ε.Κ. II** του **ΥΠ.Ε.Π.Θ.**, Μέτρο 1.1. Ενέργεια 1.1.2.Β. και συγχρηματοδοτείται από την **Ευρωπαϊκή Ένωση (Ε.Κ.Τ.)**.



ΥΠΟΥΡΓΕΙΟ ΕΘΝΙΚΗΣ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΕΑΕΚ



ΕΥΡΩΠΑΪΚΗ ΕΝΩΣΗ
ΣΥΧΡΗΜΑΤΟΔΟΤΗΣΗ
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



Η ΠΑΙΔΕΙΑ ΣΤΗΝ ΚΟΡΥΦΗ
Επιχειρησιακό Πρόγραμμα
Εκπαίδευσης και Αρχικής
Επαγγελματικής Κατάρτισης

ΠΛΗΡΟΦΟΡΙΚΗ 6

Κεφάλαιο 1^ο «Η γλώσσα σήμανσης XML»

Σκοπός και στόχοι του κεφαλαίου	1
Ενότητα 1.1 Εισαγωγή στην XML	2
Ενότητα 1.2 Μοντελοποίηση εγγράφων	3
Ενότητα 1.3 XML έγγραφα	5
-Η δομή του εγγράφου XML	5
-Σωστά διατυπωμένα XML έγγραφα	6
- Έγκυρα XML έγγραφα	8
Ενότητα 1.4 Επεξεργασία XML εγγράφων	12
Ενότητα 1.5 Δημιουργία XSLT φύλλων στυλ	17
Ενότητα 1.6 Δραστηριότητες ανακεφαλαίωσης	20

Κεφάλαιο 2^ο «Προγραμματισμός από την πλευρά του πελάτη»

Σκοπός και στόχοι του κεφαλαίου	21
Ενότητα 2.1 Εισαγωγή στον προγραμματισμό από την πλευρά του πελάτη	22
Ενότητα 2.2 Επισκόπηση της γλώσσας Javascript	23
-Εισαγωγή	23
-Ενσωμάτωση της Javascript σε ένα έγγραφο HTML	23
-Η χρήση των αντικειμένων στη Javascript	25
-Τα χειριστήρια συμβάντος (Event Handlers) της Javascript	30
-Η Javascript ως γλώσσα δομημένου προγραμματισμού	32
Ενότητα 2.3 Η γλώσσα DHTML	40
-Εισαγωγή στην DHTML	40
-Διαδοχικά φύλλα στυλ (Cascading Style Sheets – css)	40
Ενότητα 2.4 Συνδυάζοντας HTML, Javascript και διαδοχικά φύλλα στυλ	48
Ενότητα 2.5 Δραστηριότητες ανακεφαλαίωσης	56

Κεφάλαιο 3^ο «Προγραμματισμός από την πλευρά του εξυπηρετητή: PHP»

Σκοπός και στόχοι του κεφαλαίου	57
Ενότητα 3.1 Αρχιτεκτονική εφαρμογών από την πλευρά του εξυπηρετητή	59
Ενότητα 3.2 Διαχείριση βάσεων δεδομένων σε περιβάλλον MySQL	61
-Εισαγωγή	61
-Βασικά χαρακτηριστικά της MySQL	61
-Οι βάσεις δεδομένων της MySQL	62
-Η γλώσσα SQL	62
-Βασικές λειτουργίες διαχείρισης βάσης δεδομένων με τη MySQL	63
-Διαχείριση βάσης δεδομένων με το περιβάλλον PHPMyAdmin	71
Ενότητα 3.3 Επισκόπηση της γλώσσας PHP	76

-Εισαγωγή στη γλώσσα PHP	76
-Εισαγωγή στον προγραμματισμό με την PHP	76
-Οι συνιστώσες των προγραμμάτων PHP	79
-Αλληλεπίδραση με το χρήστη – Χειρισμός φόρμας με την PHP	82
Ενότητα 3.4 Δραστηριότητες ανακεφαλαίωσης	86

Κεφάλαιο 4^ο «Άλλες γλώσσες προγραμματισμού από την πλευρά του εξυπηρετητή»

Σκοπός και στόχοι του κεφαλαίου	87
Ενότητα 4.1 Active Server Pages (ASP)	89
-Εισαγωγή	89
-Βασικά χαρακτηριστικά της γλώσσας ASP	89
-Βασικές Συνιστώσες των προγραμμάτων ASP	91
-Χειρισμός φόρμας με των ASP	94
-Δραστηριότητες ανακεφαλαίωσης.....	97
Ενότητα 4.2 JSP.....	98
-Εισαγωγή	98
-Προγραμματίζοντας με τη JSP	99
-Ετικέτες (tags), Οδηγίες (directive) και Αντικείμενα (Objects).....	100
-Διαχωρίζοντας τον κώδικα από το περιεχόμενο: Η έννοια του Javabean.....	102
-Δραστηριότητες ανακεφαλαίωσης	106
Γλωσσάρι	107
Ευρετήριο	109
Βιβλιογραφία	111

ΚΕΦΑΛΑΙΟ 1ο



Η Γλώσσα Σήμανσης XML

Δυναμικές Εφαρμογές Παγκόσμιου Ιστού

Σκοπός και στόχοι του κεφαλαίου

Σκοπός του κεφαλαίου είναι η εισαγωγή των ενηλίκων στη γλώσσα σήμανσης XML.

Μετά την ολοκλήρωση του κεφαλαίου, οι μαθητές θα είναι σε θέση να:

- κατανοούν τις βασικές αρχές της γλώσσας σήμανσης XML
- γνωρίζουν τον τρόπο με τον οποίο γίνεται η μοντελοποίηση εγγράφων XML
- μπορούν να διακρίνουν τις διαφορές ανάμεσα στα σωστά διατυπωμένα και τα έγκυρα XML έγγραφα
- γνωρίζουν βασικές έννοιες σχετικά με τη δημιουργία, επεξεργασία και εμφάνιση XML εγγράφων
- κατανοούν τη σημασία των φύλλων στυλ XSL
- εφαρμόζουν τις παραπάνω γνώσεις συνδυαστικά, την για τη δημιουργία και παρουσίαση απλών XML εγγράφων.

Έννοιες-Κλειδιά

Extensible Markup Language, Έγγραφο XML, Μοντελοποίηση εγγράφων, δηλώσεις τύπου εγγράφου, Extensible Stylesheet Language,

Εισαγωγικές Παρατηρήσεις

Η Επεκτάσιμη Γλώσσα Σήμανσης (Extensible Markup Language - XML) αποτελεί σήμερα την πιο πολλά υποσχόμενη γλώσσα αποθήκευσης και μετάδοσης πληροφοριών στον Παγκόσμιο Ιστό, παρέχοντας εξελιγμένες δυνατότητες δόμησης και περιγραφής των πληροφοριών, προσφέροντας έτσι την ιδανική λύση στο πρόβλημα χειρισμού της ταχύτητα αυξανόμενης ποσότητας και πολυπλοκότητας των πληροφοριών που χρειάζεται να δημοσιευθούν στο διαδίκτυο.

Ενότητα 1.1 Εισαγωγή στην XML

Τι είναι η XML

Η Επεκτάσιμη Γλώσσα Σήμανσης (Extensible Markup Language - XML) αποτελεί σήμερα την πιο πολλά υποσχόμενη γλώσσα αποθήκευσης και μετάδοσης πληροφοριών στον Παγκόσμιο Ιστό. Αν και η Γλώσσα Σήμανσης Υπερ-κειμένου (Hypertext Markup Language - HTML) είναι προς το παρόν η πιο διαδεδομένη γλώσσα στο χώρο της δημιουργίας ιστοσελίδων, διαθέτει περιορισμένες δυνατότητες αποθήκευσης πληροφοριών. Αντίθετα, η XML διαθέτει εξαιρετικά ευέλικτο συντακτικό με το οποίο μπορούμε ουσιαστικά να περιγράψουμε οποιαδήποτε πληροφορία, από μια απλή συνταγή μαγειρικής μέχρι μια σύνθετη βάση δεδομένων. Εκτός αυτού, ένα έγγραφο XML — σε συνδυασμό με ένα φύλλο στυλ ή μια συμβατική σελίδα HTML — μπορεί να απεικονιστεί εύκολα σε ένα φυλλομετρητή (browser). Χάρη στην ικανότητα που έχει ένα έγγραφο XML να δομεί και να περιγράφει τόσο αποτελεσματικά τις πληροφορίες που περιέχει (μέσω ετικετών), ο φυλλομετρητής μπορεί να εντοπίζει, να εξάγει, να ταξινομεί, να φιλτράρει, να τακτοποιεί, και να χειρίζεται με εξαιρετικά ευέλικτους τρόπους τις πληροφορίες αυτές. Με βάση τα παραπάνω, η XML προσφέρει την ιδανική λύση στο πρόβλημα χειρισμού της ταχύτητα αυξανόμενης ποσότητας και πολυπλοκότητας των πληροφοριών που χρειάζεται να δημοσιευθούν στον Ιστό.

Γιατί είναι απαραίτητη η XML

Η HTML παρέχει ένα σταθερό σύνολο προκαθορισμένων *στοιχείων* με τα οποία μπορούμε να συμβολίζουμε τα περιεχόμενα μιας τυπικής ιστοσελίδας γενικού περιεχομένου. Παραδείγματα τέτοιων στοιχείων είναι οι επικεφαλίδες, οι ενότητες, οι λίστες, οι πίνακες, οι εικόνες και οι σύνδεσμοι. Ο φυλλομετρητής που εμφανίζει τη σελίδα HTML αναγνωρίζει αυτά τα βασικά στοιχεία και γνωρίζει πώς να τα μορφοποιήσει και να τα εμφανίσει. Για παράδειγμα, ο φυλλομετρητής εμφανίζει πάντοτε μια επικεφαλίδα <H1> με τη μεγαλύτερη γραμματοσειρά, μια επικεφαλίδα <H2> με μικρότερη γραμματοσειρά και ένα στοιχείο <P> με ακόμα μικρότερη γραμματοσειρά. Αν και το σύνολο των προκαθορισμένων στοιχείων HTML έχει εμπλουτιστεί σημαντικά από την πρώτη έκδοση της HTML, η γλώσσα αυτή εξακολουθεί να είναι ακατάλληλη για τον προσδιορισμό πολλών ειδών εγγράφων. Ακολουθούν ορισμένα παραδείγματα εγγράφων που δεν μπορούν να προσδιοριστούν επαρκώς με την HTML:

- Έγγραφα που περιλαμβάνουν εξεζητημένα στοιχεία τα οποία απαιτούνται για τη σήμανση, για παράδειγμα, μιας μουσικής παρτιτούρας ή ενός συνόλου μαθηματικών εξισώσεων.
- Έγγραφα που σχετίζονται με εργασίες βάσης δεδομένων, όπως αναζήτηση και ταξινόμηση πεδίων.
- Έγγραφα στα οποία είναι απαραίτητη η ιεραρχική αποτύπωση της δομής τους.

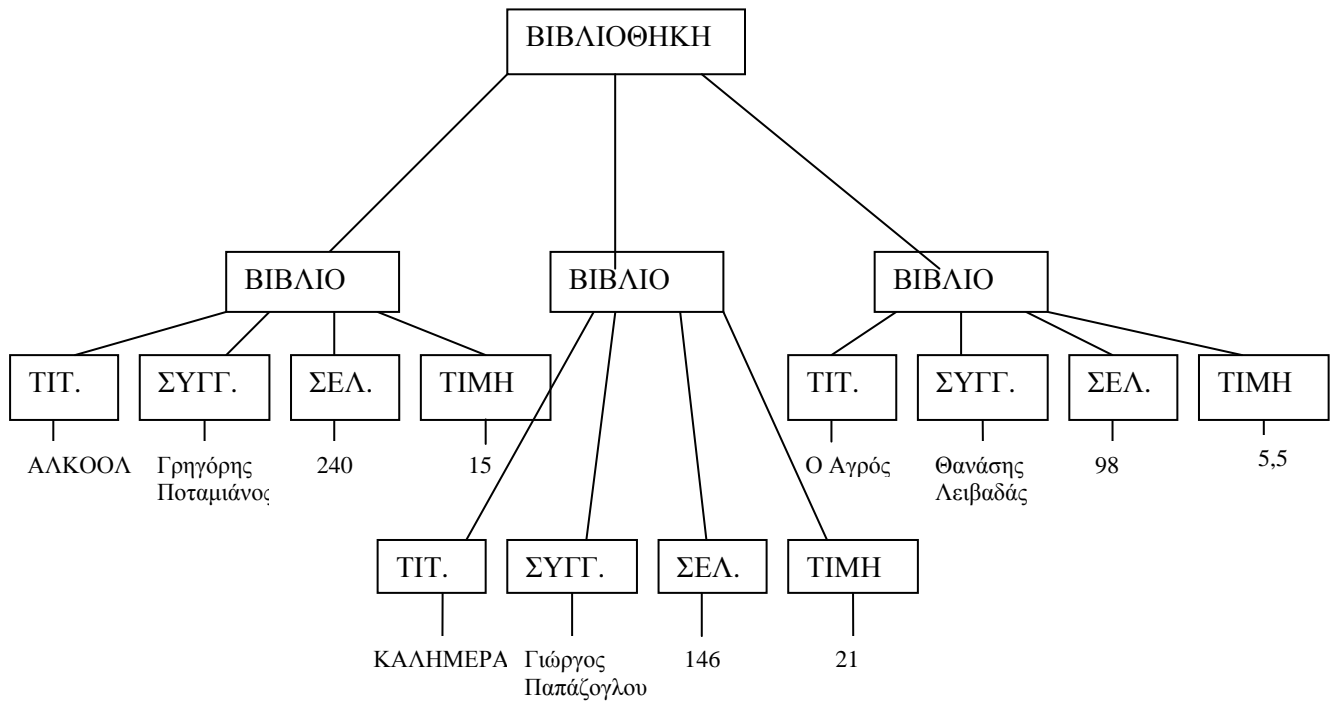
Η XML παρέχει τη λύση στους παραπάνω περιορισμούς της HTML, όπως θα δούμε στη συνέχεια.

Ενότητα 1.2 Μοντελοποίηση εγγράφων

Προκειμένου να κατανοήσουμε τις βασικές έννοιες της XML, χρειαζόμαστε ένα μοντέλο για την αναπαράσταση των εγγράφων στη μνήμη του υπολογιστή. Στην XML, τα έγγραφα αναπαρίστανται με μια δομή δέντρου. Στην κορυφή του δέντρου υπάρχει ένα βασικό στοιχείο κι αυτό το στοιχείο έχει έναν ή περισσότερους κόμβους – παιδιά (child-nodes), οι οποίοι αντιστοιχούν στους κλάδους (branches) του δέντρου. Ας υποθέσουμε ότι έχουμε το παρακάτω απλό έγγραφο το οποίο περιλαμβάνει καταχωρήσεις βιβλίων μιας βιβλιοθήκης.

Παράδειγμα 1.1	
BIBΛΙΟΘΗΚΗ	
Τίτλος:	ΑΛΚΟΟΛ
Συγγραφέας:	Γρηγόρης Ποταμιάνος
Σελίδες:	240
Τιμή:	15
Τίτλος:	ΚΑΛΗΜΕΡΑ
Συγγραφέας:	Γιώργος Παπάζογλου
Σελίδες:	146
Τιμή:	21
Τίτλος:	Ο Αγρός
Συγγραφέας:	Θανάσης Λειβαδάς
Σελίδες:	98
Τιμή:	5,5

Το μοντέλο αυτού του εγγράφου φαίνεται στο παρακάτω σχήμα:

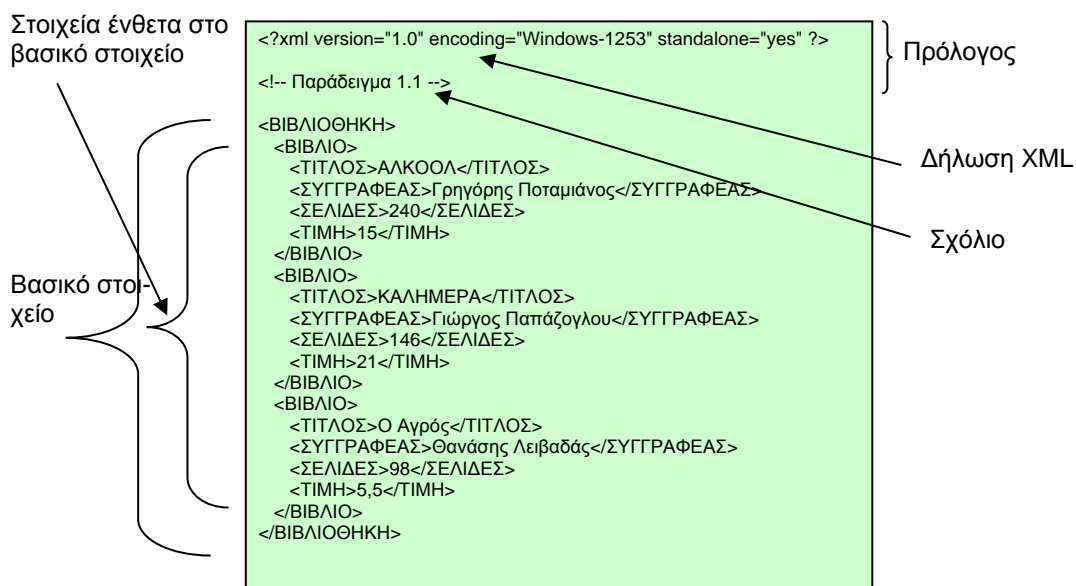


Σχήμα 1.1 Δενδροειδής αναπαράσταση του εγγράφου του Παραδείγματος 1.1

Ενότητα 1.3 XML Εγγραφα

1.3.1 Η δομή του εγγράφου XML

Σε αυτή την ενότητα θα περιγράψουμε τη δομή των XML εγγράφων με τη βοήθεια του παραπάνω παραδείγματος. Πιο συγκεκριμένα, στο παρακάτω σχήμα βλέπουμε ένα XML έγγραφο που περιλαμβάνει καταχωρήσεις βιβλίων μιας βιβλιοθήκης. Στο παράδειγμα αυτό έχουν σημειωθεί τα παρακάτω δομικά στοιχεία των αρχείων XML:



Σχήμα 1.2 XML έγγραφο καταχωρήσεων βιβλιοθήκης

Ο πρόλογος

Ο πρόλογος του παραδείγματός μας αποτελείται από δύο γραμμές:

```
<?xml version="1.0" encoding="Windows-1253" standalone="yes" ?>
<!-- Παράδειγμα 1.1 -->
```

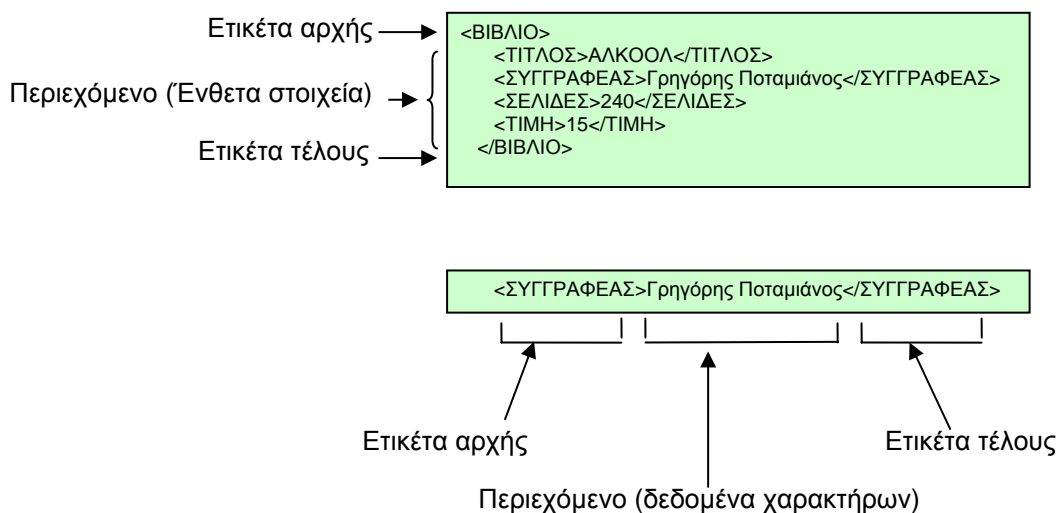
Η πρώτη γραμμή είναι η *δήλωση XML*, η οποία αναφέρει ότι το έγγραφο αυτό είναι ένα έγγραφο XML και δίνει και τον αριθμό της έκδοσης. Επίσης, με την παράμετρο *encoding* προσδιορίζει την κωδικοποίηση χαρακτήρων του εγγράφου, στην περίπτωση που αυτή είναι διαφορετική από την προεπιλεγμένη UTF-8. Τέλος, η παράμετρος *standalone* καθορίζει την ανάγκη αναδρομής σε εξωτερικά αρχεία προκειμένου να ανακτηθούν από αυτά πληροφορίες που επηρεάζουν τη μορφή του περιεχομένου του αρχείου XML, όπως θα δούμε σε επόμενη ενότητα. Πρέπει να σημειωθεί, ότι η γραμμή δήλωσης XML είναι υποχρεωτική και πρέπει να αποτελεί πάντα την πρώτη γραμμή του κώδικα.

Η δεύτερη γραμμή του προλόγου είναι ένα σχόλιο. Η προσθήκη σχολίων σε ένα έγγραφο XML είναι προαιρετική, αλλά διευκολύνει την κατανόηση του εγγράφου. Το σχόλιο ξεκινάει με τους χαρακτήρες <!-- και τελειώνει με τους χαρακτήρες -->.

Το στοιχείο εγγράφου

Το δεύτερο βασικό μέρος ενός εγγράφου XML είναι το στοιχείο εγγράφου (document element) ή βασικό στοιχείο (Root element), το οποίο περιέχει επιπλέον στοιχεία. Τα στοιχεία δείχνουν τη λογική δομή ενός εγγράφου XML και περιέχουν τις πληροφορίες του εγγράφου (στο παράδειγμά μας είναι οι πληροφορίες σχετικά με το βιβλίο, δηλαδή οι τίτλοι, τα ονόματα των συγγραφέων, οι αριθμοί σελίδων και οι τιμές). Ένα τυπικό στοιχείο αποτελείται από μια ετικέτα αρχής, το περιεχόμενο, και μια ετικέτα τέλους. Το περιεχόμενο μπορεί να αποτελείται από δεδομένα χαρακτήρων, άλλα (ένθετα) στοιχεία, ή από συνδυασμό και των δύο.

Στο παράδειγμά μας, το στοιχείο εγγράφου είναι το BIBΛΙΟΘΗΚΗ. Η ετικέτα αρχής του στοιχείου αυτού είναι <BIBΛΙΟΘΗΚΗ>, η ετικέτα τέλους </BIBΛΙΟΘΗΚΗ>, και το περιεχόμενό του είναι τα τρία ένθετα στοιχεία BIBΛΙΟ. Κάθε στοιχείο του BIBΛΙΟ περιλαμβάνει με τη σειρά του έναν αριθμό ένθετων στοιχείων όπως τα <ΤΙΤΛΟΣ>, <ΣΥΓΓΡΑΦΕΑΣ>, <ΣΕΛΙΔΕΣ> και <ΤΙΜΗ>. Καθένα από αυτά τα ένθετα στοιχεία περιέχει μόνο δεδομένα χαρακτήρων.



Σχήμα 1.3 Η δομή του στοιχείου εγγράφου για το Παράδειγμα 1.1

1.3.2 Σωστά διατυπωμένα XML έγγραφα

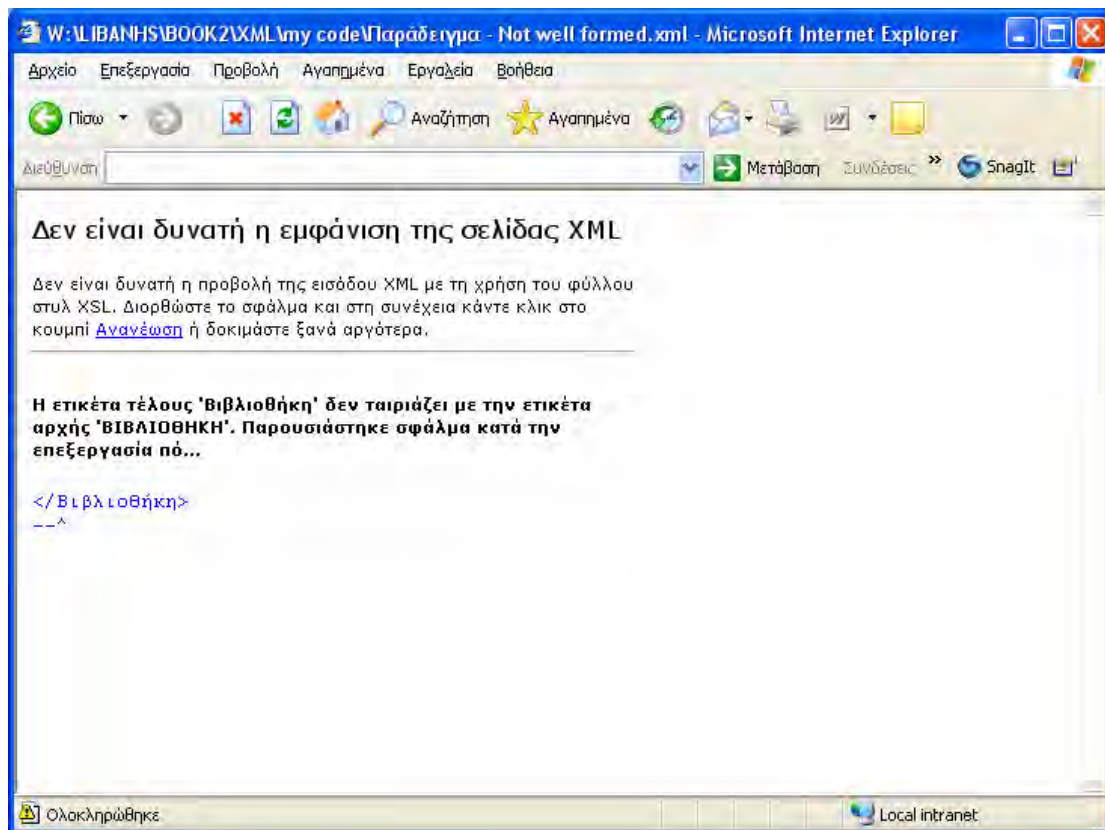
Η γλώσσα XML χαρακτηρίζεται από αυστηρούς κανόνες σύνταξης. Η τήρηση αυτών των κανόνων οδηγεί στη δημιουργία *σωστά διατυπωμένων εγγράφων XML (well-formed XML documents)*. Μερικοί από τους πιο σημαντικούς κανόνες δημιουργίας ενός σωστά διατυπωμένου εγγράφου XML είναι οι παρακάτω:

- Το έγγραφο πρέπει να περιέχει ένα μόνο στοιχείο εγγράφου (βασικό στοιχείο). Όλα τα υπόλοιπα στοιχεία πρέπει να είναι ένθετα σε αυτό.
- Η ένθεση των στοιχείων πρέπει να είναι σωστή. Αυτό σημαίνει ότι, αν ένα στοιχείο ξεκινάει μέσα σε κάποιο άλλο, πρέπει και να τελειώνει μέσα σε αυτό.
- Κάθε στοιχείο πρέπει να έχει ετικέτα αρχής και ετικέτα τέλους. Σε αντίθεση με την HTML, η XML δεν επιτρέπει την παράληψη της ετικέτας τέλους — ούτε καν στις περιπτώσεις εκείνες όπου ο φυλλομετρητής θα μπορούσε εύκολα να καταλάβει σε ποιο σημείο τελειώνει το στοιχείο.
- Το όνομα του στοιχείου στην ετικέτα αρχής πρέπει να είναι ίδιο ακριβώς με εκείνο της ετικέτας τέλους.

Η σωστή διατύπωση είναι απαραίτητη, προκειμένου το XML έγγραφο να μπορεί να εμφανίζεται και να υφίσταται επεξεργασία από ένα φυλλομετρητή ή από κάποιο άλλο πρόγραμμα. Η σημασία της σωστής διατύπωσης γίνεται φανερή με ένα παράδειγμα. Στο σωστά διατυπωμένο αρχείο της ενότητας 1.3.1, αλλάζουμε την τελευταία ετικέτα (BIBΛΙΟΘΗΚΗ) από κεφαλαία σε πεζά.

Παράδειγμα 1.2
<pre> <?xml version="1.0" encoding="Windows-1253" standalone="yes" ?> <!-- Παράδειγμα 1.2 --> <BIBΛΙΟΘΗΚΗ> <BIBΛΙΟ> <ΤΙΤΛΟΣ>ΑΛΚΟΟΛ</ΤΙΤΛΟΣ> <ΣΥΓΓΡΑΦΕΑΣ>Γρηγόρης Ποταμιάνος</ΣΥΓΓΡΑΦΕΑΣ> <ΣΕΛΙΔΕΣ>240</ΣΕΛΙΔΕΣ> <ΤΙΜΗ>15</ΤΙΜΗ> </BIBΛΙΟ> <BIBΛΙΟ> <ΤΙΤΛΟΣ>ΚΑΛΗΜΕΡΑ</ΤΙΤΛΟΣ> <ΣΥΓΓΡΑΦΕΑΣ>Γιώργος Παπαζογλου</ΣΥΓΓΡΑΦΕΑΣ> <ΣΕΛΙΔΕΣ>146</ΣΕΛΙΔΕΣ> <ΤΙΜΗ>21</ΤΙΜΗ> </BIBΛΙΟ> <BIBΛΙΟ> <ΤΙΤΛΟΣ>Ο Αγρός</ΤΙΤΛΟΣ> <ΣΥΓΓΡΑΦΕΑΣ>Θανάσης Λειβαδάς</ΣΥΓΓΡΑΦΕΑΣ> <ΣΕΛΙΔΕΣ>98</ΣΕΛΙΔΕΣ> <ΤΙΜΗ>5,5</ΤΙΜΗ> </BIBΛΙΟ> </Bιβλιοθήκη> </pre>

Εάν αποθηκεύσουμε το παραπάνω αρχείο δίνοντάς του κατάληξη .xml και επιχειρήσουμε να το ανοίξουμε με τη βοήθεια του Internet Explorer, λαμβάνουμε το ακόλουθο αποτέλεσμα:



Σχήμα 1.4 Αποτυχημένη προσπάθεια εμφάνισης του XML εγγράφου του Παραδείγματος 1.2

Αν και στη συνέχεια θα ασχοληθούμε εκτενώς με την προβολή και επεξεργασία XML εγγράφων, το παραπάνω παράδειγμα είναι χαρακτηριστικό της σημασίας της σωστής διατύπωσης στα XML έγγραφα.

1.3.3 Έγκυρα XML έγγραφα

Εκτός από τη σωστή διατύπωση ενός XML εγγράφου, είναι αναγκαίο σε πολλές περιπτώσεις να εξασφαλίσουμε ότι τα XML έγγραφα ακολουθούν μια συγκεκριμένη δομή ή σύνολο κανόνων. Για παράδειγμα, μια εκδοτική εταιρεία εγγράφων για τον Ιστό, θα μπορούσε να απαιτεί όλοι οι συγγραφείς της να δημιουργούν έγγραφα XML με κοινή δομή, εξασφαλίζοντας με τον τρόπο αυτόν ομοιομορφία σε ένα σύνολο παρόμοιων εγγράφων.

Η εξασφάλιση της ύπαρξης της απαιτούμενης δομής στα XML έγγραφα γίνεται με τη βοήθεια μιας δήλωσης τύπου εγγράφου (Document Type Definition – DTD). Το DTD μπορεί είτε να περιλαμβάνεται μέσα στο έγγραφο XML είτε να είναι διαθέσιμο εξωτερικά ως αυτοτελές αρχείο, στο οποίο το XML συνδέεται με κατάλληλη παραπομπή.

Ένα XML έγγραφο συνδεδεμένο με ένα DTD, το οποίο ακολουθεί τη δομή που υπαγορεύεται από το DTD, ονομάζεται *έγκυρο έγγραφο XML* (*valid XML document*)

Δημιουργία και εισαγωγή του DTD

Η παρουσία ενός *DTD* που θα ορίζει τη δομή αυτή επιτρέπει σε κάποιον επεξεργαστή XML να ελέγξει κατά πόσο το έγγραφο ακολουθεί την προβλεπόμενη δομή. Με άλλα λόγια, το *DTD* παρέχει στον επεξεργαστή ένα πρότυπο προσχέδιο ώστε, ελέγχοντας την εγκυρότητα του εγγράφου, να μπορεί να επιβάλει την επιθυμητή δομή και να εγγυηθεί ότι το έγγραφο πληρεί τις απαραίτητες προϋποθέσεις. Αν κάποιος από τα μέρη του εγγράφου δεν τηρεί την προδιαγραφή του *DTD*, ο επεξεργαστής μπορεί να εμφανίσει στην οθόνη ένα μήνυμα σφάλματος ώστε να διορθωθεί το έγγραφο σύμφωνα με τους κανόνες. Ας δούμε τώρα ένα απλό DTD που θα μπορούσε να περιγράψει τη δομή του XML εγγράφου της βιβλιοθήκης.

```
<!DOCTYPE ΒΙΒΛΙΟΘΗΚΗ
[
  <!ELEMENT ΒΙΒΛΙΟ (ΤΙΤΛΟΣ, ΣΥΓΓΡΑΦΕΑΣ, ΣΕΛΙΔΕΣ, ΤΙΜΗ)>
  <!ELEMENT ΤΙΤΛΟΣ (#PCDATA)>
  <!ELEMENT ΣΥΓΓΡΑΦΕΑΣ (#PCDATA)>
  <!ELEMENT ΣΕΛΙΔΕΣ (#PCDATA)>
  <!ELEMENT ΤΙΜΗ (#PCDATA)>
]
>
```

Σχήμα 1.5 Ένα DTD για το XML έγγραφο της βιβλιοθήκης

Η δήλωση DOCTYPE είναι η αρχική δήλωση του DTD και ακολουθείται από το βασικό στοιχείο του εγγράφου XML. Στη συνέχεια, ακολουθούν οι δηλώσεις στοιχείων με τη δήλωση ELEMENT στην οποία περιλαμβάνονται τα ένθετα στοιχεία του XML εγγράφου. Το όρισμα #PCDATA σημαίνει ότι το στοιχείο που δηλώνεται περιέχει δεδομένα.

Η ενσωμάτωση του παραπάνω DTD στο έγγραφο XML γίνεται αμέσως μετά τον πρόλογο, ως εξής:

DTD

```
<?xml version="1.0" encoding="Windows-1253" standalone="yes" ?>
<!-- Παράδειγμα 1.3 - Valid XML Document-->
<!DOCTYPE BIBΛΙΟΘΗΚΗ
 [
  <!ELEMENT BIBΛΙΟ (ΤΙΤΛΟΣ, ΣΥΓΓΡΑΦΕΑΣ, ΣΕΛΙΔΕΣ,
ΤΙΜΗ)>
  <!ELEMENT ΤΙΤΛΟΣ (#PCDATA)>
  <!ELEMENT ΣΥΓΓΡΑΦΕΑΣ (#PCDATA)>
  <!ELEMENT ΣΕΛΙΔΕΣ (#PCDATA)>
  <!ELEMENT ΤΙΜΗ (#PCDATA)>
 ]
>
<BIBΛΙΟΘΗΚΗ>
  <BIBΛΙΟ>
    <ΤΙΤΛΟΣ>ΑΛΚΟΟΛ</ΤΙΤΛΟΣ>
    <ΣΥΓΓΡΑΦΕΑΣ>Γρηγόρης Ποταμιάνος</ΣΥΓΓΡΑΦΕΑΣ>
    <ΣΕΛΙΔΕΣ>240</ΣΕΛΙΔΕΣ>
    <ΤΙΜΗ>15</ΤΙΜΗ>
  </BIBΛΙΟ>
  <BIBΛΙΟ>
    <ΤΙΤΛΟΣ>ΚΑΛΗΜΕΡΑ</ΤΙΤΛΟΣ>
    <ΣΥΓΓΡΑΦΕΑΣ>Γιώργος Παπάζογλου</ΣΥΓΓΡΑΦΕΑΣ>
    <ΣΕΛΙΔΕΣ>146</ΣΕΛΙΔΕΣ>
    <ΤΙΜΗ>21</ΤΙΜΗ>
  </BIBΛΙΟ>
  <BIBΛΙΟ>
    <ΤΙΤΛΟΣ>Ο Αγρός</ΤΙΤΛΟΣ>
    <ΣΥΓΓΡΑΦΕΑΣ>Θανάσης Λειβαδάς</ΣΥΓΓΡΑΦΕΑΣ>
    <ΣΕΛΙΔΕΣ>98</ΣΕΛΙΔΕΣ>
    <ΤΙΜΗ>5,5</ΤΙΜΗ>
  </BIBΛΙΟ>
</BIBΛΙΟΘΗΚΗ>
```

Χρήση εξωτερικών DTDs

Όπως έχουμε ήδη αναφέρει, τα DTDs μπορεί να βρίσκονται σε εξωτερικό αρχείο και να καλούνται από το XML έγγραφο με παραπομπή. Αναφερόμενοι στο παραπάνω παράδειγμα, δημιουργούμε ένα ξεχωριστό έγγραφο με τα εξής περιεχόμενα:

```
<?xml version="1.0" encoding="Windows-1253" ?>
<!ELEMENT BIBΛΙΟ (ΤΙΤΛΟΣ, ΣΥΓΓΡΑΦΕΑΣ, ΣΕΛΙΔΕΣ, ΤΙΜΗ)>
<!ELEMENT ΤΙΤΛΟΣ (#PCDATA)>
<!ELEMENT ΣΥΓΓΡΑΦΕΑΣ (#PCDATA)>
<!ELEMENT ΣΕΛΙΔΕΣ (#PCDATA)>
<!ELEMENT ΤΙΜΗ (#PCDATA)>
```

Αν αποθηκεύσουμε το παραπάνω κείμενο με το όνομα book.dtd, τότε έχουμε δημιουργήσει ένα DTD το οποίο μπορεί να κληθεί από το XML έγγραφο με τη βοήθεια της δήλωσης <!DOCTYPE BIBΛΙΟΘΗΚΗ SYSTEM "book.dtd">.

```
<?xml version="1.0" encoding="Windows-1253" standalone="yes" ?>

<!-- Παράδειγμα 1 - Valid XML Document-->

<!DOCTYPE BIBΛΙΟΘΗΚΗ SYSTEM "book.dtd">

<BIBΛΙΟΘΗΚΗ>
  <BIBΛΙΟ>
    <ΤΙΤΛΟΣ>ΑΛΚΟΟΛ</ΤΙΤΛΟΣ>
    <ΣΥΓΓΡΑΦΕΑΣ>Γρηγόρης Ποταμιάνος</ΣΥΓΓΡΑΦΕΑΣ>
    <ΣΕΛΙΔΕΣ>240</ΣΕΛΙΔΕΣ>
    <ΤΙΜΗ>15</ΤΙΜΗ>
  </BIBΛΙΟ>
  <BIBΛΙΟ>
    <ΤΙΤΛΟΣ>ΚΑΛΗΜΕΡΑ</ΤΙΤΛΟΣ>
    <ΣΥΓΓΡΑΦΕΑΣ>Γιώργος Παπάζογλου</ΣΥΓΓΡΑΦΕΑΣ>
    <ΣΕΛΙΔΕΣ>146</ΣΕΛΙΔΕΣ>
    <ΤΙΜΗ>21</ΤΙΜΗ>
  </BIBΛΙΟ>
  <BIBΛΙΟ>
    <ΤΙΤΛΟΣ>Ο Αγρός</ΤΙΤΛΟΣ>
    <ΣΥΓΓΡΑΦΕΑΣ>Θανάσης Λειβαδάς</ΣΥΓΓΡΑΦΕΑΣ>
    <ΣΕΛΙΔΕΣ>98</ΣΕΛΙΔΕΣ>
    <ΤΙΜΗ>5,5</ΤΙΜΗ>
  </BIBΛΙΟ>
</BIBΛΙΟΘΗΚΗ>
```

Η χρήση εξωτερικών DTDs παρέχει σημαντικές δυνατότητες, επιτρέποντας τη χρήση ενός κοινού DTD από ένα σύνολο εγγράφων, κάτι που διευκολύνει σημαντικά τη συντήρησή τους.

Ενότητα 1.4 Επεξεργασία XML εγγράφων

Στην ενότητα αυτή θα δούμε πώς μπορούμε να δημιουργήσουμε και να εμφανίσουμε ένα έγγραφο XML, χρησιμοποιώντας το παράδειγμα της παραπάνω ενότητας.

