

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 4 : ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΕΦΑΡΜΟΓΩΝ ΠΟΥ ΕΚΤΕΛΟΥΝΤΑΙ ΣΤΟΝ ΠΕΛΑΤΗ	4.1-4
4.1 Δυναμικές σελίδες HTML	4.1-4
4.2 Εφαρμογές που εκτελούνται στον πελάτη (client)	4.2-4
4.3 Η γλώσσα σεναρίων JavaScript	4.3-5
4.3.1 Η διαφορά της JavaScript και της Java.....	4.3-6
4.3.2 Γενικά για τις εφαρμογές JavaScript	4.3-7
4.3.3 Μορφή και χαρακτηριστικά των εφαρμογών JavaScript.....	4.3-8
4.3.4 Οι μεταβλητές στη JavaScript.....	4.3-14
4.3.5 Οι τελεστές στη JavaScript	4.3-16
4.3.6 Οι συναρτήσεις στη JavaScript.....	4.3-17
4.3.7 Υπό συνθήκη εντολές	4.3-18
4.3.8 Οι βρόχοι for και while.....	4.3-21
4.3.9 Γενικά για τα αντικείμενα και τις μεθόδους	4.3-23
4.3.10 Το αντικείμενο της συμβολοσειράς.....	4.3-24
4.3.11 Το αντικείμενο του πίνακα (array).....	4.3-28
4.3.12 Το αντικείμενο Math.....	4.3-29
4.3.13 Το αντικείμενο της ημερομηνίας	4.3-29
4.3.14 Το αντικείμενο του Παραθύρου.....	4.3-31
4.3.15 Το αντικείμενο του εγγράφου	4.3-43
4.3.16 Το αντικείμενο της φόρμας.....	4.3-44
4.3.17 Χειρισμός πολλαπλών φυλλομετρητών	4.3-50
4.4 Εισαγωγή στη VBScript.....	4.4-52
4.4.1 Προσθήκη VBScript σε HTML σελίδα	4.4-52
4.4.2 Διαχείριση φυλλομετρητών	4.4-55
4.4.3 Οι μεταβλητές στη VBScript	4.4-58
4.4.4 Οι πίνακες στη VBScript	4.4-62
4.4.5 Τελεστές στη VBScript.....	4.4-63
4.4.6 Συναρτήσεις στη VBScript	4.4-63
4.4.7 Εντολές υπό συνθήκη	4.4-65
4.4.8 Εντολές Βρόχων.....	4.4-68
4.4.9 Φόρμες HTML και VBScript.....	4.4-73
4.5 Εισαγωγή στα Java Applets	4.5-76
4.5.1 Γενικά χαρακτηριστικά της Java	4.5-76
4.5.2 Εφαρμογές Java και Java Applets.....	4.5-77
4.5.3 Βασικά στοιχεία της Java.....	4.5-78
4.5.4 Δημιουργία και εκτέλεση ενός Applet.....	4.5-84
4.5.5 Χρόνος ζωής ενός Applet	4.5-88
4.5.6 Παραδείγματα Applet	4.5-90
4.6 DHTML	4.6-98
4.6.1 Πλεονεκτήματα χρήσης της DHTML.....	4.6-98
4.6.2 Μειονεκτήματα	4.6-99
4.6.3 Διαφορές φυλλομετρητών	4.6-99
4.6.4 Cascading Style Sheets (CSS)	4.6-100

4.6.5 Χρήση του προτύπου CSS-P	4.6-109
4.6.6 Document Object Model (DOM).....	4.6-113
4.6.7 Διαχείριση Γεγονότων	4.6-115
4.7 Ερωτήσεις – Ασκήσεις – Θέματα για ανάπτυξη.....	4.7-117

ΕΙΚΟΝΕΣ

Εικόνα 4.3.1 - Ιεραρχία αντικειμένων ιστοσελίδας. Πηγή (Ladd, O'Donnell, et.al, 1999)	4.3-7
Εικόνα 4.3.2 - Βασική σύνταξη JavaScript	4.3-13
Εικόνα 4.3.3 - Παράδειγμα κλήσης συναρτήσεων	4.3-18
Εικόνα 4.3.4 - Διάγραμμα ροής για τη δήλωση for	4.3-22
Εικόνα 4.3.5 - Μέθοδοι ημερομηνίας.....	4.3-30
Εικόνα 4.3.6 -Προσθέτοντας ένα ρολόι Live Clock στην ιστοσελίδα.....	4.3-31
Εικόνα 4.3.7 - Φόρμα HTML που δημιουργεί και ανοίγει ένα παράθυρο	4.3-34
Εικόνα 4.3.8 - Παράθυρο που εμφανίζει στοιχεία της φόρμας από την οποία δημιουργήθηκε.....	4.3-35
Εικόνα 4.3.9 - Παράδειγμα Alert.....	4.3-40
Εικόνα 4.3.10 - Παράδειγμα Confirm	4.3-41
Εικόνα 4.3.11 - Παράδειγμα Prompt	4.3-42
Εικόνα 4.3.12 - Παράδειγμα Prompt	4.3-43
Εικόνα 4.3.13 - Αλλαγή εικόνων με την κίνηση του ποντικιού – Αρχική εικόνα.....	4.3-46
Εικόνα 4.3.14 - Αλλαγή εικόνων με την κίνηση του ποντικιού – Εικόνα που εμφανίζεται όταν ο δείκτης του ποντικιού μετακινηθεί.....	4.3-47
Εικόνα 4.3.15 - Αλλαγή εικόνων με την κίνηση του ποντικιού – Εμφάνιση νέας σελίδας HTML	4.3-47
Εικόνα 4.3.16 - Παράδειγμα ενός drop down box.....	4.3-49
Εικόνα 4.4.1 - Παράδειγμα ενός απλού VBScript κώδικα	4.4-54
Εικόνα 4.4.2 - Παράθυρο μηνύματος VBScript	4.4-55
Εικόνα 4.4.3 - Παράθυρο για εισαγωγή μηνύματος	4.4-59
Εικόνα 4.4.4 - Παράθυρο μηνύματος VBScript	4.4-59
Εικόνα 4.5.1 - Ιεραρχία αντικειμένων στη Java. Πηγή Java Tutorial (http://java.sun.com)	4.5-84
Εικόνα 4.5.2 – Αποτέλεσμα εκτέλεσης Applet σε μία σελίδα HTML	4.5-87
Εικόνα 4.5.3 – Αποτέλεσμα εκτέλεσης Applet στο AppletViewer	4.5-88
Εικόνα 4.5.4 – Applet που δέχεται παραμέτρους από τη σελίδα HTML	4.5-91
Εικόνα 4.5.5 – Applet που εμφανίζει σχήματα και χρώματα.	4.5-93
Εικόνα 4.5.6 – Applet που επιτρέπει αλληλεπίδραση με το χρήστη.	4.5-96
Εικόνα 4.6.1 – Εφαρμογή διαφορετικών στυλ σε στοιχείο τύπου επικεφαλίδας 1 ..	4.6-104
Εικόνα 4.6.2 – Εφαρμογή στυλ σε στοιχεία της σελίδας HTML	4.6-109
Εικόνα 4.6.3 – Παράδειγμα προσδιορισμού θέσης αντικειμένου	4.6-111
Εικόνα 4.6.4 – Χρήση z-index για την εμφάνιση εικόνας πίσω από κείμενο	4.6-113

ΠΙΝΑΚΕΣ

Πίνακας 4.3.1 – Τελεστές σύγκρισης στη JavaScript.....	4.3-16
Πίνακας 4.3.2 – Λογικοί τελεστές στη Java	4.3-16
Πίνακας 4.3.3 – Εκτέλεση βημάτων εντολής if.....	4.3-19
Πίνακας 4.3.4 – Αποτελέσματα βρόχου for.....	4.3-28
Πίνακας 4.3.5 – Ιδιότητες του αντικειμένου παραθύρου	4.3-32
Πίνακας 4.3.6 – Ιδιότητες εγγράφου	4.3-43
Πίνακας 4.4.1 – Περιγραφή υποτύπων της VBScript.....	4.4-58
Πίνακας 4.5.1 – Τύποι μεταβλητών της Java	4.5-78
Πίνακας 4.5.2 – Τελεστές που συνδυάζουν πράξη με ανάθεση τιμής.....	4.5-79
Πίνακας 4.5.3 – Εντολές βρόχου στη Java	4.5-83
Πίνακας 4.5.4 – Σχόλια στη Java.....	4.5-83
Πίνακας 4.6.1 – Διαφορές φυλλομετρητών στις Τεχνολογίες που υποστηρίζουν και στην υλοποίηση της DHTML. Πηγή http://www.w3schools.com/dhtml/dhtml_intro.asp	4.6-100
Πίνακας 4.6.2 – Γεγονότα που προστέθηκαν στην έκδοση 4.0 της HTML	4.6-115

ΚΕΦΑΛΑΙΟ 4 : ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΕΦΑΡΜΟΓΩΝ ΠΟΥ ΕΚΤΕΛΟΥΝΤΑΙ ΣΤΟΝ ΠΕΛΑΤΗ

4.1 Δυναμικές σελίδες HTML

Η HTML που περιγράφηκε στο προηγούμενο κεφάλαιο, επιτρέπει τη δημιουργία στατικών ιστοσελίδων, δηλαδή ιστοσελίδων των οποίων το περιεχόμενο και η μορφή παραμένουν σταθερά. Η εκτεταμένη, όμως, χρήση του Διαδικτύου οδήγησε στην ανάγκη για δημιουργία δυναμικών ιστοσελίδων, ικανών να καλύψουν πολλές και διαφορετικού είδους απαιτήσεις. Μία ιστοσελίδα είναι δυναμική όταν αλληλεπιδρά με το χρήστη (π.χ. αλλάζει η εμφάνιση ενός μενού επιλογών όταν ο δείκτης του ποντικιού τοποθετείται πάνω σ'αυτό), όταν αλλάζει η μορφή της (π.χ. μετακινούνται λέξεις και αντικείμενα, αλλάζουν δυναμικά εικόνες, γράμματα και χρώματα), είτε όταν μεταβάλλεται το περιεχόμενό της (π.χ. αλλάζουν τα περιεχόμενα ενός πίνακα). Για την ενημέρωση και την αλλαγή των περιεχομένων μιας ιστοσελίδας πολλές φορές απαιτείται απομακρυσμένη αναζήτηση δεδομένων και αλληλεπίδραση της ιστοσελίδας με αρχεία ή βάσεις δεδομένων.

Οι δυνατότητες και απαιτήσεις, που προαναφέραμε, δε μπορούν να καλυφθούν με τη χρήση της HTML. Τα έγγραφα της HTML παρέχουν περιορισμένη επικοινωνία με τους χρήστες, η οποία βασίζεται στην εμφάνιση των υπερ-συνδέσμων επιτρέποντας τη μετάβαση από τη μία ιστοσελίδα στην άλλη. Περισσότερες δυνατότητες αλληλεπίδρασης με τους χρήστες υλοποιούνται με τη χρήση επιπρόσθετων εφαρμογών. Η χρήση των CGI σεναρίων (scripts) που τρέχουν σε εξυπηρετητές διαδικτύου (Web servers) καθώς και η χρήση νεότερων τεχνολογιών όπως Java, ActiveX Controls και DHTML (Dynamic HTML), έδωσαν τη δυνατότητα να αναπτυχθούν πιο ενδιαφέροντες και αλληλεπιδραστικοί δικτυακοί τόποι. Ακόμα περισσότερες δυνατότητες δίνονται με τις γλώσσες σεναρίων (scripting languages), όπως η JavaScript και η VBScript οι οποίες επιτρέπουν τη διασύνδεση διαφορετικών αντικειμένων και τεχνολογιών στην ίδια ιστοσελίδα.

Υπάρχουν δύο τρόποι υλοποίησης δυναμικών ιστοσελίδων. Ο πρώτος βασίζεται στην ανάπτυξη εφαρμογών που εκτελούνται στον εξυπηρετητή (server), όπως σεναρία CGI καθώς και εφαρμογές σε PHP, ASP (Active Server Pages), JSP (Java Server Pages) και Java Servlets. Ο δεύτερος βασίζεται στην ανάπτυξη εφαρμογών που μεταφέρονται και εκτελούνται στην πλευρά του πελάτη (client), όπως JavaScript, VBScript, Java applets και DHTML. Σε αυτό το κεφάλαιο θα ασχοληθούμε με εφαρμογές που εκτελούνται στον πελάτη, ενώ στο επόμενο κεφάλαιο θα παρουσιάσουμε τον προγραμματισμό εφαρμογών από την πλευρά του εξυπηρετητή.

4.2 Εφαρμογές που εκτελούνται στον πελάτη (client)

Με την ανάπτυξη εφαρμογών που εκτελούνται στον πελάτη μπορούν να δημιουργηθούν δυναμικά γραφικά περιβάλλοντα και διεπαφές χρήστη, που παρέχουν αρκετές δυνατότητες αλληλεπίδρασης χρηστών με τις ιστοσελίδες (π.χ. αλλαγή χρώματος και γραμματοσειράς ή εμφάνιση ενός μενού επιλογών όταν ο δείκτης του ποντικιού τοποθετηθεί πάνω σε μία λέξη).

Σε αυτό το κεφάλαιο θα ασχοληθούμε με την ανάπτυξη δυναμικών ιστοσελίδων με τη χρήση των γλωσσών σεναρίων (scripting languages) JavaScript και VBScript καθώς και της γλώσσας προγραμματισμού Java. Ο κώδικας των γλωσσών σεναρίων είναι σχετικά απλός και ενσωματώνεται σε αυτόν της σελίδας HTML. Ο κώδικας αναλύεται (parse) και εκτελείται από το φυλλομετρητή στον οποίο φορτώνεται η σελίδα HTML. Η Java δεν είναι μία γλώσσα σεναρίων, αλλά μία γλώσσα προγραμματισμού, η οποία συμβάλλει στη δημιουργία δυναμικών ιστοσελίδων με την ενσωμάτωση αρχείων ολόκληρων εφαρμογών στη σελίδα HTML. Όπως θα επεξηγηθεί στις επόμενες ενότητες, τα αρχεία των εφαρμογών κατεβαίνουν στο φυλλομετρητή του χρήστη σε μορφή bytecodes και εκτελούνται από την Java Virtual Machine του φυλλομετρητή.

Ένα από τα βασικότερα μειονεκτήματα των εφαρμογών που εκτελούνται στον πελάτη είναι ότι εξαρτώνται από το φυλλομετρητή του πελάτη. Λόγω του ότι οι εφαρμογές μεταφέρονται και εκτελούνται στο μηχάνημα του πελάτη, το αποτέλεσμα της εκτέλεσης και η μορφή της ιστοσελίδας που εμφανίζεται εξαρτώνται από το είδος και από την έκδοση του φυλλομετρητή.

Κάποια άλλα μειονέκτημα των εφαρμογών αυτών είναι ότι καθυστερούν κατά τη μεταφορά τους και την εγκατάστασή τους στο μηχάνημα του πελάτη καθώς και ότι, κατά την εκτέλεσή τους, σπαταλούν υπολογιστική ισχύ και πόρους από το μηχάνημα του πελάτη.

4.3 Η γλώσσα σεναρίων JavaScript

Η **JavaScript** είναι μια γλώσσα σεναρίων, που εκτελείται στην πλευρά του πελάτη προσθέτοντας αλληλεπιδραστικότητα σε ιστοσελίδες. Ως γλώσσα σεναρίων είναι μία σχετικά απλή γλώσσα προγραμματισμού. Η JavaScript επιτρέπει τη συγγραφή μικρών προγραμμάτων, γνωστών ως σενάρια, τα οποία φορτώνονται και εκτελούνται στον φυλλομετρητή του χρήστη. Παραδείγματος χάριν ένα πρόγραμμα το οποίο συλλέγει τα στοιχεία μίας φόρμας και αφού ελέγξει εάν είναι πλήρη και εάν είναι σωστά τα στέλνει στον εξυπηρετητή.

Με τη JavaScript μπορούμε, επίσης, να κάνουμε πράγματα όπως, να εμφανίσουμε επιπλέον πληροφορία για συνδέσμους, να δημιουργήσουμε εφέ προερχόμενα από την κίνηση του ποντικιού, να αλλάξουμε τα περιεχόμενα σελίδων βασισμένα σε ορισμένες συνθήκες, να εμφανίσουμε τυχαία περιεχόμενο σε μια σελίδα, να αλλάξουμε το χρώμα και τη μορφή γραμματοσειρών και εικόνων, να φορτώσουμε περιεχόμενο σε νέα παράθυρα φυλλομετρητή και πλαίσια και να δημιουργήσουμε παράθυρα μηνυμάτων.

Ο κώδικας JavaScript είναι ένας εκτελέσιμος κώδικας ο οποίος ενσωματώνεται στον κώδικα της σελίδας HTML. Η JavaScript είναι μία διερμηνευτική (interpreted) γλώσσα προγραμματισμού, πράγμα που σημαίνει πως ο κώδικάς της μπορεί να εκτελεστεί χωρίς να έχει μεταγλωττιστεί εκ των προτέρων. Η JavaScript αναπτύχθηκε αρχικά από τη Netscape και στη συνέχεια, μία λίγο διαφορετική έκδοσή της με το όνομα JScript, υλοποιήθηκε και από τη Microsoft.

Μία σημαντική χρήση της JavaScript και γενικότερα των γλωσσών σεναρίων (scripting languages) που τρέχουν στους φυλλομετρητές στηρίζεται στη δυνατότητά τους να διασυνδέουν διαφορετικές τεχνολογίες. Οι γλώσσες σεναρίων χρησιμοποιούνται ως ενδιάμεσος για τη διασύνδεση τεχνολογιών όπως Java applets, Dynamic HTML, ActiveX

Controls και αντικείμενα VRML, επιτρέποντας την προσθήκη περισσότερων εφαρμογών και αλληλεπιδραστικότητας στις σελίδες του Διαδικτύου.

4.3.1 Η διαφορά της JavaScript και της Java

Η JavaScript και η Java είναι δύο αντικειμενοστραφείς γλώσσες προγραμματισμού, που δεν έχουν σχεδόν τίποτα κοινό εκτός από το όνομα τους. Η JavaScript σχεδιάστηκε αρχικά για να μοιάζει στην Java, η οποία με τη σειρά της μοιάζει στη C και στη C++. Η διαφορά είναι ότι η Java αναπτύχθηκε ως μία γλώσσα γενικού σκοπού, ενώ η JavaScript περιορίζεται κυρίως στην ανάπτυξη εμπλουτισμένων ιστοσελίδων με απλό και γρήγορο τρόπο.

Παρόλο που η Java είναι τεχνικά μια διερμηνευτική γλώσσα προγραμματισμού, ο κώδικας της μοιάζει με τη C++, με ξεχωριστά αρχεία επικεφαλίδας και κλάσης (header and class files), μεταγλωττισμένα μαζί πριν την εκτέλεση. Είναι αρκετά ισχυρή για την δημιουργία μεγάλων εφαρμογών οι οποίες μπορούν να εισαχθούν και σε μία ιστοσελίδα ως ειδικό αντικείμενο που καλείται "**applet**". Η Java έχει προκαλέσει το ενδιαφέρον λόγω της μεταφερσιμότητας των προγραμμάτων της, δηλαδή της ικανότητάς τους να εκτελούνται σε περιβάλλοντα Windows, Mac, και Unix, χωρίς να απαιτούνται αλλαγές.

Η JavaScript δεν έχει τις ίδιες δυνατότητες με την Java καθώς περιορίζεται κυρίως στην ανάπτυξη δυναμικών ιστοσελίδων. Είναι, όμως, πιο εύκολη στη χρήση της από την Java. Ο κώδικας της JavaScript ενσωματώνεται στην σελίδα HTML ως μία απλή ακολουθία μεθόδων, δηλώσεων και εντολών. Ο κώδικας JavaScript εκτελείται από τους περισσότερους φυλλομετρητές χωρίς να απαιτείται εκ των προτέρων μεταγλώττιση (compilation). Οι συμβατοί φυλλομετρητές, όπως Netscape Navigator (έκδοση 2 ή μεγαλύτερη) και Internet Explorer (έκδοση 3 ή μεγαλύτερη), κατεβάζουν από το Διαδίκτυο τη σελίδα, και ο κώδικας JavaScript φορτώνεται ως μέρος του εγγράφου HTML.

Σημείωση - Τι είναι αντικειμενοστραφής προγραμματισμός

Ο **αντικειμενοστραφής προγραμματισμός** OOP (Object Oriented Programming) είναι μια τεχνική προγραμματισμού (όχι μια δομημένη γλώσσα – δεν απαιτείται μια αντικειμενοστραφής γλώσσα για να προγραμματίσει κανείς με αντικειμενοστραφή τρόπο) σχεδιασμένη για την απλοποίηση πολύπλοκων προγραμματιστικών αρχών / ιδεών.

Ο αντικειμενοστραφής προγραμματισμός περιστρέφεται γύρω από την ιδέα των **κλάσεων (classes)** που παρέχουν ένα σχήμα για την περιγραφή αντικειμένων. Οι κλάσεις αποτελούνται από **ιδιότητες / χαρακτηριστικά (properties ή attributes)** καθώς και από **μεθόδους (methods)**, που ορίζουν ενέργειες των κλάσεων ή χρησιμοποιούνται για την προσπέλαση καθώς και την αλλαγή των τιμών των ιδιοτήτων τους. Όταν δοθούν συγκεκριμένες τιμές στα χαρακτηριστικά / ιδιότητες μιας κλάσης, δημιουργείται ένα αντίγραφο / στιγμιότυπο (**instance**) της κλάσης, το οποίο δεν είναι τίποτα άλλο από ένα συγκεκριμένο αντικείμενο (**object**).

Παραδείγματος χάριν μία κλάση στη JavaScript, είναι η κλάση του παραθύρου (window) με χαρακτηριστικά ένα url, το όνομα του παραθύρου (name), ένα μήνυμα που θα εμφανιστεί στη μπάρα κατάστασης του παραθύρου (statusbar), το πλάτος (width), το ύψος (height) κ.α. Η κλάση window έχει και κάποιες μεθόδους όπως η open() και η close(). Με την εντολή window.open ("http://www.unipi.gr", "my_window"),

δημιουργείται και ανοίγει ένα στιγμιότυπο (instance) παραθύρου, που δείχνει στη διεύθυνση “http://www.unipi.gr” και έχει όνομα “my_window”.