Δημιουργία εγγράφου XML

Για τη δημιουργία ενός εγγράφου XML μπορούμε να χρησιμοποιήσουμε έναν απλό κειμενογράφο όπως το Notepad. Ανοίξτε ένα νέο, κενό αρχείο κειμένου στο Notepad και πληκτρολογήστε το παρακάτω κείμενο:

```
<?xml version="1.0" encoding="Windows-1253" standalone="yes" ?>

<!-- Παράδειγμα 1 -->

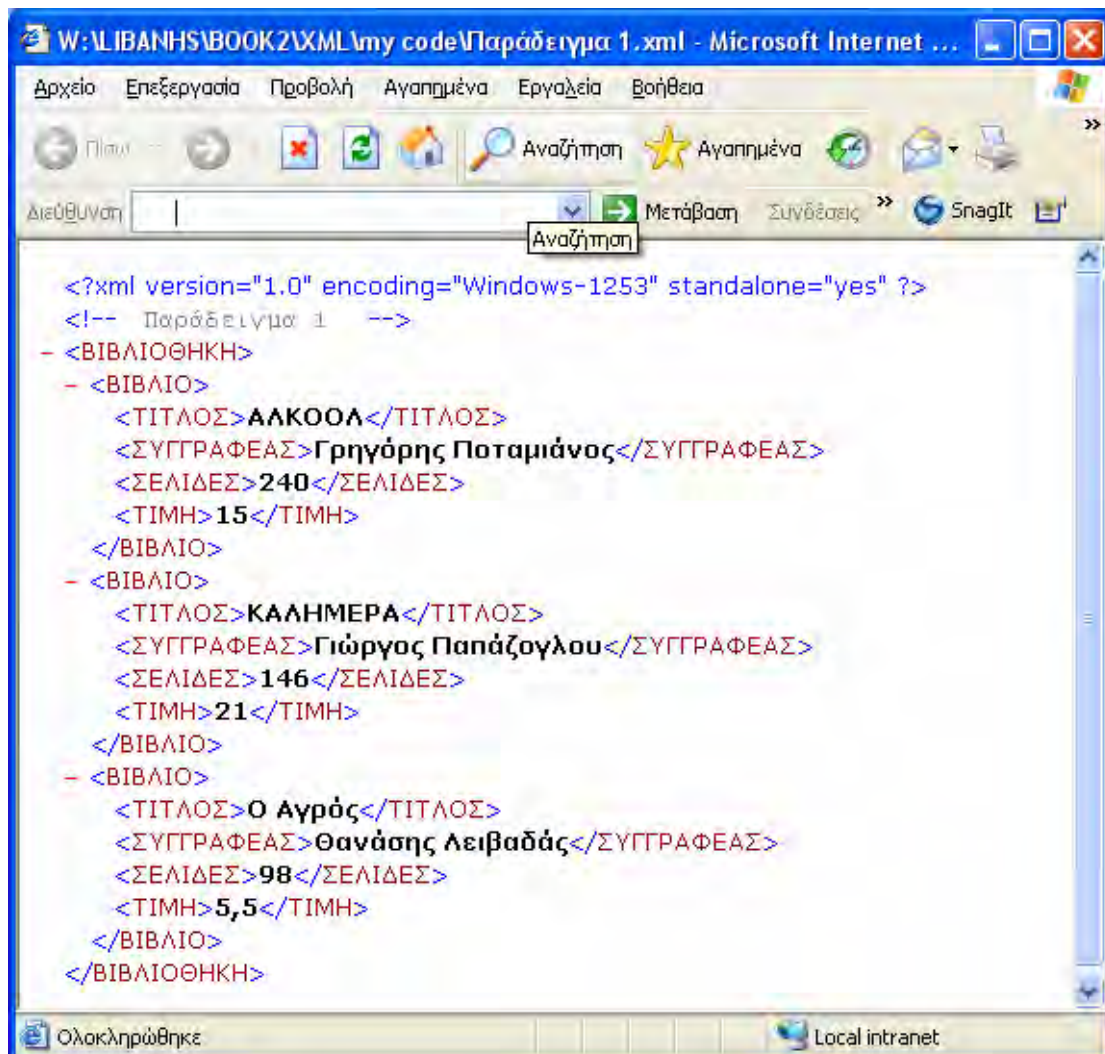
<LIBRARY>
  <BOOK>
    <TITLE>ΑΛΚΟΟΛ</TITLE>
    <AUTHOR>Γρηγόρης Ποταμιάνος</AUTHOR>
    <PAGES>240</PAGES>
    <PRICE>15</PRICE>
  </BOOK>
  <BOOK>
    <TITLE>ΚΑΛΗΜΕΡΑ</TITLE>
    <AUTHOR>Γιώργος Παπάζογλου</AUTHOR>
    <PAGES>146</PAGES>
    <PRICE>21</PRICE>
  </BOOK>
  <BOOK>
    <TITLE>Ο Αγρός</TITLE>
    <AUTHOR>Θανάσης Λειβαδάς</AUTHOR>
    <PAGES>98</PAGES>
    <PRICE>5,5</PRICE>
  </BOOK>
</LIBRARY>
```

Στη συνέχεια αποθηκεύστε το έγγραφο στο σκληρό σας δίσκο με το όνομα Παράδειγμα1.xml.

Εμφάνιση εγγράφου XML

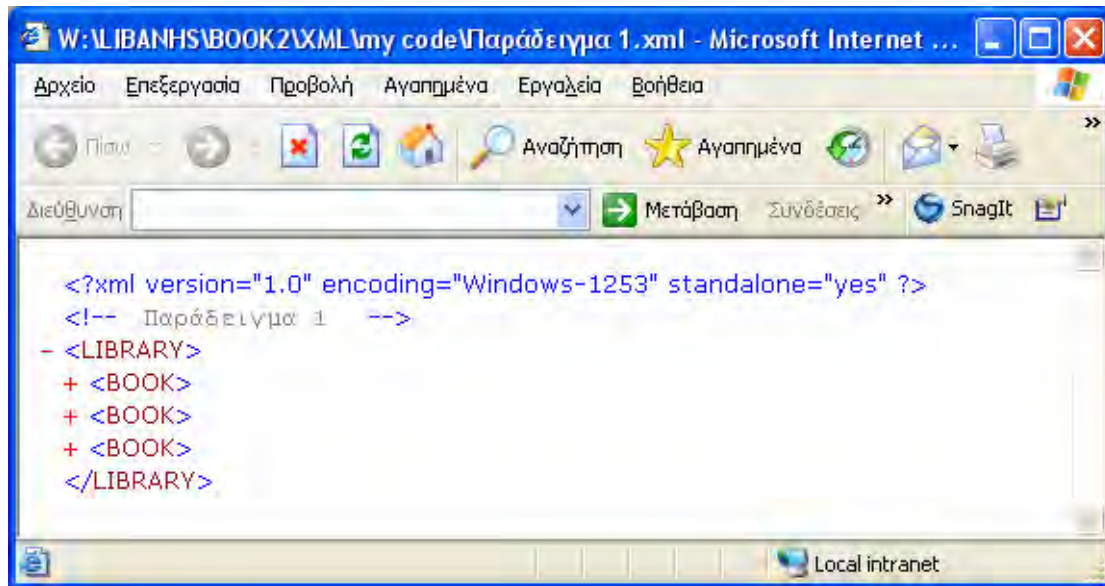
Μπορούμε να ανοίξουμε ένα έγγραφο XML κατευθείαν στο φυλλομετρητή Internet Explorer, όπως ακριβώς θα ανοίγαμε και μια ιστοσελίδα HTML. Αν το έγγραφο XML δεν είναι συνδεδεμένο με κάποιο φύλλο στυλ (Βλέπε ενότητα 1.5), ο Internet Explorer θα εμφανίσει απλώς το πλήρες έγγραφο, μαζί με τη σήμανση (για παράδειγμα, ετικέτες και σχόλια) καθώς και τα

στοιχεία χαρακτήρων. Ο Internet Explorer κωδικοποιεί με χρώματα τα διάφορα μέρη του εγγράφου για να κάνει πιο εύκολη την αναγνώρισή τους και εμφανίζει το στοιχείο εγγράφου ως συμπτύξιμο/επεκτάσιμο δέντρο, ώστε να δείξει με σαφήνεια τη λογική δομή του εγγράφου με διάφορα επίπεδα λεπτομερειών. Στο ακόλουθο σχήμα βλέπουμε τι θα εμφανίσει ο Internet Explorer με το άνοιγμα του αρχείου Παράδειγμα1.xml που δημιουργήσαμε παραπάνω.

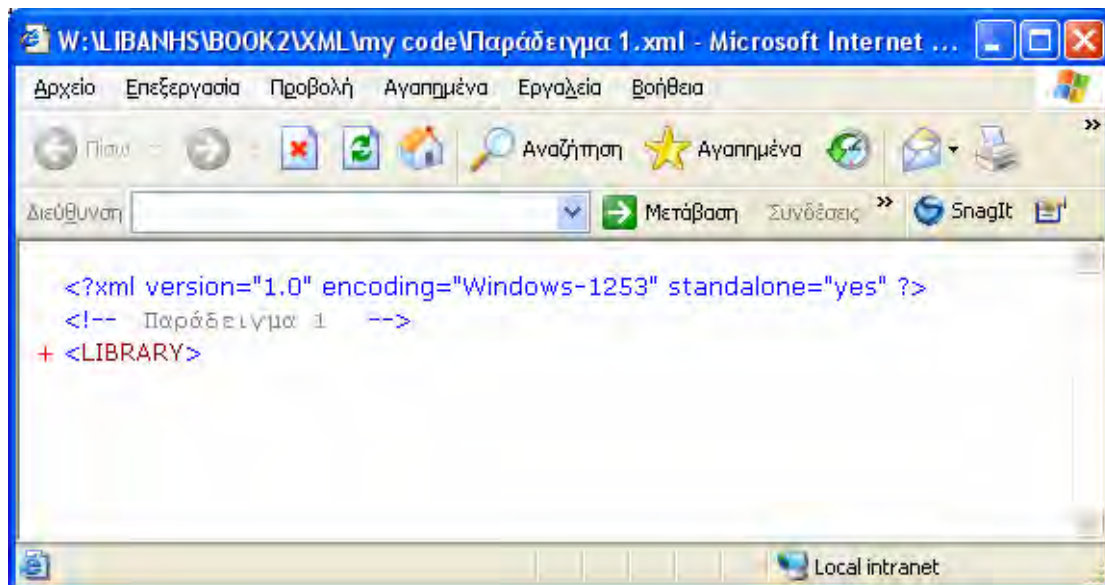


Σχήμα 1.6 Άνοιγμα XML αρχείου.

Επίσης, στα παρακάτω σχήματα βλέπουμε δύο διαφορετικά επίπεδα σύμπτυξης του XML εγγράφου που δημιουργήσαμε.



Σχήμα 1.7 Σύμπτυξη κατά ένα επίπεδο.



Σχήμα 1.5 Σύμπτυξη κατά δύο επίπεδα.

Εμφάνιση εγγράφου XML με χρήση CSS

Αν το έγγραφο **XML** είναι συνδεδεμένο με κάποιο φύλλο στυλ, ο **Internet Explorer** θα εμφανίσει μόνο τα δεδομένα χαρακτήρων των στοιχείων του εγγράφου και θα τα μορφοποιήσει ακολουθώντας τους κανόνες που παρέχονται στο φύλλο στυλ. Αυτό θα γίνει πιο κατανοητό με ένα παράδειγμα.

Ανοίξτε ένα νέο αρχείο κειμένου στο Notepad και πληκτρολογήστε το παρακάτω κείμενο:

```
/* File Name: Παράδειγμα2.css */
```

```
BOOK
```

```
{display:block;  
margin-top:12pt;  
font-size:10pt}
```

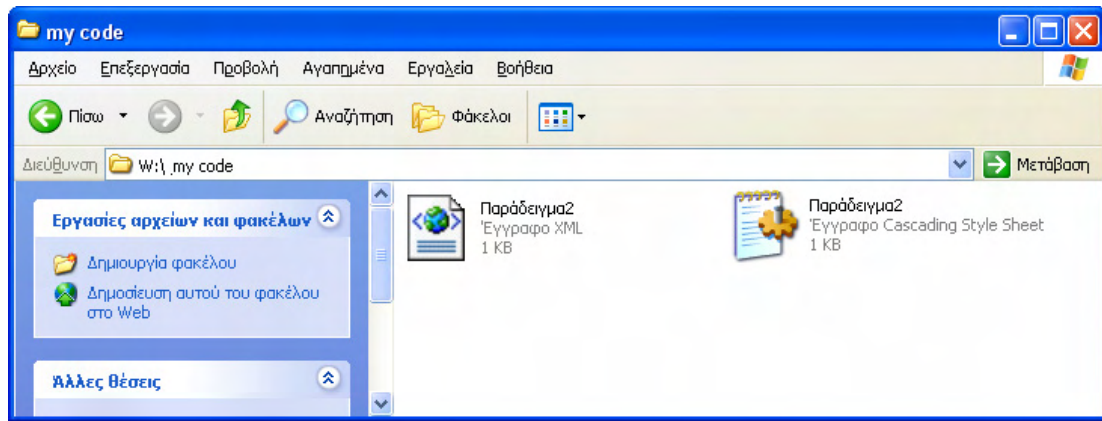
```
TITLE
```

```
{font-style:italic}
```

Στη συνέχεια αποθηκεύστε το έγγραφο στο σκληρό σας δίσκο με το όνομα Παράδειγμα2.css. Επειτα, τροποποιήστε το έγγραφο Παράδειγμα1.xml που είχατε δημιουργήσει, προσθέτοντας αμέσως πάνω από την ετικέτα <LIBRARY> τη γραμμή <?xml-stylesheet type="text/css" href="Παράδειγμα2.css"?>, όπως φαίνεται στο παρακάτω σχήμα:

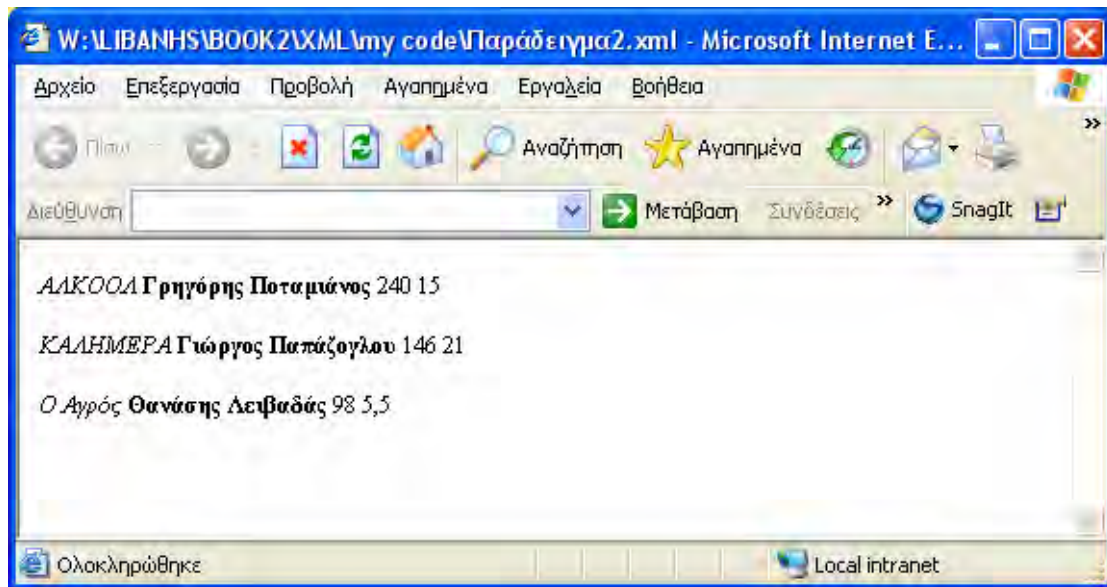
```
<?xml version="1.0" encoding="Windows-1253" standalone="yes" ?>  
  
<!-- Παράδειγμα 2 -->  
  
<?xml-stylesheet type="text/css" href="Παράδειγμα2.css"?>  
  
<LIBRARY>  
  <BOOK>  
    <TITLE>ΑΛΚΟΟΛ</TITLE>  
    <AUTHOR>Γρηγόρης Ποταμιάνος</AUTHOR>  
    <PAGES>240</PAGES>  
    <PRICE>15</PRICE>  
  </BOOK>  
  .....
```

Η γραμμή αυτή ουσιαστικά συνδέει το XML έγγραφο με το φύλλο στυλ Παράδειγμα2.css που δημιουργήσαμε. Στη συνέχεια, σώζουμε το XML έγγραφο στο σκληρό δίσκο με το όνομα Παράδειγμα2.xml.



Σχήμα 1.9 Εμφάνιση αρχείων.

Ανοίγοντας τώρα το αρχείο Παράδειγμα2.xml με τον Internet Explorer, παρατηρούμε, όπως φαίνεται στο σχήμα 1.10, ότι εμφανίζονται μόνο τα δεδομένα χαρακτήρων, ακολουθώντας τις πληροφορίες μορφής που παρέχονται στο αρχείο Παράδειγμα2.css.



Σχήμα 1.10 Εμφάνιση αρχείου XML με CSS.

Ενότητα 1.5 Δημιουργία XSLT φύλλων στυλ

Εισαγωγή

Οι όλο και αυξανόμενες απαιτήσεις μορφοποίησης και παρουσίασης των περιεχομένων των ιστοσελίδων, έκαναν επιτακτική την ανάγκη διαχωρισμού του περιεχομένου του εγγράφου και της διάταξης παρουσίασής του. Η πιο δημοφιλής μέθοδος για την αντιμετώπιση αυτής της ανάγκης στα HTML έγγραφα είναι η χρήση διαδοχικών φύλλων στυλ (Cascading Style Sheet – CSS). Τα CSS δίνουν τη δυνατότητα αλλαγής της εμφάνισης και της διάταξης όλων των σελίδων σε ένα δικτυακό τόπο (Web Site), με την απλή επεξεργασία ενός μόνου CSS εγγράφου. Εκτός από τα έγγραφα HTML, τα CSS μπορούν να χρησιμοποιηθούν και για τη μορφοποίηση και παρουσίαση εγγράφων XML όπως είδαμε στην προηγούμενη ενότητα. Ειδικότερα για τα έγγραφα XML, είναι διαθέσιμος και ένας πιο ισχυρός τύπος διαδοχικών φύλλων στυλ, η *επεκτάσιμη γλώσσα φύλλων στυλ* (Extensible Stylesheet Language – XSL). Θα δούμε στη συνέχεια παραδείγματα δημιουργίας χρήσης των XSL φύλλων στυλ, ενώ θα εμβαθύνουμε στη χρήση τους για τη μετατροπή εγγράφων XML σε διάφορες μορφές.

Τι είναι τα φύλλα στυλ XSL

Όπως συμβαίνει με τα φύλλα στυλ CSS, έτσι και ένα φύλλο στυλ XSL συνδέεται με ένα έγγραφο XML και δίνει εντολές στο φυλλομετρητή πώς να εμφανίσει τα δεδομένα XML. Για την εμφάνιση της XML όμως, η χρήση ενός φύλλου στυλ XSL αποτελεί μέθοδο πολύ πιο ισχυρή και ευέλικτη από τη χρήση CSS. Ενώ το φύλλο στυλ CSS επιτρέπει απλώς τη μορφοποίηση κάθε στοιχείου XML, το φύλλο στυλ XSL δίνει τη δυνατότητα πλήρους ελέγχου του αποτελέσματος. Συγκεκριμένα, η XSL επιτρέπει την επιλογή των δεδομένων της XML που θέλουμε να εμφανισθούν, την παρουσίασή τους με οποιαδήποτε σειρά ή διάταξη, την τροποποίησή τους και την προσθήκη πληροφοριών.

Χρήση φύλλων στυλ XSL – Βασικές αρχές

Το φύλλο στυλ XSL είναι ένα σωστά διατυπωμένο έγγραφο XML, το οποίο αντί να περιλαμβάνει κανόνες, όπως το CSS, περιλαμβάνει ένα ή περισσότερα *πρότυπα* (*templates*), καθένα από τα οποία περιέχει τις πληροφορίες για την εμφάνιση ενός συγκεκριμένου κλάδου των δεδομένων του εγγράφου XML. Όπως συμβαίνει και με τα έγγραφα XML, έτσι και τα φύλλα στυλ XSL αποτελούνται από απλό κείμενο, το οποίο μπορεί να δημιουργηθεί σε έναν απλό κειμενογράφο. Ένα φύλλο στυλ σχετικό με το παράδειγμα της βιβλιοθήκης, θα μπορούσε να είναι το ακόλουθο:

```

<?xml version="1.0" encoding="Windows-1253" ?>
<!-- File Name: Παράδειγμα1.xsl -->
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
  <xsl:template match="/">
    <H2>ΒΙΒΛΙΟΘΗΚΗ</H2>
    <SPAN STYLE="font-style:italic">Author: </SPAN>
    <xsl:value-of select="ΒΙΒΛΙΟΘΗΚΗ/ΒΙΒΛΙΟ/ΣΥΓΓΡΑΦΕΑΣ"/><BR/>
    <SPAN STYLE="font-style:italic">Title: </SPAN>
    <xsl:value-of select="ΒΙΒΛΙΟΘΗΚΗ/ΒΙΒΛΙΟ/ΤΙΤΛΟΣ"/><BR/>
    <SPAN STYLE="font-style:italic">Price: </SPAN>
    <xsl:value-of select="ΒΙΒΛΙΟΘΗΚΗ/ΒΙΒΛΙΟ/ΤΙΜΗ"/><BR/>
    <SPAN STYLE="font-style:italic">Σελίδες: </SPAN>
    <xsl:value-of select="ΒΙΒΛΙΟΘΗΚΗ/ΒΙΒΛΙΟ/ΣΕΛΙΔΕΣ"/>
  </xsl:template>
</xsl:stylesheet>

```

Στο παραπάνω παράδειγμα, αρχικά παρατηρούμε την εντολή ορισμού του φύλλου στυλ, η οποία τοποθετείται αμέσως μετά τον πρόλογο και είναι η εξής:

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
```

B) Σύνδεση του αρχείου XSL με το έγγραφο XML. Το έγγραφο XML και το φύλλο στυλ XSL φυλάγονται σε διαφορετικά αρχεία, ένα με προέκταση .xml) και ένα με προέκταση .xsl αντίστοιχα. Μπορούμε να συνδέσουμε το φύλλο στυλ XSL με το έγγραφο XML, αν συμπεριλάβουμε την εντολή επεξεργασίας *xsl-stylesheet*, η οποία έχει τη μορφή:

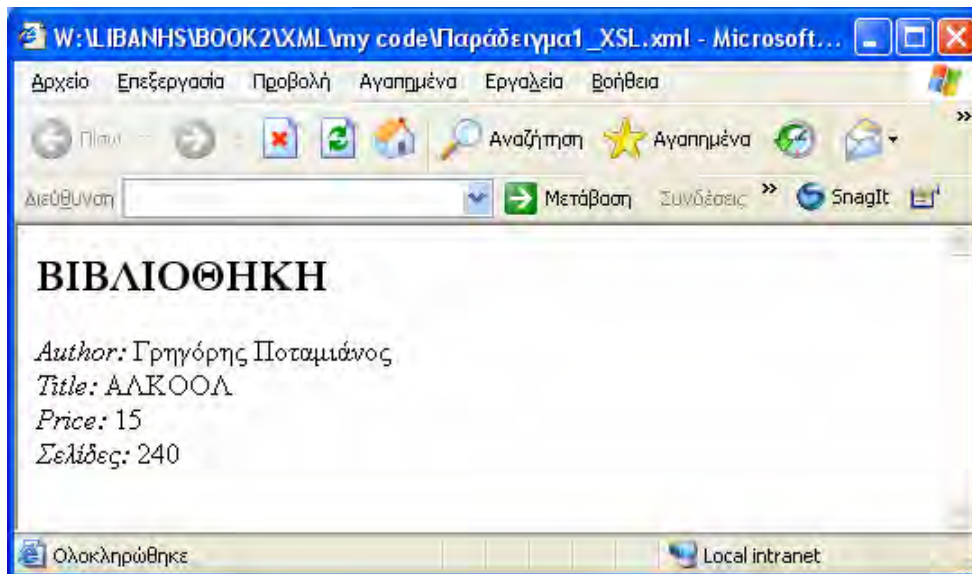
```
<?xml-stylesheet type="text/xsl" href=FilePath?>
```

Στην παραπάνω εντολή, το *FilePath* είναι η πλήρης διαδρομή του .xsl αρχείου, η οποία μπορεί να είναι είτε σε μορφή URL, είτε σε σχετική μορφή, η οποία είναι και η πιο συνηθισμένη, εφόσον στις περισσότερες περιπτώσεις το φύλλο στυλ αποθηκεύεται στον ίδιο φάκελο με το αρχείο XML.

```

<?xml version="1.0" encoding="Windows-1253" standalone="no" ?>
<!-- Παράδειγμα 1_XSL -->
<?xml-stylesheet type="text/xsl" href="Παράδειγμα1.xsl"?>
<BIBΛΙΟΘΗΚΗ>
  <BIBΛΙΟ>
    <ΤΙΤΛΟΣ>ΑΛΚΟΟΛ</ΤΙΤΛΟΣ>
    <ΣΥΓΓΡΑΦΕΑΣ>Γρηγόρης Ποταμιάνος</ΣΥΓΓΡΑΦΕΑΣ>
    <ΣΕΛΙΔΕΣ>240</ΣΕΛΙΔΕΣ>
    <ΤΙΜΗ>15</ΤΙΜΗ>
  </BIBΛΙΟ>
  <BIBΛΙΟ>
    <ΤΙΤΛΟΣ>ΚΑΛΗΜΕΡΑ</ΤΙΤΛΟΣ>
    <ΣΥΓΓΡΑΦΕΑΣ>Γιώργος Παπάζογλου</ΣΥΓΓΡΑΦΕΑΣ>
    <ΣΕΛΙΔΕΣ>146</ΣΕΛΙΔΕΣ>
    <ΤΙΜΗ>21</ΤΙΜΗ>
  </BIBΛΙΟ>
  <BIBΛΙΟ>
    <ΤΙΤΛΟΣ>Ο Αγρός</ΤΙΤΛΟΣ>
    <ΣΥΓΓΡΑΦΕΑΣ>Θανάσης Λειβαδάς</ΣΥΓΓΡΑΦΕΑΣ>
    <ΣΕΛΙΔΕΣ>98</ΣΕΛΙΔΕΣ>
    <ΤΙΜΗ>5,5</ΤΙΜΗ>
  </BIBΛΙΟ>
</BIBΛΙΟΘΗΚΗ>

```



Σχήμα 1.11 Εμφάνιση αρχείου XML.

Ενότητα 1.6 Δραστηριότητες ανακεφαλαίωσης

A) Αναφέρατε τουλάχιστον δύο τύπους κειμένων για τους οποίους η HTML παρέχει περιορισμένες δυνατότητες παρουσίασης.

B) Συμπληρώστε τα κενά με τις κατάλληλες λέξεις:

- Τα φύλλα στυλ XSL επιτρέπουν την των δεδομένων της XML που θέλουμε να εμφανισθούν, την τους με οποιαδήποτε σειρά ή διάταξη, την τροποποίησή τους και την προσθήκη πληροφοριών.
- Η γλώσσα XML χαρακτηρίζεται από κανόνες σύνταξης.
- Η περιγραφή ενός DTD ξεκινά με τη δήλωση <!.....

Γ) Επιλέξτε τη σωστή απάντηση.

<i>Η XML είναι μια γλώσσα</i>	<i>Για να είναι έγκυρο ένα έγγραφο XML πρέπει να συνδέεται με ένα</i>
<i>α) προγραμματισμού</i>	<i>α) DTD</i>
<i>β) σήμανσης</i>	<i>β) XSL</i>
<i>γ) περιγραφική</i>	<i>γ) CSS</i>

Δ) Δημιουργήστε ένα παράδειγμα τηλεφωνικού καταλόγου στη μορφή XML έγγραφου χρησιμοποιώντας το Notepad. Στη συνέχεια, ανοίξτε το αρχείο που δημιουργήσατε με το φυλλομετρητή και επιβεβαιώστε την εμφάνισή του με τη μορφή δέντρου.

Ε) Λαμβάνοντας υπόψη το αρχείο XML που δημιουργήσατε στην προηγούμενη άσκηση, δημιουργήστε ένα κατάλληλο εξωτερικό φύλλο στυλ CSS με τη βοήθεια του Notepad. Στη συνέχεια συνδέστε το αρχείο XML με το φύλλο στυλ που δημιουργήσατε. Ανοίξτε τον τηλεφωνικό κατάλογο με το φυλλομετρητή εξασφαλίζοντας ότι εμφανίζεται στην επιθυμητή μορφή.

ΚΕΦΑΛΑΙΟ 2ο



Προγραμματισμός από την Πλευρά του Πελάτη

Δυναμικές Εφαρμογές Παγκόσμιου Ιστού

Σκοπός και στόχοι του κεφαλαίου

Σκοπός του κεφαλαίου είναι η εισαγωγή των ενηλίκων στον προγραμματισμό δυναμικών εφαρμογών ιστοσελίδων, από την πλευρά του πελάτη.

Μετά την ολοκλήρωση του κεφαλαίου, οι μαθητές θα είναι σε θέση να:

- κατανοούν τη φιλοσοφία του προγραμματισμού από την πλευρά του πελάτη
- εντοπίζουν τις δυνατότητες και τα πλεονεκτήματά του, σε σχέση με τις στατικές εφαρμογές HTML
- κατανοούν τις βασικές αρχές της γλώσσας σεναρίων Javascript
- κατανοούν τις βασικές αρχές της τεχνολογίας DHTML, ως συνδυασμό της Javascript, της HTML και των διαδοχικών φύλλων στυλ CSS
- εφαρμόζουν συνδυαστικά τις παραπάνω γνώσεις για τη δημιουργία απλών εφαρμογών προγραμματισμού από την πλευρά του πελάτη, χρησιμοποιώντας HTML, σεναρία Javascript και διαδοχικά φύλλα στυλ

Εννοιες-Κλειδιά

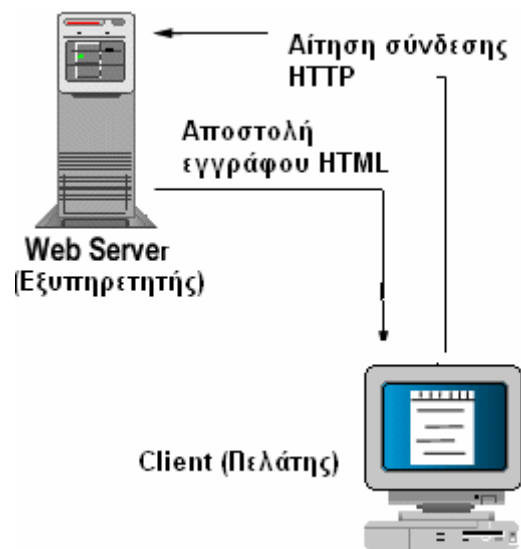
Προγραμματισμός από την πλευρά του πελάτη, Javascript, δομημένος προγραμματισμός, μοντέλο DOM, αντικείμενα, συμβάντα, DHTML.

Εισαγωγικές Παρατηρήσεις

Η ανάγκη παράκαμψης της στατικότητας του περιεχομένου της HTML οδήγησε στην ανάπτυξη της φιλοσοφίας του προγραμματισμού από την πλευρά του πελάτη. Η **JavaScript** είναι μια προγραμματισμού από την πλευρά του πελάτη, η οποία Οι εφαρμογές της Javascript στο δυναμικό προγραμματισμό παγκόσμιου ιστού είναι πάρα πολλές. Η Javascript ακολουθεί τη φιλοσοφία των γλωσσών δομημένου προγραμματισμού, και βασίζεται στην υλοποίηση *αντικειμένων (objects)*, *γεγονότων (events)* και *μεθόδων(methods)*, η χρήση των οποίων συνιστά τον κύριο μηχανισμό αλληλεπίδρασης της Javascript με τις ιστοσελίδες HTML

Ενότητα 2.1 Εισαγωγή στον προγραμματισμό από την πλευρά του πελάτη

Αν και η HTML γνώρισε αμέσως μεγάλη δημοτικότητα, οι προγραμματιστές ιστοσελίδων αντιμετώπισαν αρκετά νωρίς συγκεκριμένους περιορισμούς. Για παράδειγμα, χρησιμοποιώντας μόνο την HTML, δεν υπάρχει κάποιος τρόπος που να επιτρέπει την εμφάνιση της τρέχουσας ημερομηνίας και ώρας σε κάθε σελίδα. Ουσιαστικά η HTML από μόνη της είναι κατάλληλη και ιδιαίτερα επιτυχής για τη διανομή, κατόπιν αιτήσεως, σελίδων που έχουν ήδη προετοιμαστεί όσον αφορά το περιεχόμενο, και οι οποίες θεωρούνται *στατικές* (Σχήμα 2.1).



Σχήμα 2.1 Αίτηση λήψης στατικής σελίδας HTML

Η ανάγκη παράκαμψης της στατικότητας του περιεχομένου της HTML, οδήγησε στην ανάπτυξη της φιλοσοφίας του προγραμματισμού από την πλευρά του πελάτη. Κατά τον προγραμματισμό από την πλευρά του πελάτη, η σελίδα HTML εμπλουτίζεται με κατάλληλο κώδικα γραμμένο σε γλώσσα προγραμματισμού, ο οποίος εκτελείται από τον Η/Υ του πελάτη, επιτρέποντας τόσο το δυναμικό χειρισμό συμβάντων κατά τη διάρκεια της εμφάνισης της σελίδας, όσο και την αύξηση της διαδραστικότητας μεταξύ του χρήστη και της εφαρμογής.

Οι γλώσσες προγραμματισμού που χρησιμοποιούνται στον προγραμματισμό από την πλευρά του πελάτη, ονομάζονται γλώσσες συγγραφής σεναρίων (scripting languages) και δίνουν τη δυνατότητα επεξεργασίας των δεδομένων της ιστοσελίδας, καθώς και τη διεξαγωγή υπολογισμών και μετασχηματισμών σε αυτά τα δεδομένα, με τη βοήθεια εντολών, συναρτήσεων, αντικειμένων και συμβάντων, όπως θα δούμε στη συνέχεια. Σημαντικότερος εκπρόσωπος των γλωσσών συγγραφής σεναρίων, τον οποίο και θα μελετήσουμε σε αυτό το κεφάλαιο, είναι η γλώσσα Javascript. Εκτός από τη Javascript, θα δούμε σε αυτό το κεφάλαιο και τα διαδοχικά φύλλα φυλλομετρίας (cascading style sheets), τα οποία κάνουν δυνατό το διαχωρισμό μεταξύ του περιεχομένου και της μορφοποίησης της ιστοσελίδας. Τέλος, θα δούμε την DHTML (Dynamic HTML) ως προγραμματιστική δυνατότητα από την πλευρά του πελάτη, η οποία προκύπτει από το συνδυασμό της HTML, της Javascript και των διαδοχικών φύλλων φυλλομετρίας.

Ενότητα 2.2 Επισκόπηση της γλώσσας Javascript

2.2.1 Εισαγωγή

Η **JavaScript** είναι μια γλώσσα προγραμματισμού από την πλευρά του πελάτη, η οποία δημιουργήθηκε από την εταιρεία Netscape και χρησιμοποιείται ευρύτατα για να προσθέσει δυναμικά εφέ και διαδραστικότητα στις ιστοσελίδες. Η Javascript υποστηρίζεται από τους πιο γνωστούς φυλλομετρητές όπως τον Netscape Navigator, τον Internet Explorer και το Mozilla Firefox, καθώς και από άλλους λιγότερους γνωστούς. Η Javascript χαρακτηρίζεται και ως *γλώσσα συγγραφής σεναρίων (scripting language)*. Τα σεναρία (scripts) που γράφονται σε Javascript, είναι μικρά προγράμματα που αλληλεπιδρούν με το φυλλομετρητή και τον κώδικα HTML μιας σελίδας.

Ο κώδικας της JavaScript γράφεται σε καθαρό κείμενο (ASCII μορφή) και ενσωματώνεται μέσα στον κώδικα της HTML, μπορεί δε να εκτελεσθεί αμέσως με το φόρτωμα της σελίδας ή όταν λαμβάνει χώρα ένα *συμβάν (event)*, όπως η πίεση ενός πλήκτρου του ποντικιού ή η τοποθέτηση του ποντικιού πάνω σε ένα αντικείμενο.

Οι εφαρμογές της Javascript στο δυναμικό προγραμματισμό παγκόσμιου ιστού είναι πάρα πολλές. Ως χαρακτηριστικά παραδείγματα αναφέρουμε τον έλεγχο της εγκυρότητας των δεδομένων που πληκτρολογούνται σε μια φόρμα του πελάτη, πριν αυτά αποσταλούν στον εξυπηρετητή, τη διεξαγωγή διάφορων υπολογισμών και μετατροπών μεγεθών και την προσθήκη δυναμικών μενού στις ιστοσελίδες.

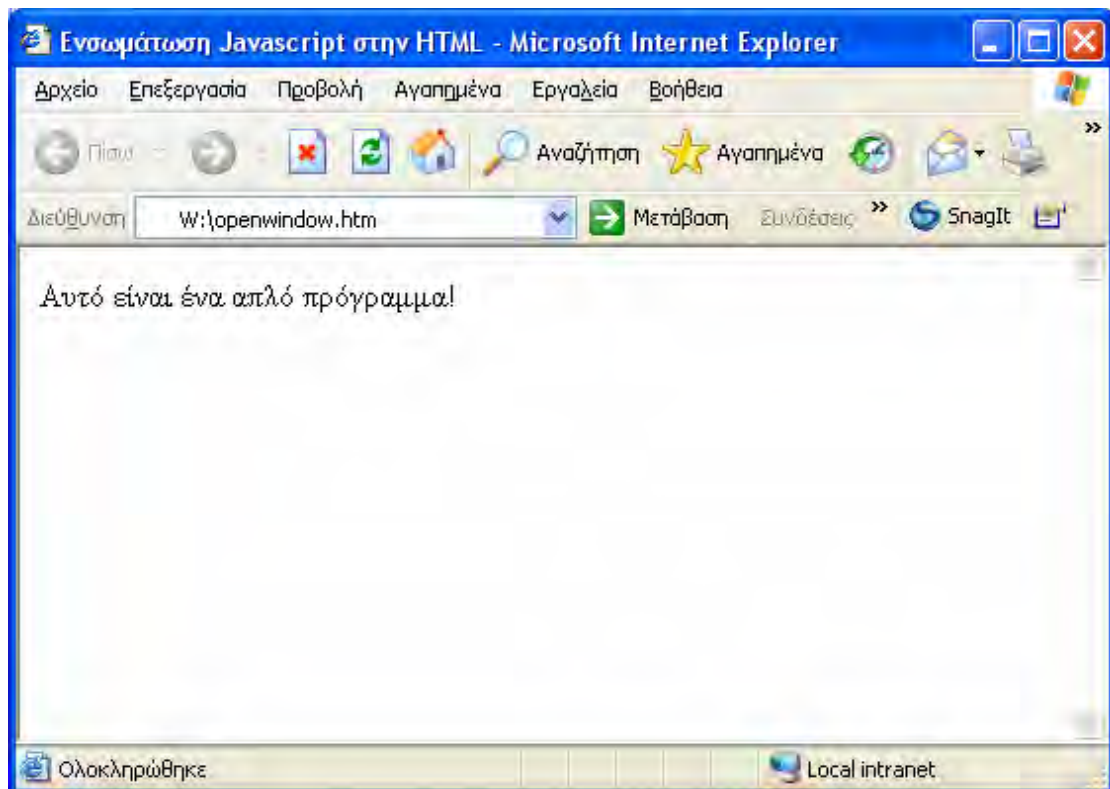
Στην παρούσα ενότητα θα παρουσιάσουμε τη Javascript ξεκινώντας από απλά παραδείγματα τα οποία αναδεικνύουν την ευκολία με την οποία τα σεναρία Javascript ενσωματώνονται στην ιστοσελίδα. Στη συνέχεια, θα παρουσιάσουμε και θα εξεικωθούμε με τη βοήθεια απλών εφαρμογών, με βασικές έννοιες των *αντικειμένων (objects)*, των *γεγονότων (events)* και των *μεθόδων (methods)*, η χρήση των οποίων συνιστά τον κύριο μηχανισμό αλληλεπίδρασης της Javascript με τις ιστοσελίδες HTML. Τέλος, θα εξετάσουμε τη Javascript από το πρίσμα του δομημένου προγραμματισμού, εισάγοντας ταυτόχρονα θεμελιώδεις έννοιες αυτού του τύπου προγραμματισμού, οι οποίες αποτελούν προϋπόθεση για τη βαθύτερη κατανόηση τόσο της Javascript, όσο και των άλλων γλωσσών που θα παρουσιαστούν στα επόμενα κεφάλαια.

2.2.2 Ενσωμάτωση της Javascript σε ένα έγγραφο HTML

Το σενάριο Javascript ενσωματώνεται στην ιστοσελίδα, ανάμεσα στις ετικέτες `<SCRIPT>` και `</SCRIPT>`. Μέσα σε ένα αρχείο HTML μπορούν να υλοποιηθούν πολλά σεναρία Javascript, χρησιμοποιώντας πολλαπλές ετικέτες SCRIPT. Ένα παράδειγμα κομματιού κώδικα, μέσα σε μια HTML σελίδα, είναι το ακόλουθο:

```
<html>
<head>
<TITLE> Ενσωμάτωση Javascript στην HTML </TITLE>
</head>
<body>
<script language="javascript">
document.write("Αυτό είναι ένα απλό πρόγραμμα!");
</script>
</body>
</html>
```

Πληκτρολογώντας το παραπάνω κείμενο στο Notepad και σώζοντας το αρχείο ως test.htm, μπορούμε να το ανοίξουμε με ένα φυλλομετρητή, οπότε και θα λάβουμε το αποτέλεσμα που φαίνεται στο παρακάτω σχήμα.



Σχήμα 2.2 Ένα απλό πρόγραμμα Javascript.

Το σενάριο Javascript του παραπάνω παραδείγματος, αν και πολύ απλό, αποτελεί μια χρήσιμη εισαγωγή στην έννοια των αντικειμένων, τα οποία θα αναλύσουμε εκτενέστερα στη συνέχεια. Όπως θα δούμε, το αντικείμενο *document* αντιπροσωπεύει την οθόνη του φυλλομετρητή, ενώ η μέθοδος *write* αυτού του αντικειμένου, η οποία καλείται από το σενάριο με την εντολή *document.write("...")*, προκαλεί την έξοδο προς το φυλλομετρητή του κειμένου που βρίσκεται ανάμεσα στις παρενθέσεις.

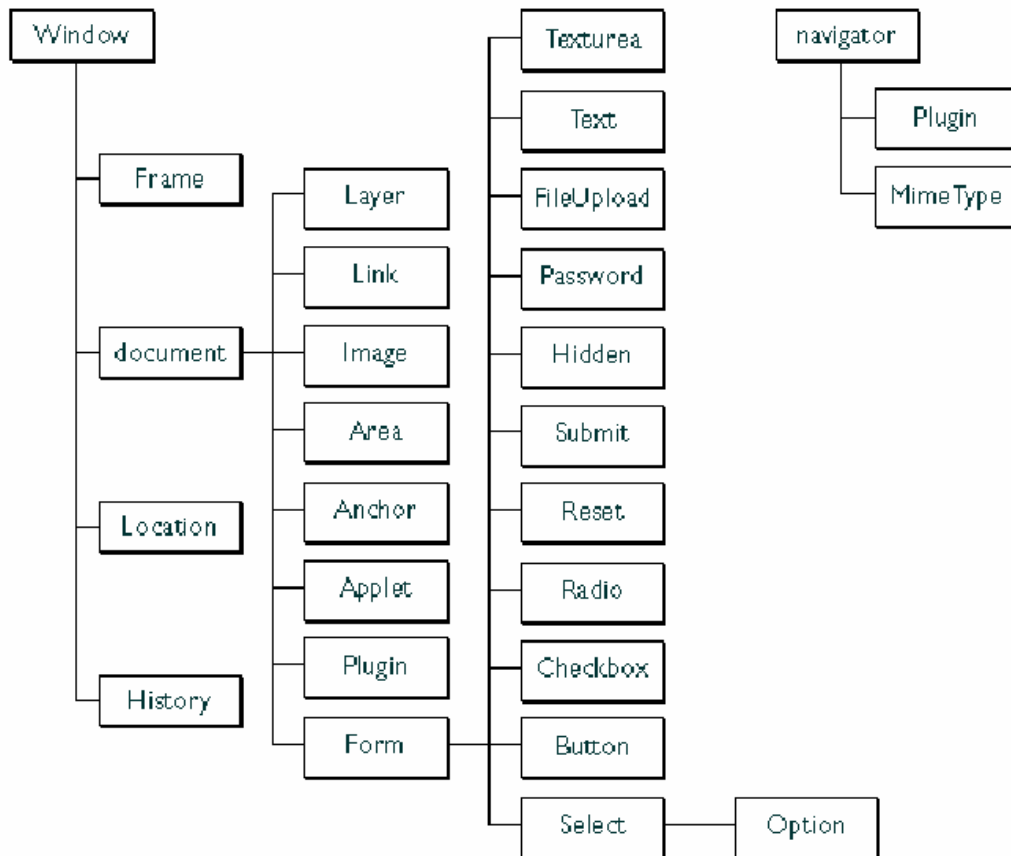
Τα σενάρια Javascript δεν είναι υποχρεωτικά ενσωματωμένα στη σελίδα HTML. Αντίθετα, μπορούν να γραφούν ως ξεχωριστά αρχεία με κατάληξη .js, τα οποία καλούνται και εκτελούνται από τη σελίδα HTML με τη βοήθεια σχετικής αναφοράς. Σε αυτή την περίπτωση, η ετικέτα SCRIPT συνοδεύεται από την παράμετρο SRC, η οποία χρησιμοποιείται για τη δήλωση του εξωτερικού αρχείου στον κώδικα HTML, όπως φαίνεται στο παρακάτω τμήμα κώδικα, στο οποίο το αρχείο global.js δηλώνεται ως εξωτερικό σενάριο Javascript :

```
<script language="JavaScript" src="global.js">  
</script>
```

2.2.3 Η χρήση των αντικειμένων στη Javascript

Στο απλό παράδειγμα της ενότητας 2.2.2, είδαμε μια εισαγωγή στην έννοια του αντικειμένου, με τη βοήθεια μιας απλής εφαρμογής, χρησιμοποιώντας τη μέθοδο *write* του αντικειμένου *document*. Όπως θα δούμε στη συνέχεια, η γλώσσα Javascript βασίζεται στην έννοια του αντικειμένου. Το αντικείμενο είναι μια οντότητα η οποία κατέχει ένα σύνολο από *ιδιότητες*, οι οποίες μπορούν να τροποποιηθούν με κατάλληλες *μεθόδους*. Ένα καλό παράδειγμα αντικειμένου είναι το αντικείμενο *document* (η επιφάνεια του φυλλομετρητή), το οποίο έχει αρκετές ιδιότητες, όπως το χρώμα παρασκηνίου, το χρώμα προσκηνίου, το χρώμα κειμένου και τον τίτλο. Οι περισσότερες ιδιότητες της Javascript σχετίζονται άμεσα ή έμμεσα με αλλαγές μιας ή περισσότερων ιδιοτήτων των αντικειμένων.

Το σύνολο των προκαθορισμένων αντικειμένων του φυλλομετρητή είναι γνωστό με το όνομα "Document Object Model" ή DOM και η ιεραρχία τους απεικονίζεται στο Σχήμα 2.3.



Σχήμα 2.3 Τα αντικείμενα του DOM

Η Javascript χρησιμοποιεί την εξής σύνταξη για να αναφερθεί σε κάποια ιδιότητα:

< όνομα αντικειμένου >.< ιδιότητα >

Για παράδειγμα, το *document.bgcolour* είναι το χρώμα παρασκηγίου για το αντικείμενο εγγράφου (document). Όπως προαναφέραμε, η έννοια του αντικειμένου είναι στενά συνδεδεμένη και με την έννοια της μεθόδου. Οι μέθοδοι είναι συναρτήσεις ή διαδικασίες που χρησιμοποιούνται για να εκτελούν μια λειτουργία σ' ένα αντικείμενο. Η κλήση και εκτέλεση των μεθόδων των αντικειμένων γίνεται με τη δήλωση *αντικείμενο.μέθοδος()*. Χαρακτηριστικό παράδειγμα κλήσης μεθόδου είναι η εντολή *document.write("Αυτό είναι ένα απλό πρόγραμμα!")*, που είδαμε σε προηγούμενο παράδειγμα.

Παραθέτουμε στη συνέχεια λεπτομέρειες για τα βασικά αντικείμενα του DOM, με αναφορά στις σχετικές ιδιότητες, μεθόδους και χειριστήρια συμβάντων που συσχετίζονται με αυτά.

- **window** - Δημιουργείται από τον φυλλομετρητή όταν φορτώνεται μια σελίδα. Αποτελεί το αντικείμενο ανωτέρου επιπέδου (top-level object) για τα αντικείμενα document, location και history. Επειδή είναι το προκαθορισμένο αντικείμενο, δεν χρειάζεται να αναφέρουμε

το όνομα του παραθύρου (window) όταν αναφερόμαστε στα αντικείμενά του, στις ιδιότητές του ή στις μεθόδους του. Για παράδειγμα, οι δηλώσεις *status* ("Hallo!") και *window.status("Hallo")* επιφέρουν το ίδιο αποτέλεσμα, δηλαδή την εμφάνιση ενός μηνύματος στη γραμμή κατάστασης (status line).

- **document** - Είναι ένα αντικείμενο που δημιουργείται από τον φυλλομετρητή όταν φορτώνεται μια ιστοσελίδα, το οποίο περιέχει πληροφορίες για το τρέχον έγγραφο, όπως είναι ο τίτλος του (ιδιότητα *title*), το χρώμα φόντου (ιδιότητα *bgcolor*) και οι φόρμες που περιέχει. Αυτές οι ιδιότητες ορίζονται μέσα στις ετικέτες της HTML.
- **frame** - Είναι ένα παράθυρο του φυλλομετρητή στο οποίο εμφανίζεται ένα έγγραφο HTML. Η επιφάνεια του φυλλομετρητή μπορεί να περιέχει πολλά frames τα οποία μπορεί να δείχνουν σε διαφορετικά URLs και μπορούν να αποτελούν στόχο (target) από άλλα πλαίσια, όπου όλα μαζί βρίσκονται στο ίδιο παράθυρο. Το κάθε πλαίσιο (frame) αποτελεί μια ιδιότητα του αντικειμένου *window*.
- **form** - Αποτελεί ιδιότητα του αντικειμένου document. Η κάθε φόρμα σ' ένα έγγραφο αποτελεί ένα ξεχωριστό αντικείμενο (object) στο οποίο μπορούμε να αναφερθούμε μέσω του αντικειμένου form. Το αντικείμενο form είναι ένας πίνακας (array) που δημιουργείται καθώς ορίζονται οι φόρμες (forms) μέσω των HTML tags. Ας υποθέσουμε ότι σε ένα HTML έγγραφο έχουμε δημιουργήσει δύο φόρμες με όνομα *firstform* και *secondform* αντίστοιχα, όπως φαίνεται στο παρακάτω πλαίσιο:

```
<form method="POST" name="firstform">
.
.
.
</form>
.
.
.
<form method="POST" name="secondform">
.
.
.
</form>
```

Στο παραπάνω παράδειγμα, μπορούμε να αναφερόμαστε στις ιδιότητες και μεθόδους της *firstform* με τη δήλωση *document.firstform* ή εναλλακτικά με τη δήλωση *document.forms[0]*. Οι ανάλογες δηλώσεις για τη φόρμα *secondform* είναι οι *document.secondform* και *document.forms[1]* αντίστοιχα. Σχετικές ιδιότητες του αντικειμένου form είναι οι *action*, *elements*, *encoding*, *forms*, *method*, *name* και *target*. Σχετική μέθοδος είναι η *submit* και σχετικό χειριστήριο συμβάντος είναι το *onSubmit*.

- **button** - Είναι ένα αντικείμενο που αποτελεί στοιχείο μιας φόρμας και μπορεί να χρησιμοποιηθεί για να εκτελεσθεί μια ενέργεια (action). Αποτελεί ιδιότητα του αντικειμένου

form. Σχετικές ιδιότητες είναι η *name* και η *value*, σχετική μέθοδος είναι η *click* και σχετικό χειριστήριο συμβάντος είναι το *onClick*.

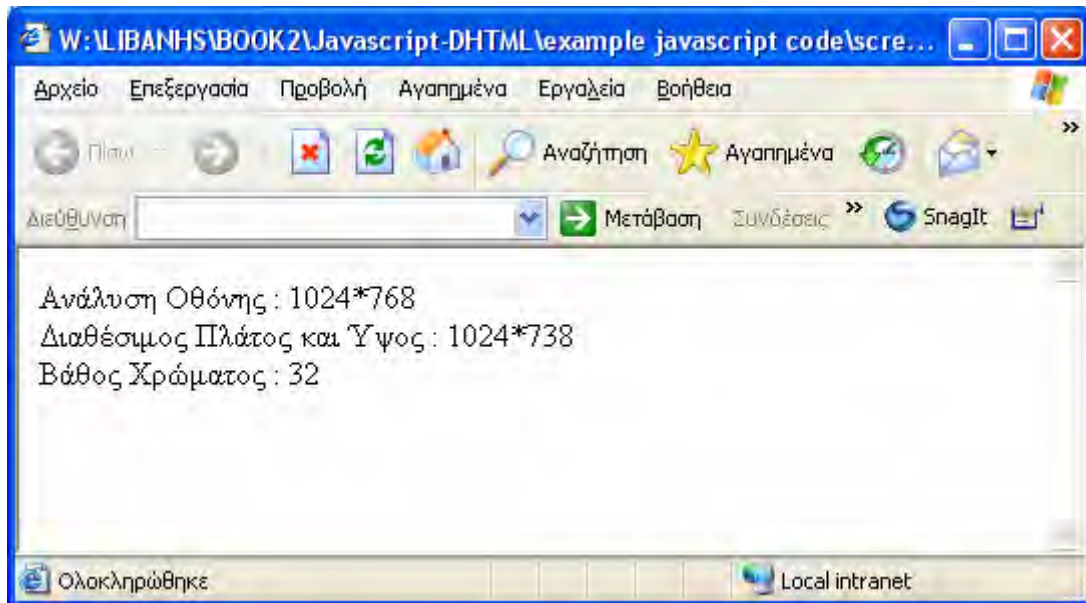
- **checkbox** - Είναι ένα στοιχείο φόρμας το οποίο ο χρήστης έχει τη δυνατότητα να κάνει ενεργό ή όχι (on ή off) κάνοντας κλικ σε αυτό. Χρησιμοποιώντας το αντικείμενο *checkbox*, μπορούμε να δούμε αν το στοιχείο είναι επιλεγμένο (ιδιότητα *checked*) καθώς και να έχουμε πρόσβαση στο όνομα (ιδιότητα *name*) και την τιμή του (ιδιότητα *value*). Σχετική μέθοδος είναι η *click* και σχετικό χειριστήριο συμβάντος είναι το *onClick*.
- **hidden** - Είναι ένα αντικείμενο κειμένου (text object) που δεν εμφανίζεται (είναι κρυφό) σε μια HTML φόρμα. Τα κρυφά αντικείμενα (hidden objects) μπορούν να χρησιμοποιηθούν μαζί με τα cookies για να μεταβιβάζονται ζευγάρια ονόματος/τιμής (name/value pairs) για την επικοινωνία μεταξύ πελάτη/εξυπηρετητή. Το αντικείμενο *hidden*, αποτελεί ιδιότητα του αντικειμένου *form* και σχετικές ιδιότητες είναι οι *cookie*, *defaultValue*, *name* και *value*.
- **history** - Αυτό το αντικείμενο περιέχει πληροφορίες σύνδεσης URL για τις ιστοσελίδες που έχουμε επισκεφθεί νωρίτερα. Αποτελεί ιδιότητα του αντικειμένου *document* και σχετική ιδιότητα είναι η *length*. Σχετικές μέθοδοι είναι οι *back*, *forward* και *go*.
- **link** - Το αντικείμενο αυτό παρέχει πληροφορίες για τους υπάρχοντες συνδέσμους υπερκειμένου. Μπορούμε επίσης να το χρησιμοποιήσουμε για να δημιουργήσουμε καινούργιους συνδέσμους. Αποτελεί ιδιότητα του αντικειμένου *document*. Σχετικές ιδιότητες είναι οι *hash*, *host*, *hostname*, *href*, *length*, *pathname*, *port*, *protocol*, *search* και *target*. Σχετική μέθοδος είναι η *link* και σχετικά χειριστήρια συμβάντος είναι τα *onClick* και *onMouseOver*.
- **location** - Το αντικείμενο αυτό περιέχει πλήρεις πληροφορίες URL για το τρέχον έγγραφο, ενώ η κάθε ιδιότητά του αναφέρεται σε ένα διαφορετικό κομμάτι του URL. Αποτελεί ιδιότητα του αντικειμένου *document*. Σχετικές ιδιότητες είναι οι *hash*, *host*, *hostname*, *href*, *location*, *pathname*, *port*, *protocol*, *search* και *target*.
- **Math** - Το αντικείμενο αυτό περιέχει ιδιότητες για μαθηματικές σταθερές και μεθόδους που χρησιμοποιούνται σε συναρτήσεις. Για παράδειγμα, η δήλωση *Math.PI* αντιπροσωπεύει την τιμή του π σε μια εξίσωση. Σχετικές ιδιότητες είναι οι *E*, *LN10*, *LN2*, *PI*, *SQRT1_2*, *SQRT2* και σχετικές μέθοδοι είναι οι *abs*, *acos*, *asin*, *atan*, *ceil*, *cos*, *exp*, *floor*, *log*, *max*, *min*, *pow*, *random*, *round*, *sin*, *sqrt* και *tan*.
- **navigator** - Το αντικείμενο αυτό περιέχει πληροφορίες για την τρέχουσα έκδοση του φυλλομετρητή που χρησιμοποιεί ο πελάτης. Σχετικές ιδιότητες είναι οι *appName*, *appCodeName*, *appVersion* και *userAgent*.
- **option** - Είναι αντικείμενα που δημιουργούνται μέσα στις φόρμες της HTML και παριστάνουν πλήκτρα επιλογής (option buttons). Από ένα σύνολο πλήκτρων επιλογής που ανήκουν στην ίδια λίστα, ο χρήστης μπορεί να επιλέξει μόνο ένα. Σχετικές ιδιότητες είναι οι *checked*, *defaultChecked*, *index*, *length*, *name* και *value*, σχετική μέθοδος είναι η *click* και σχετικό χειριστήριο συμβάντος είναι το *onClick*.
- **password** - Είναι ειδικά πεδία κειμένου της HTML τα οποία ορίζονται μέσα σε μια φόρμα και στα οποία το κείμενο που καταχωρείται από τον χρήστη είναι κρυμμένο συνήθως με το χαρακτήρα “*”. Σχετικές ιδιότητες είναι οι *defaultValue*, *name* και *value* και σχετικές μέθοδοι είναι οι *focus*, *blur* και *select*.

- **reset** - Είναι ένα πλήκτρο εντολής το οποίο ορίζεται μέσα σε μια φόρμα και επαναφέρει τα περιεχόμενα όλων των αντικειμένων της φόρμας στις προκαθορισμένες τους τιμές. Αποτελεί ιδιότητα του αντικειμένου form. Σχετικές ιδιότητες είναι οι *name* και *value*, σχετική μέθοδος είναι η *click* και σχετικό χειριστήριο συμβάντος είναι το *onClick*.
- **select** - Είναι μια λίστα επιλογής (selection list) ή λίστα κύλισης (scrolling list) σε μια HTML φόρμα. Μια λίστα επιλογής δίνει τη δυνατότητα στο χρήστη να επιλέξει ένα στοιχείο από μια λίστα, ενώ μια λίστα κύλισης δίνει τη δυνατότητα επιλογής ενός ή περισσότερων στοιχείων από μια λίστα. Αποτελεί ιδιότητα του αντικειμένου form. Σχετικές ιδιότητες είναι οι *length*, *name*, *options* και *selectedIndex*, σχετικές μέθοδοι είναι οι *blur* και *focus* και σχετικά χειριστήρια συμβάντος είναι τα *onBlur*, *onChange* και *onFocus*.
- **submit** - Είναι ένα ειδικό πλήκτρο εντολής που δημιουργείται μέσα σε μια φόρμα και που προκαλεί την υποβολή της στο πρόγραμμα που καθορίζεται από την ιδιότητα *action* της φόρμας. Αποτελεί ιδιότητα του αντικειμένου form. Σχετικές ιδιότητες είναι οι *name* και *value*, σχετική μέθοδος είναι η *click* και σχετικό χειριστήριο συμβάντος είναι το *onClick*. Τυπική εφαρμογή του πλήκτρου *submit* είναι η υποβολή τιμών μιας φόρμας σε ένα πρόγραμμα του εξυπηρετητή, όπως θα δούμε στο επόμενο κεφάλαιο.
- **text** - Είναι ένα πεδίο καταχώρισης κειμένου μίας γραμμής σε μια HTML φόρμα το οποίο δέχεται χαρακτήρες ή και αριθμούς. Αποτελεί ιδιότητα του αντικειμένου form. Σχετικές ιδιότητες είναι οι *defaultValue*, *name* και *value*, σχετικές μέθοδοι είναι οι *focus*, *blur* και *select* και σχετικά χειριστήρια συμβάντος είναι τα *onBlur*, *onChange*, *onFocus*, και *onSelect*.
- **textarea** - Είναι παρόμοιο με το πεδίο κειμένου, με τη διαφορά ότι μπορεί να περιέχει πολλές γραμμές. Αποτελεί ιδιότητα του αντικειμένου form. Σχετικές ιδιότητες είναι οι *defaultValue*, *name* και *value*, σχετικές μέθοδοι είναι οι *focus*, *blur* και *select* και σχετικά χειριστήρια συμβάντος είναι τα *onBlur*, *onChange*, *onFocus* και *onSelect*.

Το ακόλουθο απλό παράδειγμα χρησιμοποιεί ιδιότητες και μεθόδους των αντικειμένων *window* και *document* για να εμφανίσει στο φυλλομετρητή πληροφορίες σχετικά με την οθόνη του υπολογιστικού συστήματος. Το σχετικό σενάριο φαίνεται στο παρακάτω πλαίσιο:

```
<html><body>
<script type="text/javascript">
    document.write("Ανάλυση Οθόνης : ")
    document.write(screen.width + " * ")
    document.write(screen.height + "<br>")
    document.write("Διαθέσιμος Πλάτος και Ύψος : ")
    document.write(window.screen.availWidth + " * ")
    document.write(window.screen.availHeight + "<br>")
        document.write("Βάθος Χρώματος : ")
    document.write(window.screen.colorDepth + "<br>")
</script></body></html>
```

Πληκτρολογώντας το παραπάνω κείμενο και αποθηκευόντάς το ως αρχείο HTML, μπορούμε να το ανοίξουμε από το φυλλομετρητή οπότε και θα λάβουμε το ακόλουθο αποτέλεσμα:



Σχήμα 2.4 Εμφάνιση πληροφοριών του συστήματος με τη Javascript.

2.2.4 Τα χειριστήρια συμβάντος (Event Handlers) της Javascript

Τα *χειριστήρια συμβάντος* (*event handlers*) είναι παράμετροι που επιτρέπουν τη συσχέτιση του κώδικα Javascript με συγκεκριμένα γεγονότα που προκύπτουν από τις κινήσεις του φυλλομετρητή ή του χρήστη. Παρακολουθώντας τις ενέργειες του χρήστη, η JavaScript μπορεί να εκτελεί σενάρια στο Η/Υ του πελάτη, χωρίς να χρειασθεί να μεσολαβήσει ο εξυπηρετητής. Οι ενέργειες του χρήστη σε μια ιστοσελίδα οι οποίες καλύπτονται από τα χειριστήρια συμβάντων, σχετίζονται με την κίνηση του ποντικιού και τις καταχωρήσεις στα στοιχεία (πεδία) των φορμών. Τα κυριότερα χειριστήρια συμβάντων της Javascript συνοψίζονται στον ακόλουθο πίνακα:

Χειριστήριο Συμβάντος	Κριτήριο ενεργοποίησης
<i>onBlur</i>	Λαμβάνει χώρα όταν χάνεται η εστίαση (lost focus), δηλαδή όταν απομακρυνόμαστε είτε με κλικ με το ποντίκι είτε πατώντας το πλήκτρο tab σ' ένα πεδίο κειμένου ή σε μια περιοχή κειμένου ή και σ' ένα πλαίσιο λίστας (select) μιας φόρμας.
<i>onChange</i>	Λαμβάνει χώρα όταν τροποποιείται το περιεχόμενο ενός πεδίου κειμένου ή μιας περιοχής κειμένου ή και ενός πλαισίου λίστας μιας φόρμας, πριν ακόμα απομακρυνθούμε από το στοιχείο αυτό.
<i>onClick</i>	Λαμβάνει χώρα όταν κάνουμε κλικ με το ποντίκι σ' ένα αντικείμενο, όπως είναι για παράδειγμα ένα πλήκτρο εντολής (button) ή ένα πλαίσιο ελέγχου (checkbox)
<i>onFocus</i>	Λαμβάνει χώρα όταν κάνουμε εστίαση (focus) μέσα σ' ένα πεδίο μιας φόρμας.
<i>onLoad</i>	Λαμβάνει χώρα όταν τελειώσει το φόρτωμα μιας ιστοσελίδας σ' ένα παράθυρο ή όταν τελειώσει το φόρτωμα όλων των πλαισίων (frames) που βρίσκονται μέσα σε μια σελίδα.
<i>onMouseOver</i>	Λαμβάνει χώρα όταν ο δείκτης του ποντικιού τοποθετείται πάνω από ένα αντικείμενο ή από ένα σύνδεσμο.
<i>onSelect</i>	Λαμβάνει χώρα όταν επιλέγουμε ένα μέρος ή όλο το κείμενο ενός πεδίου κειμένου (text field) ή μιας περιοχής κειμένου (textarea).
<i>onSubmit</i>	Λαμβάνει χώρα όταν κάνουμε κλικ με το ποντίκι σ' ένα πλήκτρο υποβολής (submit button) για να υποβάλλουμε μια φόρμα.
<i>onUnload</i>	Λαμβάνει χώρα όταν φεύγουμε από ένα έγγραφο (ιστοσελίδα).

Σαν παράδειγμα χρήσης των χειριστηρίων συμβάντων, θα χρησιμοποιήσουμε το συμβάν *OnClick* ενός πλήκτρου *button* της οθόνης για να αποθηκεύσουμε μια συγκεκριμένη ιστοσελίδα στα αγαπημένα. Η αποθήκευση στα αγαπημένα γίνεται με την εκτέλεση της συνάρτησης *bookmark()* η οποία εκτελείται με την πίεση του πλήκτρου του ποντικιού στο *button* της οθόνης, όπως δηλώνεται με τη φράση *onclick="bookmark()"* του σεναρίου. Ολόκληρο το σενάριο παρουσιάζεται στο παρακάτω πλαίσιο:

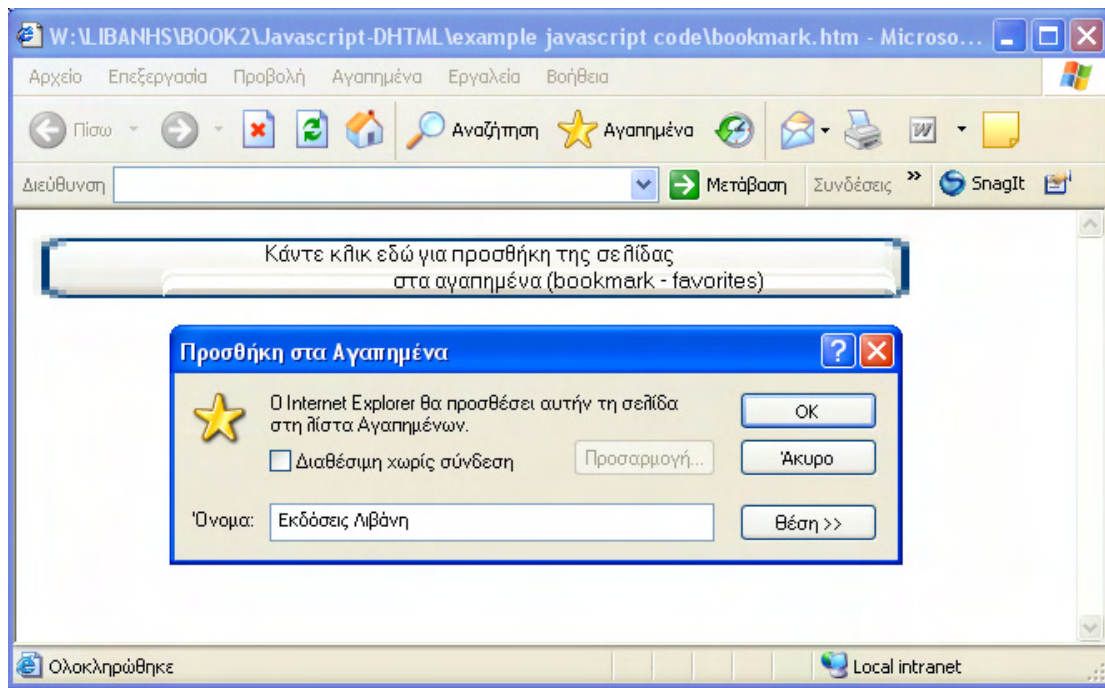
```
<html>
<head>
<script type="text/javascript">
    function bookmark()
    {
        window.external.AddFavorite("http://www.livanis.gr","Εκδόσεις Λιβάνη")
    }
</script>
</head>
```

```

<body>
  <form>
    <input type="button" onclick="bookmark()"
      value="Κάντε κλικ εδώ για προσθήκη της σελίδας
      στα αγαπημένα (bookmark – favorites) ">
  </form>
</body>
</html>

```

Πληκτρολογώντας το παραπάνω σενάριο και ανοίγοντάς το από το φυλλομετρητή, θα λάβουμε το ακόλουθο αποτέλεσμα:



Σχήμα 2.5 Προσθήκη στα αγαπημένα με Javascript.

2.2.5 Η Javascript ως γλώσσα δομημένου προγραμματισμού

Ως γλώσσα προγραμματισμού, η Javascript ανήκει στην κατηγορία γλωσσών δομημένου προγραμματισμού, στην οποία ανήκουν και πολλές άλλες ευρέως διαδεδομένες γλώσσες όπως η C, η Basic, η Pascal κλπ. Ο δομημένος προγραμματισμός βασίζεται σε εντολές που χρησιμοποιούνται για αναθέσεις και συγκρίσεις τιμών και για τον έλεγχο της ροής και την εκτέλεση συγκεκριμένων τμημάτων του κώδικα. Στη συνέχεια θα αναλύσουμε τις βασικές συνιστώσες του δομημένου προγραμματισμού, όπως αυτές υλοποιούνται στη γλώσσα Javascript.

Η έννοια της μεταβλητής

Στο δομημένο προγραμματισμό, οι *μεταβλητές* (*variables*) είναι σύμβολα τα οποία αναπαριστούν δεδομένα, και των οποίων οι τιμές *αλλάζουν* κατά τη διάρκεια εκτέλεσης του προγράμματος. Ας εξετάσουμε το παρακάτω τμήμα σεναρίου Javascript:

```
.  
.   
.   
Y=5;  
X=Y+10;  
X=X+Y;  
document.write(X);  
.   
.   
.
```

Στο παραπάνω τμήμα κώδικα, η μεταβλητή *Y* παίρνει αρχικά την τιμή 5. Στη συνέχεια, εκτελείται η πρόσθεση $Y + 10$ και το αποτέλεσμα εκχωρείται στη μεταβλητή *X*, η οποία λαμβάνει πλέον την τιμή 15. Στη συνέχεια, η εντολή $X=X+Y$ προσθέτει αρχικά τις τιμές των μεταβλητών *X* και *Y* και στη συνέχεια εκχωρεί το αποτέλεσμα της πρόσθεσης στη μεταβλητή *X*, η οποία έχει πλέον την τιμή 20. Τέλος, η εντολή `document.write(X);` προκαλεί την εμφάνιση του περιεχομένου της μεταβλητής *X*, δηλαδή του 20, στην οθόνη του φυλλομετρητή.

Σε αντίθεση με άλλες γλώσσες προγραμματισμού, στη JavaScript δεν είναι απαραίτητο να δηλωθεί ο τύπος των δεδομένων μιας μεταβλητής πριν την ανάθεση τιμής σε αυτή και οι τύποι δεδομένων μετατρέπονται αυτόματα κατά την εκτέλεση του προγράμματος. Έτσι μπορεί να γίνεται ανάθεση οποιασδήποτε τιμής σε οποιαδήποτε μεταβλητή. Η μόνη στιγμή που εξετάζονται θέματα τύπων δεδομένων είναι όταν γίνονται λειτουργίες μεταξύ των εμπλεκόμενων δεδομένων. Για παράδειγμα, η εντολή `X="5"+"10";` εκχωρεί στη μεταβλητή *X* την τιμή "510", εφόσον τα δεδομένα ανάμεσα σε εισαγωγικά αντιμετωπίζονται πλέον ως χαρακτήρες.

Οι τελεστές

Οι τελεστές είναι σύμβολα που ενεργούν πάνω σε μία ή περισσότερες μεταβλητές ή τιμές δεδομένων και εκτελούν αριθμητικές ή λογικές πράξεις επιστρέφοντας μια νέα τιμή. Οι μεταβλητές ή οι τιμές δεδομένων στις οποίες επενεργούν οι τελεστές ονομάζονται *ορίσματα*. Οι τελεστές χωρίζονται σε διάφορες κατηγορίες ανάλογα την πράξη που εκτελούν. Πιο συγκεκριμένα, οι *αριθμητικοί τελεστές* επιστρέφουν τιμές που προκύπτουν από την εκτέλεση αριθμητικών πράξεων μεταξύ των ορισμάτων τους. Οι *τελεστές σύγκρισης* επιστρέφουν τις τιμές «Αληθής» ή «Ψευδής», ανάλογα με το αποτέλεσμα της σύγκρισης μεταξύ των ορισμάτων τους. Οι *λογικοί τελεστές* εκτελούν λογικές πράξεις μεταξύ των ορισμάτων, ενώ οι *τελεστές*

ανάθεσης χρησιμοποιούνται για την εκχώρηση τιμών σε μεταβλητές. Οι κυριότεροι τελεστές της Javascript συνοψίζονται στον παρακάτω πίνακα:

Είδος τελεστή	Σύμβολο	Λειτουργία
<i>Αριθμητικοί</i>	+ - * / % ++ --	Πρόσθεση Αφαίρεση Πολλαπλασιασμός Διαίρεση Ακέραιο μέρος Μοναδιαία Αύξηση Μοναδιαία Μείωση
<i>Σύγκρισης</i>	== != > >= < <=	επιστρέφει αληθές αν τα ορίσματα είναι ίσα επιστρέφει αληθές αν τα ορίσματα δεν είναι ίσα επιστρέφει αληθές όταν το αριστερό όρισμα είναι μεγαλύτερο από το δεξί επιστρέφει αληθές όταν το αριστερό όρισμα είναι μεγαλύτερο ή ίσο με το δεξί επιστρέφει αληθές όταν το αριστερό όρισμα είναι μικρότερο από το δεξί επιστρέφει αληθές όταν το αριστερό όρισμα είναι μικρότερο ή ίσο με το δεξί
<i>Λογικοί</i>	&& 	επιστρέφει αληθές αν και τα δύο ορίσματα είναι αληθή επιστρέφει αληθές αν οποιοδήποτε από τα δύο ορίσματα είναι αληθές
<i>Ανάθεσης</i>	= +=, -=, *=/	Θέτει την τιμή της έκφρασης στα δεξιά στη μεταβλητή του αριστερού μέρους Προσθέτει/Αφαιρεί/Πολλαπλασιάζει/Διαιρεί την τιμή της έκφρασης στα δεξιά και τη θέτει στη μεταβλητή του αριστερού μέρους

Συναρτήσεις

Μία συνάρτηση ομαδοποιεί ένα σύνολο εντολών υπό μια ονομαζόμενη υπορουτίνα. Αυτό επιτρέπει στο χρήστη να καλεί τη συνάρτηση αυτή όταν χρειαστεί τις λειτουργίες της. Οι συναρτήσεις είναι ένας βασικός τρόπος προγραμματισμού στα περισσότερα προγράμματα σε JavaScript. Φυσικά πριν την κλήση θα πρέπει να έχει γίνει ο ορισμός της συνάρτησης. Η

χρήση των συναρτήσεων μπορεί να χωριστεί σε δύο λογικές κατηγορίες: Ορισμός συναρτήσεων και κλήση συναρτήσεων.

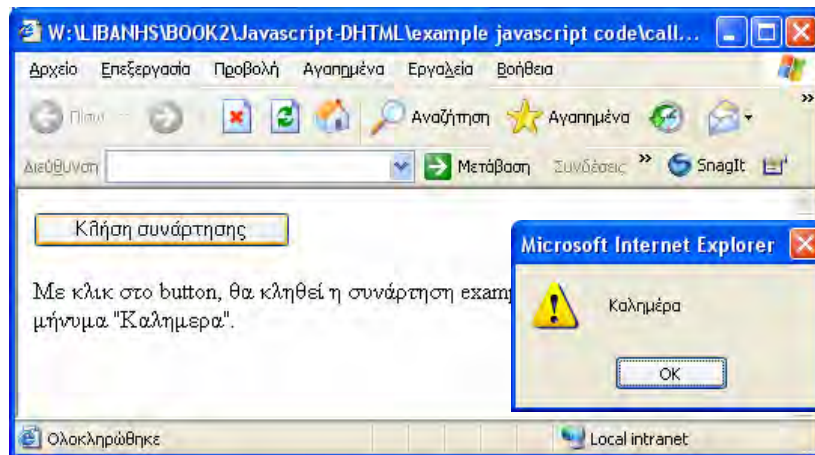
Η Javascript παρέχει τη δυνατότητα δημιουργίας συναρτήσεων, οι οποίες, ουσιαστικά, είναι σενάρια τα οποία εκτελούνται όταν κληθούν, με το όνομά τους, από κάποιο σημείο του HTML κώδικα, ή από κάποιο άλλο σενάριο Javascript. Οι συναρτήσεις της Javascript ενσωματώνονται ανάμεσα στις ετικέτες <HEAD> και </HEAD> και η δήλωσή τους γίνεται με την εντολή

function Ονομα_Συνάρτησης(Ορισμα ή Ορίσματα)

Το όρισμα της συνάρτησης είναι προαιρετικό και παρέχει τη δυνατότητα μεταβίβασης συγκεκριμένων παραμέτρων στη συνάρτηση κατά την κλήση της. Στο ακόλουθο παράδειγμα, ορίζεται η συνάρτηση example(), η οποία στη συνέχεια καλείται πατώντας ένα button που έχει δημιουργηθεί με κώδικα HTML.