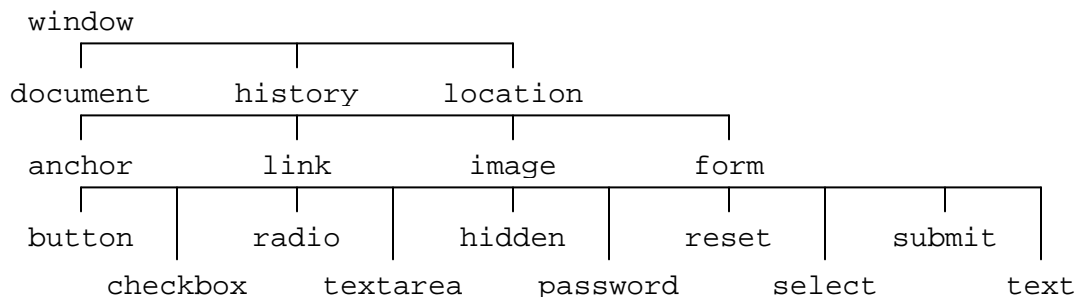
Ένα αντικείμενο είναι ένα μαύρο κουτί, που αποθηκεύει πληροφορία. Κάποια από αυτήν την πληροφορία μπορεί να είναι άμεσα προσβάσιμη, όπως οι τιμές των ιδιοτήτων του αντικειμένου (π.χ. window.status = "A message"), και άλλη ίσως να απαιτεί την χρήση μιας μεθόδου για την πρόσβασή της. Οι μέθοδοι χρησιμοποιούνται κυρίως στις περιπτώσεις που απαιτούνται έλεγχοι τόσο για το είδος όσο και για το μέγεθος της πληροφορίας που θα αποθηκευθεί στο αντικείμενο. Χρησιμοποιούνται, δηλαδή, για να ελέγχουν τις αλλαγές και τα δεδομένα που εισάγει ο χρήστης, καθώς και για να διαβάζουν, να μετατρέπουν και να επιστρέφουν την πληροφορία στο χρήστη.

Οι μέθοδοι χρησιμοποιούνται, επίσης, για να ορίσουν κάποιες ενέργειες των κλάσεων, όπως η μέθοδος open() της κλάσης window που παρουσιάστηκε παραπάνω.

4.3.2 Γενικά για τις εφαρμογές JavaScript

Η JavaScript παρέχει μία ολοκληρωμένη συλλογή από συναρτήσεις και εντολές, που επιτρέπουν την διαχείριση αριθμητικών δεδομένων ή συμβολοσειρών, την υλοποίηση μαθηματικών υπολογισμών, το άνοιγμα παραθύρων, τη μετάβαση σε νέες ιστοσελίδες, καθώς και την προσθήκη ήχων. Με την JavaScript, μπορούμε, επίσης, να ελέγξουμε μια φόρμα για λάθη, γράφοντας μια απλή ακολουθία εντολών / δηλώσεων, ένα μπλοκ εντολών if-then, στην κορυφή της ιστοσελίδας.

Η JavaScript στηρίζεται σε αντικειμενοστραφή λογική. Για την JavaScript όλα τα στοιχεία της ιστοσελίδας είναι αντικείμενα και κάθε αναφορά σε αυτά γίνεται με βάση την ιεραρχική δομή των αντικειμένων (Εικόνα 4.3-1) που υποστηρίζουν οι φυλλομετρητές.



Εικόνα 4.3.1 - Ιεραρχία αντικειμένων ιστοσελίδας. Πηγή (Ladd, O'Donnell, et.al, 1999)

Ολόκληρο το παράθυρο της ιστοσελίδας είναι ένα αντικείμενο window ενώ το περιεχόμενο της ιστοσελίδας είναι ένα αντικείμενο με όνομα document. Κάθε στοιχείο μιας σελίδας, δηλαδή κάθε πίνακας, φόρμα, πλήκτρο (button), εικόνα ή σύνδεσμος στην ιστοσελίδα είναι επίσης ένα αντικείμενο. Με βάση την ιεραρχία αυτή για να αναφερθούμε στην τιμή (value) ενός πεδίου κειμένου (text1) μιας φόρμας (form1), γράφουμε window.document.form1.text1.value. Στην JavaScript λόγω του ότι η εμβέλεια του κώδικα περιορίζεται στο επίπεδο του παραθύρου, δε χρειάζεται να αναφερθούμε στο αντικείμενο window, αλλά στα αντικείμενα που βρίσκονται κάτω από αυτό, δηλαδή document.form1.text1.value.

Κάθε αντικείμενο έχει ορισμένες ιδιότητες (attributes) στις οποίες αποθηκεύεται πληροφορία για το αντικείμενο (τιμές των ιδιοτήτων). Για παράδειγμα, το αντικείμενο “έγγραφο” μιας ιστοσελίδας (document) έχει διάφορες ιδιότητες όπως π.χ. το χρώμα του φόντου (bgcolor). Για να προσπελάσουμε την τιμή του χρώματος του φόντου ενός εγγράφου (document) γράφουμε document.bgcolor. Μπορούμε, επίσης, να αλλάξουμε το χρώμα της σελίδας σε κόκκινο γράφοντας: document.bgcolor="red". Αντίστοιχα, για να προσπελάσουμε ή να αλλάξουμε τα περιεχόμενα (ή την τιμή) ενός πεδίου κειμένου με το όνομα "password" σε μια φόρμα με το όνομα "entryform" καλούμε το document.entryform.password.value.

Ο κώδικας JavaScript ενσωματώνεται σε μία ιστοσελίδα και οι εντολές του εκτελούνται είτε αμέσως μόλις φορτωθεί η ιστοσελίδα στο φυλλομετρητή, είτε όταν συμβούν ορισμένα γεγονότα τα οποία έχουν εκ των προτέρων οριστεί.

Γεγονότα

Η JavaScript μπορεί να παρακολουθεί γεγονότα (ενέργειες του χρήστη) που συμβαίνουν πάνω σε αντικείμενα της σελίδας HTML. Η εμφάνιση / εκτέλεση κάποιων γεγονότων μπορούν να προκαλέσουν την εκτέλεση ενεργειών ή συναρτήσεων. Εάν για παράδειγμα θέλουμε όταν πατηθεί ένα πλήκτρο (δηλαδή όταν συμβεί ένα γεγονός onClick) να εκτελεστεί μία συνάρτηση με όνομα my_function(), θα πρέπει στη δήλωση του πλήκτρου (button) να συμπεριλάβουμε την εντολή onclick="my_function()". Άλλα γεγονότα που μπορούν να συμβούν στα αντικείμενα της σελίδας HTML είναι τα ακόλουθα: onMouseOver (όταν ο δείκτης του ποντικιού βρεθεί πάνω από κάποιο αντικείμενο), onMouseOut (όταν ο δείκτης του ποντικιού φύγει πάνω από κάποιο αντικείμενο), onFocus (όταν εστιάσουμε πάνω σε κάποιο αντικείμενο / όταν ένα αντικείμενο γίνει ενεργό) , onBlur (όταν χαθεί η εστίαση από κάποιο αντικείμενο), onLoad (όταν “φορτωθεί” κάποιο αντικείμενο (π.χ. μία φόρμα)) και unload .

4.3.3 Μορφή και χαρακτηριστικά των εφαρμογών JavaScript

Ο κώδικας JavaScript συνήθως τοποθετείται πάνω στο έγγραφο HTML, είτε στην επικεφαλίδα (HEAD) είτε στο σώμα (BODY) της σελίδας HTML. Σε ένα έγγραφο HTML μπορούν να υπάρχουν περισσότερα του ενός σενάρια. Η γενική σύνταξη είναι η ακόλουθη:

```
<SCRIPT LANGUAGE="JavaScript">
εδώ γράφουμε τις εντολές του Script
</SCRIPT>
```

Παρατηρούμε ότι για να ενσωματώσουμε τον κώδικα JavaScript μέσα σε αυτόν της HTML χρησιμοποιούμε την ετικέτα <SCRIPT> με το χαρακτηριστικό LANGUAGE να είναι ίσο με JavaScript. Ένας άλλος τρόπος δήλωσης της γλώσσας μπορεί να γίνει με το χαρακτηριστικό TYPE όπως φαίνεται στην παρακάτω γραμμή κώδικα:

```
<SCRIPT TYPE="text/javascript">
```


Παρόλο που η JavaScript είναι μία πολύ απλή γλώσσα, έχει πολλές δυνατότητες και ο κώδικάς της μοιάζει με τον κώδικα άλλων γλωσσών προγραμματισμού όπως της Java, C++ και της Visual Basic. Κάποια χαρακτηριστικά της δομής της JavaScript είναι τα εξής:

- Υπάρχει διαφορά ανάμεσα στα πεζά και τα κεφαλαία γράμματα
- Υπάρχει ευελιξία στις δηλώσεις των μεταβλητών. Κάποιες δηλώσεις μπορούν να καταλαμβάνουν περισσότερες από μία γραμμές, αλλά και σε μία γραμμή μπορούν να τοποθετούνται περισσότερες από μία δηλώσεις. Ανάμεσα στις δηλώσεις πρέπει απαραίτητως να μπαίνει ερωτηματικό.
- Τα άγκιστρα ({ και }) χρησιμοποιούνται για να ομαδοποιούν εντολές, όπως το να διαχωρίζουν τις εντολές μίας συνάρτησης, είτε τις εντολές ενός for ή while βρόχου.
- Στις περισσότερες γλώσσες προγραμματισμού, στο τέλος κάθε εντολής υπάρχει ερωτηματικό. Στη JavaScript αυτό δεν είναι απαραίτητο παρά μόνο στις δηλώσεις μεταβλητών.

Οι εφαρμογές JavaScript απαρτίζονται από:

- τις εντολές και τις συναρτήσεις που ορίζουν τις ενέργειες που θα εκτελέσει ο φυλλομετρητής και
- τις αναφορές σε αυτές τις συναρτήσεις (κλήσεις των συναρτήσεων).

Στην επικεφαλίδα (HEAD) ή στο BODY της σελίδας HTML περιγράφονται οι εντολές JavaScript που θα εκτελέσει ο φυλλομετρητής. Όπως φαίνεται στο παρακάτω δείγμα κώδικα JavaScript, οι εντολές περιγράφονται πάντα ανάμεσα σε σχόλια (<!-- and //-->). Είναι σημαντικό ο κώδικας σχολίων να είναι σε ξεχωριστές δικές του σειρές. Αν βάλουμε τα σχόλια στην ίδια γραμμή με κάποιον κώδικα, τότε αυτή η γραμμή θα μετρηθεί ως σχόλιο και το όλο σενάριο δε θα λειτουργεί.

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
  <!-- Beginning of JavaScript -
  (JavaScript code)
  // - End of JavaScript - -->
</SCRIPT>
</HEAD>
<BODY> ... </BODY>
</HTML>
```

Η JavaScript δίνει, επίσης τη δυνατότητα δημιουργίας εξωτερικών scripts. Δηλαδή scripts των οποίων ο κώδικας τοποθετείται σε ένα εξωτερικό αρχείο. Το αρχείο αυτό καλείται μέσα από τη σελίδα HTML, όπως φαίνεται στο παρακάτω παράδειγμα:

```
<HTML>
<HEAD>
<SCRIPT src="my_script.js" >
</SCRIPT>
</HEAD>
<BODY> ... </BODY>
</HTML>
```

Η σύνταξη και τα βασικά χαρακτηριστικά των εφαρμογών JavaScript παρουσιάζονται στα παραδείγματα που ακολουθούν.

Παράδειγμα απλού σεναρίου JavaScript:

Το παράδειγμα αυτό χρησιμοποιεί την εντολή `document.write` για να εμφανίσει το κείμενο “Welcome to the world of JavaScript” στη σελίδα HTML.

```
<HTML>
<HEAD><TITLE>JavaScript Example</TITLE>
<SCRIPT Language = "JavaScript">
    document.write("Welcome to the world of Java");
</SCRIPT>
</HEAD>
<BODY></BODY>
</HTML>
```

Το ίδιο αποτέλεσμα προκύπτει εάν ο κώδικας JavaScript τοποθετηθεί μέσα στο `BODY` της σελίδας HTML:

```
<HTML>
<HEAD><TITLE>JavaScript Example</TITLE>
</HEAD>
<BODY>
<SCRIPT Language = "JavaScript">
    document.write("Welcome to the world of Java");
</SCRIPT>
</BODY>
</HTML>
```

Το ίδιο κείμενο θα μπορούσε, επίσης, να εμφανιστεί στη σελίδα με κάποια συγκεκριμένη μορφοποίηση (format). Εάν για παράδειγμα θέλουμε το κείμενο να έχει τη μορφή επικεφαλίδας μπορούμε να γράψουμε:

```
document.write("<H1> Welcome to the world of Java´</H1>")
```

Παράδειγμα σεναρίου που περιλαμβάνει συνάρτηση:

Στον κώδικα JavaScript, εκτός από απλές εντολές, μπορούν να περιλαμβάνονται και συναρτήσεις (functions). Οι συναρτήσεις που περιγράφονται στο HEAD της HTML σελίδας, καλούνται μέσα από το κύριο τμήμα της σελίδας (BODY) με μία απλή αναφορά στο όνομά τους.

Οι συναρτήσεις JavaScript μπορούν να κληθούν και να ενεργοποιηθούν από γεγονότα που συμβαίνουν σε αντικείμενα της σελίδας. Στο παράδειγμα αυτό, όταν πατηθεί το πλήκτρο (button) μίας φόρμας (form), καλείται μία συνάρτηση με όνομα MsgBox(). Η MsgBox() χρησιμοποιεί την εντολή Alert για να δημιουργήσει ένα παράθυρο / κουτί μηνύματος (message box) που εμφανίζει κάποιο κείμενο ή κάποιον αριθμό.

- Το Alert("Hello world") θα εμφανίσει "Hello World" σε ένα κουτί μηνύματος.
- Το Alert(SomeText) θα υποθέσει ότι το SomeText είναι μια μεταβλητή και θα εμφανίσει την τιμή της μεταβλητής αυτής. Έστω ότι στην μεταβλητή SomeText εισάγουμε την τιμή "JavaScript course", χρησιμοποιώντας την εντολή SomeText="JavaScript course". Τότε η κλήση της Alert με παράμετρο το SomeText, δηλαδή η εντολή Alert(SomeText) θα εμφανίσει ένα κουτί μηνύματος με το κείμενο "JavaScript course".

Στο παράδειγμα που ακολουθεί η φόρμα έχει 2 αντικείμενα, ένα πεδίο κειμένου και ένα κουμπί. Όταν ο χρήστης πατήσει το πλήκτρο της φόρμας, θα ενεργοποιηθεί η MsgBox και θα εμφανίσει σ' ένα κουτί μηνύματος το κείμενο που πληκτρολογήθηκε στο πεδίο κειμένου.

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
    <!-- Beginning of JavaScript -
    function MsgBox (textstring) {
        alert (textstring) }
    // - End of JavaScript - -->
</SCRIPT>
</HEAD>
<BODY>
```

```
<FORM>
<INPUT NAME="text1" TYPE=Text>
<INPUT NAME="submit" TYPE=Button VALUE="Show Me"
onClick="MsgBox(form.text1.value)">
</FORM>
</BODY>
</HTML>
```

Όλες οι φόρμες ξεκινούν με μια ετικέτα `<FORM>` και τελειώνουν με την ετικέτα `</FORM>`. Μέσα στη φόρμα μπορούν να τοποθετηθούν διάφορα αντικείμενα όπως πεδία κειμένων, λίστες επιλογής και πλήκτρα. Τα αντικείμενα αυτά με τα οποία μπορεί να αλληλεπιδράσει ο χρήστης ορίζονται μέσα σε μία ετικέτα `<INPUT . . . >` και προσδιορίζονται / περιγράφονται από πεδία όπως τα `NAME`, `TYPE` και `VALUE`.

```
<INPUT NAME="text1" TYPE=Text>
```

Το πεδίο `TYPE` προσδιορίζει το είδος του αντικειμένου. Το `TYPE` δείχνει στο φυλλομετρητή εάν θα δημιουργήσει ένα πεδίο κειμένου ή ένα πλήκτρο (`button`) ή ένα `check box`.

Στο πεδίο `NAME` τοποθετείται το όνομα του αντικειμένου, το οποίο χρησιμοποιούμε όταν θέλουμε να αναφερθούμε στο συγκεκριμένο αντικείμενο. Κάθε αντικείμενο έχει ένα όνομα. Το πεδίο κειμένου στο συγκεκριμένο παράδειγμα έχει όνομα `text1` και το χρησιμοποιούμε για να δηλώσουμε στη συνάρτηση ποιο από τα πεδία έχει το κείμενο που θέλουμε να εμφανίσουμε στο κουτί μηνύματος. Ολόκληρο το όνομα του πεδίου κειμένου είναι `form.text1`.

Το πεδίο `VALUE` αναφέρεται στο κείμενο που εμφανίζει το κάθε αντικείμενο. Στον ορισμό του πεδίου κειμένου `text1` δεν είχε δοθεί κάποιο `value`, και το πεδίο κειμένου εμφανίζεται κενό. Το `value` παίρνει την τιμή του κειμένου που πληκτρολογεί ο χρήστης μέσα στο πεδίο. Η τιμή ή το κείμενο, που θα πληκτρολογηθεί, καλείται με την εντολή:

```
form.text1.value
```

Πατώντας το πλήκτρο (`button`) της φόρμας ο χρήστης μπορεί να ενεργοποιήσει την εκτέλεση μιας συνάρτησης. Το πλήκτρο είναι ένα αντικείμενο φόρμας και ορίζεται με τα χαρακτηριστικά `NAME`, `TYPE` και `VALUE`.

```
<INPUT NAME="submit" TYPE=Button VALUE="Show Me"
onClick="MsgBox(form.text1.value)">
```

Το `TYPE` δείχνει ότι πρόκειται για αντικείμενο τύπου πλήκτρου (`button`).

Το `NAME` είναι το όνομα του πλήκτρου και χρησιμοποιείται στη JavaScript για αναφορά στο πλήκτρο, αλλά δεν είναι συνήθως σημαντικό.

Το VALUE προσδιορίζει το κείμενο που θα αναγράφεται στο πλήκτρο.

Στο πλήκτρο αυτό ορίζεται, επίσης, ένα γεγονός onClick, που συμβαίνει όταν ο χρήστης πατήσει το πλήκτρο αυτό. Η εντολή:

```
onClick="MsgBox(form.text1.value)"
```

δηλώνει την ενέργεια / συνάρτηση που θα εκτελεστεί μετά το πάτημα του πλήκτρου. Στο παράδειγμα αυτό ενεργοποιείται η MsgBox, και ακολουθούνται τα βήματα που φαίνονται στην Εικόνα 4.3-2:

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
<!-- Beginning of JavaScript -
function MsgBox (textstring) {
alert (textstring) }
// - End of JavaScript - -->
</SCRIPT>
</HEAD>
<BODY>
<FORM>
<INPUT NAME='text1' TYPE=Text>
<INPUT NAME="submit" TYPE=Button VALUE="Show Me" onClick="MsgBox(form.text1.value)">
</FORM>
</BODY>
</HTML>
```

Εικόνα 4.3.2 - Βασική σύνταξη JavaScript

Όταν ο χρήστης πατήσει το πλήκτρο, το πρόγραμμα θα καλέσει την συνάρτηση MsgBox και θα χρησιμοποιήσει την τιμή του αντικειμένου της φόρμας με το όνομα text1 σαν τη μεταβλητή το υ (1) Η τιμή το υ text1 (form.text1.value) θα δοθεί στη συνάρτηση MsgBox (2) και αυτή θα το περάσει στην εντολή Alert που με τη σειρά της θα εμφανίσει το κείμενο σε ένα κουτί μηνύματος (3).

Παράδειγμα σεναρίου που εκτελείται με φόρμα:

Στο παράδειγμα αυτό παρουσιάζεται μία φόρμα η οποία περιλαμβάνει 4 πλήκτρα. Κάθε φορά που ο χρήστης επιλέγει ένα από αυτά τα πλήκτρα, αλλάζει το χρώμα του φόντου της σελίδας HTML.

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
<!-- Beginning of JavaScript -
function changecolor(code) {
```

```

        document.bgColor=code
    }
    // - End of JavaScript - -->
</SCRIPT>
</HEAD>
<BODY>
<form>
    <input type="button" name="Button1" value="RED"
    onclick="changecolor('red')">
    <input type="button" name="Button2" value="GREEN"
    onclick="changecolor('green')">
    <input type="button" name="Button3" value="BLUE"
    onclick="changecolor('blue')">
    <input type="button" name="Button4" value="WHITE"
    onclick="changecolor('white')">
</form>
</BODY>
</HTML>

```

Η συνάρτηση του παραδείγματος είναι η `changecolor(code)` και αποτελείται από την εντολή: `document.bgColor=code`.

Το `code` είναι μία μεταβλητή η οποία παίρνει την τιμή οποιουδήποτε χρώματος ορίσουμε. Το `bgColor` είναι το χρώμα του φόντου. Αν γράψουμε `document.bgColor=code`, το φόντο θα πάρει την τιμή της μεταβλητής `code`. Εάν, δηλαδή, το `code` έχει την τιμή "red" η εντολή `document.bgColor=code` (= "red") κάνει το φόντο κόκκινο.

Στη φόρμα δηλώνουμε τέσσερα πλήκτρα που καλούν την `changecolor()` με διαφορετικό χρώμα. Όταν, για παράδειγμα, πατηθεί το δεύτερο πλήκτρο (`button2`), ενεργοποιείται η εντολή:

```

OnClick="changecolor('green')"

```

η οποία λέει στο πρόγραμμα να εκτελέσει τη συνάρτηση με όνομα `changecolor(code)`, όπου `code='green'`.

4.3.4 Οι μεταβλητές στη JavaScript

Οι μεταβλητές δεσμεύουν ένα χώρο μνήμης στον οποίο αποθηκεύουν πληροφορία. Οι τιμές που παίρνουν οι μεταβλητές μπορούν να αλλάξουν κατά τη

διάρκεια του προγράμματος. Για να ορίσουμε μεταβλητές στη JavaScript χρησιμοποιούμε τη δήλωση var:

```
var my_number = a value
```

μπορούμε όμως να τις ορίσουμε και χωρίς το var δηλαδή όπως παρακάτω:

```
my_number = a value
```

Τα ονόματα των μεταβλητών θα πρέπει να ξεκινάνε πάντα με κάποιο γράμμα ή το χαρακτήρα της υπογράμμισης. Επίσης, θα πρέπει να λαμβάνεται υπόψη πως στη JavaScript υπάρχει διαφορά ανάμεσα στα κεφαλαία και τα μικρά γράμματα.