```
<html>
<head>
<script type="javascript">
    function example()
    {
        alert("Καλημέρα")
    }
</script>
</head>
<body>
    <form>
        <input type="button" onclick="example()" value="Κλήση συνάρτησης">
    </form>
<p>Με κλικ στο button, θα κληθεί η συνάρτηση example που θα εμφανίσει το μήνυμα "Καλημερα".</p>
</body>
</html>
```

Αποθηκεύοντας το παραπάνω κείμενο ως αρχείο HTML και ανοίγοντας στη συνέχεια το αρχείο αυτό με το φυλλομετρητή, εάν πιάσουμε το button «Κλήση συνάρτησης», καλείται η συνάρτηση example του κώδικα Javascript η οποία εμφανίζει ένα μήνυμα, όπως φαίνεται στο παρακάτω στιγμιότυπο:



Σχήμα 2.6 Εμφάνιση μηνύματος με Javascript.

Ένα άλλο παράδειγμα το οποίο συνδυάζει την έννοια της συνάρτησης με τη δυνατότητα δυναμικής τροποποίησης των ιδιοτήτων των αντικειμένων, περιλαμβάνεται στον παρακάτω κώδικα:

```

<html>

<head>
<title>Μετατροπéας δραχμών σε Ευρώ</title>
</head>

<body>

<script language=javascript>

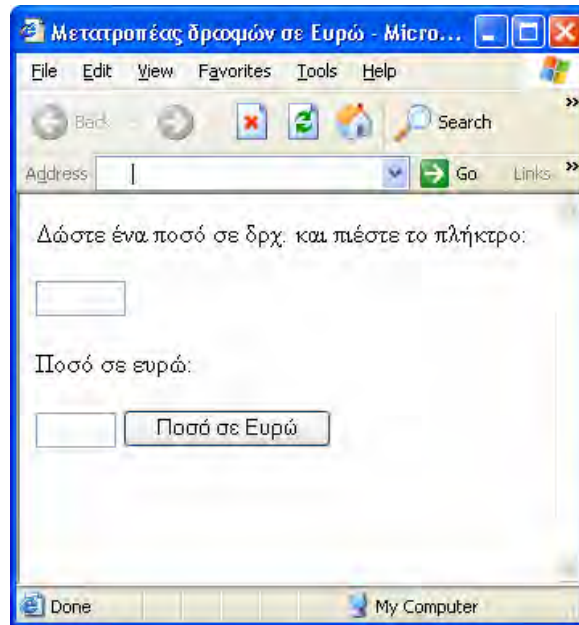
function calculate()
{
D=document.myform.T1.value;
E=D/340.75;
document.myform.T2.value=E;
}
</script>
<form name="myform" >
<p>Δώστε ένα ποσό σε δρχ. και πίεςτε το πλήκτρο:<p>
<input type="text" name="T1" size="5"><p>
<p>Ποσό σε ευρώ:<p>
<input type="text" name="T2" size="4">
<input type="button" value="Ποσό σε Ευρώ" onClick="calculate();" name="Execute" >
</form>

</body>

</html>

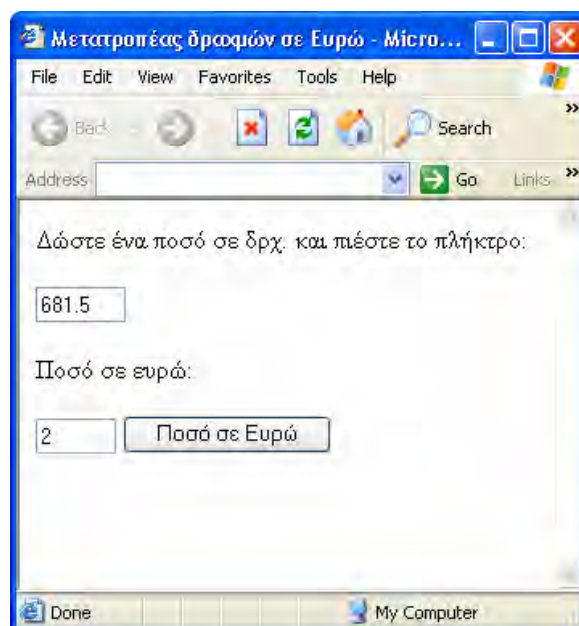
```


Στο παραπάνω αρχείο, έχουμε τοποθετήσει δύο πεδία κειμένου με ονόματα T1 και T2. Στο πεδίο T1, θα πληκτρολογήσουμε έναν αριθμό που αντιστοιχεί σε ένα ποσό σε δραχμές, το οποίο θέλουμε να μετατρέψουμε σε Ευρώ. Πιέζοντας το πλήκτρο «Ποσό σε Ευρώ», με τη βοήθεια του συμβάντος onClick καλείται η συνάρτηση calculate(), η οποία πραγματοποιεί τον υπολογισμό και καταχωρεί το αποτέλεσμα στο πεδίο κειμένου T2. Αποθηκεύοντας το παραπάνω κείμενο και ανοίγοντάς το με το φυλλομετρητή λαμβάνουμε το αποτέλεσμα που φαίνεται στο ακόλουθο σχήμα:



Σχήμα 2.7

Παρατηρούμε ότι μπορούμε πράγματι να αλλάξουμε δυναμικά τις ιδιότητες των αντικειμένων του φυλλομετρητή με τη βοήθεια της συνάρτησης Javascript που δημιουργήσαμε, όπως φαίνεται στο ακόλουθο στιγμιότυπο:



Σχήμα 2.8

Εντολές ελέγχου ροής

Οι εντολές ελέγχου ροής κατευθύνουν τη ροή του προγράμματος ανάλογα με την τιμή των μεταβλητών που συμμετέχουν. Η προγραμματιστική δομή που μας επιτρέπει αυτή την αλλαγή ροής του προγράμματος ονομάζεται δομή επιλογής και υλοποιείται στη Javascript με τις παρακάτω εντολές:

```
if (συνθήκη)  
{Τμήμα Εντολών 1}  
else  
{Τμήμα Εντολών 2}
```

Ως συνθήκη εννοούμε οποιαδήποτε ποσότητα δεδομένων ή έκφραση, των οποίων το αποτέλεσμα μπορεί να έχει τις τιμές Αληθής ή Ψευδής. Στην παραπάνω δομή, αν η συνθήκη είναι Αληθής, εκτελείται το Τμήμα Εντολών 1. Διαφορετικά, εκτελείται το Τμήμα Εντολών 2. Μια άλλη μορφή της δομής επιλογής η οποία επιτρέπει την αλλαγή ροής για περισσότερες από δύο περιπτώσεις υλοποιείται με τις παρακάτω εντολές:

```
if (συνθήκη1)  
{Τμήμα Εντολών 1}  
else if (συνθήκη2)  
{Τμήμα Εντολών 2}  
.  
.  
.  
else if (συνθήκη n)  
{Τμήμα Εντολών n}
```

Η παραπάνω δομή ονομάζεται δομή πολλαπλής επιλογής και περιλαμβάνει περισσότερες από μία συνθήκες, δίνοντας τη δυνατότητα επιλεκτικής εκτέλεσης αντίστοιχων διαφορετικών τμημάτων κώδικα. Χρησιμοποιώντας τη δομή πολλαπλής επιλογής, ο κώδικας του παρακάτω παραδείγματος αλλάζει το χρώμα φόντου, ανάλογα με τον αριθμό που θα πληκτρολογήσουμε στο πεδίο κειμένου.

```
<html>
<head>
<title>Δομή επιλογής</title>
</head>
<body>

<script language=javascript>

function changebackground(arg)
{
if (arg==1)
    {
    document.body.style.background='red';
    }
else if (arg==2)
    {
    document.body.style.background='green';
    }
else if (arg==3)
    {
    document.body.style.background='blue';
    }

}
</script>

<form name="testform">
<input type="text" name="T1" size="20" onChange="changebg (this.value)"></p>

</form>
</body>
```

Ενότητα 2.3 Η γλώσσα DHTML

2.3.1 Εισαγωγή στην DHTML

Η DHTML (*Dynamic HTML*) είναι ένας συνδυασμός τεχνικών ο οποίος επιτρέπει τη δημιουργία δυναμικών σελίδων με τη χρήση σεναρίων και με κατάλληλη διαχείριση του στυλ (style), της διάταξης (layout) και των περιεχομένων (contents) τους. Η *DHTML*, δεν αποτελεί κάποιο πρότυπο που έχει ορισθεί από κάποιο διεθνή οργανισμό, ούτε είναι μια καινούργια γλώσσα προγραμματισμού ή μορφοποίησης. Η DHTML είναι απλώς ένας συνδυασμός της HTML, των Φύλλων Στυλ (Style Sheets) και των σεναρίων της JavaScript ή/και της VBScript, που δίνει στα έγγραφα σημαντικές δυνατότητες αλληλεπίδρασης, επιτρέποντας παράλληλα τη δημιουργία εντυπωσιακών εφέ. Πιο συγκεκριμένα, η DHTML συνδυάζει τη γλώσσα HTML με τα παρακάτω στοιχεία:

- **Διαδοχικά φύλλα στυλ (Cascading Style Sheets -CSS):** Τα CSS παρέχουν ένα μοντέλο στυλ (style) και διάταξης (layout) για τα HTML έγγραφα. Η δημιουργία δυναμικών HTML εγγράφων δεν θα ήταν δυνατή χωρίς τα CSS.
- **Document Object Model (DOM):** Όπως είδαμε στην προηγούμενη ενότητα, το DOM παρέχει ένα μοντέλο περιεχομένου εγγράφου για τα HTML έγγραφα.
- **JavaScript (και VBScript):** Η δημιουργία δυναμικών HTML εγγράφων είναι άρρηκτα συνδεδεμένη με τη δυνατότητα δυναμικής αλλαγής του στυλ, της διάταξης και του περιεχομένου τους μέσω σεναρίων Javascript ή VBScript.

2.3.2 Διαδοχικά φύλλα στυλ (Cascading Style Sheets – CSS)

Τα CSS παρέχουν ένα μοντέλο στυλ (style) και διάταξης (layout) για τα HTML έγγραφα, ορίζοντας τον τρόπο εμφάνισης των στοιχείων της HTML. Τα CSS προστέθηκαν στην HTML 4.0 προκειμένου να καταστεί δυνατός ο διαχωρισμός μεταξύ του περιεχομένου ενός HTML εγγράφου και της μορφής που αυτό λαμβάνει κατά την εμφάνισή του. Ο διαχωρισμός αυτός επιτρέπει την ανεξάρτητη επεξεργασία του περιεχομένου και της μορφής των αρχείων HTML, απλουστεύοντας σημαντικά τις σχετικές εργασίες, όπως θα δούμε στη συνέχεια.

Δημιουργία και χρήση εξωτερικών φύλλων στυλ CSS

Τα φύλλα στυλ CSS μπορούν να ορισθούν μέσα σε ένα HTML αρχείο. Παρόλ' αυτά, είναι συνήθης πρακτική η δημιουργία εξωτερικών φύλλων στυλ CSS, με τα οποία τα αρχεία HTML συνδέονται με σχετική παραπομπή. Τα εξωτερικά φύλλα στυλ παρέχουν μεγάλα πλεονεκτήματα, καθώς επιτρέπουν την αλλαγή της μορφής πολλών ή ακόμα και όλων των ιστοσελίδων ενός δικτυακού τόπου με την αλλαγή παραμέτρων σε ένα μόνο αρχείο CSS. Αυτό γίνεται φανερό με το παρακάτω παράδειγμα.

Δημιουργούμε με το Notepad τα αρχεία Athens.htm και thessaloniki.htm ως εξής:

Athens.htm

```
<html>
<head>
<link rel="stylesheet" type="text/css" href="ex1.css">
</head>
<body>
<h1> Τουριστικός Οδηγός Αθήνας </h1>
<h2> Αξιοθέατα </h2>
<h2> Συγκοινωνίες </h2>
<h2> Ψυχαγωγία </h2>
</body>
</html>
```

Thessaloniki.htm

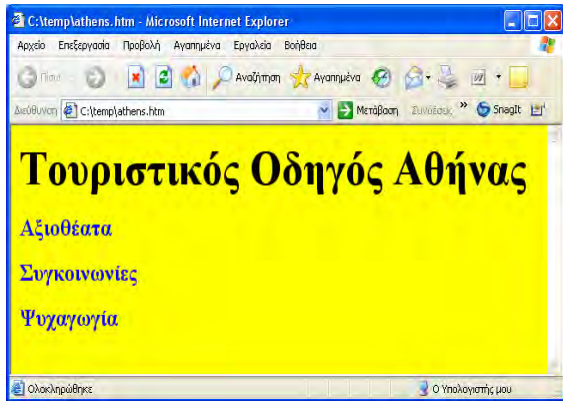
```
<html>
<head>
<link rel="stylesheet" type="text/css" href="ex1.css">
</head>
<body>
<h1> Τουριστικός Οδηγός Θεσσαλονίκης </h1>
<h2> Αξιοθέατα </h2>
<h2> Συγκοινωνίες </h2>
<h2> Ψυχαγωγία </h2>
</body>
</html>
```

Παρατηρούμε στην τρίτη σειρά των παραπάνω αρχείων την αναφορά στο εξωτερικό φύλλο στυλ ex1.css. Δημιουργούμε στη συνέχεια, πάλι με το Notepad, το αρχείο ex1.css με το παρακάτω περιεχόμενο:

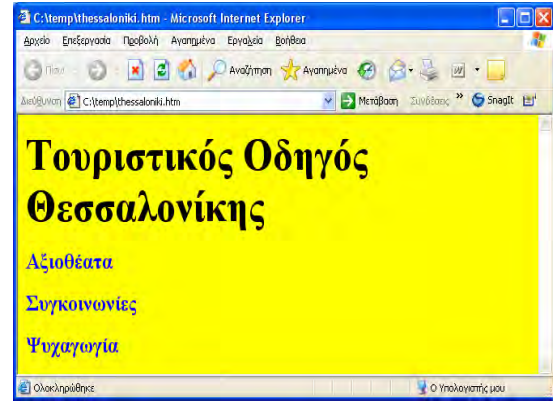
Ex1.css

```
body {background-color: yellow}
h1 {font-size: 36pt}
h2 {color: blue}
```

Ανοίγοντας τα αρχεία Athens.htm και Thessaloniki.htm με το φυλλομετρητή παρατηρούμε τα εξής:



Σχήμα 2.9(α)

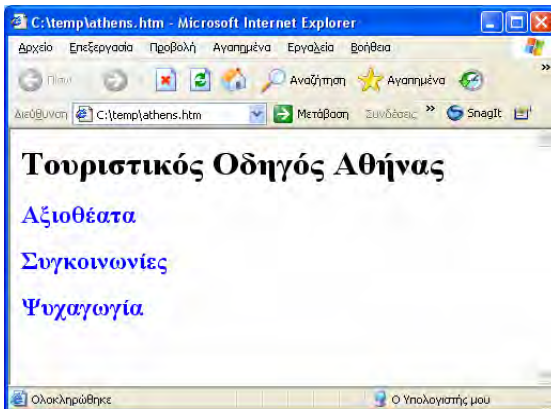


Σχήμα 2.9(β)

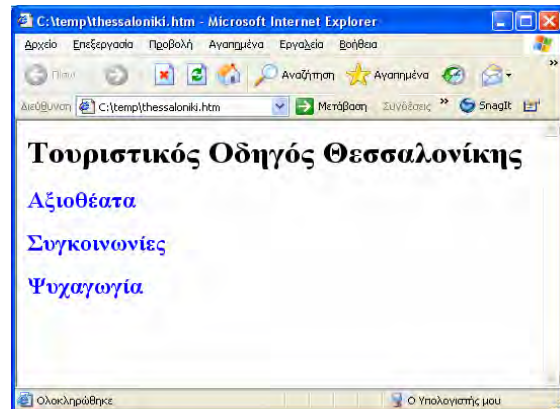
Στη συνέχεια τροποποιούμε το αρχείο ex1.css αλλάζοντας το χρώμα φόντου σε λευκό και το μέγεθος της επικεφαλίδας h1 σε 24pt και αποθηκεύουμε το αρχείο.

```
Ex1.css
body {background-color: white}
h1 {font-size: 24pt}
h2 {color: blue}
```

Ανοίγοντας πάλι τα αρχεία Athens.htm και Thessaloniki.htm με το φυλλομετρητή, παρατηρούμε ότι και στα δύο έχει αλλάξει τόσο το χρώμα φόντου όσο και το μέγεθος γραμματοσειράς της επικεφαλίδας.



Σχήμα 2.10(α)



Σχήμα 2.10(β)

Από το παραπάνω παράδειγμα, είναι φανερό οι δυνατότητες που παρέχουν τα εξωτερικά φύλλα στυλ, εφόσον επιτρέπουν τη μορφοποίηση ενός μεγάλου αριθμού εγγράφων HTML με την τροποποίηση ενός και μόνο αρχείου .css.

Ιδιότητες φόντου των CSS

Οι ιδιότητες φόντου (background properties) μάς δίνουν τη δυνατότητα να ελέγξουμε το χρώμα φόντου ενός στοιχείου, να ορίσουμε μια εικόνα σαν φόντο, να επαναλάβουμε μια εικόνα φόντου κατακόρυφα ή οριζόντια και να τοποθετήσουμε μια εικόνα σε μια σελίδα.

Ιδιότητα	Περιγραφή	Τιμές
background	Ορίζει όλες τις ιδιότητες φόντου σε μια δήλωση	<i>background-color</i> <i>background-image</i> <i>background-repeat</i> <i>background-attachment</i> <i>background-position</i>
background-attachment	Ορίζει αν μια εικόνα φόντου είναι σταθερή (fixed) ή κυλάει μαζί με την υπόλοιπη σελίδα (scroll)	<i>scroll, fixed</i>
background-color	Ορίζει το χρώμα φόντου ενός στοιχείου	<i>Color-rgb, color-hex,color-name,transparent</i>
background-image	Ορίζει μια εικόνα σαν φόντο	<i>url, none</i>
background-position	Ορίζει τη θέση εκκίνησης μια εικόνας φόντου	<i>top left, top center, top right, center left, center center, center right, bottom left bottom center, bottom right</i> <i>x-%, y-%, x-pos y-pos</i>
background-repeat	Ορίζει αν και πώς θα επαναλαμβάνεται μια εικόνα φόντου	<i>repeat, repeat-x, no-repeat</i>

Ιδιότητες κειμένου των CSS

Οι ιδιότητες κειμένου (text properties) μάς δίνουν τη δυνατότητα να ελέγξουμε την εμφάνιση του κειμένου. Είναι δυνατό να αλλάξουμε το χρώμα ενός κειμένου, να αυξήσουμε ή να ελαττώσουμε το διάστημα ανάμεσα στους χαρακτήρες ενός κειμένου, να ευθυγραμμίσουμε ένα κείμενο, να διακοσμήσουμε ένα κείμενο, να δημιουργήσουμε εσοχή στην πρώτη γραμμή ενός κειμένου κ.ά.

Ιδιότητα	Περιγραφή	Τιμές
color	Ορίζει το χρώμα του κειμένου	<i>color</i>
direction	Ορίζει την κατεύθυνση του κειμένου	<i>ltr, rtl</i>
letter-spacing	Αυξάνει ή ελαττώνει το διάστημα ανάμεσα στους χαρακτήρες	<i>normal, length</i>
text-align	Ευθυγραμμίζει το κείμενο σ' ένα στοιχείο	<i>Left, right, center, justify</i>
text-decoration	Προσθέτει διακόσμηση στο κείμενο	<i>none, underline, overline line-through, blink</i>
text-indent	Δημιουργεί εσοχή στην πρώτη γραμμή του κειμένου ενός στοιχείου	<i>length</i>

text-transform	Ελέγχει τα γράμματα σ' ένα στοιχείο	<i>none, capitalize, uppercase, lowercase</i>
white-space	Ορίζει το πώς θα αντιμετωπίζεται το λευκό κενό (white space) μέσα σ' ένα στοιχείο	<i>normal, pre, nowrap</i>
word-spacing	Αυξάνει ή ελαττώνει το διάστημα ανάμεσα στις λέξεις	<i>normal, length</i>

Ιδιότητες γραμματοσειράς των CSS

Οι ιδιότητες γραμματοσειρών μάς δίνουν τη δυνατότητα να αλλάξουμε την οικογένεια της γραμματοσειράς (font family), το έντονο (boldness), το μέγεθος (size) και το στυλ (style) ενός κειμένου.

Ιδιότητα	Περιγραφή	Τιμές
Font	Ορίζει όλες τις ιδιότητες μιας γραμματοσειράς σε μια δήλωση	<i>font-style, font-variant, font-weight, font-size/line-height, font-family, caption, icon, menu, message-box, small-caption, status-bar</i>
font-family	Μια λίστα προτεραιότητας οικογενειών γραμματοσειράς και/ή generic ονομάτων οικογένειας για ένα στοιχείο	<i>family-name, generic-family</i>
font-size	Ορίζει το μέγεθος μιας γραμματοσειράς	<i>xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger, length</i>
font-size-adjust	Καθορίζει μια τιμή άποψης (aspect value) για ένα στοιχείο που θα διατηρήσει το x-height της πρώτης επιλεγμένης γραμματοσειράς	<i>None, number</i>
font-stretch	Συμπυκνώνει ή επεκτείνει την τρέχουσα οικογένεια γραμματοσειράς	<i>Normal, wider, narrower, ultra-condensed, extra-condensed, condensed, semi-condensed, semi-expanded, expanded, extra-expanded, ultra-expanded</i>
font-style	Ορίζει το στυλ της γραμματοσειράς	<i>normal, italic, oblique</i>

font-variant	Εμφανίζει το κείμενο με μικρά κεφαλαία (small-caps) ή κανονικά (normal)	<i>normal, small-caps</i>
font-weight	Ορίζει το βάρος (weight) μιας γραμματοσειράς	<i>normal, bold, bolder, lighter 100, 200, ...,900</i>

Ιδιότητες περιγραμμάτων των CSS

Οι ιδιότητες περιγραμμάτων μάς δίνουν τη δυνατότητα να καθορίσουμε το στυλ, το χρώμα και το πλάτος του περιγράμματος ενός στοιχείου. Στην HTML χρησιμοποιούμε πίνακες για να δημιουργούμε περιγράμματα γύρω από ένα κείμενο, αλλά με τις ιδιότητες περιγραμμάτων μπορούμε να δημιουργήσουμε περιγράμματα με ωραία εφέ και τα οποία μπορούν να εφαρμοσθούν σ' ένα οποιοδήποτε στοιχείο.

Ιδιότητα	Περιγραφή	Τιμές
border	Ορίζει όλες τις ιδιότητες για τα τέσσερα περιγράμματα σε μια δήλωση	border-width, border-style border-color
border-bottom	Ορίζει όλες τις ιδιότητες του κάτω περιγράμματος σε μια δήλωση	border-bottom-width, border-style, border-color
border-bottom-color	Ορίζει το χρώμα του κάτω περιγράμματος	border-color
border-bottom-style	Ορίζει το στυλ του κάτω περιγράμματος	border-style
border-bottom-width	Ορίζει το πλάτος του κάτω περιγράμματος	thin, medium, thick, length
border-color	Ορίζει το χρώμα των τεσσάρων περιγραμμάτων και μπορεί να έχει τιμή από ένα έως τέσσερα χρώματα	color
border-left	Ορίζει όλες τις ιδιότητες του αριστερού περιγράμματος σε μια δήλωση	border-left-width, border-style, border-color
border-left-color	Ορίζει το χρώμα του αριστερού περιγράμματος	border-color,
border-left-style	Ορίζει το στυλ του αριστερού περιγράμματος	border-style
border-left-width	Ορίζει το πλάτος του αριστερού περιγράμματος	thin, medium, thick, length

border-right	Ορίζει όλες τις ιδιότητες του δεξιού περιγράμματος σε μια δήλωση	border-right-width border-style, border-color
border-right-color	Ορίζει το χρώμα του δεξιού περιγράμματος	border-color
border-right-style	Ορίζει το στυλ του δεξιού περιγράμματος	border-style
border-right-width	Ορίζει το πλάτος του δεξιού περιγράμματος	thin, medium, thick, length
border-style	Ορίζει το στυλ των τεσσάρων περιγραμμάτων και μπορεί να έχει τιμή από ένα έως τέσσερα στυλ	none, hidden, dotted, dashed, solid, double, groove, ridge, inset, outset
border-top	Ορίζει όλες τις ιδιότητες του πάνω περιγράμματος σε μια δήλωση	border-top-width border-style, border-color
border-top-color	Ορίζει το χρώμα του πάνω περιγράμματος	border-color
border-top-style	Ορίζει το στυλ του πάνω περιγράμματος	border-style
border-top-width	Ορίζει το πλάτος του πάνω περιγράμματος	thin, medium, thick, length
border-width	Ορίζει το πλάτος των τεσσάρων περιγραμμάτων σε μια δήλωση και μπορεί να έχει από μία έως τέσσερις τιμές	thin, medium, thick, length

Ιδιότητες περιθωρίου των CSS

Οι ιδιότητες περιθωρίου ορίζουν το διάστημα γύρω από τα στοιχεία. Μπορούμε να χρησιμοποιήσουμε αρνητικές τιμές για να επικαλύψουμε (overlap) το περιεχόμενο.

Το πάνω, δεξιά, κάτω και αριστερό περιθώριο μπορούν να αλλάξουν ανεξάρτητα χρησιμοποιώντας ξεχωριστές ιδιότητες. Μπορούμε να χρησιμοποιήσουμε την ιδιότητα margin για να αλλάξουμε όλα τα περιθώρια μονομιάς.

Ιδιότητα	Περιγραφή	Τιμές
margin	Ορίζει τις ιδιότητες περιθωρίου σε μια δήλωση	margin-top, margin-right margin-bottom, margin-left

margin-bottom	Ορίζει το κάτω περιθώριο ενός στοιχείου	auto, length, %
margin-left	Ορίζει το αριστερό περιθώριο ενός στοιχείου	auto, length, %
margin-right	Ορίζει το δεξιό περιθώριο ενός στοιχείου	auto, length, %
margin-top	Ορίζει το πάνω περιθώριο ενός στοιχείου	auto, length, %

Ενότητα 2.4 Συνδυάζοντας HTML, Javascript και διαδοχικά φύλλα στυλ

Είδαμε στις παραπάνω ενότητες ότι τόσο η Javascript, όσο και τα διαδοχικά φύλλα στυλ μπορούν να ενσωματωθούν σε ένα αρχείο HTML είτε απευθείας είτε ως εξωτερικά αρχεία. Σε αυτή την ενότητα θα δούμε ένα ολοκληρωμένο παράδειγμα το οποίο συνδυάζει την HTML, τη Javascript και τα διαδοχικά φύλλα στυλ.

Το ζητούμενο

Θα επιδιώξουμε να δημιουργήσουμε μια ιστοσελίδα με τις εξής προδιαγραφές:

A) Η σελίδα θα πρέπει να περιλαμβάνει μια εικόνα φόντου η οποία να παραμένει ακίνητη όταν κυλάμε τη σελίδα.

B) Η σελίδα θα πρέπει να περιλαμβάνει μια εικόνα και να δίνει τη δυνατότητα, κάνοντας κλικ στην εικόνα, να εμφανίζονται διαδοχικά τρεις διαφορετικές εικόνες.

Γ) Η σελίδα θα πρέπει να δίνει τη δυνατότητα αλλαγής του χρώματος φόντου με την τοποθέτηση του ποντικιού σε ανάλογα χρωματισμένο κελί ενός πίνακα. Συνολικά απαιτούνται 3 διαφορετικά χρώματα φόντου.

Δ) Η σελίδα θα περιλαμβάνει μια φράση η οποία θα κινείται προς τα δεξιά όσο μετακινούμε το ποντίκι πάνω της.

Ανάλυση απαιτήσεων

Ζητούμενο A) Το ζητούμενο μπορεί να υλοποιηθεί χρησιμοποιώντας τις ιδιότητες φόντου των CSS. Επιλέγοντας για ευκολία το εσωτερικό φύλλο στυλ, αρκεί να ενσωματώσουμε στην ιστοσελίδα το παρακάτω τμήμα:

```
<style>
  body
  {
    background-attachment: fixed;
    background-image: url('img0.jpg');
    background-repeat: no-repeat
  }
</style>
```

Ζητούμενο B) Κατ'αρχήν, χρειαζόμαστε ένα πεδίο εικόνας στην ιστοσελίδα. Εφόσον όμως απαιτείται η αλλαγή της εικόνας αυτής με το κλικ του ποντικιού, θα χρειαστεί να δώσουμε στο πεδίο αυτό μια ονομασία ώστε να μπορέσουμε στη συνέχεια με τη βοήθεια του συμβά-

ντος Onclick να καλέσουμε μια συνάρτηση Javascript η οποία θα αλλάζει την εικόνα του πεδίου εκ περιτροπής.

Ο ορισμός του πεδίου εικόνας, ο χαρακτηρισμός του με το όνομα myimage και η σύνδεσή του με την αρχική εικόνα η οποία βρίσκεται στο αρχείο img1.jpg, υλοποιείται με το ακόλουθο τμήμα κώδικα:

```
<p align="center"></p>
```

Παρατηρούμε τη χρήση του χειριστηρίου συμβάντων onclick το οποίο παραπέμπει στη συνάρτηση *changeimage* της Javascript, την οποία θα χρειαστεί να δημιουργήσουμε. Για το σκοπό αυτό, θα χρησιμοποιήσουμε μια μεταβλητή την οποία ονομάζουμε M και της δίνουμε αρχική τιμή 0. Στη συνέχεια, γράφουμε κώδικα ο οποίος κάθε φορά που εκτελείται φορτώνει διαφορετική εικόνα στο πεδίο myimage, ανάλογα με την τιμή της μεταβλητής M, η οποία παίρνει κυκλικά τις τιμές 0, 1, 2. Αυτό γίνεται με τη χρήση της εντολής επιλογών *if... else...* Ο κώδικας της παραπάνω συνάρτησης παρουσιάζεται στο ακόλουθο πλαίσιο.

```
<script language="JavaScript">
  M=0
  function changeimage()
  {
    if (M==0)
    {
      cc=1
      myimage.src="img2.jpg"
    }
    else {
      if (M==1)
      {
        M=2
        myimage.src="img3.jpg"
      }
      else {
        M=0
        myimage.src="img1.jpg"
      }
    }
  }
</script>
```

Ζητούμενο Γ) Κατ' αρχήν χρειαζόμαστε έναν πίνακα με τρία πεδία, το καθένα με διαφορετικό χρώμα, ας υποθέσουμε κόκκινο, μπλε και πράσινο. Στη συνέχεια, σε κάθε πεδίο θα πρέπει με το χειριστήριο συμβάντος *onmouseover* να καλούμε τη Javascript συνάρτηση *bgChange*, την οποία θα χρειαστεί να δημιουργήσουμε και η οποία θα πρέπει να δέχεται ως όρισμα το χρώμα που θέλουμε να παίρνει το φόντο. Ο κώδικας HTML ακολουθεί τη δημιουργία του παραπάνω πίνακα ακολουθεί:

```
<table border="1" width="100%" height="34">
  <tr>
    <td onmouseover="bgChange('red')" onmouseout="bgChange('transparent')"
    bgcolor="red" width="33%" align="center" height="28"></td>
    <td onmouseover="bgChange('blue')" onmouseout="bgChange('transparent')"
    bgcolor="blue" width="33%" align="center" height="28"></td>
    <td onmouseover="bgChange('green')" onmouseout="bgChange('transparent')"
    bgcolor="green" width="34%" align="center" height="28"></td>
  </tr>
</table>
```

Η Javascript συνάρτηση *bgChange* είναι αρκετά απλή, όπως φαίνεται στο παρακάτω πεδίο.

```
function bgChange(bg)
{
    document.body.style.background=bg
}
```

Ζητούμενο Δ) Κατ' αρχήν θα ορίσουμε μια φράση στην οποία θα δώσουμε την ονομασία *Mytext*, συνδέοντας ταυτόχρονα το συμβάν *onmouseover* με τη Javascript συνάρτηση *moveright* την οποία θα κατασκευάσουμε στη συνέχεια. Ο HTML ορισμός της φράσης θα μπορούσε να είναι ο ακόλουθος:

```
<h3 id="Mytext" onmouseover="moveright()">Μετακινείστε το ποντίκι
πάνω απ' αυτό το κείμενο για να μετακινηθεί</h3>
```

Η ακόλουθη Javascript συνάρτηση χρησιμοποιεί τις ιδιότητες *position* και *left* του αντικειμένου *Mytext* που ορίσαμε, αλλάζοντας συνέχεια την τιμή της ιδιότητας *left*, η οποία αυξάνει κατά 1 κάθε φορά που κινούμε το ποντίκι πάνω από τη φράση.

```
var i=1
function moveright()
{
    Mytext.style.position="relative"
    Mytext.style.left=i
```

```
        i++
    }
```

Ο τελικός κώδικας της σελίδας

Συνδυάζοντας όλα τα τμήματα κώδικα των ζητημάτων Α – Δ σε μια λίστα, προκύπτει ο παρακάτω συνολικός κώδικας:

```
<html>

<head>
<title>Συνδυάζοντας HTML, CSS και Javascript</title>

<style>
    body
    {
        background-attachment: fixed;
        background-image: url('img0.jpg');
        background-repeat: no-repeat
    }
</style>

<script language="JavaScript">
    aa=0
    cc=0

    function changeimage()
    {
        if (cc==0)
        {
            cc=1
            myimage.src="img2.jpg"
        }
        else {
            if (cc==1)
            {
                cc=2
                myimage.src="img3.jpg"
            }
            else {
```

```

        cc=0
        myimage.src="img1.jpg"
    }

    }
}
function bgChange(bg)
{
    document.body.style.background=bg
}
var i=1
function moveright()
{
    Mytext.style.position="relative"
    Mytext.style.left=i

        i++
}
</script>

<p align="center">&nbsp;</p>
<p align="center">H<br>
    εικόνα φόντου (πάνω αριστερά) που περιέχει αυτή η<br>
σελίδα δεν μετακινείται όταν κυλάμε τη σελίδα</p>
<p align="center">&nbsp;</p>
<p align="center">Κάνοντας κλικ σε<br>
αυτή την εικόνα εμφανίζονται διαδοχικά<br>
διαφορετικές εικόνες</p>
<p align="center">&nbsp;</p>
<p align="center"></p>

<p>&nbsp;</p>
<p align="center">Κινώντας το ποντίκι<br>
στα κελιά του πίνακα αλλάζει το Χρώμα<br>
Φόντου</p>
<table border="1" width="100%" height="34">
<tr>
<td onmouseover="bgChange('red')" onmouseout="bgChange('transparent')"
bgcolor="red"
width="33%" align="center" height="28"></td>

```



```

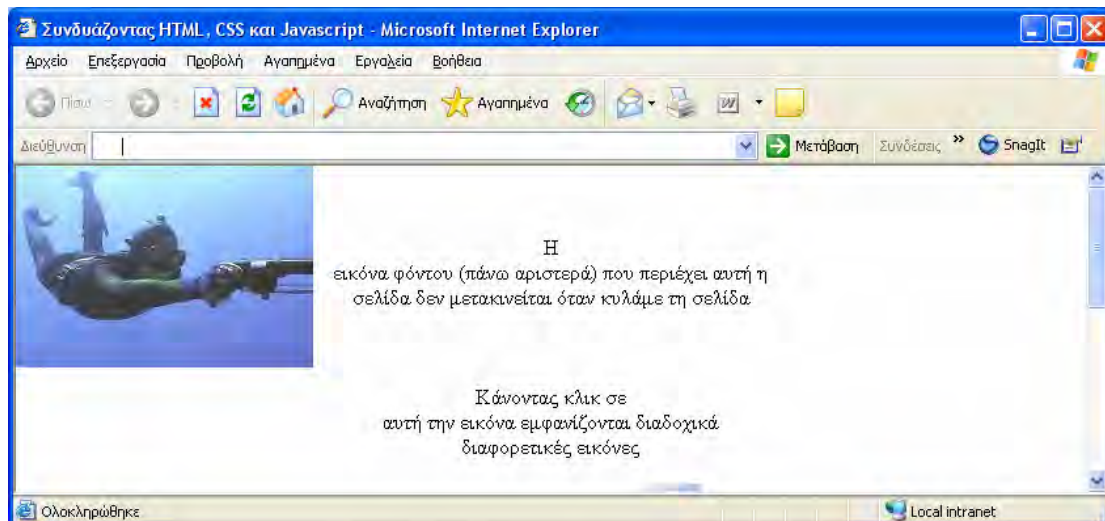
<td onmouseover="bgChange('blue')" onmouseout="bgChange('transparent')"
bgcolor="blue"
width="33%" align="center" height="28"></td>
<td onmouseover="bgChange('green')" onmouseout="bgChange('transparent')"
bgcolor="green"
width="34%" align="center" height="28"></td>
</tr>
</table>

<h3 id="Mytext" onmousemove="moveright()">Μετακινείστε το ποντίκι
πάνω απ' αυτό το κείμενο για να μετακινηθεί</h3>

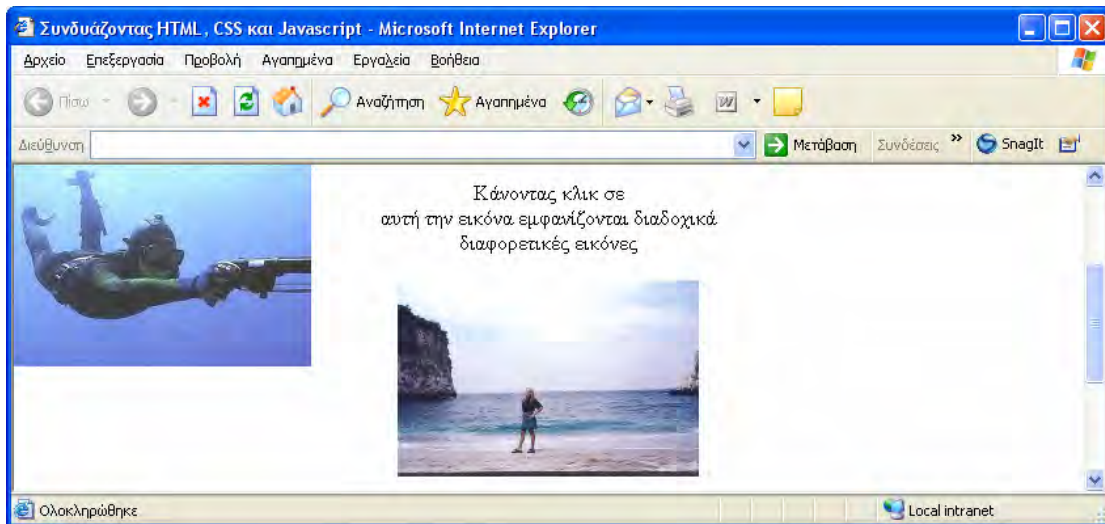
</body>
</html>

```

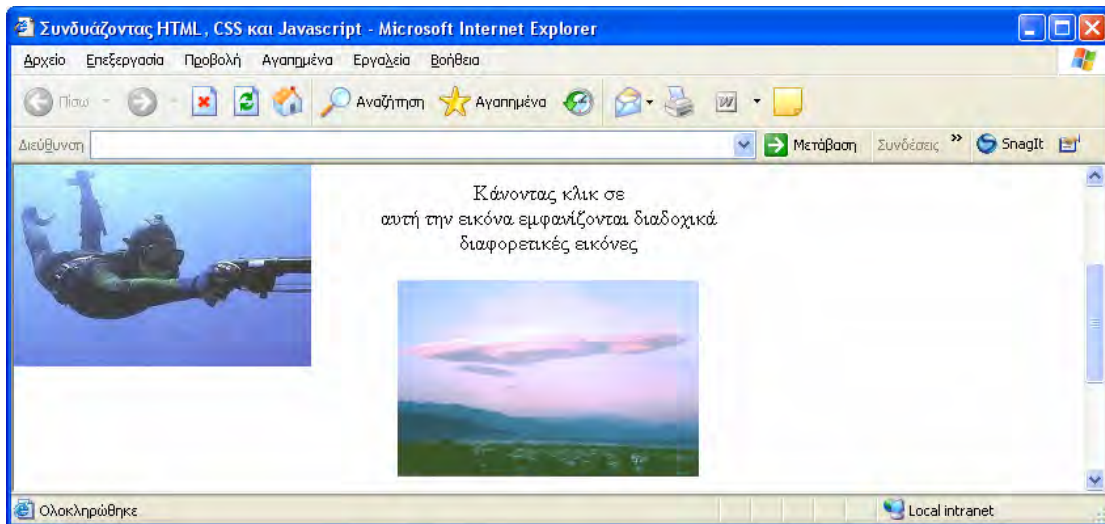
Σώζοντας τον παραπάνω κώδικα ως αρχείο .htm στον ίδιο φάκελο με τις εικόνες img0.jpg, img1.jpg, img2.jpg και img3.jpg, μπορούμε να πειραματιστούμε με τις λειτουργίες της σελίδας μας χρησιμοποιώντας τον Internet Explorer. Χαρακτηριστικά παραθέτουμε τα ακόλουθα στιγμιότυπα.



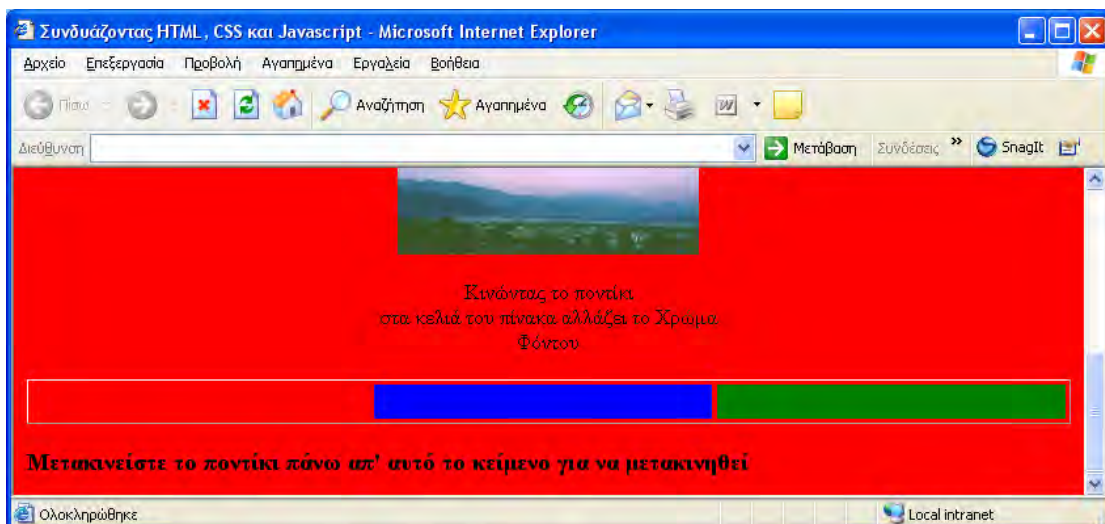
Σχήμα 2.11(α)



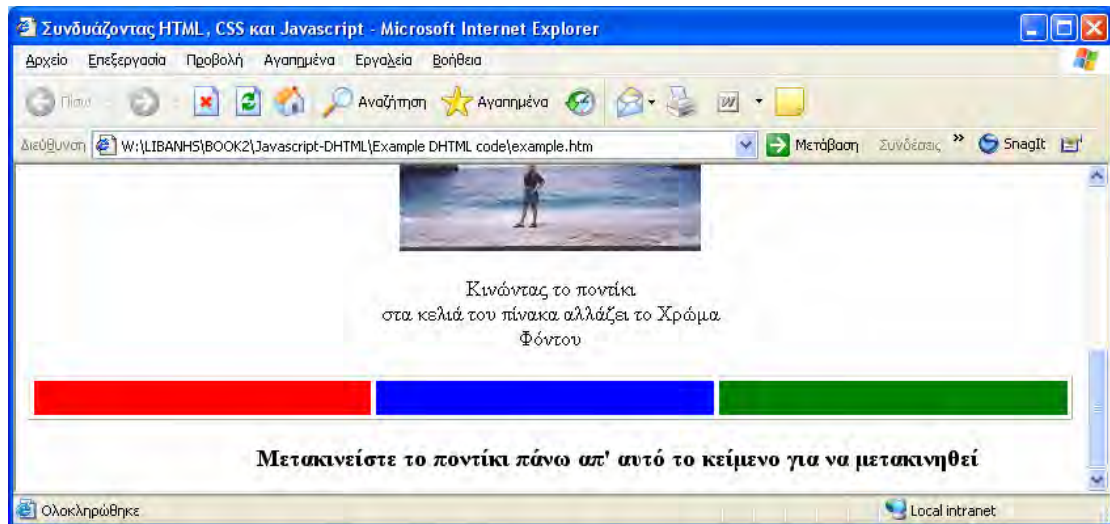
Σχήμα 2.11(β)



Σχήμα 2.11(γ)



Σχήμα 2.11(δ)



Σχήμα 2.11(ε)

Ενότητα 2.5 Δραστηριότητες ανακεφαλαίωσης

A) Ποια είναι τα πλεονεκτήματα και ποια τα μειονεκτήματα των γλωσσών σεναρίων από την πλευρά του πελάτη;

B) Ποιές είναι οι συνήθειες εφαρμογές των γλωσσών σεναρίων από την πλευρά του πελάτη;

Γ) Δίνεται το παρακάτω τμήμα σεναρίου Javascript