Για να αναθέσουμε μία τιμή σε μία μεταβλητή γράφουμε:

```
var my_number= 2 ή my_number= 2
```

Με παρόμοιο τρόπο μπορούμε, επίσης, να ορίσουμε και να δώσουμε τιμή σε μία μεταβλητή string:

```
var my_string= "A string" ή my_string= "A string"
```

Σημείωση:

Όταν μία μεταβλητή δηλωθεί μέσα σε μία συνάρτηση ονομάζεται τοπική μεταβλητή και έχει ισχύ μόνο μέσα στην συνάρτηση αυτή. Όταν ολοκληρωθεί η εκτέλεση της συνάρτησης η μεταβλητή αυτή καταστρέφεται. Επομένως μπορούμε σε διαφορετικές συναρτήσεις να έχουμε τοπικές μεταβλητές με το ίδιο όνομα.

Εάν μία μεταβλητή δηλωθεί εκτός των συναρτήσεων της JavaScript μπορεί να κληθεί και να χρησιμοποιηθεί από όλες τις συναρτήσεις. Η μεταβλητή ξεκινάει να έχει ισχύ από τη στιγμή που δηλώνεται (δηλαδή από τη στιγμή που φορτώνεται η σελίδα HTML) και σταματάει να έχει ισχύ όταν κλείσει η σελίδα.

Παράδειγμα

Στο παράδειγμα αυτό δηλώνεται μία μεταβλητή WelcomeText στην οποία δίνεται η τιμή "Welcome to the world of Java". Η μεταβλητή αυτή καλείται από τη μέθοδο writeln η οποία τυπώνει, σε μία γραμμή του εγγράφου (document) της σελίδας τη συμβολοσειρά: "<H1> Welcome to the world of Java </H1>".

```
<HTML>
<HEAD><TITLE>JavaScript Example</TITLE>
<SCRIPT Language = "JavaScript">
    var WelcomeText;
    WelcomeText="Welcome to the world of Java";
    document.writeln("<H1>" + WelcomeText + "</H1>");
```

```

</SCRIPT>

</HEAD>

<BODY></BODY>

</HTML>

```

Το παραπάνω πρόγραμμα έχει ως αποτέλεσμα, η πρόταση “Welcome to the world of Java” να εμφανιστεί στη σελίδα HTML με τη μορφή επικεφαλίδας.

4.3.5 Οι τελεστές στη JavaScript

Η JavaScript υποστηρίζει τους συνηθισμένους μαθηματικούς τελεστές (+, -, *, /, %). Το (%) επιστρέφει το υπόλοιπο μιας διαίρεσης. (π.χ. 10%3 επιστρέφει 1) Η JavaScript υποστηρίζει, επίσης, τους τελεστές αυτόματης αύξησης (++) και αυτόματης μείωσης (--), οι οποίοι αυξάνουν ή μειώνουν αντίστοιχα την τιμή μιας μεταβλητής κατά 1. Έστω ότι έχουμε τη μεταβλητή x, το x++ θα είναι ισοδύναμο με x+1 ενώ το x-- με x-1.

Όπως είδαμε στα προηγούμενα παραδείγματα η ανάθεση τιμής γίνεται με τον τελεστή “=” (π.χ. το x=2 δίνει στη μεταβλητή x την τιμή 2). Στη JavaScript υπάρχουν και σύνθετοι τελεστές ανάθεσης (+=, -=, *=, /=, %=), όπου το x+=y είναι ισοδύναμο με x=x+y, το x-=y με x=x-y, κ.λ.π.

Η JavaScript υποστηρίζει τους ακόλουθους τελεστές σύγκρισης:

Πίνακας 4.3.1 – Τελεστές σύγκρισης στη JavaScript

Τελεστής	Επεξήγηση	Παράδειγμα
==	Ελέγχει εάν δύο τιμές είναι ίσες	Το 5==2 επιστρέφει false
!=	Ελέγχει εάν δύο τιμές είναι άνισες	Το 5!=2 επιστρέφει true
>	Ελέγχει εάν μία τιμή είναι μεγαλύτερη από μία άλλη	Το 5>2 επιστρέφει true
>=	Ελέγχει εάν μία τιμή είναι μεγαλύτερη ή ίση από μία άλλη	Το 5>=2 επιστρέφει true
<	Ελέγχει εάν μία τιμή είναι μικρότερη από μία άλλη	Το 5<2 επιστρέφει false
<=	Ελέγχει εάν μία τιμή είναι μικρότερη ή ίση από μία άλλη	Το 5<=2 επιστρέφει false

καθώς και τους ακόλουθους λογικούς τελεστές:

Πίνακας 4.3.2 – Λογικοί τελεστές στη Java

Τελεστής	Επεξήγηση	Παράδειγμα
&&	και (and)	Εάν x=2 και y=3 το (x==2 && y>1) επιστρέφει true
	ή (or)	Εάν x=2 και y=3 το (x<2 y>1) επιστρέφει true
!	όχι (not)	Εάν x=2 το !(x==3) επιστρέφει true

4.3.6 Οι συναρτήσεις στη JavaScript

Μία συνάρτηση περιλαμβάνει κάποιες εντολές / ενέργειες οι οποίες εκτελούνται όταν συμβεί ένα γεγονός είτε όταν κληθεί μία συνάρτηση. Η JavaScript υποστηρίζει δικές της συναρτήσεις όπως την “alert” (π.χ. alert(“A message”)) αλλά επιτρέπει και τον προσδιορισμό νέων συναρτήσεων. Στη JavaScript υπάρχουν δύο είδη συναρτήσεων. Οι συναρτήσεις που εκτελούν κάποιες ενέργειες χωρίς να επιστρέφουν τιμή και οι συναρτήσεις που επιστρέφουν κάποια τιμή. Και τα δύο είδη συναρτήσεων ορίζονται με τη δήλωση function, όπως φαίνεται στα παρακάτω παραδείγματα:

```
function WriteText() {  
    document.writeln(“Welcome to the world of JavaScript”);  
}
```

Οι συναρτήσεις που επιστρέφουν τιμή χρησιμοποιούν τη δήλωση “return”:

```
function SumNumbers(){  
    var x=2;  
    var y=3;  
    result=x+y;  
    return result;  
}
```

Οι συναρτήσεις μπορούν, επίσης, να περιλαμβάνουν ορίσματα:

```
function WriteText(WelcomeText) {  
    document.writeln(WelcomeText);  
}
```

```
function SumNumbers(x,y){  
    result=x+y;  
    return result;  
}
```

Για να καλέσουμε μία συνάρτηση γράφουμε το όνομά της (π.χ. WriteText ()) και δίνουμε τιμή στα ορίσματά της εάν υπάρχουν (π.χ. WriteText (“Welcome to JavaScript”)). Οι συναρτήσεις που επιστρέφουν τιμή καλούνται συνήθως με ανάθεση της τιμής τους σε μία μεταβλητή. Το παράδειγμα που ακολουθεί παρουσιάζει τον τρόπο με τον οποίο καλούνται οι συναρτήσεις:

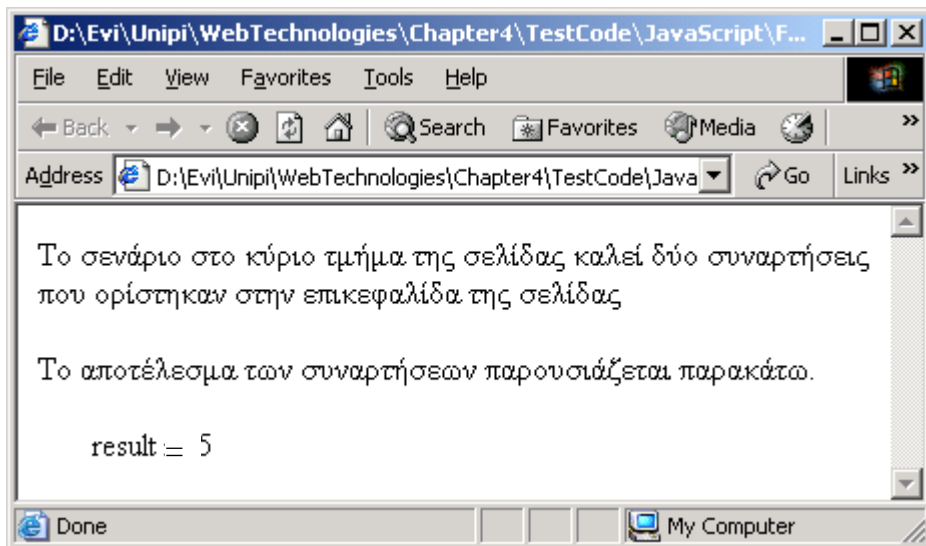
```
<html>  
<head>  
<script language="JavaScript">  
function WriteText() {  
    document.writeln(“result = ”);  
}
```

```

}
function total(x,y){
    return x+y;
}
</script>
</head>
<body>
<p>Το σενάριο στο κύριο τμήμα της σελίδας καλεί δύο συναρτήσεις
που ορίστηκαν στην επικεφαλίδα της σελίδας</p>
<p>Το αποτέλεσμα των συναρτήσεων παρουσιάζεται παρακάτω.</p>
<script language="JavaScript">
    WriteText();
    result=total(2,3);
    document.write(result);
</script>
</body>
</html>

```

Με την εκτέλεση του JavaScript κώδικα στο φυλλομετρητή θα εμφανιστεί η εικόνα 4.3-3:



Εικόνα 4.3.3 - Παράδειγμα κλήσης συναρτήσεων

4.3.7 Υπό συνθήκη εντολές

Οι υπό συνθήκη εντολές επιτρέπουν τον έλεγχο ορισμένων συνθηκών και ανάλογα με το αν οι συνθήκες είναι αληθείς ή ψευδείς εκτελούν κάποιες εντολές. Η JavaScript υποστηρίζει τις υπο συνθήκη εντολές: if, if...else και switch.

Οι εντολές if και if...else

Οι if και if...else χρησιμοποιούνται στην περίπτωση που έχουμε να ελέγξουμε κάποιες συνθήκες και ανάλογα με τα αποτελέσματα να εκτελέσουμε κάποιες εντολές.

IF (μια σύγκριση)

{μια ακολουθία εντολών αν η σύγκριση είναι αληθής}

IF (μια σύγκριση)

{μια ακολουθία εντολών αν η σύγκριση είναι αληθής}

ELSE {ακολουθία εντολών αν η σύγκριση είναι ψευδής}

Παράδειγμα:

Ας υποθέσουμε ότι έχουμε μία φόρμα αίτησης για μία υπηρεσία. Οι χρήστες συμπληρώνουν τα στοιχεία τους και την ηλικία τους και επιλέγουν το πλήκτρο παράδοσης της φόρμας (submit). Θέλουμε όταν πατηθεί το πλήκτρο παράδοσης (submit) όλοι οι μεγάλοι σε ηλικία πολίτες (>64) να λάβουν διαφορετικό μήνυμα από τους υπόλοιπους.

```
<SCRIPT LANGUAGE= "JavaScript">
function PresentMessage() {
    age=form.age.value;
    if (age>64){
        alert("Your form has been submitted. Ask about our
        Senior Discounts")
    }
    else{
        alert("Your form has been submitted.")
    }
}
</SCRIPT>
```

Για την υλοποίηση του σεναρίου αυτού ακολουθήθηκαν τα βήματα που περιγράφονται στον Πίνακα 3.6.3:

Πίνακας 4.3.3 – Εκτέλεση βημάτων εντολής if

age=form.age.value	Μεταφορά των περιεχομένων ενός πεδίου ηλικίας της φόρμας σε μια μεταβλητή με το όνομα age.
if (age>64){ alert("Your form has been submitted. Ask	Ελέγχεται εάν η ηλικία του χρήστη είναι μεγαλύτερη από 64 και εάν ναι, δηλαδή εάν η συνθήκη (age>64) είναι αληθής, εμφανίζεται ένα κουτί μηνύματος με το κείμενο

<pre>about our Senior Discounts") }</pre>	<p>"Your form has been submitted. Ask about our Senior Discounts".</p>
<pre>ELSE { alert("Your form has been submitted.") }</pre>	<p>Εάν η συνθήκη είναι ψευδής εκτελείται η εντολή που βρίσκεται μέσα στο ELSE. Δηλαδή εμφανίζεται ένα κουτί μηνύματος με το κείμενο "Your form has been submitted."</p>

Η εντολή switch

Η switch χρησιμοποιείται στην περίπτωση που ισχύει μία μόνο συνθήκη από ένα σύνολο πολλών συνθηκών.