```
A=5;  
B=A+3  
If (B>=A){  
    C=A+B  
} else{  
    C=A-B  
}
```

Ποια θα είναι η τιμή της μεταβλητής C μετά την εκτέλεση του σεναρίου;

Δ) Με ποιες ετικέτες της γλώσσας HTML προστίθενται σεναρία Javascript στον κώδικα μιας ιστοσελίδας;

E) Επιλέξτε τη σωστή απάντηση

Το DOM είναι ένα μοντέλο α) Αναπαράστασης εγγράφων στη μνήμη του Η/Υ β) Παρουσίασης πληροφοριών στο φυλλομετρητή γ) Γλώσσας προγραμματισμού από την πλευρά του πελάτη	Η DHTML είναι α) Μια γλώσσα προγραμματισμού από την πλευρά του πελάτη β) ένας συνδυασμός τεχνικών παρουσίασης και δυναμικού προγραμματισμού ιστοσελίδων γ) Μια εξελιγμένη έκδοση της γλώσσας HTML.
--	---

ΣΤ) Χρησιμοποιώντας HTML και Javascript δημιουργήστε μια ιστοσελίδα με τίτλο «ΔΕΙΚΤΗΣ ΜΑΖΑΣ ΣΩΜΑΤΟΣ» η οποία θα δέχεται σε δύο διαφορετικά πλαίσια κειμένου το ύψος και το βάρος ενός ατόμου και με την πίεση ενός button θα υπολογίζει και θα εμφανίζει σε ένα τρίτο πεδίο κειμένου το δείκτη μάζας σώματος, ο οποίος προκύπτει από τη σχέση:

$$\text{Δείκτης Μάζας Σώματος} = \text{Βάρος} / \text{Υψος}^2$$

Z) Δημιουργήστε ένα έγγραφο HTML, το οποίο θα πρέπει να περιέχει κατάλληλες εντολές Javascript ώστε με το άνοιγμά του να ανοίγει και ένα δεύτερο παράθυρο στο οποίο να εμφανίζεται η ιστοσελίδα της προτίμησής σας.

H) Δημιουργήστε μια σελίδα HTML η οποία θα πρέπει να περιέχει 2 πεδία κειμένου και μια συνάρτηση Javascript, η οποία θα πρέπει να δέχεται ένα αριθμό (αντιστοιχεί σε δραχμές) και να επιστρέφει την αντίστοιχη τιμή σε € (1€=340.75 δρχ.). Στο πρώτο από τα παραπάνω πεδία κειμένου θα πρέπει να πληκτρολογείται το ποσό σε € και αυτόματα να εμφανίζεται το ποσό σε δραχμές στο δεύτερο πεδίο κειμένου.

ΚΕΦΑΛΑΙΟ 3ο



**Προγραμματισμός από
την Πλευρά του
Εξυπηρετητή: PHP**

Δυναμικές Εφαρμογές Παγκόσμιου Ιστού

Σκοπός και στόχοι του κεφαλαίου

Σκοπός του κεφαλαίου είναι η εισαγωγή των ενηλίκων στον προγραμματισμό δυναμικών εφαρμογών ιστοσελίδων, από την πλευρά του εξυπηρετητή, σε περιβάλλον PHP & MySQL.

Μετά την ολοκλήρωση του κεφαλαίου, οι μαθητές θα είναι σε θέση να:

- κατανοούν τη φιλοσοφία του προγραμματισμού από την πλευρά του εξυπηρετητή
- εντοπίζουν τις δυνατότητες και τα πλεονεκτήματά του, τόσο σε σχέση με τις στατικές εφαρμογές HTML, όσο και με τις εφαρμογές προγραμματισμού από την πλευρά του πελάτη.
- κατανοούν τις βασικές αρχές και λειτουργίες της βάσης δεδομένων MySQL.
- δημιουργούν βάσεις δεδομένων σε περιβάλλον MySQL και να εκτελούν στοιχειώδεις λειτουργίες διαχείρισης, όπως προσθήκη, τροποποίηση και διαγραφή εγγραφών.
- κατανοούν τις βασικές αρχές της γλώσσας προγραμματισμού PHP.
- δημιουργούν απλές εφαρμογές δυναμικού προγραμματισμού από την πλευρά του εξυπηρετητή χρησιμοποιώντας την PHP.

Εννοιες-Κλειδιά

Προγραμματισμός από την πλευρά του εξυπηρετητή, βάσεις δεδομένων, MySQL, PHP

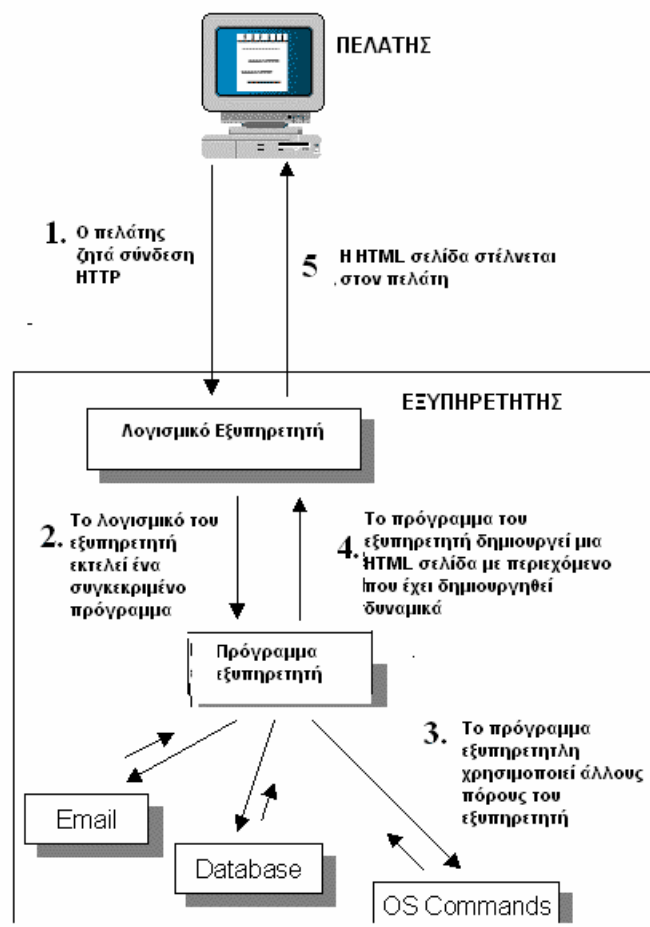
Εισαγωγικές Παρατηρήσεις

Ο προγραμματισμός από την πλευρά του πελάτη έχει από τη φύση του περιορισμένες δυνατότητες εφαρμογής σε συγκεκριμένους τομείς, ενώ η υλοποίηση σχετικών λύσεων συχνά καθίσταται προβληματική, τόσο λόγω ασυμβατότητας των φυλλομετρητών, όσο και λόγω συγκεκριμένων ρυθμίσεων που ενδέχεται να έχει κάνει ο χρήστης, οι οποίες να μην επιτρέπουν την εκτέλεση των σεναρίων. Τους παραπάνω περιορισμούς έρχεται να αντιμετωπίσει ο προγραμματισμός από την πλευρά του εξυπηρετητή, ο οποίος παρέχει σημαντικά πλεονεκτήματα. Στα πιο δημοφιλή περιβάλλοντα προγραμματισμού από την πλευρά του εξυπηρετητή συγκαταλέγονται ο συνδυασμός της βάσης δεδομένων MySQL και της γλώσσα προγραμματισμού PHP. . Τόσο η MySQL όσο και η PHP διαθέτουν ανοικτό κώδικα και μπορούν να εγκατασταθούν πρακτικά σε όλους τους τύπους εξυπηρετητών, ενώ η χρήση τους στις δυναμικές εφαρμογές είναι δωρεάν.

Ενότητα 3.1 Αρχιτεκτονική εφαρμογών από την πλευρά του εξυπηρετητή

Ο προγραμματισμός από την πλευρά του πελάτη, ο οποίος αναλύθηκε στο προηγούμενο κεφάλαιο, συνείσφερε σημαντικά στη βελτίωση της διαδραστικότητας των ιστοσελίδων και στην παράκαμψη της στατικότητας του αμιγώς HTML περιεχομένου. Παρόλ' αυτά, ο προγραμματισμός από την πλευρά του πελάτη έχει από τη φύση του περιορισμένες δυνατότητες εφαρμογής σε συγκεκριμένους τομείς όπως η πρόσβαση σε βάσεις δεδομένων και η αποστολή ή λήψη ηλεκτρονικού ταχυδρομείου. Επίσης, η υλοποίηση λύσεων με προγραμματισμό από την πλευρά του πελάτη συχνά καθίσταται προβληματική λόγω ασυμβατότητας των φυλλομετρητών, αλλά και λόγω συγκεκριμένων ρυθμίσεων που ενδέχεται να έχει κάνει ο χρήστης, οι οποίες να μην επιτρέπουν την εκτέλεση των σχετικών σεναρίων, όπως η απενεργοποίηση από το χρήστη της εκτέλεσης των σεναρίων JavaScript (ρυθμίσεις ασφαλείας του Internet Explorer).

Τους παραπάνω περιορισμούς έρχεται να αντιμετωπίσει ο προγραμματισμός από την πλευρά του εξυπηρετητή, η φιλοσοφία του οποίου απεικονίζεται στο Σχήμα 3.1:



Σχήμα 3.1 Η φιλοσοφία του προγραμματισμού από την πλευρά του εξυπηρετητή

Σύμφωνα με το παραπάνω σχήμα, η πρόσβαση σε δυναμικό περιεχόμενο που παρέχεται από την πλευρά του εξυπηρετητή, είναι μια ακολουθία των παρακάτω διακριτών βημάτων:

Βήμα 1^ο : Ο πελάτης ζητά σύνδεση HTTP με τον εξυπηρετητή.

Βήμα 2^ο : Ο εξυπηρετητής εκτελεί ένα συγκεκριμένο πρόγραμμα.

Βήμα 3^ο : Το πρόγραμμα του εξυπηρετητή χρησιμοποιεί άλλους πόρους του εξυπηρετητή, όπως εφαρμογές email, βάσεις δεδομένων (Databases) και εντολές του λειτουργικού συστήματος (OS Commands).

Βήμα 4^ο : Το πρόγραμμα του εξυπηρετητή ενσωματώνει τα αποτελέσματα της εκτέλεσής του σε μια σελίδα HTML την οποία επιστρέφει στον πελάτη. Τονίζεται ότι ο πελάτης λαμβάνει το αποτέλεσμα της εκτέλεσης του προγράμματος του εξυπηρετητή και όχι το πρόγραμμα αυτό καθαυτό.

Σε σχέση με τις εφαρμογές που τρέχουν από την πλευρά του πελάτη, οι εφαρμογές του εξυπηρετητή παρέχουν πλεονεκτήματα στα οποία περιλαμβάνονται:

- Μειωμένο κόστος της επέκτασης ανά χρήστη. Οι τελικοί χρήστες μιας εφαρμογής εξυπηρετητή μπορούν να τρέξουν την εφαρμογή χρησιμοποιώντας μόνο ένα φυλλομετρητή, ενώ κανένα ειδικό λογισμικό δεν πρέπει να εγκατασταθεί στους υπολογιστές τους για την εφαρμογή για να εργαστεί.
- Πρόσβαση σε ένα ευρύ ακροατήριο. Οι εφαρμογές εξυπηρετητή λειτουργούν με μια ευρεία ποικιλία των φυλλομετρητών και των λειτουργικών συστημάτων και έτσι απευθύνονται σε ένα ευρύ ακροατήριο.
- Εύκολη διαχείριση βάσης δεδομένων από τον πελάτη, εφόσον η βάση ανοίγεται μόνο από τον εξυπηρετητή.

Οι εφαρμογές από την πλευρά του εξυπηρετητή παρέχουν σημαντικές προγραμματιστικές δυνατότητες, στις οποίες περιλαμβάνονται η δυναμική τροποποίηση του περιεχομένου των ιστοσελίδων, η αλληλεπίδραση με το χρήστη ως αποτέλεσμα επεξεργασίας δεδομένων που υποβάλλονται από φόρμες HTML και η πρόσβαση σε βάσεις δεδομένων και επιστροφή αποτελεσμάτων στο φυλλομετρητή του χρήστη.

Στη συνέχεια αυτού του κεφαλαίου θα ασχοληθούμε αρχικά με τη MySQL, η οποία είναι μια από τις πιο δημοφιλείς εφαρμογές βάσεων δεδομένων και χρησιμοποιείται ευρέως στον προγραμματισμό από την πλευρά του εξυπηρετητή. Έπειτα, θα παρουσιάσουμε τη γλώσσα προγραμματισμού PHP, η οποία επίσης χρησιμοποιείται ευρέως στον προγραμματισμό από την πλευρά του εξυπηρετητή, συχνά συνεργαζόμενη με βάσεις δεδομένων MySQL. Εναλλακτικά προγραμματιστικά περιβάλλοντα από την πλευρά του εξυπηρετητή, όπως είναι οι γλώσσες Active Server Pages (ASP) και Java Server Pages (JSP), αποτελούν αντικείμενο του επόμενου κεφαλαίου.

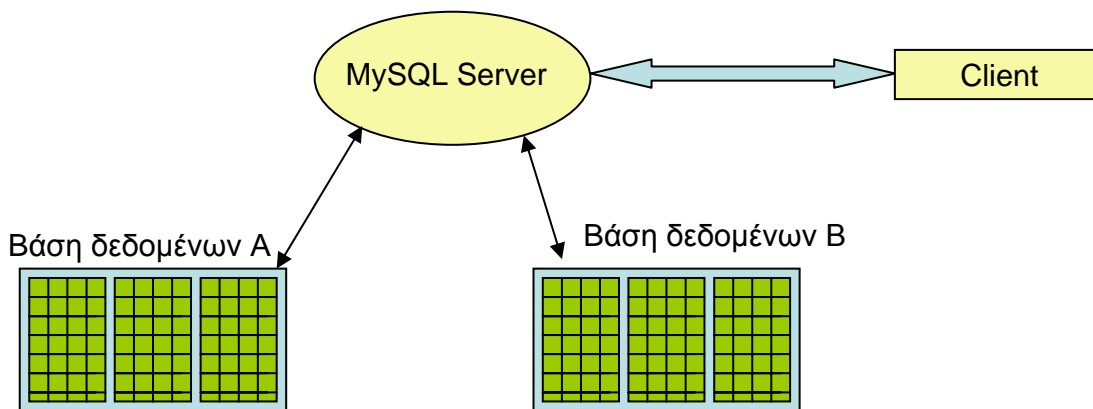
Ενότητα 3.2 Διαχείριση Βάσεων Δεδομένων σε περιβάλλον MySQL

3.2.1 Εισαγωγή

Η MySQL είναι ίσως η δημοφιλέστερη ανοικτή εφαρμογή διαχείρισης βάσεων δεδομένων (Database). Η MySQL αναπτύσσεται, διανέμεται, και υποστηρίζεται από την εταιρεία MySQL AB (<http://www.mysql.com>), η οποία είναι μια εμπορική επιχείρηση ανάπτυξης ανοικτού λογισμικού. Ανοικτό λογισμικό, σημαίνει ότι είναι δυνατό για οποιονδήποτε να χρησιμοποιήσει και να τροποποιήσει το λογισμικό, χωρίς καμία οικονομική επιβάρυνση. Το λογισμικό MySQL υπόκειται στους περιορισμούς της άδειας ευρέος κοινού (General Public License – GPL), ενώ σχετική άδεια απαιτείται για τη χρήση του σε εμπορικές εφαρμογές, οι όροι της οποίας περιγράφονται στη διεύθυνση <http://www.mysql.com/company/legal/licensing/>

3.2.2 Βασικά χαρακτηριστικά της MySQL

Η MySQL λειτουργεί με το μοντέλο Πελάτη/Εξυπηρετητή (Client/Server). Ο εξυπηρετητής δέχεται αιτήματα από τους πελάτες σχετικά με τη διαχείριση μίας ή περισσότερων βάσεων δεδομένων και προβαίνει στις απαιτούμενες ενέργειες, όπως φαίνεται στο Σχήμα 3.2.



Σχήμα 3.2 Το μοντέλο Πελάτη/Εξυπηρετητή της MySQL

Οι κύριες λειτουργίες διαχείρισης βάσεων δεδομένων τις οποίες υλοποιεί ο εξυπηρετητής κατόπιν αιτήσεως από τον πελάτη, δε διαφέρουν σημαντικά από τις αντίστοιχες λειτουργίες των άλλων εφαρμογών βάσεων δεδομένων και περιλαμβάνουν τη δημιουργία-διαγραφή βάσεων δεδομένων, την εισαγωγή-τροποποίηση-διαγραφή πινάκων (tables) και πεδίων (fields), την εισαγωγή-τροποποίηση-διαγραφή εγγραφών (records) και τέλος την ανάκτηση δεδομένων από τη βάση με τη χρήση συγκεκριμένων κριτηρίων.

Για την υλοποίηση των παραπάνω λειτουργιών, η MySQL χρησιμοποιεί την SQL (Δομημένη Γλώσσα Ερωτήσεων – Structured Query Language). Η SQL είναι η ευρύτερα χρησιμοποιού-

μενη τυποποιημένη γλώσσα πρόσβασης στις βάσεις δεδομένων και αποτελείται από εντολές οι οποίες επιτρέπουν την ανάκτηση και ενημέρωση δεδομένων σε μια βάση. Σημειώνουμε ότι, εκτός από τη MySQL, η γλώσσα SQL συνεργάζεται με άλλα προγράμματα βάσεων δεδομένων όπως είναι η Access, Microsoft SQL Server, Oracle, Sybase και πολλά άλλα.

3.2.3 Οι βάσεις δεδομένων της MySQL

Μια βάση δεδομένων της MySQL αποτελείται από έναν ή περισσότερους πίνακες, ο καθένας από τους οποίους περιέχει μια λίστα από δεδομένα. Αν, για παράδειγμα, θελήσουμε να δημιουργήσουμε μια βάση δεδομένων για να αποθηκεύσουμε έναν τηλεφωνικό κατάλογο, θα μπορούσαμε να κατασκευάσουμε έναν πίνακα με το όνομα *ΕΠΑΦΕΣ*. Η μορφή που θα μπορούσε να έχει αυτός ο πίνακας φαίνεται στο σχήμα 3.3:

<i>ΕΠΑΦΕΣ</i>			
ID	Επώνυμο	Όνομα	Τηλέφωνο
1	Αντωνιάδης	Γεώργιος	2101234567
2	Παπαδόπουλος	Σωκράτης	2101288888

Σχήμα 3.3 Παράδειγμα μορφής πίνακα βάσης δεδομένων

Όπως φαίνεται από το παραπάνω παράδειγμα, ο κάθε πίνακας σε μια βάση δεδομένων περιέχει μια ή περισσότερες *στήλες* (*columns*) ή *πεδία* (*fields*), όπου η κάθε στήλη περιέχει μια συγκεκριμένη πληροφορία για την κάθε εγγραφή που καταχωρείται στη βάση δεδομένων. Παρατηρούμε ότι, εκτός από τις στήλες για το όνομα, επώνυμο και τηλέφωνο του ατόμου που καταχωρούμε, υπάρχει και μια στήλη με όνομα *ID*, ο σκοπός της οποίας είναι να εκχωρήσει έναν μοναδικό αριθμό στο κάθε άτομο, έτσι ώστε να έχουμε έναν εύκολο τρόπο αναφοράς σ' αυτό και να μπορούμε να το ξεχωρίσουμε από τα άλλα άτομα που είναι καταχωρημένα στον πίνακα.

3.2.4 Η γλώσσα SQL

Το σύνολο των εντολών που θα χρησιμοποιήσουμε στη συνέχεια για την αλληλεπίδραση με τη MySQL, αποτελεί μέρος ενός προτύπου το οποίο ονομάζεται *Δομημένη Γλώσσα Ερωτημάτων* (*Structured Query Language*) ή *SQL*. Οι εντολές της SQL αποκαλούνται επίσης και *ερωτήματα* (*queries*).

Η SQL αποτελεί την ευρύτερα διαδεδομένη γλώσσα για αλληλεπίδραση με τις περισσότερες βάσεις δεδομένων. Δεν πρέπει να συγχέουμε την SQL με την MySQL. Η MySQL είναι το λογισμικό του *διακομιστή βάσεων δεδομένων* (*database server software*) που χρησιμοποιούμε, ενώ η SQL είναι η γλώσσα που χρησιμοποιούμε για τη διαχείριση της βάσης δεδομένων.

3.2.5 Βασικές λειτουργίες διαχείρισης βάσης δεδομένων με τη MySQL

Έχοντας δει τη δομή και τα βασικά χαρακτηριστικά της MySQL, θα προχωρήσουμε στη συνέχεια, με τη βοήθεια παραδειγμάτων, στην εξοικείωση με τις βασικές λειτουργίες της MySQL, όπως είναι η δημιουργία και διαγραφή βάσεων δεδομένων, και η δημιουργία, ενημέρωση και διαγραφή πινάκων. Απαραίτητη προϋπόθεση για να μπορέσουμε να υλοποιήσουμε τα παραδείγματα που ακολουθούν είναι η πρόσβαση σε κάποιο MySQL Server. Αυτό μπορεί να γίνει απλά με το κατέβασμα από τη διεύθυνση <http://www.mysql.com/> και εγκατάσταση τόσο του λογισμικού MySQL Server όσο και του MySQL client στον υπολογιστή μας.

Όποια μέθοδο κι αν επιλέξουμε και όποιο λειτουργικό σύστημα και αν χρησιμοποιούμε, θα καταλήξουμε σε μια γραμμή εντολών (*command line*), έτοιμοι να εκτελέσουμε το πρόγραμμα πελάτη της MySQL για να συνδεθούμε στον MySQL server. Η σχετική εντολή είναι η εξής:

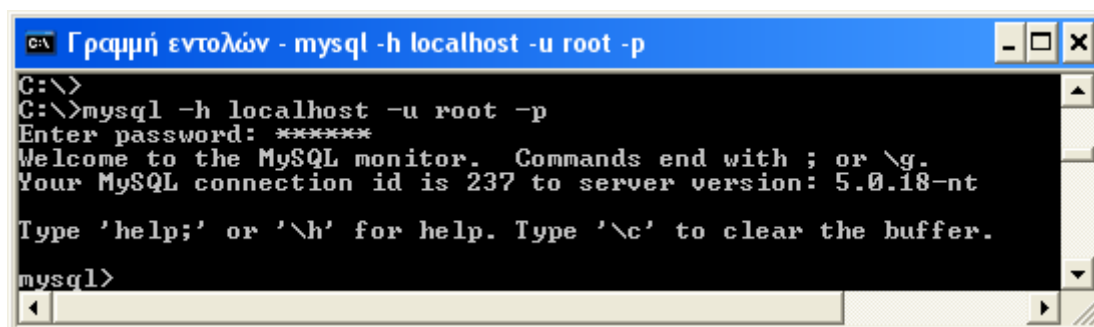
```
mysql -h <hostname> -u <username> -p
```

Θα πρέπει να αντικαταστήσουμε το *<hostname>* με το όνομα ή την IP διεύθυνση του υπολογιστή στον οποίο εκτελείται ο MySQL server. Για παράδειγμα αν εκτελούμε το πρόγραμμα πελάτη στον ίδιο υπολογιστή με τον MySQL server, μπορούμε να παραλείψουμε το τμήμα *-h <hostname>* της εντολής αντί να γράψουμε *-h localhost*. Το *<username>* πρέπει να είναι το δικό μας όνομα χρήστη στην MySQL.

Αν εγκαταστήσαμε εμείς οι ίδιοι τον MySQL server, αυτό θα είναι το *root*, ενώ αν χρησιμοποιούμε τον MySQL server του Web host που μας φιλοξενεί, αυτό θα πρέπει να είναι το όνομα χρήστη της MySQL που μας έχει δοθεί.

Το όρισμα *-p* λέει στο πρόγραμμα να ζητήσει από εμάς τον κωδικό εισόδου (password), κάτι που θα συμβεί μόλις δώσουμε την παραπάνω εντολή.

Αν τα γράψαμε όλα σωστά, το πρόγραμμα πελάτη της MySQL θα παρουσιάσει τον εαυτό του και θα εμφανίσει τη γραμμή εντολών της MySQL, ως εξής:



```
C:\>mysql -h localhost -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 237 to server version: 5.0.18-nt

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

mysql>
```

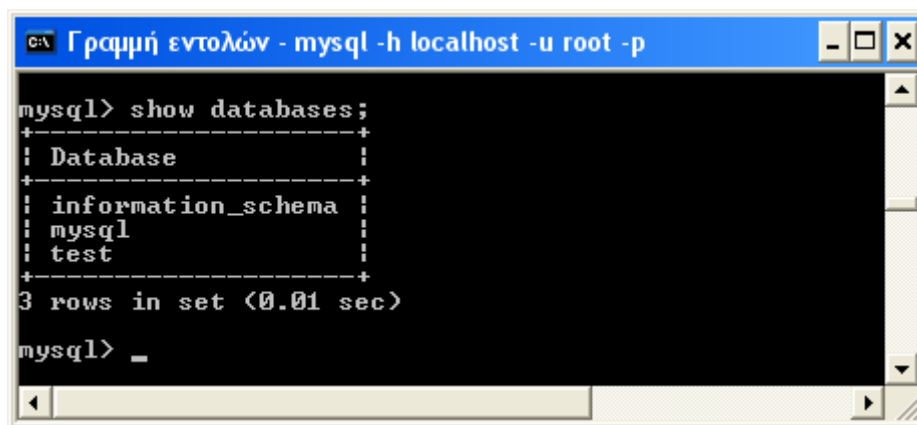
Σχήμα 3.4 Εμφάνιση της γραμμής εντολών της MySQL

Πρόσβαση, δημιουργία και διαγραφή σε βάσεις δεδομένων MySQL

Εχοντας αποκτήσει πρόσβαση στο MySQL Server, το επόμενο βήμα είναι η εξοικείωσή μας με βασικές εντολές πρόσβασης σε βάσεις δεδομένων του MySQL Server. Πρώτα απ' όλα, θα δούμε μια λίστα των βάσεων δεδομένων που υπάρχουν στον τρέχοντα server. Δίνουμε την επόμενη εντολή και μετά πατάμε ENTER:

```
mysql> SHOW DATABASES;
```

Η MySQL θα εμφανίσει μια λίστα με τις βάσεις δεδομένων που υπάρχουν στον server, ως εξής:



```
C:\ Γραμμή εντολών - mysql -h localhost -u root -p
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql      |
| test      |
+-----+
3 rows in set (0.01 sec)

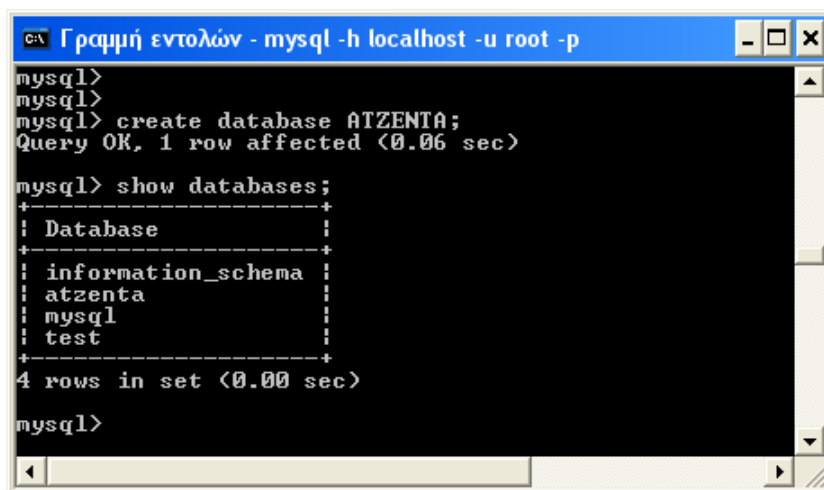
mysql> _
```

Σχήμα 3.5 Εμφάνιση λίστας βάσεων δεδομένων της MySQL

Στο παραπάνω παράδειγμα, ο MySQL Server διαχειρίζεται τρεις βάσεις δεδομένων, εκ των οποίων οι *mysql* και *test* δημιουργούνται με την εγκατάσταση του Server. Η βάση *mysql* χρησιμοποιείται από το MySQL Server για τη διαχείριση των χρηστών (δικαιώματα πρόσβασης, κωδικοί κλπ), ενώ η βάση δεδομένων *test* αποτελεί ένα προκατασκευασμένο δείγμα βάσης δεδομένων.

Δημιουργία μιας Βάσης Δεδομένων (DataBase)

Η δημιουργία μιας νέας βάσης δεδομένων είναι εύκολη και χρησιμοποιεί την εντολή `CREATE DATABASE database_name`. Θα χρησιμοποιήσουμε αυτή την εντολή για να δημιουργήσουμε μια νέα βάση δεδομένων με το όνομα *ATZENTA*, όπως φαίνεται στο ακόλουθο Σχήμα:



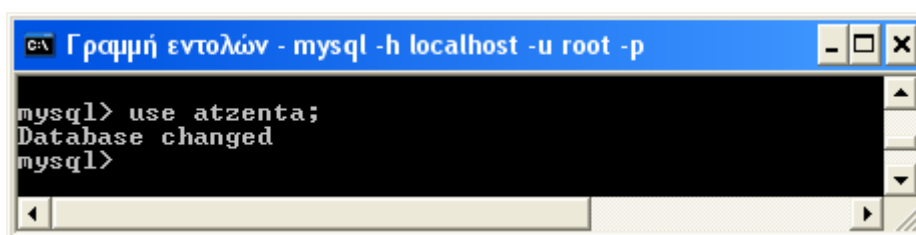
```
C:\ Γραμμή εντολών - mysql -h localhost -u root -p
mysql>
mysql>
mysql> create database ATZENTA;
Query OK, 1 row affected (0.06 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| atzenta |
| mysql |
| test |
+-----+
4 rows in set (0.00 sec)

mysql>
```

Σχήμα 3.6 Δημιουργία της βάσεως δεδομένων ATZENTA

Για να μπορέσουμε να χρησιμοποιήσουμε μια βάση δεδομένων, θα πρέπει να πληκτρολογήσουμε την εντολή *USE database*;



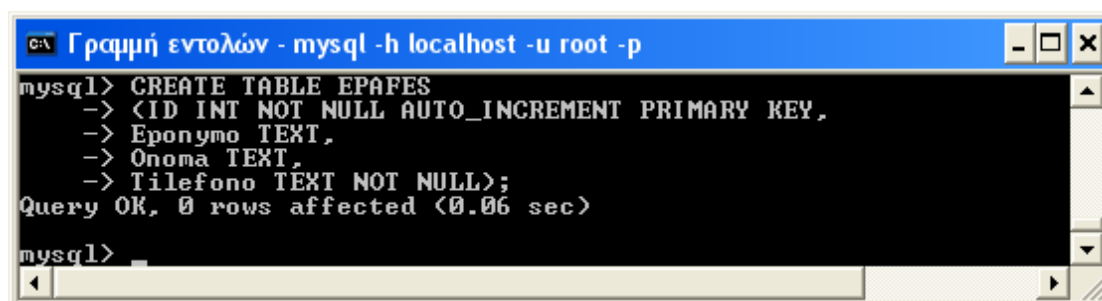
```
C:\ Γραμμή εντολών - mysql -h localhost -u root -p
mysql> use atzenta;
Database changed
mysql>
```

Σχήμα 3.7 Εναρξη χρήσης της βάσεως δεδομένων ATZENTA

Μπορούμε τώρα να αρχίσουμε να χρησιμοποιούμε τη βάση δεδομένων και να δημιουργήσουμε τον πρώτο μας πίνακα.

Δημιουργία ενός Πίνακα

Η σύνταξη της εντολής SQL για τη δημιουργία ενός πίνακα (table) είναι η *CREATE TABLE*. Η χρήση αυτής της εντολής για τη δημιουργία του πίνακα EPAFES, φαίνεται στο παρακάτω στιγμιότυπο:



```
C:\ Γραμμή εντολών - mysql -h localhost -u root -p
mysql> CREATE TABLE EPAFES
-> (ID INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
-> Eponymo TEXT,
-> Onoma TEXT,
-> Tilefono TEXT NOT NULL);
Query OK, 0 rows affected (0.06 sec)

mysql>
```

Σχήμα 3.8 Δημιουργία πίνακα σε MySQL

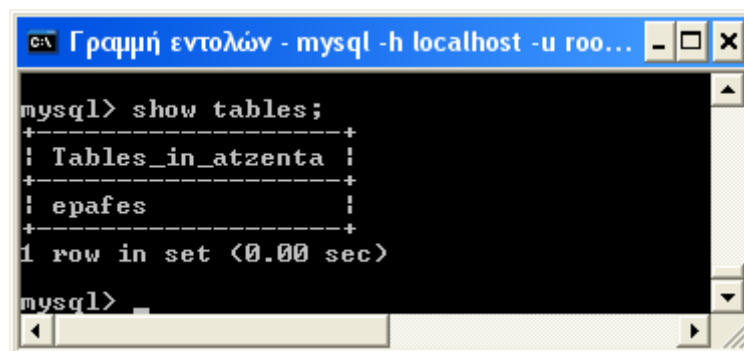
Η εντολή που πληκτρολογήσαμε παραπάνω αναλύεται ως εξής:

- Η πρώτη γραμμή είναι αρκετά απλή και ορίζει το όνομα του νέου πίνακα.
- Η δεύτερη γραμμή ορίζει στον πίνακα μια στήλη (column) με όνομα *ID* που θα περιέχει μια ακέραια τιμή (integer, *INT*). Ακόμη, αυτή η στήλη δεν μπορεί να είναι κενή (*NOT NULL*). Επίσης, αν δεν καθορίσουμε κάποια συγκεκριμένη τιμή, όταν κάνουμε μια νέα καταχώρηση, η MySQL θα επιλέξει η ίδια μια τιμή που θα είναι κατά ένα, μεγαλύτερη από τη μεγαλύτερη τιμή του πίνακα μέχρι τώρα (*AUTO_INCREMENT*). Τέλος, αυτή η στήλη θα ενεργεί σαν ένα μοναδικό αναγνωριστικό (unique identifier) για τις καταχωρήσεις του πίνακα, έτσι όλες οι τιμές αυτής της στήλης θα πρέπει να είναι μοναδικές (*PRIMARY KEY*).
- Η τρίτη και τέταρτη γραμμή είναι πολύ απλές: λένε ότι θέλουμε μία στήλη με όνομα *Eponymo* και μία με το όνομα *Onoma* αντίστοιχα, που θα περιέχουν κείμενο (*TEXT*).
- Η τέταρτη γραμμή ορίζει την τελευταία στήλη, με όνομα *Telefono*, η οποία θα περιέχει δεδομένα κειμένου και δεν θα μπορεί να είναι κενή (*NOT NULL*).

Για να βεβαιωθούμε ότι πράγματι δημιουργήθηκε ο πίνακας, δίνουμε την εξής εντολή:

```
mysql> SHOW TABLES;
```

Η απάντηση θα είναι ως εξής:



```
C:\> Γραμμή εντολών - mysql -h localhost -u roo...
mysql> show tables;
+-----+
| Tables_in_atzenta |
+-----+
| epafes            |
+-----+
1 row in set (0.00 sec)
mysql>
```

Σχήμα 3.9 Εκτύπωση λίστας διαθέσιμων πινάκων

Αυτή είναι η λίστα όλων των πινάκων που υπάρχουν στη βάση δεδομένων *ATZENTA* και περιέχει μόνο τον πίνακα *EPAFES*. Μπορούμε να δούμε αναλυτικά τη δομή του πίνακα που δημιουργήσαμε με την εντολή *DESCRIBE*, ως εξής:

```

c:\ Γραμμή εντολών - mysql -h localhost -u root -p
mysql>
mysql>
mysql> describe EPAPES;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra      |
+-----+-----+-----+-----+-----+-----+
| ID         | int(11)   | NO   | PRI | NULL     | auto_increment |
| Eponymo    | text      | YES  |     | NULL     |              |
| Onoma      | text      | YES  |     | NULL     |              |
| Tilefono   | text      | NO   |     |          |              |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>

```

Σχήμα 3.10 Εμφάνιση της δομής επιλεγμένου πίνακα

Η παραπάνω εντολή εμφανίζει μια λίστα των στηλών (πεδίων) που υπάρχουν στον πίνακα. Όπως μπορούμε να δούμε, υπάρχουν τέσσερις στήλες σ' αυτόν τον πίνακα που εμφανίζονται σαν τέσσερις γραμμές στο αποτέλεσμα.

Εισαγωγή Δεδομένων σε Πίνακα

Η εντολή για να εισάγουμε δεδομένα σε μια βάση δεδομένων αποκαλείται **INSERT** και υπάρχουν οι εξής δύο βασικές μορφές αυτής της εντολής:

1^η μορφή εντολής INSERT

```

mysql> INSERT INTO <table name> SET

-> columnName1 = value1,

-> columnName2 = value2,

-> ...

-> ;

```

και

2^η μορφή εντολής INSERT

```

mysql> INSERT INTO <table name>

-> (columnName1, columnName2, ...)