Παράδειγμα

Το παράδειγμα αυτό ελέγχει το βαθμό ενός φοιτητή και, ανάλογα με την επίδοσή του, εμφανίζει σχετικό μήνυμα στην οθόνη.

```
switch (grade)
{
case 5:
document.write("You passed")
break
case 6:
document.write("Moderate")
break
case 7:
document.write("Good")
break
case 8:
document.write("Very good")
break
case 9:
document.write("Very very good")
break
case 10:
document.write("Excellent")
break
default:
document.write("You failed")
}
```

Η εντολή switch ελέγχει μία μία τις συνθήκες και εάν κάποια από αυτές είναι αληθής εκτελεί τον αντίστοιχο κώδικα. Η εντολή break που βρίσκεται μέσα στις συνθήκες διακόπτει την εκτέλεση της switch και αποτρέπει τον έλεγχο των υπόλοιπων συνθηκών όταν μία συνθήκη βρεθεί αληθής.

Οι εντολές της default εκτελούνται όταν όλες οι υπόλοιπες συνθήκες είναι ψευδείς.

4.3.8 Οι βρόχοι for και while

Οι βρόχοι (LOOP) επιτρέπουν σε μία συνάρτηση να επαναλάβει την ίδια ενέργεια περισσότερες από μία φορές. Αν θέλουμε να γράψουμε ένα όνομα 5 φορές, δημιουργούμε ένα βρόχο ως εξής:

```
START LOOP (REPEAT 5 TIMES)
    Write the Name
END OF LOOP
```

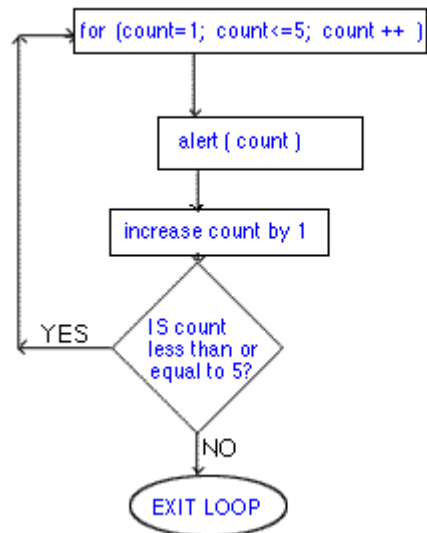
Αυτός φυσικά είναι ψευδοκώδικας και για την υλοποίησή του σε JavaScript μπορούμε να χρησιμοποιήσουμε τις δηλώσεις: for, while και do...while.

```
For ( count=1; count <=5; count++ ) {
    document.writeln(name);
}
```

Η δήλωση διαβάζεται ως εξής:

"Για count ίσο με ένα, όσο το count είναι μικρότερο ή ίσο με πέντε, το count αυξάνει κατά ένα"

Η παραπάνω δήλωση FOR χρησιμοποιεί μία μεταβλητή που ονομάζεται "count" (μπορούμε να χρησιμοποιήσουμε οποιοδήποτε άλλο όνομα). Αρχικά, το count θα είναι ίσο με 1. Ο υπολογιστής θα εκτελέσει τις εντολές που είναι μεταξύ των αγκίστρων, μέχρι να φτάσει στο τέλος του βρόχου όπου το count θα αυξηθεί κατά 1. Εάν η νέα τιμή του count είναι μικρότερη ή ίση του 5, ο βρόχος θα επαναληφθεί. Διαφορετικά, δηλαδή εάν $count > 5$, τότε το πρόγραμμα θα εξέλθει από το βρόχο και θα συνεχίσει στις υπόλοιπες εντολές (Εικόνα 4.3-4).



Εικόνα 4.3.4 - Διάγραμμα ροής για τη δήλωση for

Παράδειγμα

```

<HTML>
<HEAD>
  <SCRIPT LANGUAGE="JavaScript">
  <!-- Beginning of JavaScript -
    function repeat(text) {
      for (Count=0; Count<5; Count++) {
        document.writeln(text)
      }
    }
  // - End of JavaScript - -->
</SCRIPT>
</HEAD>
<BODY bgcolor="beige">
<FORM name="form1">
<table border=0>
<tr><td>Type your word
<td><INPUT NAME="input" TYPE=Text>
<td><INPUT TYPE=Button VALUE="Repeat word"
onClick="repeat(form1.input.value)">

```

```
</table>
</FORM>
</BODY>
</HTML>
```

Ο χρήστης πληκτρολογεί μία λέξη στο πεδίο κειμένου της φόρμας. Επιλέγοντας το πλήκτρο (button) με τίτλο “Repeat word”, εκτελείται η συνάρτηση repeat με παράμετρο τη λέξη που πληκτρολόγησε ο χρήστης στο πεδίο κειμένου (form1.input.value). Η συνάρτηση repeat, χρησιμοποιώντας την εντολή for, τυπώνει πέντε φορές τη λέξη σε μία νέα σελίδα.

Η ίδια ενέργεια θα μπορούσε να εκτελεστεί και με τη χρήση της εντολής **while** ή με τη χρήση της εντολής do...while. Η συνάρτηση repeat θα είχε τότε τη μορφή:

```
function repeat(text) {
    count=0;
    while (Count<5) {
        document.writeln(text)
        count++;
    }
}
```

ή τη μορφή:

```
function repeat(text) {
    count=0;
    Do{
        document.writeln(text)
        count++;
    } while (Count<5)
}
```

Η διαφορά της while από τη do...while είναι ότι στην πρώτη περίπτωση μπορεί οι εντολές του βρόχου να μην εκτελεστούν καθόλου, ενώ στη δεύτερη θα εκτελεστούν τουλάχιστον μία φορά.

4.3.9 Γενικά για τα αντικείμενα και τις μεθόδους

Όπως προαναφέρθηκε η JavaScript είναι μία αντικειμενοστραφής γλώσσα προγραμματισμού και τα βασικότερα στοιχεία της υλοποιούνται ως αντίγραφα κλάσεων, δηλαδή αντικείμενα. Τα πιο βασικά αντικείμενα της JavaScript είναι οι πίνακες, οι συμβολοσειρές, το αντικείμενο της ημερομηνίας, τα έγγραφα, οι φόρμες και τα παράθυρα.

Τα αντικείμενα αυτά έχουν ιδιότητες καθώς και μεθόδους. Όταν θέλουμε να κάνουμε κάποιες ενέργειες, όπως να ανοίξουμε κάποιο παράθυρο, να γράψουμε κείμενο στην οθόνη, να απομονώσουμε το πρώτο γράμμα μιας λέξης, να αντιστοιχίσουμε την σημερινή ημερομηνία σε μια μεταβλητή, να εμφανίσουμε στο χρήστη την προηγούμενη

(πίσω) σελίδα ή να εμφανίσουμε ένα κουτί προειδοποίησης, χρησιμοποιούμε μια μέθοδο.

Οι μέθοδοι εκτελούν ενέργειες και δε θα πρέπει να τις μπερδεύουμε με τις ιδιότητες που απλά αλλάζουν τιμές. Το `document.bgcolor` είναι μία ιδιότητα του εγγράφου η οποία μπορεί να πάρει διάφορες τιμές όπως `document.bgcolor= "red"`. Αντίθετα η `document.open()` είναι μία μέθοδος η οποία ανοίγει ένα νέο έγγραφο.

Μερικά από τα αντικείμενα που υποστηρίζει η JavaScript αναλύονται στις ενότητες που ακολουθούν.

4.3.10 Το αντικείμενο της συμβολοσειράς

Στην ενότητα αυτή περιγράφονται κάποιες από τις ιδιότητες και τις μεθόδους των συμβολοσειρών που υποστηρίζει η JavaScript. Για καλύτερη κατανόηση των ιδιοτήτων και των μεθόδων χρησιμοποιούνται παραδείγματα συναρτήσεων.

Η συνάρτηση που ακολουθεί εμφανίζει σε ένα παράθυρο μηνύματος το μήκος του κειμένου μιας μεταβλητής `text1`. Το μήκος ενός κειμένου ή αλφαριθμητικού δίνεται από την ιδιότητα `length` η οποία δηλώνει πόσους χαρακτήρες έχει το κείμενο ή το αλφαριθμητικό.

```
function DisplayLength() {  
    text1="A book on JavaScript"  
    alert (text1.length )  
}
```

Η συμβολοσειρά `text1` αποτελείται από 20 χαρακτήρες και το αποτέλεσμα που εμφανίζεται στο παράθυρο μηνύματος είναι 20.

Αν θελήσουμε να βρούμε ένα τμήμα ή ένα υποσύνολο μιας συμβολοσειράς, θα χρησιμοποιούσαμε την μέθοδο `substring` η οποία συντάσσεται ως `substring(start, stop)`. Η μέθοδος `substring` χρειάζεται δύο παραμέτρους `start` και `stop`, οι οποίες προσδιορίζουν τη θέση του πρώτου και του τελευταίου χαρακτήρα του τμήματος της συμβολοσειράς που θα επιστραφεί από την `substring`. Να θυμάστε ότι η JavaScript ξεκινάει πάντα το μέτρημα από το μηδέν. Δηλαδή, το πρώτο γράμμα είναι η θέση μηδέν.

Η συνάρτηση που ακολουθεί επιστρέφει τον πρώτο χαρακτήρα δηλαδή το 'A' της μεταβλητής `text1`.

```
function DisplayFirstChar() {  
    text1="A book on JavaScript"  
    alert (text1.substring(0,1) )  
}
```

Το αντίστροφο της ιδιότητας `substring` είναι η ιδιότητα `indexOf`, η οποία αντί να επιστρέφει το κείμενο μιας συγκεκριμένης θέσης, επιστρέφει τη θέση ενός χαρακτήρα κάποιου κειμένου. Για παράδειγμα, αν θέσουμε το `text1` ίσο με "Computer Systems" τότε η ιδιότητα `text1.indexOf("r")` θα επέστρεφε την τιμή 7.

Ο τελεστής += επιτρέπει την προσθήκη κειμένου στο τέλος μιας υπάρχουσας μεταβλητής. Για παράδειγμα, αν έχουμε τη μεταβλητή MyText="cat", τότε η εντολή MyText+="erpillar" θα έδινε σαν αποτέλεσμα τη MyText ίση "caterpillar." Το παράδειγμα που ακολουθεί συνενώνει δύο μεταβλητές και παρουσιάζει το αποτέλεσμα σε ένα παράθυρο μηνύματος

```
function Concatenate_Strings() {  
    text1="Hello "  
    text2="everybody!"  
    text1 += text2  
    alert (text1)  
}
```

Το αποτέλεσμα της συνάρτησης είναι ένα παράθυρο μηνύματος που γράφει "Hello everybody!".

Παράδειγμα:

Το παράδειγμα αυτό χρησιμοποιεί τις προαναφερθείσες ιδιότητες αλφαριθμητικών καθώς και ένα βρόχο for για να κωδικοποιήσει το κείμενο που εισάγει ένας χρήστης, αντικαθιστώντας κάθε αριθμό ή χαρακτήρα με τον επόμενο του:

```

<HTML>
<HEAD>
  <SCRIPT LANGUAGE="JavaScript">
  <!-- Beginning of JavaScript -
  function encode(text) {
  Ref="0123456789abcdefghijklmnopqrstuvwyz.-
  ~ABCDEFGHIJKLMNOPQRSTUVWXYZ";
  Result="";
  for (Count=0; Count<text.length; Count++) {
    Char=text.substring (Count, Count+1);
    Num=Ref.indexOf (Char);
    EncodeChar=Ref.substring(Num+1, Num+2);
    Result += EncodeChar;
  }
  document.form1.result.value=Result;
  }
  // - End of JavaScript - -->
</SCRIPT>
</HEAD>
<BODY bgcolor="beige">
<FORM>
<table border=0>
<tr><td>Type your word<td><INPUT NAME="input" TYPE=Text>
<td><INPUT TYPE=Button VALUE="submit"
onClick="encode(this.form.input.value)">
<tr><td>Result:<td><INPUT NAME="result" TYPE=Text>
</table>
</FORM>
</BODY>
</HTML>

```

Στην σελίδα HTML, που εμφανίζεται, υπάρχουν δύο πεδία κειμένου. Ο χρήστης πληκτρολογεί μία λέξη στο πρώτο πεδίο κειμένου και επιλέγει το πλήκτρο (button) “submit”. Τότε ενεργοποιείται η συνάρτηση encode η οποία κωδικοποιεί την λέξη του χρήστη. Το αποτέλεσμα της κωδικοποίησης εμφανίζεται στο δεύτερο πεδίο κειμένου.

Ανάλυση της συνάρτησης encode:

```

function encode(text) {
  Ref="0123456789abcdefghijklmnopqrstuvwyz.-
  ~ABCDEFGHIJKLMNOPQRSTUVWXYZ";
  Result="";
  for (Count=0; Count<text.length; Count++) {
    Char=text.substring (Count, Count+1);
    Num=Ref.indexOf (Char);
    EncodeChar=Ref.substring(Num+1, Num+2);
    Result += EncodeChar;
  }
  document.form1.result.value=Result

```

```
}

```

Αρχικά ορίζεται μια μεταβλητή Ref που περιέχει μια σειρά με όλους τους δυνατούς χαρακτήρες που μπορεί να πληκτρολογήσει ο χρήστης.

```
Ref="0123456789abcdefghijklmnopqrstuvwxyz.-
~ABCDEFGHIJKLMNOPQRSTUVWXYZ"
```

Κάθε γράμμα ή αριθμός έχει μία συγκεκριμένη θέση στη Ref. Π.χ. το γράμμα 'a' επιστρέφει την τιμή 10, επειδή είναι το δέκατο γράμμα στη Ref από τα αριστερά (στην ουσία είναι το ενδέκατο, αλλά η JavaScript αρχίζει να μετράει από το μηδέν).

Η δήλωση `Result=""` στη δεύτερη γραμμή, δείχνει ότι υπάρχει η μεταβλητή κειμένου με το όνομα `Result`, η οποία, όμως δεν περιέχει κάποιο κείμενο εκείνη τη χρονική στιγμή. Καθώς στο βρόχο `for` θα προσθέσουμε γράμματα στο τέλος του `Result`, είναι ανάγκη να έχει δηλωθεί η μεταβλητή από πριν.

Στη συνάρτηση υπάρχει ένας βρόχος, που χρησιμοποιεί το `Count` ως μεταβλητή αρίθμησης. Ο βρόχος θα επαναληφθεί τόσες φορές όσο το μήκος της λέξης που πληκτρολόγησε ο χρήστης `text.length`. Οι εντολές εντός του βρόχου θα εκτελεστούν μια φορά για κάθε γράμμα στο `text`.

Εντός του βρόχου υπάρχουν τέσσερις εντολές που εκτελούνται κάθε φορά που επαναλαμβάνεται ο βρόχος. Η πρώτη απομονώνει το χαρακτήρα της λέξης που βρίσκεται στη θέση `count` (`char=text.substring(count, count+1)`). Εάν `count=0` το `char` παίρνει τιμή ίση με τον πρώτο χαρακτήρα της λέξης που πληκτρολόγησε ο χρήστης, ενώ εάν `count=1` το `char` ισούται με το δεύτερο χαρακτήρα της λέξης.

Στη συνέχεια με την εντολή `Num=Ref.indexOf(Char)` η μεταβλητή `Num` γίνεται ίση με την τιμή της θέσης του χαρακτήρα `Char` μέσα στη `Ref`.

Με βάση τη θέση του χαρακτήρα στο `Ref`, βρίσκουμε τον αμέσως επόμενο χαρακτήρα `Ref.substring(Num+1, Num+2)` και τον αποθηκεύουμε στη μεταβλητή `EncodeChar` με την εντολή `EncodeChar=Ref.substring(Num+1, Num+2)`. Ο κωδικοποιημένος χαρακτήρας προστίθεται στο τέλος της μεταβλητής `result`.

Ας υποθέσουμε ότι ο χρήστης πληκτρολογεί τη λέξη "HAL". Η μεταβλητή `text` παίρνει την τιμή "HAL" και καλείται η `encode(text)`. Το `text.length` είναι 3 και ο βρόχος επαναλαμβάνεται 3 φορές. Για `count=0, 1` και `2`. Τα αποτελέσματα της συνάρτησης φαίνονται στον Πίνακα είναι τα ακόλουθα.

(Αρχική τιμή της μεταβλητής `Result=""`)

Πίνακας 4.3.4 – Αποτελέσματα βρόχου for

Εντολή	Βρόχος 1 Count=0	Βρόχος 2 Count=1	Βρόχος 3 Count=2
Char=text.substring (Count, Count+1);	Char = "H".	Char = "A".	Char = "L".
Num=Ref.indexOf (Char);	Num = 46.	Num = 38.	Num = 50.
EncodeChar=Ref.substring(Num+1, Num+2)	EncodeChar = "I"	EncodeChar = "B"	EncodeChar = "M"
Result += EncodeChar	Result = "I"	Result = "IB"	Result = "IBM"

4.3.11 Το αντικείμενο του πίνακα (array)

Οι πίνακες χρησιμοποιούνται για να αποθηκεύουν ένα σύνολο από τιμές στην ίδια μεταβλητή. Κάθε τιμή είναι ένα στοιχείο του πίνακα και καταλαμβάνει μία συγκεκριμένη θέση η οποία συμβολίζεται με κάποιο δείκτη. Στη JavaScript, όπως και σε άλλες αντικειμενοστραφείς γλώσσες, ο πίνακας δεν είναι μία απλή μεταβλητή, αλλά ένα αντικείμενο με ιδιότητες και μεθόδους.

Για να ορίσουμε ένα πίνακα με όνομα “colors” και 4 θέσεις, στη JavaScript γράφουμε:

```
var mycolors=new Array(4)
```

Για να αναφερθούμε σε κάποιο στοιχείο του πίνακα γράφουμε το όνομα του πίνακα και τον δείκτη του στοιχείου μέσα σε αγκύλες “[” και “]”. Ο δείκτης του πίνακα ξεκινάει πάντα στην τιμή 0. Για να θέσουμε τιμές στα στοιχεία πίνακα γράφουμε:

```
mycolors[0]="blue"  
mycolors[1]="red"  
mycolors[2]="yellow"  
mycolors[3]="green"
```

Μπορούμε , επίσης να ανακτήσουμε την τιμή ενός στοιχείου πίνακα και να την αποθηκεύσουμε σε κάποια άλλη μεταβλητή ως εξής:

```
document.bgcolor=mycolors[1]
```

Μια ιδιότητα του αντικειμένου array είναι η length η οποία επιστρέφει το μήκος, δηλαδή τον αριθμό των στοιχείων του array. Για παράδειγμα το mycolors.length είναι 4.

Το αντικείμενο array έχει και κάποιες μεθόδους όπως την join() η οποία ενώνει όλα τα στοιχεία του πίνακα σε ένα αλφαριθμητικό (string). Η μέθοδος reverse() αντιστρέφει τα στοιχεία του πίνακα και η μέθοδος sort() τα ταξινομεί.

4.3.12 Το αντικείμενο Math

Στη JavaScript όλες οι μαθηματικές συναρτήσεις (π.χ. στρογγυλοποίηση, ημίτονο, συνημίτονο, απόλυτη τιμή) ορίζονται ως μέθοδοι του αντικειμένου `math`. Στα παραδείγματα που ακολουθούν ορίζονται κάποιες από τις μεθόδους της `math`.

Η μέθοδος `round()` επιστρέφει τη στρογγυλοποίηση ενός αριθμού. Στην ακόλουθη γραμμή κώδικα η μεταβλητή `rounded_number` παίρνει την τιμή 7.

```
rounded_number= math.round(7.2)
```

Η μέθοδος `random()` επιστρέφει έναν τυχαίο αριθμό μεταξύ του 0 και του 1.

Η μέθοδος `max()` επιστρέφει τον μεγαλύτερο μεταξύ δύο αριθμών ενώ η `min()` επιστρέφει τον μικρότερο. Στην παρακάτω γραμμή κώδικα η τιμή του `min_number` είναι 4.

```
min_number=math.min(4,6)
```

4.3.13 Το αντικείμενο της ημερομηνίας

Το αντικείμενο της ημερομηνίας περιλαμβάνει ιδιότητες και μεθόδους που διευκολύνουν τη διαχείριση της ημερομηνίας και της ώρας. Οι μέθοδοι της ημερομηνίας παρουσιάζονται αναλυτικά στο Παράρτημα II.