-> VALUES (value1, value2, ...);

```

Έτσι, για να προσθέσουμε μια καταχώρηση στον πίνακα *Erafes* που δημιουργήσαμε, μπορούμε να επιλέξουμε μία από τις εξής εντολές :

```
mysql> INSERT INTO Erafes SET
-> Eponimo = "Αλεξόπουλος",
-> Onoma = "Γεώργιος",
-> Tilefono = "210121315";
```

ή

```
mysql> INSERT INTO Erafes
-> (Eponimo, Onoma, Tilefono) VALUES (
-> " Αλεξόπουλος",
-> "Γεώργιος",
-> "210121315");
```

Πρέπει να έχουμε υπόψη μας ότι στη δεύτερη μορφή της εντολής *INSERT*, η σειρά με την οποία γράφουμε τις στήλες πρέπει να ταιριάζει με τη σειρά που γράφουμε τις αντίστοιχες τιμές.

Εμφάνιση των Αποθηκευμένων Δεδομένων

Η εντολή για να δούμε τα δεδομένα που είναι αποθηκευμένα στους πίνακες μιας βάσης δεδομένων είναι η *SELECT*, η οποία είναι η συχνότερα χρησιμοποιούμενη, αλλά ταυτόχρονα και πιο πολύπλοκη εντολή της SQL.

Η επόμενη εντολή θα εμφανίσει τις καταχωρήσεις όλων των στηλών του πίνακα *Erafes*:

```
mysql> SELECT * FROM Erafes;
```

Αν θέλουμε να εμφανισθούν οι τιμές ορισμένων μόνο στηλών, για παράδειγμα των στηλών *ID* και *Eponimo*, δίνουμε την εξής εντολή :

```
mysql> SELECT ID, Eponimo FROM Erafes;
```


Διαγραφή πινάκων – Βάσεων δεδομένων

Η διαδικασία της διαγραφής στην MySQL αποκαλείται *dropping* (απόρριψη) και η εντολή για να διαγράψουμε μια βάση δεδομένων, π.χ. την *test*, είναι η εξής:

```
mysql> DROP DATABASE test;
```

Αν δώσουμε αυτή την εντολή και πατήσουμε Enter, η MySQL θα διαγράψει τη βάση δεδομένων *test* και θα εμφανίσει το μήνυμα *Query OK* σαν επιβεβαίωση. Επειδή αυτή η εντολή δεν εμφανίζει κάποιο μήνυμα προειδοποίησης, πρέπει να είμαστε πολύ προσεκτικοί όταν τη δίνουμε.

Για να διαγράψουμε έναν πίνακα στη MySQL, η εντολή είναι παρόμοια με την εντολή για την διαγραφή μιας βάσης δεδομένων:

```
mysql> DROP TABLE <tableName>;
```

Γραμμή εντολών – σύνταξη εντολών της MySQL

Στα προηγούμενα παραδείγματα παρατηρήσαμε διάφορα χαρακτηριστικά της MySQL σχετικά με τον τρόπο εισαγωγής των εντολών στην οθόνη. Συνοψίζοντας, όλες οι εντολές στη MySQL τελειώνουν με τον χαρακτήρα «;». Μέχρι να συναντήσει αυτό το χαρακτήρα, η MySQL θεωρεί το κείμενο που πληκτρολογούμε ως τμήμα μιας εντολής. Έτσι, μπορούμε να επεκτείνουμε τις εντολές σε περισσότερες από μία γραμμές, όπως φαίνεται στο ακόλουθο παράδειγμα:

```
mysql> SHOW  
-> DATABASES;
```

Η MySQL δείχνει ότι περιμένει από εμάς να ολοκληρώσουμε την εντολή, αλλάζοντας την προτροπή (prompt) από *mysql>* σε *->*.

Για να ακυρώσουμε την τρέχουσα εντολή και να αρχίσουμε να τη γράφουμε από την αρχή, γράφουμε τους χαρακτήρες *\c* και πατάμε ENTER, ως εξής:

```
mysql> DROP DATABASE\c  
mysql>
```

Η MySQL θα αγνοήσει την εντολή που είχαμε ξεκινήσει και θα περιμένει να δώσουμε μια άλλη εντολή.

Τέλος, αν θέλουμε να εξέλθουμε από το πρόγραμμα πελάτη της MySQL, μπορούμε απλά να γράψουμε *quit* ή *exit*. Είναι οι μόνες εντολές που δεν χρειάζονται τον χαρακτήρα «;».

Φιλτράρισμα δεδομένων με κριτήρια: Η εντολή WHERE

Χρησιμοποιώντας την δήλωση **WHERE** σε μια εντολή SELECT, μπορούμε να περιορίσουμε τα επιστρεφόμενα αποτελέσματα χρησιμοποιώντας κάποια συνθήκη, ως εξής :

```
mysql> SELECT Eponimo FROM Erafes
```

```
-> WHERE Onoma = «Γιάννης»;
```

Το παραπάνω ερώτημα θα επιστρέψει τα επώνυμα των καταχωρήσεων στις οποίες το μικρό όνομα είναι "Γιάννης".

Τροποποίηση των Αποθηκευμένων Δεδομένων

Για να κάνουμε αλλαγές στις τιμές ενός πίνακα μιας βάσης δεδομένων, χρησιμοποιούμε την εντολή UPDATE, η σύνταξη της οποίας είναι:

```
mysql> UPDATE <tableName> SET
```

```
-> <col_name>=<new_value>, ...
```

```
-> WHERE <where clause>;
```

Ετσι, αν θελήσουμε για παράδειγμα, να αλλάξουμε το τηλέφωνο της πρώτης καταχώρησης στη βάση EPAFES, θα πρέπει να δώσουμε την εξής εντολή:

```
mysql> UPDATE Erafes SET Telefono="2109876541" WHERE ID=1;
```

Η στήλη ID είναι βολική σ' αυτή την περίπτωση, καθώς μας δίνει την δυνατότητα να διαχωρίσουμε εύκολα και σίγουρα την καταχώρηση που θέλουμε να τροποποιήσουμε.

Διαγραφή Αποθηκευμένων Δεδομένων

Για να διαγράψουμε γραμμές (εγγραφές) στην SQL, η σύνταξη είναι η εξής:

```
mysql> DELETE FROM <tableName> WHERE <where clause>;
```

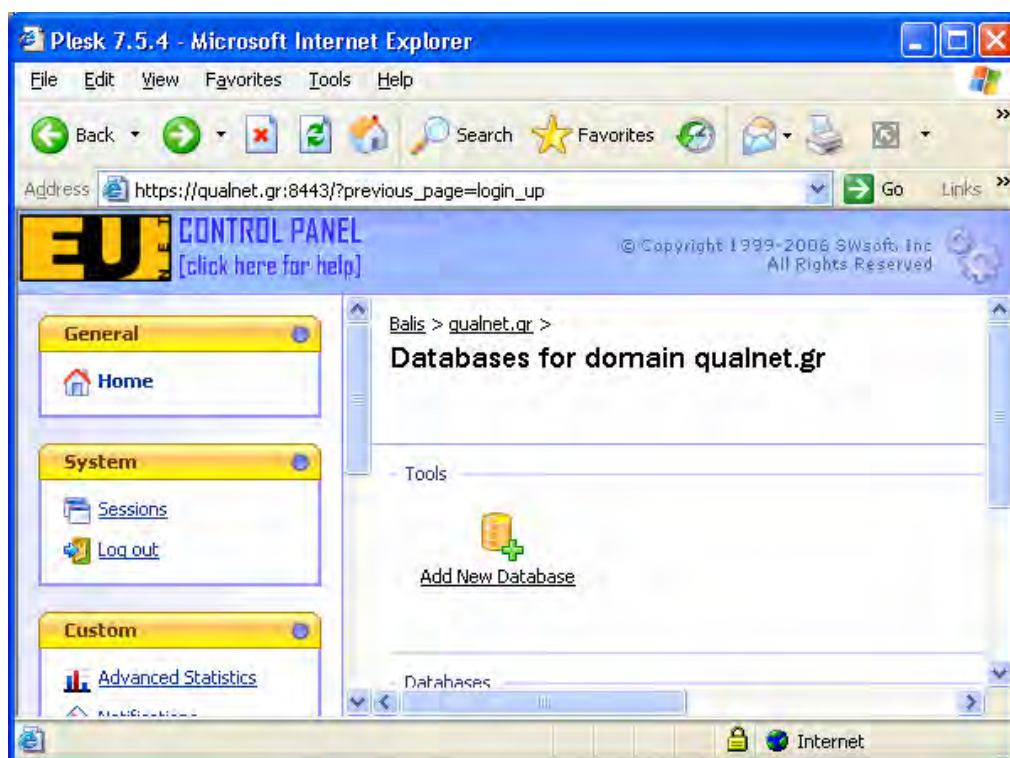
Επίσης, η επόμενη εντολή θα αδειάσει τελείως τον πίνακα EPAFES:

```
mysql> DELETE FROM Erafes;
```

3.2.6 Διαχείριση βάσης δεδομένων με το περιβάλλον PHPMyAdmin

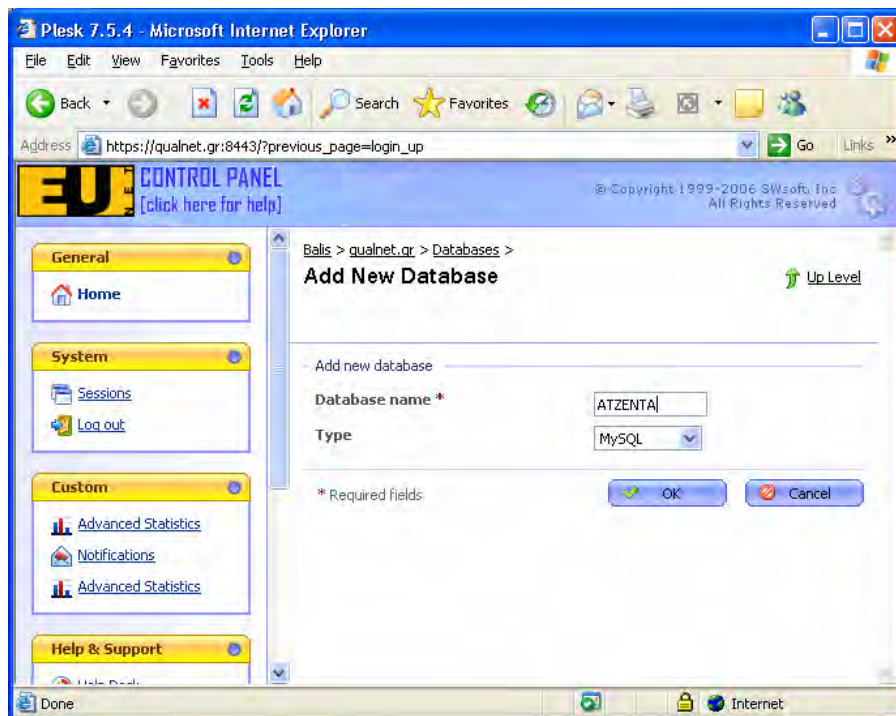
Είδαμε στην προηγούμενη ενότητα τις βασικές εντολές SQL που μας επιτρέπουν τη διαχείριση των βάσεων δεδομένων MySQL. Για το σκοπό αυτό χρησιμοποιήσαμε το MySQL Client που εκτελείται μέσω της γραμμής εντολών (Command Line). Αυτός ο τρόπος διαχείρισης βασίζεται στην εκτέλεση εντολών SQL που πληκτρολογούνται από το χρήστη, διευκολύνοντας την εξοικείωσή του με αυτές. Παρόλ' αυτά, αυτός ο τρόπος διαχείρισης είναι ιδιαίτερα δύσχρηστος και σπανίως χρησιμοποιείται σε πρακτικές εφαρμογές, όπου είναι σαφώς προτιμότερο κάποιο γραφικό περιβάλλον διαχείρισης. Ένα ιδιαίτερα δημοφιλές γραφικό περιβάλλον διαχείρισης βάσεων MySQL, είναι το PHPMyAdmin, το οποίο είναι μια ανοικτή πλατφόρμα διαχείρισης βάσεων δεδομένων MySQL, και είναι δωρεάν διαθέσιμο στη διεύθυνση www.phpmyadmin.net. Η εφαρμογή PHPMyAdmin είναι συνήθως εγκατεστημένη στους εξυπηρετητές που υποστηρίζουν βάσεις δεδομένων MySQL, επιτρέποντας τη διαχείρισή τους με τη γλώσσα PHP την οποία θα δούμε στη συνέχεια του κεφαλαίου.

Για την εξοικείωσή μας με το περιβάλλον PHPMyAdmin, θα υλοποιήσουμε ένα τμήμα του παραδείγματος της προηγούμενης ενότητας σε ένα δοκιμαστικό δικτυακό τόπο. Αυτό μπορεί να γίνει με την επίσκεψη στο λεγόμενο Control Panel, το οποίο περιέχει επιλογή για τη δημιουργία βάσεων δεδομένων, όπως φαίνεται στο παρακάτω σχήμα.



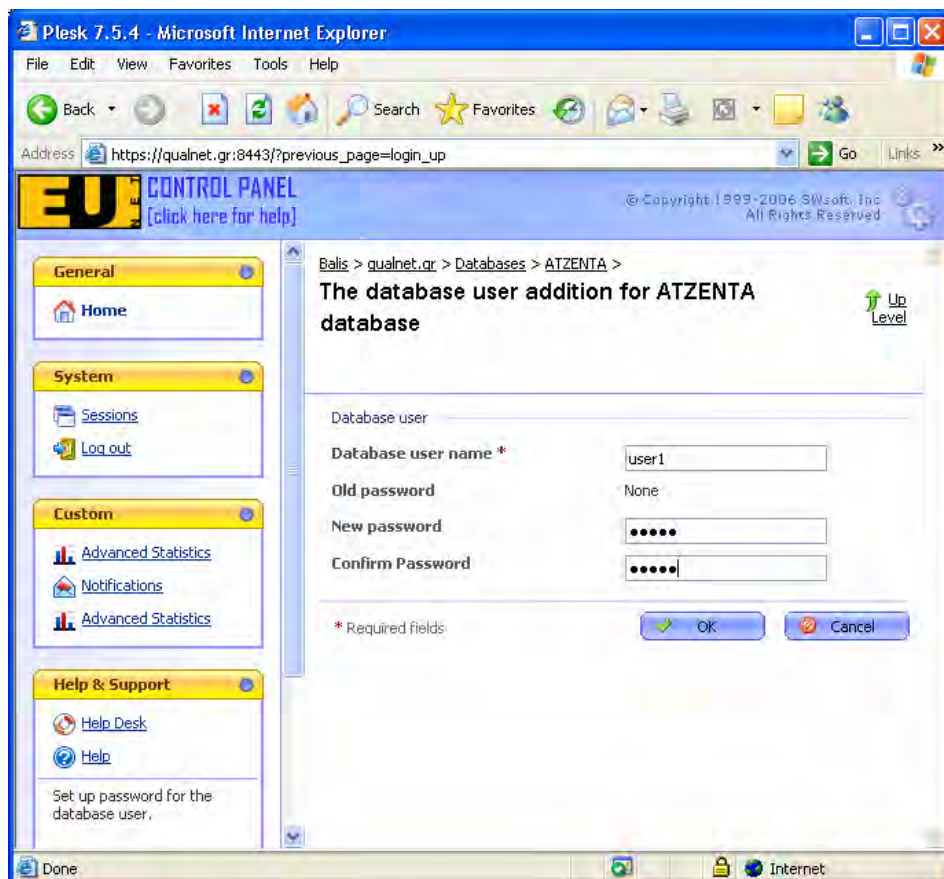
Σχήμα 3.11 Επιλογή προσθήκης νέας βάσεως δεδομένων σε ένα δοκιμαστικό δικτυακό τόπο

Επιλέγοντας Add New Database, μπορούμε να δημιουργήσουμε τη βάση δεδομένων ATZENTA, όπως φαίνεται στο επόμενο σχήμα:



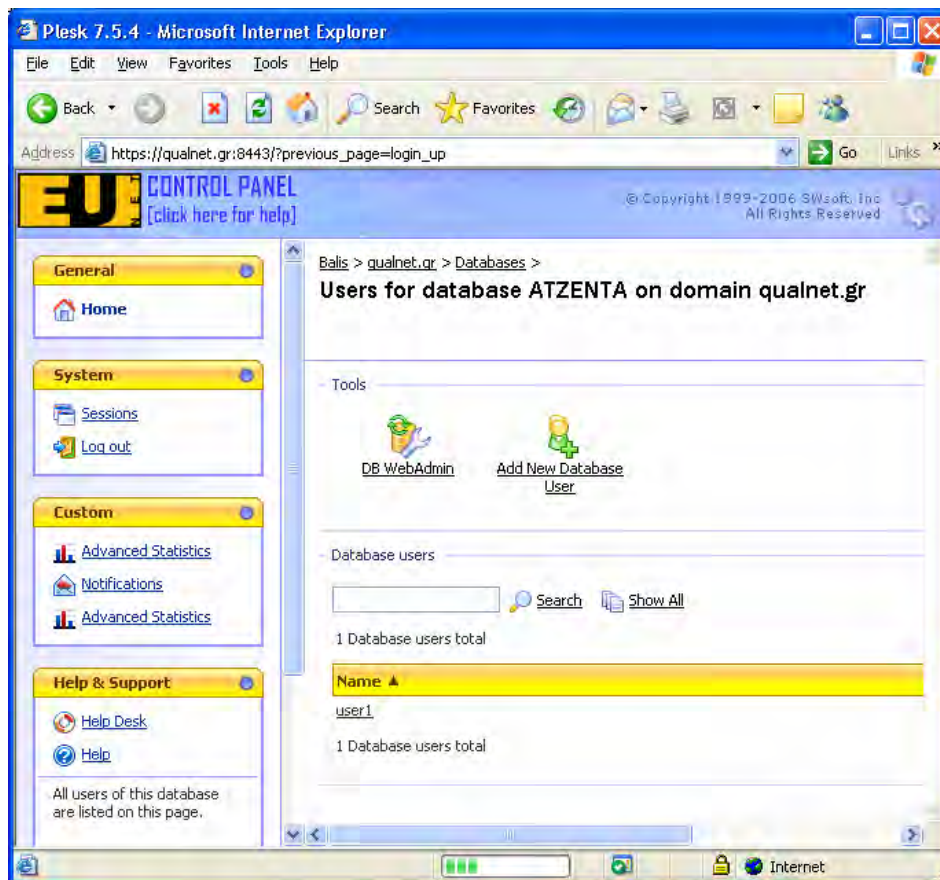
Σχήμα 3.12 Δημιουργία νέας βάσεως δεδομένων

Στη συνέχεια, θα πρέπει να ορίσουμε ένα χρήστη για τη βάση δεδομένων που δημιουργήσαμε, ως εξής:



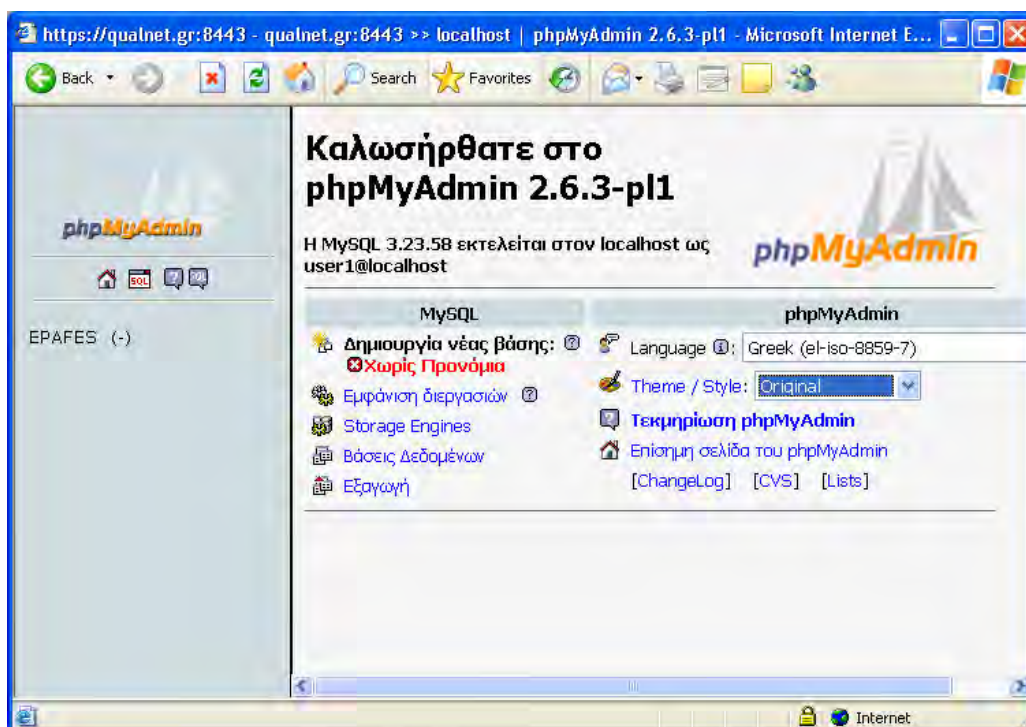
Σχήμα 3.13 Δημιουργία προφίλ χρήστη για τη βάση δεδομένων ATZENTA

Μπορούμε τώρα να διαχειριστούμε τη βάση δεδομένων που δημιουργήσαμε, με την επιλογή DB WebAdmin του παρακάτω σχήματος:



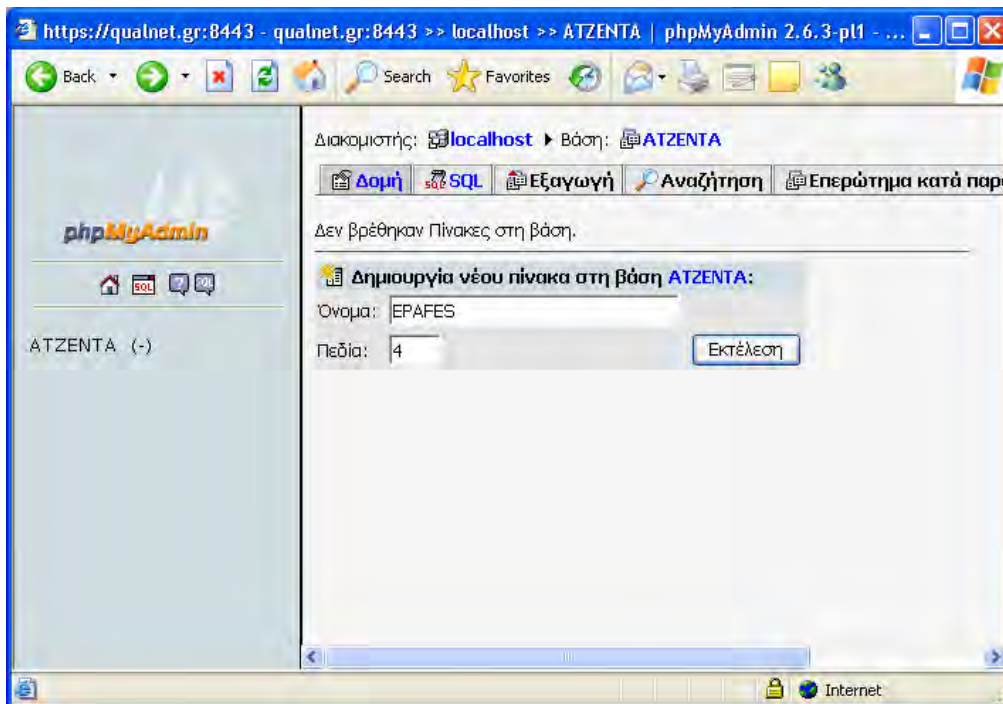
Σχήμα 3.14 Η επιλογή DB WebAdmin

Πιέζοντας την επιλογή DB WebAdmin, εμφανίζεται η κεντρική σελίδα της PHPMYAdmin, όπως φαίνεται στο παρακάτω σχήμα:



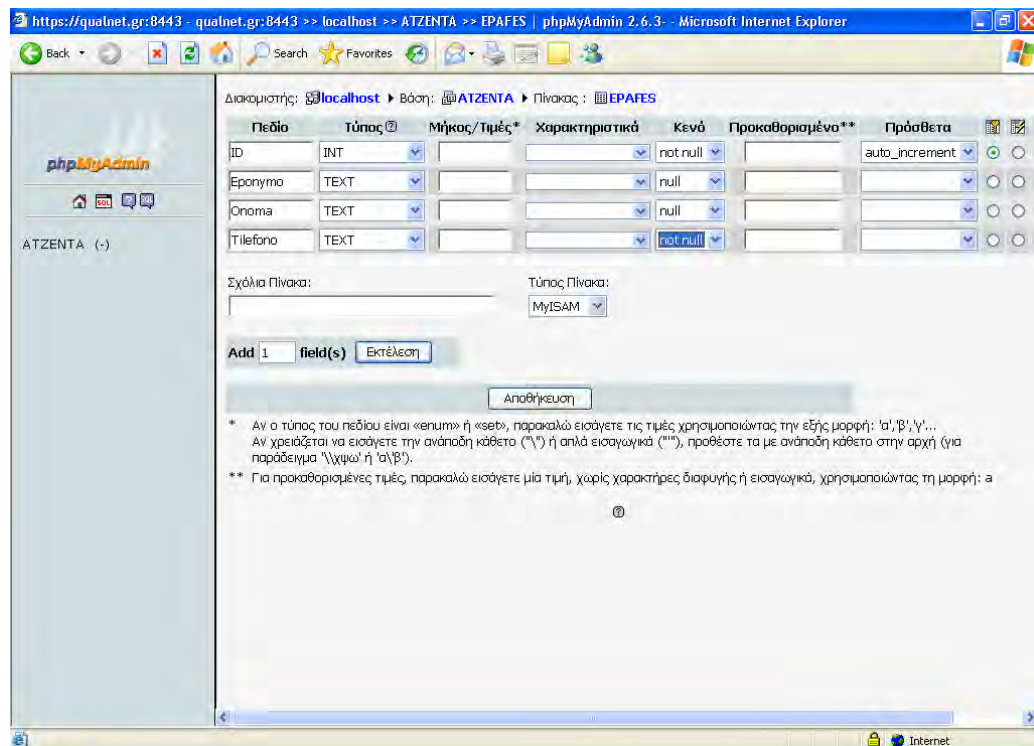
Σχήμα 3.15 Το κεντρικό μενού της phpMyAdmin

Το παραπάνω κεντρικό μενού παρέχει διάφορες λειτουργίες, με βασικότερη εκείνη της δημιουργίας νέας βάσεως δεδομένων. Δημιουργώντας τη βάση δεδομένων ATZENTA, μπορούμε στη συνέχεια να δημιουργήσουμε τον πίνακα EPAFES, όπως φαίνεται στο Σχήμα 3.16.



Σχήμα 3.16 Δημιουργία του πίνακα EPAFES με PhPMyAdmin

Η υποβολή των αρχικών στοιχείων του πίνακα EPAFES οδηγεί στην εμφάνιση παραθύρου για τον καθορισμό των ορισμάτων των πεδίων αυτού του πίνακα, όπως φαίνεται στο Σχήμα 3.17.



Σχήμα 3.17 Καθορισμός ορισμάτων των πεδίων του πίνακα EPAFES

Πέρα από τις παραπάνω βασικές λειτουργίες, η phpMyAdmin παρέχει πολλές δυνατότητες εισαγωγής, αναζήτησης και τροποποίησης των δεδομένων, με ιδιαίτερα φιλικό προς το χρήστη τρόπο. Για το λόγο αυτό, η phpMyAdmin είναι ίσως η δημοφιλέστερη εφαρμογή διαχείρισης βάσεων δεδομένων MySQL και η περαιτέρω εξοικείωση με αυτήν προτείνεται ανεπιφύλακτα στους ενδιαφερόμενους αναγνώστες.

Ενότητα 3.3 Επισκόπηση της γλώσσας PHP

3.3.1 Εισαγωγή στη γλώσσα PHP

Η γλώσσα PHP, της οποίας τα αρχικά προκύπτουν από τη φράση *Hypertext Preprocessor* (προεπεξεργαστής υπερκειμένου) είναι μια ευρύτατα διαδεδομένη γλώσσα προγραμματισμού από την πλευρά του εξυπηρετητή. Η γλώσσα PHP είναι ανοικτού κώδικα και μπορεί να εγκατασταθεί πρακτικά σε όλους τους τύπους εξυπηρετητών, ενώ η εγκατάσταση και χρήση της στις δυναμικές εφαρμογές είναι δωρεάν.

Όπως η Javascript, την οποία είδαμε στο προηγούμενο κεφάλαιο, έτσι και η PHP μπορεί να ενσωματωθεί στη σελίδα HTML. Αυτό όμως που διαφοροποιεί τη λειτουργικότητα της PHP σε σχέση με τις γλώσσες που εκτελούνται από την πλευρά του πελάτη, είναι ότι το πρόγραμμα εκτελείται στον εξυπηρετητή, ενώ ο πελάτης λαμβάνει μόνο το αποτέλεσμα της εκτέλεσης του προγράμματος, σε μορφή HTML. Η εκτέλεση του προγράμματος στον εξυπηρετητή γίνεται με τη βοήθεια του μεταγλωτιστή PHP (PHP parser), ο οποίος εγκαθίσταται στον εξυπηρετητή προσδίδοντάς του τη δυνατότητα εκτέλεσης δυναμικών εφαρμογών PHP. Ο μεταγλωτιστής PHP είναι διαθέσιμος δωρεάν στη διεύθυνση www.php.net, σε διάφορους τύπους, ανάλογα με τον τύπο του εξυπηρετητή και το λειτουργικό σύστημα του υπολογιστή στον οποίο θα γίνει η εγκατάσταση.

Η PHP χαρακτηρίζεται από απλότητα, η οποία παρέχει τις προϋποθέσεις γρήγορης εκμάθησης, σε σχέση με άλλες γλώσσες προγραμματισμού από την πλευρά του εξυπηρετητή. Παρόλη την απλότητά της, η PHP προσφέρει προηγμένα χαρακτηριστικά τα οποία συχνά την καθιστούν μία από τις πρώτες επιλογές για επαγγελματικές εφαρμογές. Ένα άλλο σημαντικό χαρακτηριστικό της PHP είναι η υποστήριξη που παρέχει για ένα μεγάλο εύρος βάσεων δεδομένων, στις οποίες περιλαμβάνονται η MySQL, η Oracle, η Informix, η Sybase και πολλές άλλες.

Στη συνέχεια της ενότητας, θα δούμε τα βασικά προγραμματιστικά χαρακτηριστικά της PHP και θα εξοικειωθούμε με τις βασικές δυνατότητες που παρέχει, με τη βοήθεια απλών παραδειγμάτων.

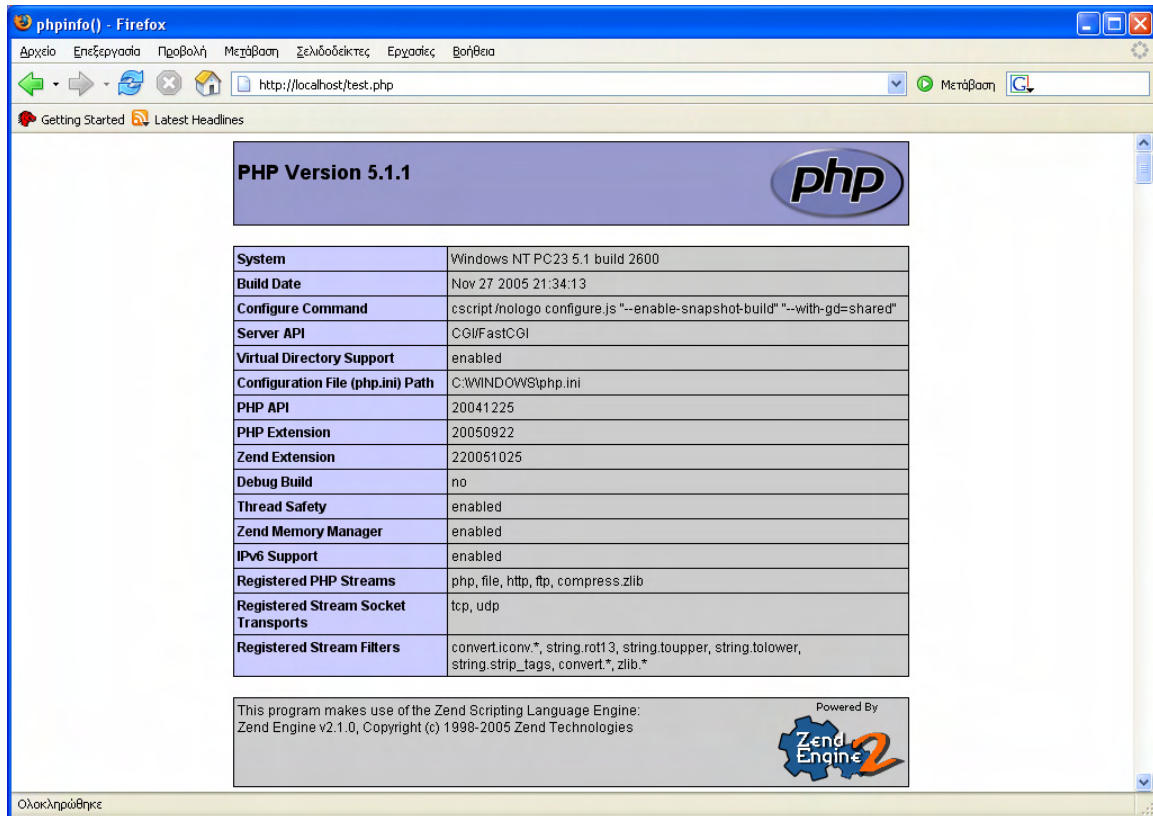
3.3.2 Εισαγωγή στον προγραμματισμό με την PHP

Για τη συνέχεια αυτού του κεφαλαίου, υποθέτουμε ότι οι μαθητές έχουν πρόσβαση σε εξυπηρετητή με ενεργή την υποστήριξη για PHP. Αυτό μπορεί να επαληθευτεί ως εξής: Πληκτρολογούμε το παρακάτω τμήμα κώδικα στο notepad και το αποθηκεύουμε με το όνομα `test.php` στο root directory του web server.

Παράδειγμα 3.1 Test.php

```
<?php  
echo phpinfo();  
?>
```

Στη συνέχεια, καλούμε τη σελίδα Test.php από τη γραμμή διευθύνσεων του φυλλομετρητή. Εάν η PHP είναι σωστά εγκατεστημένη, η σελίδα που θα δούμε θα μοιάζει με την παρακάτω:



Σχήμα 3.18 Αποτέλεσμα παραδείγματος 3.1

Ουσιαστικά το παραπάνω αποτέλεσμα προέρχεται από τη συνάρτηση phpinfo, η οποία ανακτά από τον εξυπηρετητή πληροφορίες σχετικά με την εγκατάσταση της PHP, οι οποίες στη συνέχεια προωθούνται στο φυλλομετρητή με την εντολή echo. Θα μιλήσουμε αναλυτικότερα για τις συναρτήσεις της PHP στη συνέχεια.

Μια πρώτη εφαρμογή

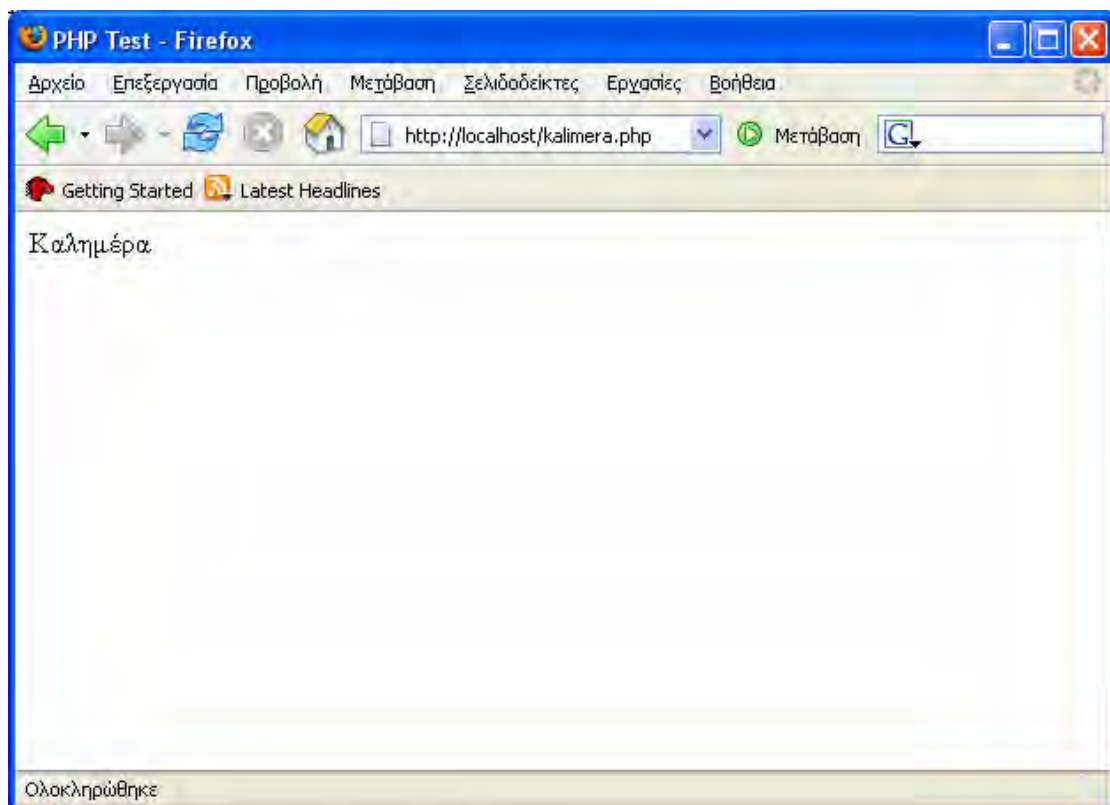
Ένας καλός τρόπος για να εξοικιωθούμε με τα βασικά χαρακτηριστικά της PHP είναι η δημιουργία και εκτέλεση μιας απλής εφαρμογής. Ας δούμε το παρακάτω παράδειγμα:

Δημιουργήστε με το Notepad ένα αρχείο με το ακόλουθο περιεχόμενο και αποθηκεύστε το με το όνομα kalimera.php στο root directory του web server.

Παράδειγμα 3.2 kalimera.php

```
<html>
<head>
  <title>PHP Test</title>
</head>
<body>
  <?php echo "Καλημέρα<p>"; ?>
</body>
</html>
```

Στη συνέχεια, καλέστε τη σελίδα kalimera.php από τη γραμμή διευθύνσεων του φυλλομετρητή. Θα πρέπει να λάβετε το παρακάτω αποτέλεσμα:



Σχήμα 3.19 Αποτέλεσμα παραδείγματος 3.2

Ο πηγαίος κώδικας που αντιστοιχεί στην παραπάνω οθόνη του χρήστη (Client) είναι:

```
<html>
<head>
<title>PHP Test</title>
</head>
<body>
Καλημέρα<p></body>
</html>
```

Αν και το παραπάνω πρόγραμμα είναι ιδιαίτερα απλό, ουσιαστικά μας εισάγει στην πρακτική του προγραμματισμού από την πλευρά του εξυπηρετητή. Αυτό γίνεται φανερό παρατηρώντας ότι ο Server εκτελεί την εντολή echo() της γλώσσας PHP και επιστρέφει στο χρήστη το αποτέλεσμα της εντολής σε μορφή HTML.

Ενσωμάτωση των σεναρίων PHP στην ιστοσελίδα

Όπως είδαμε και στο παραπάνω παράδειγμα, ο κώδικας PHP γράφεται μέσα στον κώδικα HTML και περικλείεται από τις ετικέτες `<?php` και `?>`. Για να γίνει η επεξεργασία μιας ιστοσελίδας που περιέχει PHP από τον εξυπηρετητή, η σελίδα θα πρέπει να αποθηκευθεί ως αρχείο με κατάληξη `.php`. Εάν η σελίδα έχει κατάληξη `.htm`, τότε ο εξυπηρετητής απλά στέλνει τις HTML δηλώσεις στον πελάτη, ο οποίος τις επεξεργάζεται και εμφανίζει το αποτέλεσμα της επεξεργασίας στο φυλλομετρητή. Αν όμως η ιστοσελίδα έχει την κατάληξη `.php`, τότε αυτή ανοίγεται και επεξεργάζεται από τον εξυπηρετητή, ο οποίος αναζητά στη σελίδα τμήματα κώδικα PHP. Μόλις βρεθούν τέτοια τμήματα, η εφαρμογή του εξυπηρετητή προχωρά στις απαιτούμενες επεξεργασίες και μετατρέπει το αποτέλεσμά τους σε μορφή HTML, προωθώντας το στον πελάτη.

3.3.3 Οι συνιστώσες των προγραμμάτων PHP

Η σύνταξη της PHP είναι πολύ κοντά στις εξής γλώσσες: C, C++, Java, JavaScript και Perl. Ένα σενάριο της PHP αποτελείται από μια σειρά εντολών (commands ή statements), η καθεμία από τις οποίες είναι μια οδηγία (instruction) που πρέπει να ακολουθήσει ο Web server πριν προχωρήσει στην επόμενη. Οι εντολές της PHP τερματίζονται πάντα με τον χαρακτήρα semicolon (;). Όπως σε κάθε γλώσσα προγραμματισμού, έτσι και στην PHP βασικά δομικά στοιχεία αποτελούν οι μεταβλητές (Variables) οι τελεστές (operators) και οι συναρτήσεις (functions).

Μεταβλητές

Οι *μεταβλητές* της PHP είναι ίδιες με τις μεταβλητές που υπάρχουν στις περισσότερες γλώσσες προγραμματισμού. Η επόμενη εντολή δημιουργεί μια μεταβλητή με όνομα *\$testvariable* και της εκχωρεί την τιμή 3:

```
$testvariable = 3;
```

Πρέπει να έχουμε υπόψη μας ότι όλα τα ονόματα μεταβλητών στην PHP ξεκινούν με το σύμβολο του δολαρίου (\$).

Τελεστές

Όπως και στις άλλες γλώσσες προγραμματισμού, έτσι και στην PHP χρησιμοποιούνται τελεστές για τη διεξαγωγή αριθμητικών πράξεων και συγκρίσεων και για την εκχώρηση τιμών σε μεταβλητές. Οι τελεστές της PHP δε διαφέρουν σημαντικά από τους αντίστοιχους της Javascript τους οποίους είδαμε στο δεύτερο κεφάλαιο. Οι σημαντικότεροι από αυτούς συνοψίζονται στον παρακάτω πίνακα:

Είδος τελεστή	Σύμβολο	Λειτουργία
<i>Αριθμητικοί</i>	+ - * / % ++ --	Πρόσθεση Αφαίρεση Πολλαπλασιασμός Διαίρεση Ακέραιο μέρος Μοναδιαία Αύξηση Μοναδιαία Μείωση
<i>Σύγκρισης</i>	== !=,<> > >= < <=	επιστρέφει αληθές αν τα ορίσματα είναι ίσα. επιστρέφει αληθές αν τα ορίσματα δεν είναι ίσα. επιστρέφει αληθές όταν το αριστερό όρισμα είναι μεγαλύτερο από το δεξί. επιστρέφει αληθές όταν το αριστερό όρισμα είναι μεγαλύτερο ή ίσο με το δεξί. επιστρέφει αληθές όταν το αριστερό όρισμα είναι μικρότερο από το δεξί. επιστρέφει αληθές όταν το αριστερό όρισμα είναι μικρότερο ή ίσο με το δεξί.
<i>Λογικοί</i>	&& 	επιστρέφει αληθές αν και τα δύο τα ορίσματα είναι αληθή. επιστρέφει αληθές αν οποιοδήποτε από τα δύο ορίσματα είναι αληθές.

<i>Ανάθεση</i>	=	Θέτει την τιμή της έκφρασης στα δεξιά στη μεταβλητή του αριστερού μέρους.
	+ =, - =, * =, / =	Προσθέτει/Αφαιρεί/Πολλαπλασιάζει/Διαιρεί την τιμή της έκφρασης στα δεξιά και τη θέτει στη μεταβλητή του αριστερού μέρους.

Εκτός από τους παραπάνω τελεστές, ένας άλλος χρήσιμος τελεστής της PHP είναι ο τελεστής «.», που επιτρέπει τη συνένωση δύο δεδομένων αλφαριθμητικού τύπου (χαρακτήρες). Για παράδειγμα, η εντολή `$testvariable = «Γεια » . «σου»!` εκχωρεί το κείμενο «Γειά σου!» στη μεταβλητή `testvariable`.

Εντολές ελέγχου ροής

Οι εντολές ελέγχου ροής της PHP είναι παρόμοιες με αυτές της Javascript, τις οποίες αναλύσαμε στο προηγούμενο κεφάλαιο. Η δομή απλής επιλογής της PHP συντάσσεται όπως φαίνεται στο παρακάτω πλαίσιο:

```
if (συνθήκη)
{Τμήμα Εντολών 1}
else
{Τμήμα Εντολών 2}
```

Επίσης, στο παρακάτω πλαίσιο βλέπουμε τη σύνταξη της δομής πολλαπλής επιλογής της PHP.

```
if (συνθήκη1)
{Τμήμα Εντολών 1}
else if (συνθήκη2)
{Τμήμα Εντολών 2}
.
.
.
else if (συνθήκη n)
{Τμήμα Εντολών n}
```

Συναρτήσεις

Ως γλώσσα δομημένου προγραμματισμού, η PHP χρησιμοποιεί την έννοια της συνάρτησης με την οποία εξοικειωθήκαμε στο δεύτερο κεφάλαιο. Κατ' αρχήν, η PHP διαθέτει ένα μεγάλο πλήθος ενσωματωμένων συναρτήσεων, όπως είναι η συνάρτηση `echo` που συναντήσαμε στο παράδειγμα της ενότητας 3.2.2, οι οποίες διευκολύνουν σημαντικότερα τον προγραμματιστή,

επιτρέποντας την εύκολη υλοποίηση ενός μεγάλου εύρους λειτουργιών. Σε αυτές τις συναρτήσεις περιλαμβάνονται και αυτές που επιτρέπουν την αποστολή ηλεκτρονικού ταχυδρομείου, τη σύνδεση και διαχείριση εγγράφων βάσης δεδομένων, την επεξεργασία αλφαριθμητικών δεδομένων, το χειρισμό δεδομένων ημερομηνίας και ώρας, καθώς και πολλές άλλες λειτουργίες. Για τους ενδιαφερόμενους αναγνώστες, πλήρεις αναφορές και παραδείγματα για όλες τις συναρτήσεις της PHP, είναι διαθέσιμα στην ηλεκτρονική διεύθυνση <http://www.php.net>.

Εκτός από τις ενσωματωμένες συναρτήσεις, η PHP επιτρέπει τη δημιουργία συναρτήσεων από τον προγραμματιστή, με μια διαδικασία παρόμοια με εκείνη που είδαμε κατά τον ορισμό των συναρτήσεων στη Javascript, στο δεύτερο κεφάλαιο. Στο παρακάτω παράδειγμα, το οποίο καλείστε να υλοποιήσετε και να δοκιμάσετε, ορίζεται η συνάρτηση “Enosis” η οποία δέχεται δύο ορίσματα και επιστρέφει τη συνένωση των χαρακτήρων τους. Στη συνέχεια, δημιουργούνται οι μεταβλητές \$first και \$second, οι οποίες χρησιμοποιούνται ως ορίσματα στην κλήση της συνάρτησης, η οποία γίνεται με την εντολή

echo Enosis(\$first, \$second); που επιστρέφει το αποτέλεσμα της συνένωσης του περιεχομένου των μεταβλητών \$first και \$second στην οθόνη του φυλλομετρητή.

```
<?php
function Enosis($wordA, $wordB) {
return $wordA . $wordB;
}

$first = "Αυτή είναι μία ";
$second = " πλήρης πρόταση";
echo Enosis($first, $second);
?>
```

3.3.4 Αλληλεπίδραση με το χρήστη – Χειρισμός φόρμας με την PHP

Μία από τις σημαντικότερες δυνατότητες της PHP είναι η αλληλεπίδραση μεταξύ πελάτη – εξυπηρετητή. Αυτή η αλληλεπίδραση γίνεται με τη βοήθεια των λεγόμενων *μεταβλητών εξυπηρετητή* (*server variables*) της PHP. Από αυτές τις μεταβλητές, οι \$_GET και \$_POST μπορούν να καταχωρήσουν τιμές που εισάγονται από το χρήστη σε μια HTML φόρμα, η οποία υποβάλλεται στο πρόγραμμα PHP με τις μεθόδους GET ή POST αντίστοιχα. Ουσιαστικά, οι \$_GET και \$_POST είναι πίνακες στους οποίους καταχωρούνται οι ονομασίες και οι τιμές όλων των αντικειμένων της φόρμας που υποβάλλεται στο πρόγραμμα PHP. Ας δούμε το ακόλουθο παράδειγμα:

Παράδειγμα 3.3 *Submitform.htm*

```
<HTML>
<BODY>

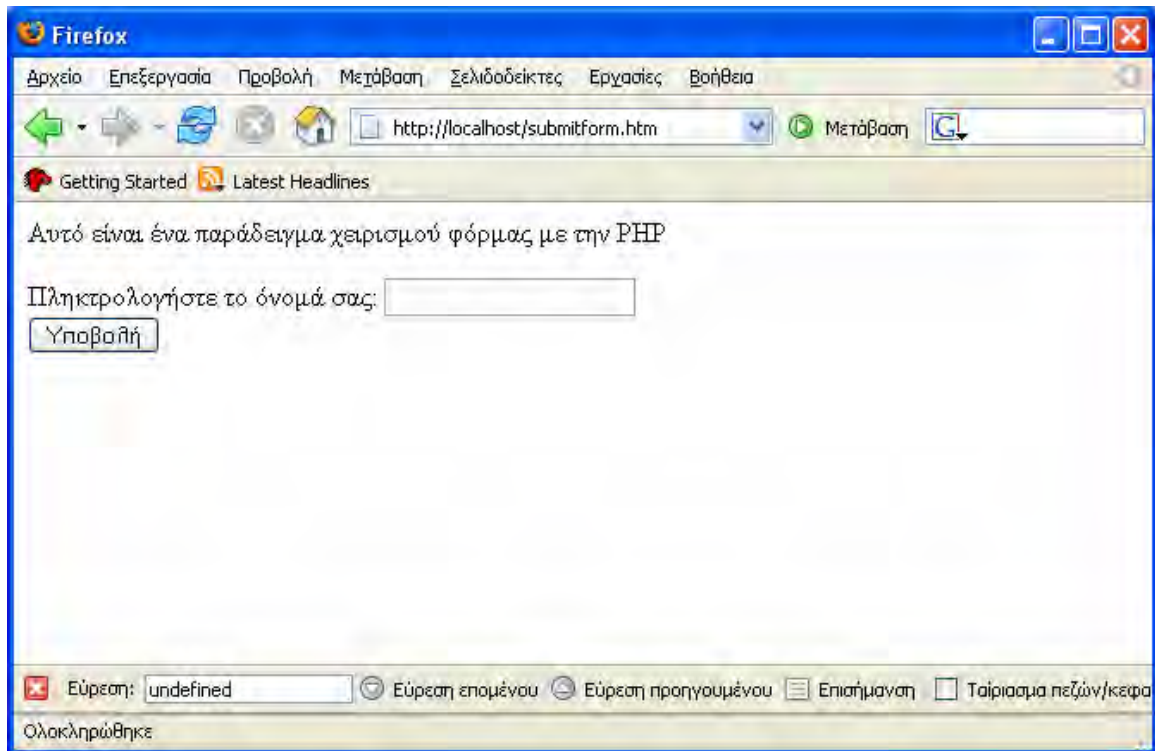
<P>Αυτό είναι ένα παράδειγμα χειρισμού φόρμας με την PHP
</P>
<FORM NAME='frmMain' METHOD=post ACTION="action.php">
Πληκτρολογήστε το όνομά σας:
<INPUT NAME=applicant TYPE=text><BR>
<INPUT NAME=enter TYPE=submit VALUE="Υποβολή">
</FORM>
</BODY>
</HTML>
```

Στο παραπάνω αρχείο HTML δημιουργείται μια φόρμα η οποία περιέχει ένα πεδίο κειμένου με όνομα “applicant” και ένα Button υποβολής της φόρμας, η οποία υποβάλλεται με τη μέθοδο POST, στο αρχείο action.php που ακολουθεί.

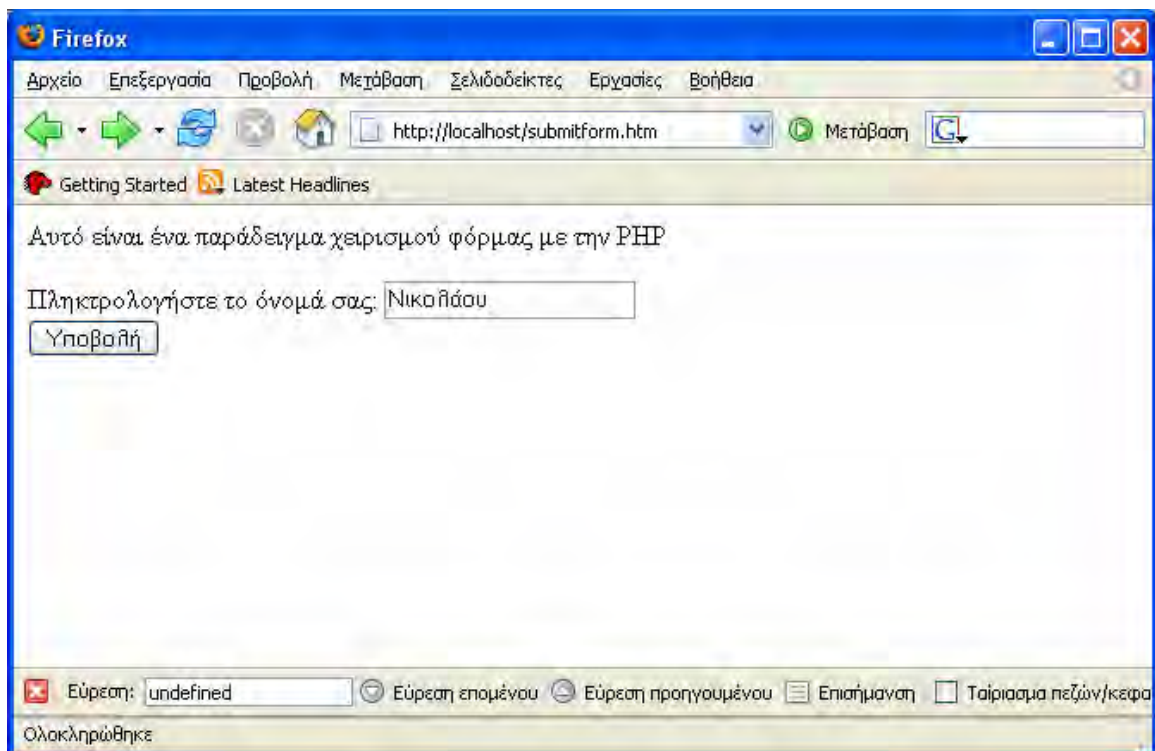
Παράδειγμα 3.4 *Action.php*

```
<HTML>
<BODY>
<P>Καλώς ήλθες <?php echo ($_POST['applicant']); ?>!</P>
</BODY>
</HTML>
```

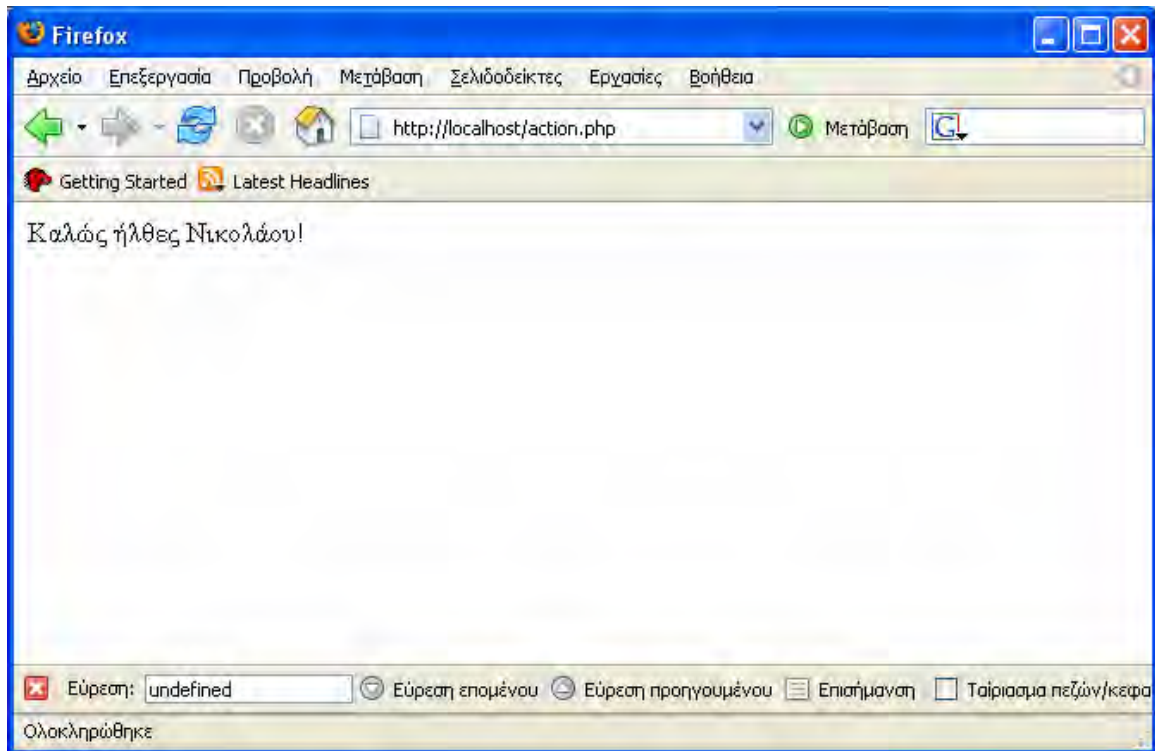
Στο παραπάνω αρχείο, η τιμή του πλαισίου κειμένου εξάγεται ως `$_POST['applicant']` και στη συνέχεια προβάλλεται στο φυλλομετρητή με την `echo`. Πληκτρολογώντας τα δύο παραπάνω αρχεία και αποθηκεύτάς τα στο Root Directory του εξυπηρετητή, μπορούμε να εκτελέσουμε το πρόγραμμα λαμβάνοντας τα παρακάτω αποτελέσματα:



Σχήμα 3.20 Εμφάνιση φόρμας παραδείγματος 3.3



Σχήμα 3.21 Εισαγωγή στοιχείων στη φόρμα του παραδείγματος 3.3



Σχήμα 3.22 Απόκριση του εξυπηρετητή με τη χρήση του παραδείγματος 3.4

Ενότητα 3.4 Δραστηριότητες ανακεφαλαίωσης

A) Περιγράψτε δύο βασικά πλεονεκτήματα που παρέχουν οι εφαρμογές από την πλευρά του εξυπηρετητή, σε σχέση με τις εφαρμογές από την πλευρά του πελάτη.

B) Χαρακτηρίστε τα παρακάτω με Σωστό ή Λάθος:

- Η λειτουργία των εφαρμογών από την πλευρά του εξυπηρετητή δεν εξαρτάται από τον τύπο του φυλλομετρητή των χρηστών.
- Η MySQL υλοποιείται με την αρχιτεκτονική πελάτη/εξυπηρετητή (client/server).
- Στη MySQL μπορούμε να δημιουργήσουμε μια βάση δεδομένων χωρίς να υπάρχει επικοινωνία με το MySQL Server.
- Η γλώσσα PHP μπορεί να ενσωματωθεί στη σελίδα HTML.

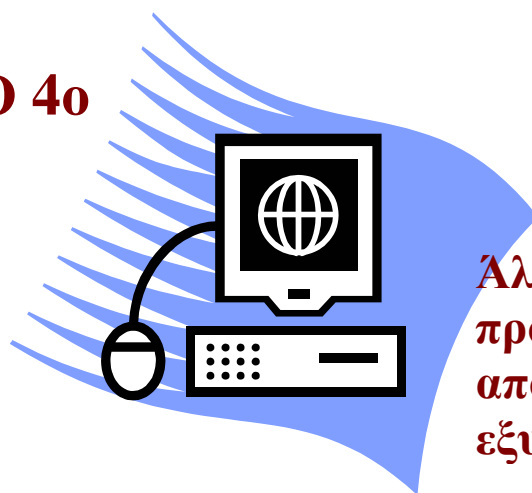
Γ) Επιλέξτε τη σωστή απάντηση

Ποια από τις παρακάτω συναρτήσεις της PHP μπορεί να χρησιμοποιηθεί για την εμφάνιση ενός κειμένου στην οθόνη του πελάτη; <ul style="list-style-type: none">• date• echo• mail	Η λήψη δεδομένων από μια φόρμα που υποβάλλεται στον εξυπηρετητή μπορεί να γίνει με τη βοήθεια των <ul style="list-style-type: none">• ενσωματωμένων συναρτήσεων• μεταβλητών εξυπηρετητή• σεναρίων Javascript
---	--

Δ) Δημιουργήστε μια σελίδα HTML στην οποία θα πρέπει να περιλαμβάνεται μια φόρμα, η οποία θα περιέχει δύο πεδία κειμένου, ένα για το όνομα και ένα για το επώνυμο, καθώς και ένα πλήκτρο υποβολής. Με την πίεση του πλήκτρου, η σελίδα θα πρέπει να υποβάλλεται σε ένα πρόγραμμα PHP, το οποίο επίσης θα πρέπει να δημιουργήσετε και το οποίο θα πρέπει να επεξεργάζεται τις τιμές των πλαισίων κειμένου, επιστρέφοντας την ένδειξη “Δεν πληκτρολογήσατε όνομα” ή “Δεν πληκτρολογήσατε επώνυμο”, στην περίπτωση που κάποιο από τα πεδία είναι κενό, διαφορετικά, θα πρέπει να εμφανίζει την ένδειξη “Καλημέρα” συνοδευμένη από το όνομα και το επώνυμο που δόθηκαν.

E) Στη διεύθυνση <http://www.php.net/manual/el>, θα βρείτε πολύ χρήσιμες πληροφορίες τεκμηρίωσης της PHP, μεταφρασμένες στην ελληνική γλώσσα. Χρησιμοποιώντας το πεδίο αναζήτησης αυτής της ιστοσελίδας, εντοπίστε τη συνάρτηση mail και διαβάστε προσεκτικά την περιγραφή της και το παράδειγμα χρήσης της. Με βάση αυτή τη συνάρτηση, καλείστε να δημιουργήσετε την ιδιαίτερα δημοφιλή εφαρμογή αποστολής μηνύματος φόρμας προς κάποια συγκεκριμένη διεύθυνση ηλεκτρονικού ταχυδρομείου. Ποιο συγκεκριμένα, η φόρμα που θα δημιουργήσετε σε HTML, θα πρέπει να περιέχει ένα πεδίο κειμένου για το θέμα και ένα για το κείμενο του μηνύματος. Η φόρμα αυτή θα πρέπει να υποβάλλεται σε ένα PHP script που θα δημιουργήσετε, το οποίο θα δέχεται τις τιμές των πεδίων κειμένου της φόρμας και στη συνέχεια, χρησιμοποιώντας κατάλληλα τη συνάρτηση mail, θα στέλνει ένα μήνυμα ηλεκτρονικού ταχυδρομείου.

ΚΕΦΑΛΑΙΟ 4ο



**Άλλες Γλώσσες
προγραμματισμού
από την πλευρά του
εξυπηρετητή**

Δυναμικές Εφαρμογές Παγκόσμιου Ιστού

Σκοπός και στόχοι του κεφαλαίου

Σκοπός του κεφαλαίου είναι η εισαγωγή των ενηλίκων στον προγραμματισμό δυναμικών εφαρμογών ιστοσελίδων, από την πλευρά του εξυπηρετητή, με τις τεχνολογίες ASP και JSP.

Μετά την ολοκλήρωση του κεφαλαίου, οι μαθητές θα είναι σε θέση να:

- κατανοούν τις βασικές αρχές της γλώσσας προγραμματισμού ASP, τόσο αυτόνομα, όσο και σε σύγκριση με τις αντίστοιχες βασικές αρχές της γλώσσας PHP.
- δημιουργούν απλές εφαρμογές δυναμικού προγραμματισμού από την πλευρά του εξυπηρετητή χρησιμοποιώντας την ASP.
- κατανοούν τις βασικές αρχές της γλώσσας προγραμματισμού JSP και να αντιλαμβάνονται την έννοια του Javabean ως μια βασική προγραμματιστική οντότητα η οποία διαφοροποιεί τη JSP από τις γλώσσες προγραμματισμού PHP και ASP.
- δημιουργούν απλές εφαρμογές δυναμικού προγραμματισμού από την πλευρά του εξυπηρετητή χρησιμοποιώντας την JSP.

Έννοιες-Κλειδιά

ActiveServer Pages, VBScript, Java Server Pages, Javabean

Εισαγωγικές Παρατηρήσεις

Η αρχιτεκτονική των εφαρμογών ASP (ActiveServer Pages) παρουσιάζει σημαντικές ομοιότητες με αυτή των εφαρμογών PHP, με τη βασική διαφορά ότι η ASP αναπτύχθηκε από τη Microsoft προκειμένου να παράσχει βελτιστοποιημένες δυνατότητες παροχής δυναμικού περιεχομένου στους εξυπηρετητές Web τύπου IIS της Microsoft. Από την άλλη πλευρά, η JSP (Java Server Pages) είναι μια τεχνολογία προγραμματι-

σμού από την πλευρά του εξυπηρετητή, η οποία αναπτυχθηκε από τη Sun Microsystems και βασίζεται στη γλώσσα προγραμματισμού Java, ενώ σε σχέση με την PHP και την ASP παρέχει εξελιγμένες δυνατότητες διαχωρισμού μεταξύ του περιεχομένου της ιστοσελίδας και της μορφής με την οποία αυτό παρουσιάζεται.

Ενότητα 4.1 Active Server Pages (ASP)

4.1.1 Εισαγωγή

Η γλώσσα προγραμματισμού ASP παρέχει ένα προγραμματιστικό περιβάλλον ανάπτυξης εφαρμογών οι οποίες εκτελούνται από την πλευρά του εξυπηρετητή. Η αρχιτεκτονική των εφαρμογών ASP παρουσιάζει σημαντικές ομοιότητες με αυτή των εφαρμογών PHP που αναλύσαμε στο προηγούμενο κεφάλαιο. Όμως, σε αντίθεση με την PHP η οποία είναι γλώσσα ανοιχτού κώδικα και μπορεί να εγκατασταθεί σε ένα μεγάλο εύρος λειτουργικών συστημάτων και τύπων εξυπηρετητών, η ASP αναπτύχθηκε από τη Microsoft προκειμένου να παράσχει βελτιστοποιημένες δυνατότητες παροχής δυναμικού περιεχομένου στους εξυπηρετητές Web τύπου IIS της Microsoft.

Όπως ισχύει στην PHP, έτσι και στην ASP, ο κώδικας πρώτα εκτελείται στον εξυπηρετητή και το αποτέλεσμα της εκτέλεσης επιστρέφει στο χρήστη με τη μορφή εγγράφου HTML. Επίσης, όπως και στην PHP, έτσι και στην ASP ο κώδικας ενσωματώνεται στη σελίδα και περικλείεται από κατάλληλες ετικέτες.

Στη συνέχεια του κεφαλαίου, θα αναλύσουμε τα βασικά προγραμματιστικά χαρακτηριστικά της γλώσσας ASP και θα εξοικειωθούμε με τις δυνατότητες που παρέχει στον προγραμματιστή δυναμικών εφαρμογών, με τη βοήθεια απλών παραδειγμάτων.

4.1.2 Βασικά χαρακτηριστικά της γλώσσας ASP

Σε αναλογία με την PHP, ένα αρχείο ASP περιέχει ετικέτες HTML, καθώς και σενάρια που εκτελούνται στον εξυπηρετητή, τα οποία περικλείονται από τις ετικέτες `<%` και `%>`. Τα σενάρια αυτά μπορούν να περιέχουν οποιεσδήποτε εκφράσεις, εντολές, συναρτήσεις ή τελεστές, ανάλογα με τη γλώσσα σεναρίων που χρησιμοποιούμε. Τα αρχεία ASP έχουν κατάληξη `.asp`. Σε αντίθεση με την PHP, στην ASP μπορούμε να χρησιμοποιήσουμε διάφορες γλώσσες συγγραφής σεναρίων, όπως είναι η VBScript και η JavaScript. Η προκαθορισμένη γλώσσα συγγραφής σεναρίων της ASP είναι η VBScript. Στο σημείο αυτό θα βοηθούσε η υλοποίηση ενός απλού παραδείγματος σε ASP.

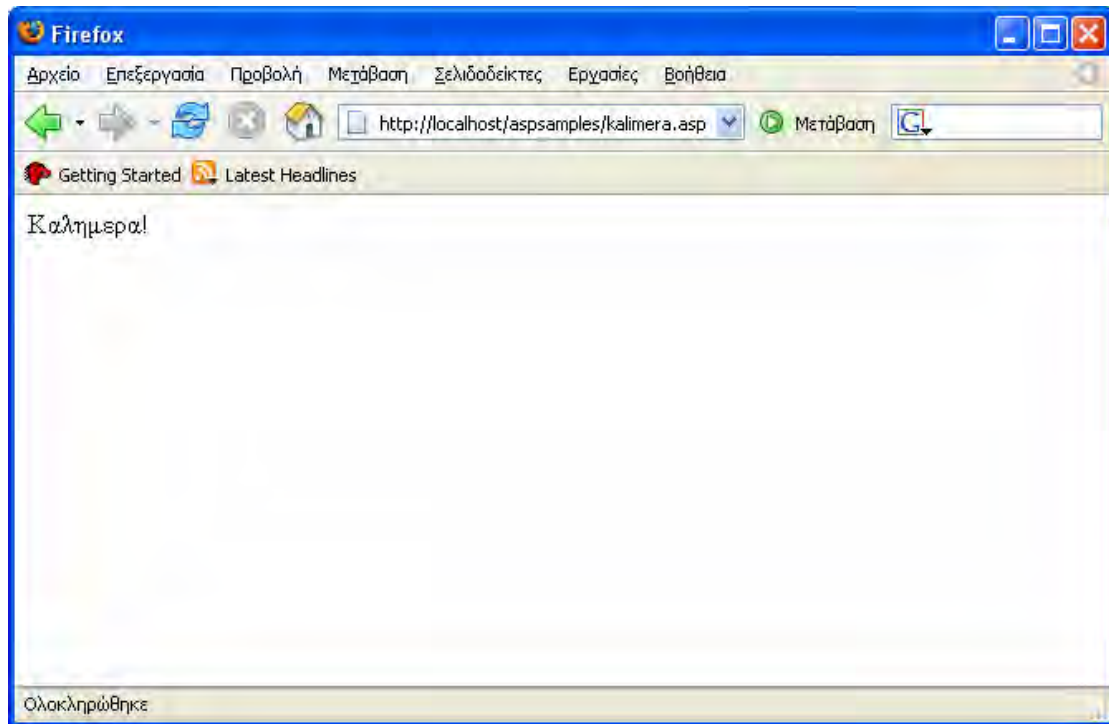
Μια απλή εφαρμογή σε ASP

Πληκτρολογήστε το παρακάτω κείμενο στο Notepad και αποθηκεύστε το ως αρχείο με το όνομα «kalimera.asp», στο root directory του web server.

```
Kalimera.asp
<html>
<body>
<%
response.write("Hello World!")
```

```
%>
</body>
</html>
```

Στη συνέχεια δώστε τη διεύθυνση του αρχείου στο φυλλομετρητή. Θα πρέπει να λάβετε το ακόλουθο αποτέλεσμα:



Σχήμα 4.1 Μια απλή εφαρμογή ASP

Το παραπάνω παράδειγμα χρησιμοποιεί τη συνάρτηση *response.write* για να εμφανίσει το μήνυμα *Καλημέρα!* στο φυλλομετρητή. Όπως και στο αντίστοιχο παράδειγμα που είδαμε στο προηγούμενο κεφάλαιο με την PHP, ο πηγαίος κώδικας που αντιστοιχεί στην παραπάνω οθόνη του φυλλομετρητή δεν περιέχει το σενάριο ASP που δημιουργήσαμε, παρά μόνο το αποτέλεσμα του, το οποίο φαίνεται στο ακόλουθο πλαίσιο:

```
<html>
<head>
<title></title>
</head>
<body>
Καλημέρα!<p></body>
</html>
```

4.1.3 Βασικές συνιστώσες των προγραμμάτων ASP

Όπως οι περισσότερες γλώσσες προγραμματισμού, έτσι και η ASP χρησιμοποιεί στοιχεία προγραμματισμού όπως μεταβλητές, τελεστές, εντολές, συναρτήσεις και αντικείμενα. Θα περιγράψουμε στη συνέχεια τα παραπάνω στοιχεία προγραμματισμού και θα εξοικιωθούμε με τη χρήση τους με τη βοήθεια παραδειγμάτων.

Μεταβλητές

Η χρήση των μεταβλητών στην ASP δε διαφέρει σημαντικά σε σχέση τόσο με την PHP όσο και με τις άλλες γλώσσες προγραμματισμού. Σε σύγκριση με την PHP, οι βασικές διαφορές είναι αφενός η χρήση του ονόματος της μεταβλητής χωρίς να προηγείται το σύμβολο \$ και αφετέρου η δήλωση της μεταβλητής πριν από τη χρήση της, με τη βοήθεια της εντολής Dim, στην περίπτωση που χρησιμοποιείται η προκαθορισμένη γλώσσα VBScript. Στο παρακάτω παράδειγμα ορίζεται με την εντολή Dim η μεταβλητή *Pi*, και στη συνέχεια αποδίδεται σε αυτήν η τιμή 3.1415.

```
<%  
Option Explicit  
Dim Pi  
Pi = 3.1415  
%>
```

Αξίζει να σημειωθεί ότι στη VBScript δεν είναι υποχρεωτική η δήλωση των μεταβλητών, όμως συνίσταται ανεπιφύλακτα, εφόσον στην αντίθετη περίπτωση ο εντοπισμός σφαλμάτων του προγράμματος καθίσταται ιδιαίτερα προβληματικός. Η δήλωση Option Explicit του παραπάνω παραδείγματος καθιστά υποχρεωτική τη δήλωση των μεταβλητών, προκαλώντας σφάλμα στην περίπτωση που αυτό δε γίνει. Για το λόγο αυτό, συνίσταται όλα τα σενάρια ASP να ξεκινούν με αυτή τη δήλωση.

Εντολές ελέγχου ροής

Οι εντολές ελέγχου ροής της ASP ακολουθούν τη φιλοσοφία των αντίστοιχων εντολών της Javascript και της PHP, όμως διαφέρουν στη σύνταξη. Έτσι, η δομή απλής επιλογής της PHP συντάσσεται όπως φαίνεται στο παρακάτω πλαίσιο:

```
if (συνθήκη) then  
{Τμήμα Εντολών 1}  
else  
{Τμήμα Εντολών 2}  
End if
```

Αντίστοιχα, η σύνταξη της δομής πολλαπλής επιλογής της ASP φαίνεται στο παρακάτω πλαίσιο:

```
if (συνθήκη1) then
{Τμήμα Εντολών 1}
else if (συνθήκη2)
{Τμήμα Εντολών 2}
.
.
.
else if (συνθήκη n)
{Τμήμα Εντολών n}
End If
```

Συναρτήσεις

Όπως η PHP, έτσι και η ASP διαθέτει ένα μεγάλο πλήθος ενσωματωμένων συναρτήσεων, οι οποίες περιλαμβάνουν λειτουργίες όπως την αποστολή ηλεκτρονικού ταχυδρομείου, τη σύνδεση και διαχείριση εγγραφών βάσης δεδομένων, την επεξεργασία αλφαριθμητικών δεδομένων, το χειρισμό δεδομένων ημερομηνίας και ώρας και πολλές άλλες. Επίσης, είναι δυνατή η δημιουργία συναρτήσεων από τον προγραμματιστή, με τη χρήση κατάλληλης σύνταξης. Στο παρακάτω παράδειγμα, το οποίο καλείστε να υλοποιήσετε και να δοκιμάσετε, ορίζεται η συνάρτηση «Addition» η οποία δέχεται δύο αριθμούς και επιστρέφει το άθροισμά τους. Στη συνέχεια, η συνάρτηση αυτή καλείται με δύο διαφορετικά ορίσματα, επιστρέφοντας το αποτέλεσμα της πρόσθεσης στο φυλλομετρητή του πελάτη.

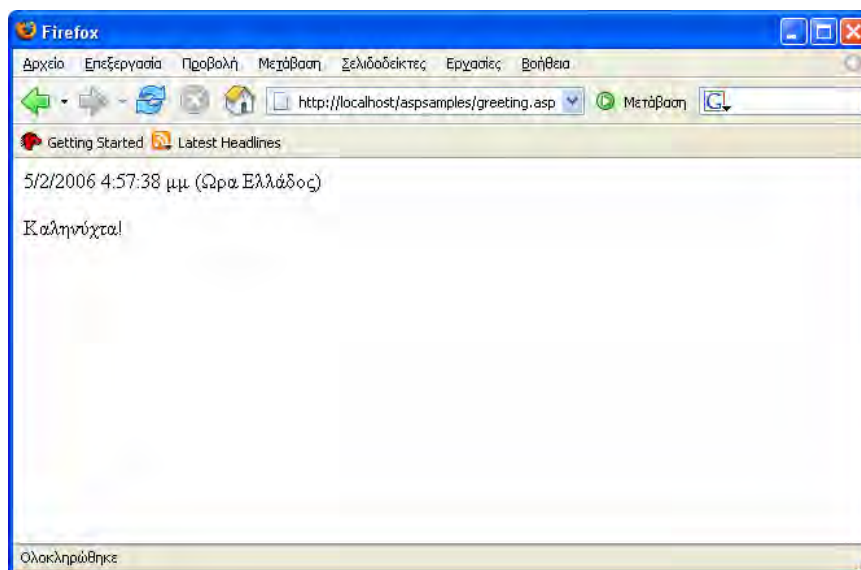
```
<%
Function Addition(A, B)
Addition = A + B
End Function
%>
.
.
.
<%
Response.Write Addition(2, 3)
Response.Write Addition(5, 8)
%>
```


Συνδυάζοντας μεταβλητές, συναρτήσεις και εντολές ελέγχου ροής

Με τη βοήθεια των παραπάνω γνώσεων σχετικά με τις μεταβλητές, τις εντολές ελέγχου ροής και τις συναρτήσεις της ASP, μπορούμε να υλοποιήσουμε ένα πιο σύνθετο παράδειγμα. Στο παράδειγμα αυτό, θα χρησιμοποιήσουμε την ενσωματωμένη συνάρτηση `hour(now)` που επιστρέφει το ακέραιο μέρος της ώρας, για να εμφανίσουμε ένα διαφορετικό μήνυμα στον πελάτη ανάλογα με την ώρα του εξυπηρετητή. Ο σχετικός κώδικας παρατίθεται στο ακόλουθο πλαίσιο:

```
Greeting.asp
<html>
<body>
<%
Dim h
h = hour(now())
response.write("<p>" & now())
response.write(" (Ωρα Ελλάδος) </p>")
If h < 12 then
response.write("Καλημέρα!")
else
response.write("Καληνύχτα!")
end if
%>
</body>
</html>
```

Αποθηκεύοντας το παραπάνω κείμενο ως αρχείο με όνομα `greeting.asp` στο Root Directory του εξυπηρετητή, μπορούμε να το προσπελάσουμε με το φυλλομετρητή, οπότε και θα λάβουμε το ακόλουθο αποτέλεσμα:



Σχήμα 4.2 Χρήση συναρτήσεων – εμφάνιση του παραδείγματος `Greeting.asp`

Τα αντικείμενα της ASP

Η ASP περιλαμβάνει ένα σύνολο προκαθορισμένων αντικειμένων τα οποία επιτρέπουν την πρόσβαση και διαχείριση ενός μεγάλου εύρους δεδομένων και παραμέτρων, ενώ εμπεριέχουν ενσωματωμένες συναρτήσεις οι οποίες εκτελούν συγκεκριμένες εντολές κατά την εκτέλεση του κώδικα ASP. Χαρακτηριστικό παράδειγμα αντικειμένου της ASP είναι το αντικείμενο *Response*, το οποίο εμπεριέχει τη συνάρτηση *Response.Write* την οποία είδαμε στην απλή εφαρμογή της προηγούμενης ενότητας. Τα βασικά αντικείμενα της ASP συνοψίζονται στον παρακάτω πίνακα:

Αντικείμενο	Χρήση
<i>Application</i>	Διανομή πληροφοριών ανάμεσα σ' όλους τους χρήστες μιας συγκεκριμένης εφαρμογής.
<i>Response</i>	Αποστολή δεδομένων στο φυλλομετρητή του πελάτη.
<i>Request</i>	Λήψη δεδομένων από το φυλλομετρητή του πελάτη.
<i>Server</i>	Πρόσβαση σε ιδιότητες και μεθόδους του εξυπηρετητή.
<i>Session</i>	Αποθηκεύση πληροφοριών και τροποποίηση ρυθμίσεων, σχετικά με τη σύνοδο ενός χρήστη.

Τα πιο σημαντικά από τα παραπάνω αντικείμενα, τα οποία και θα εξετάσουμε στη συνέχεια με τη βοήθεια παραδειγμάτων, είναι τα αντικείμενα *Response* και *Request*, τα οποία επιτρέπουν την άμεση αλληλεπίδραση του χρήστη με τον εξυπηρετητή με τη βοήθεια φορμών HTML.

4.1.4 Χειρισμός φόρμας με την ASP

Η δήλωση *Request.QueryString* επιτρέπει τη λήψη δεδομένων που αποστέλλονται στον κώδικα ASP από μια φόρμα HTML. Αυτό γίνεται ιδιαίτερα κατανοητό με τη βοήθεια ενός παραδείγματος. Κατ' αρχήν, δημιουργούμε το παρακάτω HTML έγγραφο.

```
Form.htm
<html>

<body>

<form action="form_action.asp" method="get">

Παρακαλώ, γράψτε το μικρό σας όνομα :

<input type="text" name="fname"><br><br>

<input type="submit" value="Submit">

</form>
```

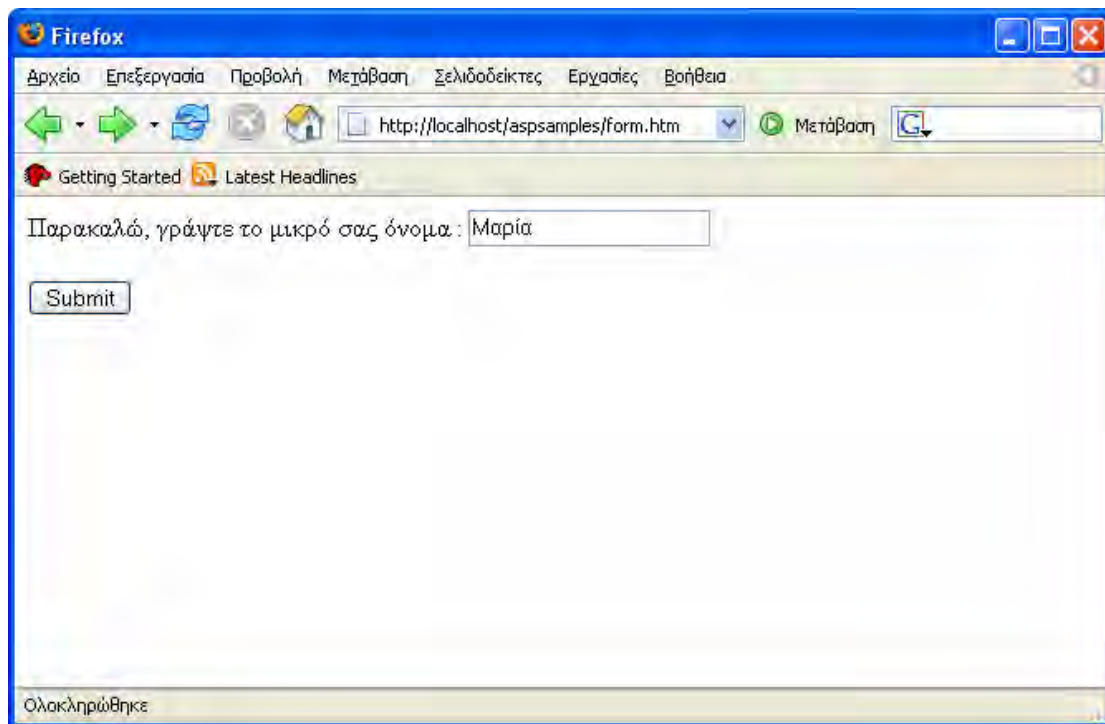
```
</body>
</html>
```

Το παραπάνω έγγραφο περιέχει μια φόρμα, της οποίας τα αντικείμενα υποβάλλονται στο σενάριο `form_action.asp`, όπως καθορίζεται από την παράμετρο `action`. Το σενάριο αυτό είναι το εξής:

```
Form_action.asp
<%
If Request.QueryString("fname")<>"" Then
Response.Write ("Γεια σου " & Request.QueryString("fname") & "!")
Response.Write ("<br> Τι κάνεις;")
End If
%>
```

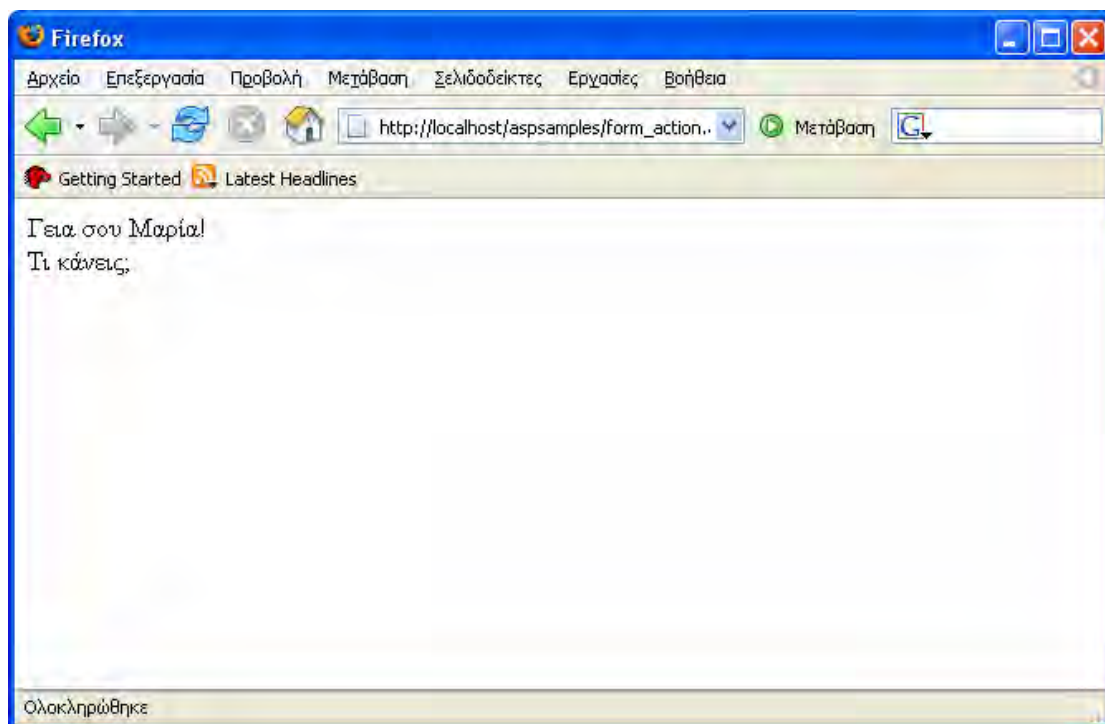
Στο παραπάνω σενάριο, η δήλωση `Request.QueryString("fname")` ανακτά την τιμή του πεδίου κειμένου `fname` που ορίσαμε στον κώδικα HTML. Στη συνέχεια, στην περίπτωση που η τιμή αυτή δεν είναι κενή, το σενάριο επιστρέφει ένα μήνυμα στο φυλλομετρητή του πελάτη.

Αποθηκεύοντας τα δύο παραπάνω αρχεία στο `Root Directory` του εξυπηρετητή, μπορούμε να προσπελάσουμε το αρχείο `Form.htm` με το φυλλομετρητή, οπότε και θα λάβουμε το ακόλουθο αποτέλεσμα:



Σχήμα 4.3 Παράδειγμα υποβολής φόρμας

Πληκτρολογώντας ένα όνομα και πιέζοντας το πλήκτρο Submit, το σενάριο αποκρίνεται στο φυλλομετρητή ως εξής:



Σχήμα 4.4 Αποτέλεσμα επεξεργασίας δεδομένων φόρμας

4.1.5 Δραστηριότητες ανακεφαλαίωσης

A) Συμπληρώστε τα κενά με την κατάλληλη φράση ή σύμβολο.

- Το αντικείμενο της ASP επιτρέπει τη λήψη δεδομένων στο φυλλομετρητή.
- Η είναι η προκαθορισμένη γλώσσα συγγραφής σεναρίων ASP.
- Στην ASP οι μεταβλητές πριν από τη χρήση τους πρέπει να δηλώνονται με την εντολή

B) Επιλέξτε τη σωστή απάντηση.

Ο κώδικας ASP περικλείεται από τα σύμβολα α) <% και %> β) <! και !> γ) <* και *> δ) <@ και @>	Με την ASP μπορούμε να στείλουμε ένα κείμενο στο φυλλομετρητή με τη συνάρτηση α) Request.QueryString β) Response.Write γ) Server.CreateObject
---	--

Γ) Καλείστε να δημιουργήσετε σε περιβάλλον ASP μια εφαρμογή μετατροπής δραχμών σε Ευρώ η οποία θα λειτουργεί ως εξής:

- Θα παρουσιάζεται στον πελάτη μια φόρμα στην οποία θα εισάγετε σε πλαίσιο κειμένου το προς μετατροπή ποσό σε δραχμές.
- Η φόρμα θα πρέπει να περιλαμβάνει ένα κουμπί υποβολής του αιτήματος μετατροπής.
- Η μετατροπή θα πρέπει να γίνεται με την κλήση κατάλληλου ASP αρχείου που θα δημιουργήσετε, το οποίο θα κάνει τον υπολογισμό και θα επιστρέφει το αποτέλεσμα στον πελάτη.

Δ) Δημιουργήστε μια σελίδα HTML στην οποία θα πρέπει να περιλαμβάνεται μια φόρμα, η οποία θα περιέχει δύο πεδία κειμένου, ένα για το όνομα και ένα για το επώνυμο, καθώς και ένα πλήκτρο υποβολής. Με την πίεση του πλήκτρου, η σελίδα θα πρέπει να υποβάλλεται σε ένα πρόγραμμα ASP, το οποίο επίσης θα πρέπει να δημιουργήσετε και το οποίο θα πρέπει να επεξεργάζεται τις τιμές των πλαισίων κειμένου, επιστρέφοντας την ένδειξη “Δεν πληκτρολογήσατε όνομα” ή “Δεν πληκτρολογήσατε επώνυμο”, στην περίπτωση που κάποιο από τα πεδία είναι κενό. Διαφορετικά, θα πρέπει να εμφανίζει την ένδειξη “Καλημέρα” συνοδευμένη από το όνομα και το επώνυμο που δόθηκαν, καθώς και από την ημερομηνία του εξυπηρετητή

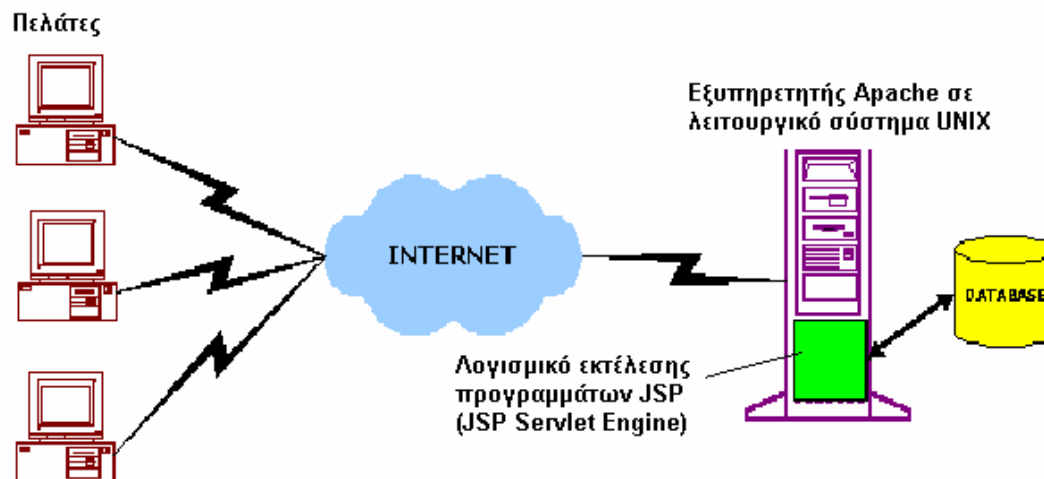
Ενότητα 4.2 JSP

4.2.1 Εισαγωγή

Η JSP (Java Server Pages) είναι μια τεχνολογία προγραμματισμού από την πλευρά του εξυπηρετητή, η οποία αναπτύχθηκε από τη Sun Microsystems και βασίζεται στη γλώσσα προγραμματισμού Java. Όπως με την PHP και την ASP, τα αρχεία JSP είναι αρχεία HTML, τα οποία περιέχουν ειδικές ετικέτες οι οποίες περικλείουν κώδικα γραμμένο σε Java, που είναι υπεύθυνος για την παροχή δυναμικού περιεχομένου στο χρήστη.

Από την πλευρά του προγραμματισμού, η JSP, βασιζόμενη στις αρχές του αντικειμενοστραφούς προγραμματισμού, παρέχει σημαντικές δυνατότητες διαχωρισμού του προγράμματος από το περιεχόμενο της ιστοσελίδας, σε σχέση με τις τεχνολογίες PHP και ASP, όπως θα δούμε στη συνέχεια. Από εμπορικής πλευράς, η JSP θα μπορούσε να χαρακτηριστεί ως η απάντηση της Sun Microsystems στην ASP της Microsoft. Όσον αφορά τη χρήση, η ASP προτιμάται στις περιπτώσεις που ο Web Server είναι υλοποιημένος στην πλατφόρμα της Microsoft, ενώ η JSP είναι ανεξάρτητη πλατφόρμας, αν και συνεργάζεται άμεσα με συστήματα UNIX.

Ο τρόπος με τον οποίο ο εξυπηρετητής εκτελεί τα προγράμματα JSP παρέχοντας δυναμικό περιεχόμενο στους πελάτες, δε διαφέρει από τον αντίστοιχο των τεχνολογιών PHP και ASP που είδαμε στις προηγούμενες ενότητες. Η σχετική διάταξη απεικονίζεται στο παρακάτω σχήμα.



Σχήμα 4.5 Διάταξη πελάτη – εξυπηρετητή JSP

Όπως φαίνεται, ο κώδικας JSP εκτελείται από το λογισμικό εκτέλεσης προγραμμάτων JSP (JSP Servlet Engine) που είναι εγκατεστημένο στον εξυπηρετητή. Το λογισμικό αυτό δημιουργεί δυναμικά τον κώδικα HTML ο οποίος επιστρέφεται στο φυλλομετρητή του πελάτη.

Όπως προαναφέρθηκε, η JSP είναι συμβατή με τις πλατφόρμες εξυπηρετητών που βασίζονται τόσο στα λειτουργικά συστήματα της Microsoft (π.χ. IIS), όσο και στο UNIX (π.χ. Apache). Αυτό ισχύει υπό την προϋπόθεση ότι έχει εγκατασταθεί στον εξυπηρετητή ο ανάλογος τύπος του λογισμικού JSP Servlet Engine, συγκεκριμένα ο JRUN ή ο Tomcat. Η παραπάνω δυνατότητα εξασφαλίζει την εύκολη μετάβαση και μετεγκατάσταση των εφαρμογών JSP από τη μια αρχιτεκτονική εξυπηρετητή στην άλλη, κάτι το οποίο θεωρείται ένα από τα πλεονεκτήματα του JSP προγραμματιστικού περιβάλλοντος.

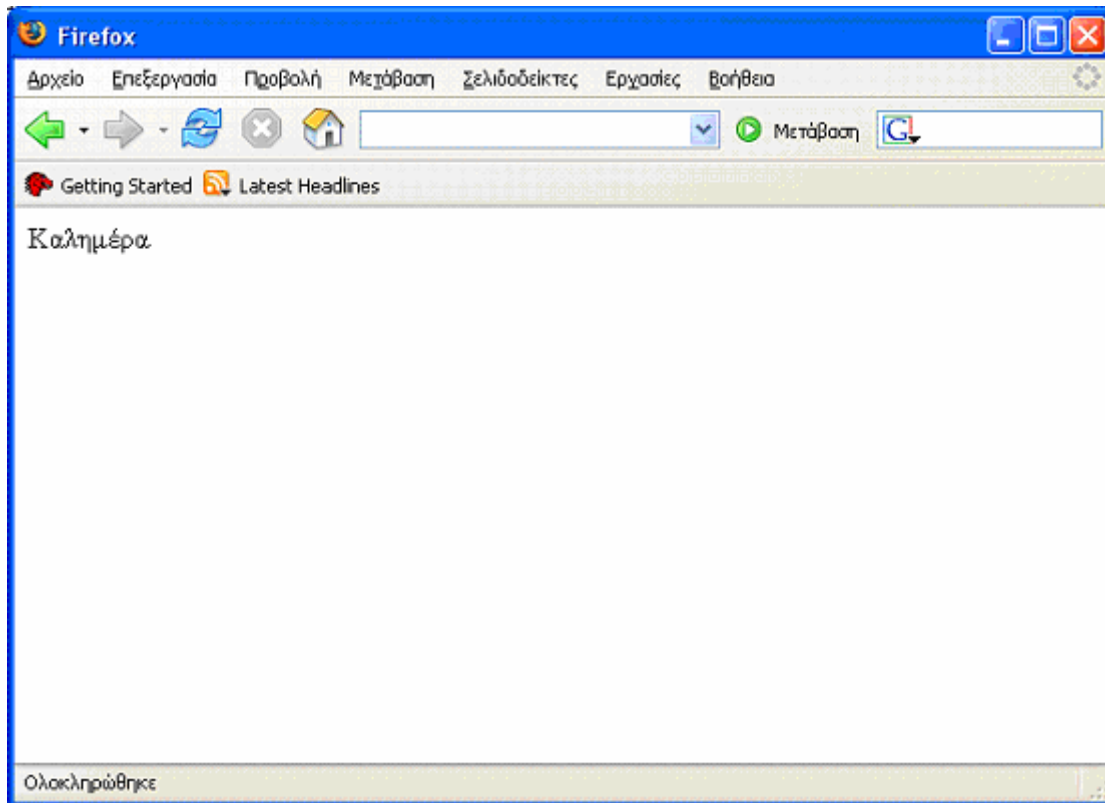
Στη συνέχεια του κεφαλαίου, θα παρουσιάσουμε τα βασικά χαρακτηριστικά της JSP και θα εξοικειωθούμε με τον προγραμματισμό σε περιβάλλον JSP με τη βοήθεια απλών παραδειγμάτων.

4.2.2 Προγραμματίζοντας με τη JSP

Έχοντας πρόσβαση σε εξυπηρετητή ο οποίος υποστηρίζει τη JSP, μπορούμε να δημιουργήσουμε την πρώτη μας απλή εφαρμογή, όπως κάναμε και στις γλώσσες PHP και ASP. Ο σχετικός κώδικας παρουσιάζεται στο ακόλουθο πλαίσιο:

```
kalimera.jsp  
<HTML>  
<BODY>  
<% out.println("Καλημέρα"); %>  
</BODY>  
</HTML>
```

Πληκτρολογώντας το παραπάνω κείμενο και αποθηκεύοντάς το στον εξυπηρετητή ως αρχείο με το όνομα “kalimera.jsp”, μπορούμε να ανοίξουμε το αρχείο αυτό από το φυλλομετρητή, οπότε και θα λάβουμε το παρακάτω αποτέλεσμα:



Σχήμα 4.6 Εμφάνιση παραδείγματος kalimera.jsp

Από το παραπάνω απλό παράδειγμα, μπορούμε αρχικά να παρατηρήσουμε ότι ο κώδικας JSP μπορεί να ενσωματωθεί στην ιστοσελίδα περικλειόμενος από κατάλληλες ετικέτες, κάτι που ισχύει τόσο στην PHP όσο και στην ASP. Επίσης, ο τρόπος με τον οποίο στέλνεται η λέξη «Καλημέρα» στο φυλλομετρητή με την εντολή `out.println("Καλημέρα")` μοιάζει περισσότερο με τον αντίστοιχο τρόπο που είδαμε στην ASP με την εντολή `Response.Write` η οποία χρησιμοποιεί το αντικείμενο `Response`, παρά με την PHP στην οποία χρησιμοποιείται η συνάρτηση `echo()`. Αυτή η παρατήρηση είναι μια πρώτη ένδειξη της χρήσης των αντικειμένων από τη JSP, την οποία θα περιγράψουμε εκτενέστερα στη συνέχεια.

4.2.3 Ετικέτες (tags), Οδηγίες (directives) και Αντικείμενα (Objects)

Αν και η JSP παρουσιάζει κοινά σημεία με τις άλλες τεχνολογίες προγραμματισμού από την πλευρά του εξυπηρετητή, έχει σημαντικά ιδιαίτερα χαρακτηριστικά τα οποία αποτελούν αναπόσπαστο μέρος των JSP προγραμμάτων. Ενα από αυτά τα ιδιαίτερα χαρακτηριστικά της JSP είναι η χρήση ετικετών, οι οποίες ομαδοποιούν συγκεκριμένες λειτουργίες του κώδικα JSP, επιτρέποντας την ευέλικτη ενσωμάτωσή του στο αρχείο HTML. Οι βασικές ετικέτες της JSP συνοψίζονται στον παρακάτω πίνακα:

Ετικέτα	Συμβολισμός	Χρήση
Declaration	<code><%! ... %></code>	Ορισμός μεταβλητών ή μεθόδων αντικειμένων
Expression	<code><%=... %></code>	Ενσωμάτωση εκφράσεων και έξοδος αποτελέσματος στο φυλλομετρητή

Directive	<% @ directive ... %>	Παροχή ειδικών πληροφοριών της σελίδας στο JSP Servlet Engine
Scriptlet	<%...%>	Ενσωμάτωση κώδικα JSP
Action	<jsp:/>	Κλήση εξωτερικών αντικειμένων

Εκτός από τις ετικέτες, άλλο ιδιαίτερο χαρακτηριστικό της JSP είναι οι οδηγίες (*directives*). Οι οδηγίες είναι στοιχεία της JSP τα οποία παρέχουν γενικές πληροφορίες για μια ολόκληρη σελίδα JSP. Για παράδειγμα, η δήλωση οδηγίας <%@ page language="java" %> χρησιμοποιεί την οδηγία *page* και την παράμετρο *language* για να καθορίσει τη γλώσσα της σελίδας JSP. Εκτός από την οδηγία *page*, η JSP χρησιμοποιεί επίσης τις οδηγίες *include* και *taglib*. Οι οδηγίες αυτές συνοδεύονται από σχετικές παραμέτρους και ο ρόλος τους συνοψίζεται στον παρακάτω πίνακα:

Οδηγία	Ρόλος
Page	Ορίζει βασικές πληροφορίες που ρυθμίζουν τη μεταγλώτιση ολόκληρης της σελίδας JSP, όπως τη γλώσσα, την αποθήκευση ή όχι των μεταβλητών συνόδου, την αποθήκευση (buffering) των δεδομένων που στέλνονται στο φυλλομετρητή κλπ.
Include	Χρησιμοποιείται για την εισαγωγή εξωτερικού κειμένου ή κώδικα στη σελίδα JSP κατά τη διάρκεια της εκτέλεσης του JSP κώδικα από το φυλλομετρητή.
Taglib	Παρέχει ένα μηχανισμό χρησιμοποίησης μη προκαθορισμένων ετικετών, οι οποίες δημιουργούνται από τον προγραμματιστή και αποθηκεύονται σε εξωτερικές βιβλιοθήκες. Για τη χρήση των ετικετών αυτών σε μια σελίδα JSP είναι απαραίτητη η αναφορά στη σχετική βιβλιοθήκη με την οδηγία <i>taglib</i> .

Εκτός από τις ετικέτες και τις οδηγίες, η JSP χρησιμοποιεί προκαθορισμένα αντικείμενα, τα οποία αναλαμβάνουν σημαντικό ρόλο κατά την εκτέλεση του προγράμματος. Τα προκαθορισμένα αυτά αντικείμενα παρουσιάζουν αναλογίες με τα αντίστοιχα της ASP, όπως φαίνεται από τον παρακάτω πίνακα που συνοψίζει τα βασικά τους χαρακτηριστικά.

Αντικείμενο	Χρήση
Request	Λήψη δεδομένων από το φυλλομετρητή του πελάτη. Server
Response	Επεξεργασία και αποστολή δεδομένων HTML στο φυλλομετρητή του πελάτη.
Out	Αποστολή δεδομένων στο φυλλομετρητή του πελάτη
Session	Αποθήκευση πληροφοριών και τροποποίηση ρυθμίσεων, σχετικά με τη σύνοδο ενός χρήστη.
Application	Διανομή πληροφοριών ανάμεσα σ' όλους τους χρήστες μιας συγκεκριμένης εφαρμογής.

Page	Represents the JSP page and is used to call any methods defined by the servlet class.
Config	Πρόσβαση σε ιδιότητες και μεθόδους του εξυπηρετητή.

4.2.4 Διαχωρίζοντας τον κώδικα από το περιεχόμενο: Η έννοια του Javabean

Κατά την παρουσίαση των γλωσσών PHP και ASP που είδαμε στις προηγούμενες ενότητες, παρουσιάσαμε τον τρόπο με τον οποίο οι γλώσσες αυτές ενσωματώνονται στην ιστοσελίδα, περικλειόμενες από κατάλληλες ετικέτες. Όπως είναι φανερό, ο συγκεκριμένος τρόπος προγραμματισμού βασίζεται στην ανάμιξη του προγράμματος με το περιεχόμενο της ιστοσελίδας. Η δυσκολία που αυτό συνεπάγεται στην ανάπτυξη και υποστήριξη εφαρμογών μεγάλης κλίμακας είναι προφανής: Ο προγραμματιστής καλείται να έχει άμεση εμπλοκή στον τρόπο με τον οποίο παρουσιάζεται το περιεχόμενο της σελίδας στους χρήστες, κάτι που ουσιαστικά είναι αντικείμενο εξειδικευμένων ατόμων στους οποίους μπορεί να περιλαμβάνονται γραφίστες, φιλόλογοι κλπ. Το πρόβλημα αυτό έρχεται να αντιμετωπίσει η JSP, παρέχοντας σημαντικές δυνατότητες διαχωρισμού μεταξύ του περιεχομένου και του προγράμματος των ιστοσελίδων. Αυτός ο διαχωρισμός πραγματοποιείται με τη χρήση των μεγάλων δυνατοτήτων αντικειμενοστραφούς προγραμματισμού που παρέχει η JSP, η χρήση των οποίων προϋποθέτει την κατανόηση της θεμελιώδους έννοιας των JavaBeans. Τα JavaBeans είναι μια συγκεκριμένη κατηγορία αντικειμένων της JSP που δημιουργούνται από τον προγραμματιστή και αποθηκεύονται στον εξυπηρετητή. Τα αντικείμενα αυτά μπορούν να κληθούν από διαφορετικές σελίδες JSP εκτελώντας προκαθορισμένες λειτουργίες. Η προγραμματιστική υλοποίηση των JavaBeans γίνεται με τέτοιο τρόπο ώστε το κάθε JavaBean να είναι ένα αυτοτελές κομμάτι κώδικα το οποίο εκτελεί συγκεκριμένες λειτουργίες και επιστρέφει αποτελέσματα, επικοινωνώντας για το σκοπό αυτό με άλλα JSP προγράμματα του εξυπηρετητή, με τη βοήθεια ιδιοτήτων και μεθόδων τις οποίες ενσωματώνει στο JavaBean ο προγραμματιστής.

Ο εξοικείωση με τη χρήση των JavaBeans διευκολύνεται με ένα παράδειγμα. Πιο συγκεκριμένα, θα αναπτύξουμε με τη JSP μια απλή εφαρμογή χειρισμού φόρμας. Σε αντίθεση με παρόμοιες εφαρμογές που είδαμε τόσο στην PHP όσο και στην ASP, εδώ θα υλοποιήσουμε την εφαρμογή με τη βοήθεια ενός JavaBean. Αυτό το JavaBean κωδικοποιείται όπως φαίνεται στο παρακάτω πλαίσιο:

```
Namehandler.java
package kalimera;
public class NameHandler {
private String username;
public NameHandler() {
```

```

username = null;
}
public void setUsername( String name ) {
username = name;
}
public String getUsername() {
return username;
}
}
}

```

Στον παραπάνω κώδικα, δημιουργείται ένα *JavaBean* με το όνομα *NameHandler*, το οποίο ανήκει στο πακέτο *kalimera*, με τις δηλώσεις *public class NameHandler* και *package kalimera*, αντίστοιχα. Το πακέτο *kalimera* θα μπορούσε να περιέχει και άλλα *JavaBeans*. Στο *JavaBean NameHandler* ορίζουμε την ιδιότητα *username* με τη δήλωση *private String username*. Στη συνέχεια, ορίζουμε τις μεθόδους *setUsername* και *getUsername*, οι οποίες χρησιμοποιούνται για την απόδοση τιμής και την ανάκτηση τιμής, αντίστοιχα, της ιδιότητας *username*.

Ουσιαστικά, το παραπάνω *JavaBean* καλείται από ένα πρόγραμμα το οποίο του υποβάλλει ένα όνομα, και απλά επιστρέφει το όνομα στο πρόγραμμα από το οποίο κλήθηκε. Ας δούμε λοιπόν στη συνέχεια το κυρίως πρόγραμμα του παραδείγματος, ο κώδικας του οποίου παρουσιάζεται στο παρακάτω πλαίσιο.

main.jsp

```

<%@ page import="kalimera.NameHandler" %>
<jsp:useBean id="abean" scope="page" class="kalimera.NameHandler" />
<jsp:setProperty name="abean" property="*" />
<html>
<h1>Με λένε Γιώργο. Εσένα πως σε λένε;</h1>
<form method="get">
<input type="text" name="username" size="25">
<br>
<input type="submit" value="Υποβολή">
<input type="reset" value="Επαναφορά">
</form>

```

```
<%  
if ( request.getParameter("username") != null ) {  
%>  
<% @ include file="response.jsp" %>  
%>  
}  
%>  
</html>
```

Στο παραπάνω αρχείο, καθορίζουμε κατ' αρχήν ότι θα χρησιμοποιηθεί το JavaBean name-handler. Στη συνέχεια δημιουργούμε μια φόρμα με ένα πεδίο κειμένου και με πλήκτρα υποβολής και επαναφοράς. Στη φόρμα αυτή, αντί να χρησιμοποιήσουμε την παράμετρο action, κατευθύνουμε τη ροή του προγράμματος στο σενάριο response.jsp, με τη δήλωση

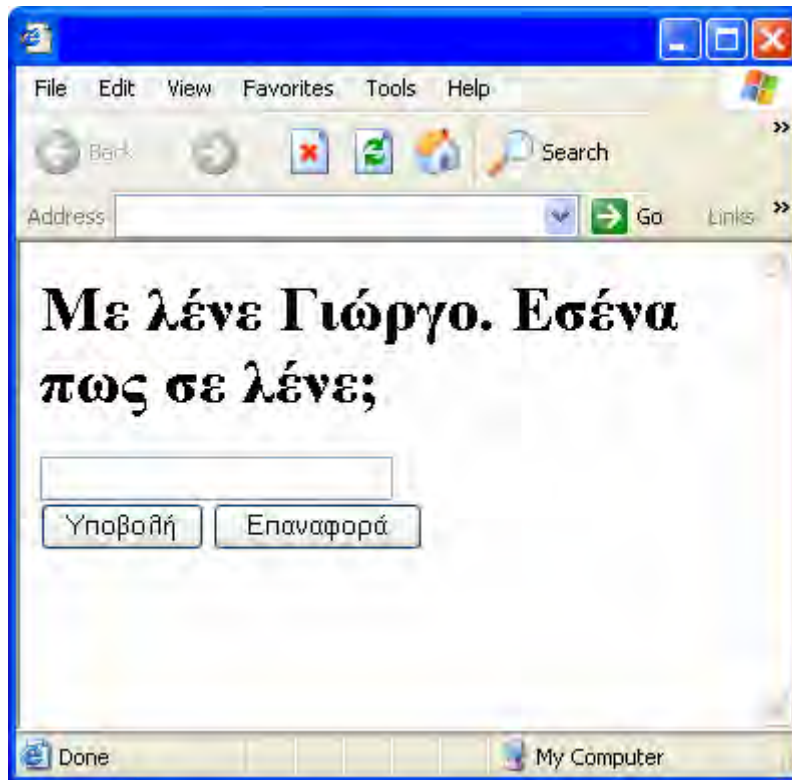
```
<% @ include file="response.jsp" %>
```

Το σενάριο response.jsp, το οποίο απεικονίζεται στο παρακάτω πλαίσιο, χρησιμοποιεί το JavaBean για να ανακτήσει την τιμή του πεδίου κειμένου της φόρμας και να επιστρέψει το μήνυμα στο φυλλομετρητή του πελάτη.

response.jsp

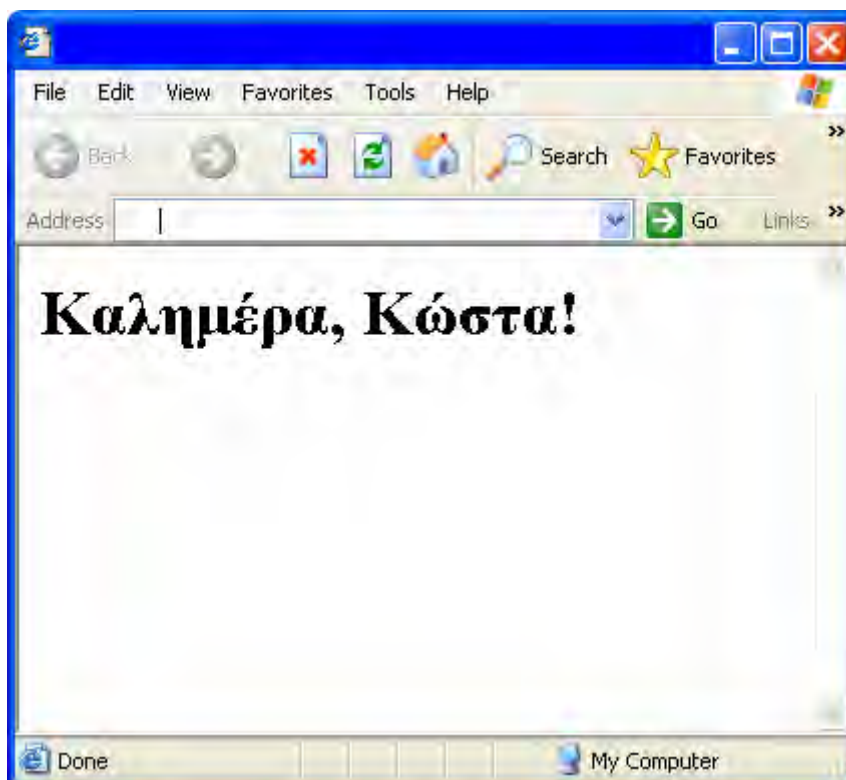
```
<html>  
<table border="0" width="700">  
<h1>Καλημέρα, <jsp:getProperty name="abean" property="username" />!  
</h1>  
</html>
```

Αποθηκεύοντας τα παραπάνω αρχεία στο Root Directory του εξυπηρετητή, μπορούμε να προσπελάσουμε το αρχείο main.jsp με το φυλλομετρητή, οπότε και θα λάβουμε το ακόλουθο αποτέλεσμα:



Σχήμα 4.7

Πληκτρολογώντας στη συνέχεια ένα όνομα και πιέζοντας το πλήκτρο «Υποβολή», λαμβάνουμε το αποτέλεσμα του παρακάτω σχήματος:



Σχήμα 4.8

4.2.5 Δραστηριότητες ανακεφαλαίωσης

A) Συμπληρώστε τα κενά με την κατάλληλη φράση ή σύμβολο.

- Το αντικείμενο της JSP επιτρέπει την αποθήκευση ρυθμίσεων του Java Servlet Engine.
- Στην ASP οι μεταβλητές πριν από τη χρήση τους πρέπει να δηλώνονται με την εντολή

B) Επιλέξτε τη σωστή απάντηση.

<p>Ο κώδικας ASP περικλείεται από τα σύμβολα</p> <p>α) <% και %> β) <! και !> γ) <*> και *> δ) <@ και @></p>	<p>Με την ASP μπορούμε να στείλουμε ένα κείμενο στο φυλλομετρητή με τη συνάρτηση</p> <p>α) Request.QueryString β) Response.Write γ) Server.CreateObject</p>
<p>Ο κώδικας JSP περικλείεται από τα σύμβολα</p> <p>α) <% και %> β) <! και !> γ) <*> και *> δ) <@ και @></p>	<p>Με τη JSP μπορούμε να στείλουμε ένα κείμενο στο φυλλομετρητή με το αντικείμενο</p> <p>α) Request β) Response γ) Session</p>

Γ) Καλείστε να επαναλάβετε την υλοποίηση της εφαρμογής μετατροπής δραχμών σε ευρώ που σας ζητήθηκε στην ενότητα 4.1.4, αυτή τη φορά σε περιβάλλον JSP. Σε ποιο από τα δύο περιβάλλοντα, ASP και JSP, σας φάνηκε η ανάπτυξη πιο εύκολη και γιατί;

Δ) Χρησιμοποιώντας τον παγκόσμιο ιστό, επισκεφθείτε όσο περισσότερους δικτυακούς τόπους μπορείτε, για περίπου 20 λεπτά. Κατά την περιήγησή σας, εντοπίστε από τις καταλήξεις των αρχείων ιστοσελίδων την τεχνολογία που χρησιμοποιούν οι δικτυακοί τόποι που επισκεφθήκατε. Στη συνέχεια, υπολογίστε τα ποσοστά των δικτυακών τόπων που χρησιμοποιούν:

- α) μόνο στατικές σελίδες
- β) τεχνολογία PHP
- γ) τεχνολογία ASP
- δ) τεχνολογία JSP
- ε) άλλες τεχνολογίες (σημειώστε τις αντίστοιχες καταλήξεις των ιστοσελίδων)

Συζητήστε τα ποσοστά που βρήκατε στην τάξη, συγκρίνοντας τα με αυτά των άλλων εκπαιδευομένων.

ΓΛΩΣΣΑΡΙ

αντικείμενο: είναι μια οντότητα η οποία κατέχει ένα σύνολο από *ιδιότητες*, οι οποίες μπορούν να τροποποιηθούν με κατάλληλες *μεθόδους*.

γλώσσες συγγραφής σεναρίων Οι γλώσσες προγραμματισμού που χρησιμοποιούνται στον προγραμματισμό από την πλευρά του πελάτη, ονομάζονται γλώσσες συγγραφής σεναρίων (scripting languages)

Δομημένη Γλώσσα Ερωτήσεων τυποποιημένη γλώσσα πρόσβασης στις βάσεις δεδομένων η οποία αποτελείται από εντολές οι οποίες επιτρέπουν την ανάκτηση και ενημέρωση δεδομένων σε μια βάση.

Δομημένος προγραμματισμός Προγραμματιστική τεχνική η οποία βασίζεται σε εντολές που χρησιμοποιούνται για αναθέσεις και συγκρίσεις τιμών και για τον έλεγχο της ροής και την εκτέλεση συγκεκριμένων τμημάτων του κώδικα.

Επεκτάσιμη Γλώσσα Σήμανσης (Extensible Markup Language - XML) γλώσσα αποθήκευσης και μετάδοσης πληροφοριών στον Παγκόσμιο Ιστό, η οποία διαθέτει εξαιρετικά ευέλικτο συντακτικό και ιδιαίτερη δυνατότητα δόμησης των πληροφοριών.

μεταβλητές Σύμβολα τα οποία αναπαριστούν δεδομένα, των οποίων οι τιμές αλλάζουν κατά τη διάρκεια εκτέλεσης του προγράμματος.

στοιχείο εγγράφου τμήμα εγγράφου XML το οποίο καθορίζει τη λογική δομή του

τελεστές: είναι σύμβολα που ενεργούν πάνω σε μια ή περισσότερες μεταβλητές ή τιμές δεδομένων και εκτελούν αριθμητικές ή λογικές πράξεις επιστρέφοντας μια νέα τιμή.

φύλλα στυλ XSL Τμήματα προγράμματος τα καθορίζουν τη μορφή και τη σειρά παρουσίασης των εγγράφων XML.

χειριστήρια συμβάντος (event handlers) παράμετροι που επιτρέπουν τη συσχέτιση του κώδικα Javascript με συγκεκριμένα γεγονότα που προκύπτουν από τις κινήσεις του φυλλομετρητή ή του χρήστη.

ΕΥΡΕΤΗΡΙΟ

XML	2, 5
Επεκτάσιμη Γλώσσα Σήμανσης	Βλ. XML
HTML	2, 12, 17, 20, 30, 35
Μοντελοποίηση εγγράφων	3
δομή δένδρου	3
XML Εγγραφα	5
DTD	8, 20
Document Type Definition	Βλ. DTD
XSLT	17
γλώσσες συγγραφής σεναρίων	22, 89, 107
Javascript	22, 30, 40
CSS	14, 40
DHTML	40, 56
<i>Dynamic HTML</i>	Βλ. DHTML
<i>συμβάν</i>	23
Αντικείμενα	25, 26, 94, 100
DOM	25, 40, 56
Document Object Model	Βλ. DOM
Event Handlers	30, 107
δομημένος προγραμματισμος	32, 107
μεταβλητές	33, 79, 80, 91, 93, 107
τελεστές	33, 79, 80, 91, 93, 107
συναρτήσεις	34, 81, 92, 93
Έλεγχος ροής	38, 81, 92, 93
MySQL	61,
SQL	62
Structured Query Language	Βλ. SQL
πίνακες	45, 62
PHPMyAdmin	71
PHP	76
phpinfo	77
echo	77, 79, 81
server variables	82
ASP	89
IIS	89, 99
VBScript	40, 89, 91
JSP	98
Java	23, 98
Servlet Engine	98, 101, 106
Apache	99
JRUN	99
Tomcat	99
Ετικέτες	100
Οδηγίες	100
Javabeen	102

ΒΙΒΛΙΟΓΡΑΦΙΑ

- T. Converse, J. Park, C. Morgan, *Php5 and MySQL Bible*, Wiley, 2004.
- S. Koch, *Javascript*, Wiley, 2003.
- Arciniegas, Fabio, *XML Developpers Guide*, McGraw-Hill, 2001.
- *Προγραμματιστικά Εργαλεία για το Διαδίκτυο*, Σχολικό σύγγραμμα ΤΕΕ-Τομέας Πληροφορικής και Δικτύων Η/Υ, ΟΕΔΒ.
- Manas Tungare, *A practical guide to Microsoft Active Server Pages 3.0*, Manas Tungare, 2001.
- *Η Γλώσσα Προγραμματισμού Javascript*, ΚΕΠΛΗΝΕΤ ΦΛΩΡΙΝΑΣ.
- Γ. Γαροφαλάκης, Π. Δεστούνης, Π. Κάπος, Γ. Τζήμας, *Τεχνολογίες Διαδικτύου*, Πανεπιστήμιο Πατρών, Τμήμα Μηχ/κών Η/Υ και Πληροφορικής, 1999.
- D. Goodman, *Javascript and DHTML Cookbook*, O'Reilly, 2003.
- *VisualBuilder JSP Tutorial*, VisualBuilder, 2001 (on-line: <http://visualbuilder.com>)
- H. Bergsten, *JavaServer Pages – 2nd Edition*, O'Reilly, 2002.
- R. Lerdorf, K. Tatroe, *Programming PHP*, O'Reilly, 2002.
- ΚΕΠΛΗΝΕΤ Φλώρινας - <http://dide.flo.sch.gr/Plinet/plinet.html>