Κάποιες μέθοδοι και τα αποτελέσματά τους παρουσιάζονται, επίσης, στο ακόλουθο σχήμα:

```
If now = Thu Dec 5 23:55:16 UTC+0200 2002

now.getDay()= 4 Day of week (0=Sunday)
now.getMonth()= 11 Month (0 to 11)
now.getDate()= 5 Day of month
now.getYear()= 2002 Year
now.getHours()= 23 Hours
now.getMinutes()= 55 Minutes
now.getSeconds()= 16 Seconds
now.getTime()= 1039125316419 Milliseconds since January 1, 1990 at
Midnight.
now.getTimezoneOffset()= -120 minutes off from GMT
```

Εικόνα 4.3.5 - Μέθοδοι ημερομηνίας

Για να θέσουμε σε μία μεταβλητή τη σημερινή ημερομηνία γράφουμε:

```
now = new Date()
```

Μπορούμε, επίσης, να προσδιορίσουμε οποιαδήποτε ημερομηνία και να την περάσουμε ως τιμή σε μία μεταβλητή με τους ακόλουθους τρόπους:

```
a_date =new Date(" October 14, 1971 12:35:00")
```

```
a_date = new Date(1971, 10, 14)
```

```
a_date=new Date(1971, 10, 14, 12, 35, 0)
```

Παράδειγμα - Δημιουργία και εμφάνιση ρολογιού σε μία σελίδα HTML

Το παρακάτω script εμφανίζει ένα ρολόι στην ιστοσελίδα:

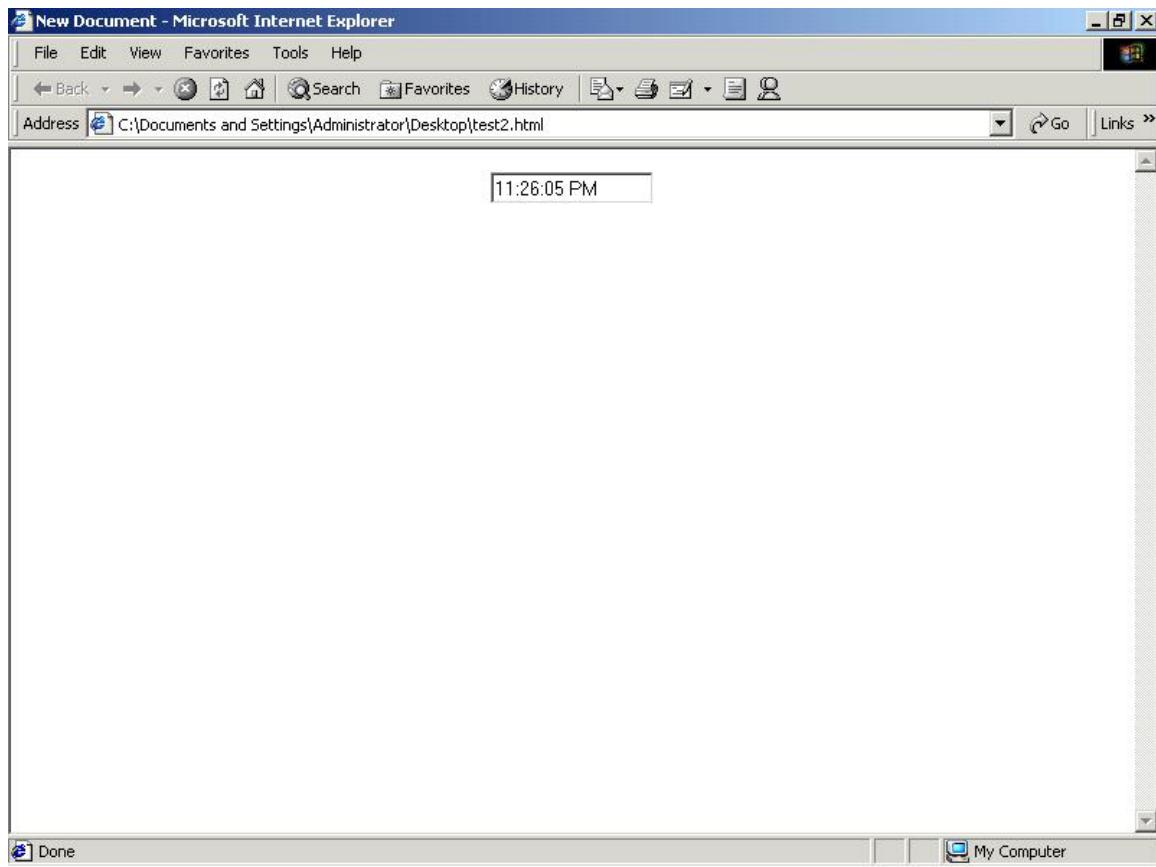
```
<HTML>
<BODY>
  <form name="time">
    <input type="text" name="time2" size=15>
  </form>
  <SCRIPT>
  function liveclock(){
    var curdate=new Date();
    var hours=curdate.getHours();
    var minutes=curdate.getMinutes();
    var seconds=curdate.getSeconds();
    var suffix="AM";
    if (hours>=12){
      suffix="PM";
      if (hours>=13)
        hours-=12;
    }
    if (minutes<10)
      minutes="0"+minutes;
    if (seconds<10)
```

```

        seconds="0"+seconds;
        var thetime=hours+":"+minutes+":"+seconds+" "+suffix;
        document.time.time2.value=thetime;
        setTimeout("liveclock()",1000);
    }
    liveclock()
</SCRIPT>
</BODY>
</HTML>

```

Στο παράδειγμα αυτό το script τοποθετείται μέσα στο BODY της HTML σελίδας, μετά τον ορισμό της φόρμας. Αυτό γίνεται διότι το script χρησιμοποιεί στοιχεία της φόρμας για να τρέξει. Το αποτέλεσμα του παραδείγματος φαίνεται στην Εικόνα 4.3-6.



Εικόνα 4.3.6 -Προσθέτοντας ένα ρολόι Live Clock στην ιστοσελίδα

4.3.14 Το αντικείμενο του Παραθύρου

Για κάθε έγγραφο, δηλαδή για κάθε σελίδα HTML, ο φυλλομετρητής δημιουργεί ένα παράθυρο. Το παράθυρο (window) είναι ένα αντικείμενο και το περιεχόμενό του, έγγραφο (document) είναι, επίσης, αντικείμενο. Από ένα έγγραφο μπορούν να

δημιουργηθούν περισσότερα του ενός παράθυρα με τη χρήση της μεθόδου `open()`, του αντικειμένου του παραθύρου.

Το αντικείμενο του παραθύρου έχει τις ιδιότητες που περιγράφονται στον Πίνακα 4.3.5:

Πίνακας 4.3.5 – Ιδιότητες του αντικειμένου παραθύρου

Ιδιότητα	Περιγραφή
<code>name</code>	Το όνομα του παραθύρου
<code>closed</code>	Δείχνει εάν ένα παράθυρο είναι κλειστό
<code>opener</code>	Το όνομα του παραθύρου από το οποίο ανοίχτηκε αυτό το παράθυρο.
<code>status</code>	Προσδιορίζει το κείμενο που θα εμφανιστεί στη μπάρα κατάστασης
<code>defaultStatus</code>	Το προκαθορισμένο κείμενο που εμφανίζεται στη μπάρα κατάστασης
<code>top</code> ή <code>self</code>	Αναφέρεται στο ίδιο το παράθυρο
<code>length</code>	Προσδιορίζει τον αριθμό των πλαισίων (<code>frames</code>) που υπάρχουν στο παράθυρο.
<code>parent</code>	Αναφέρεται σε ένα παράθυρο που περιλαμβάνει πλαίσια (<code>frames</code>)

Το αντικείμενο του παραθύρου έχει, επίσης, μεθόδους που βοηθούν στη διαχείρισή του. Οι πιο σημαντικές μέθοδοι του παραθύρου είναι η `open(arguments)`, η οποία ανοίγει ένα παράθυρο βασισμένη στις τιμές που δίνονται στα ορίσματα της και η `close()` η οποία κλείνει ένα παράθυρο. Υπάρχουν, όμως, και οι μέθοδοι `alert(string)`, `confirm(string)` και `prompt(string, [inputDefault])` οι οποίες εμφανίζουν στην οθόνη παράθυρα μηνυμάτων με την τιμή της συμβολοσειράς, που δίνεται ως όρισμά τους.

Οι ιδιότητες και οι μέθοδοι του αντικειμένου του παραθύρου αναλύονται στις εφαρμογές και τα παραδείγματα που ακολουθούν.

Άνοιγμα και κλείσιμο ενός παραθύρου

Το παράδειγμα αυτό παρουσιάζει αρχικά μία φόρμα HTML με αντικείμενα ένα πεδίο κειμένου (`text1`) και ένα πλήκτρο (`button`). Όταν πατηθεί το πλήκτρο της φόρμας ανοίγει ένα νέο παράθυρο που εμφανίζει την ημερομηνία και την ώρα δημιουργίας του καθώς και το κείμενο του `text1`. Η τιμή του πεδίου κειμένου (`text1`) της φόρμας εμφανίζεται με τη μορφή συνδέσμου, που, όταν επιλεγεί, κλείνει το παράθυρο.

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
    <!-- Beginning of JavaScript -
    function MyWindow(message) {
        //Define Date to display in local syntax
```



```

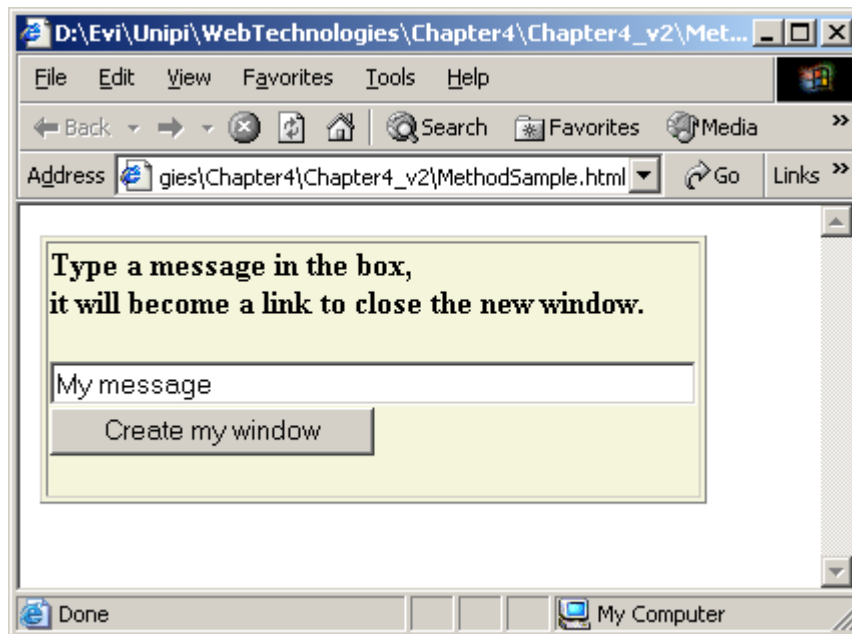
        now = new Date();
        LocalTime = now.toLocaleString();
//Define contents of page
        contents='<body bgcolor="beige">' + '<h2>Hello</h2>' +
        'Click on the message below to close this window <br>'
        + '<A HREF = "javascript:window.close()" >'+ message +
        '</A>'

//Create new Window
        options = "toolbar=0,status=0,menubar=0,scrollbars=0,"
        + "resizable=0,width=300,height=200";
        newwindow=window.open("", "mywindow", options);
        newwindow.document.writeln(LocalTime);
        newwindow.document.write(contents);
        newwindow.document.close();

    }
    // - End of JavaScript - -->
</SCRIPT>
</HEAD>
<BODY bgcolor="white">
<TABLE BORDER=1 bgcolor="beige">
<tr><td><B>Type a message in the box, <br>it will become a link
to close the new window.</B>
<FORM NAME="form1">
<INPUT NAME="text1" TYPE=Text SIZE="50" MAXLENGTH="50"><br>
<INPUT TYPE=Button VALUE="Create my window"
onClick="MyWindow(form.text1.value)">
</FORM></TABLE>
</BODY>
</HTML>

```

Όταν εκτελεστεί ο παραπάνω κώδικας, δημιουργείται ένας πίνακας (table) χρώματος μπεζ μέσα στον οποίο τοποθετούνται τα αντικείμενα (πεδίο κειμένου και πλήκτρο) μίας φόρμας (form), με όνομα form1. Τα αποτελέσματα του παραδείγματος φαίνονται στην Εικόνα 4.3.7. Όταν εμφανίζεται η HTML σελίδα το πεδίο κειμένου, με όνομα text1, είναι αρχικά κενό. Ο χρήστης πληκτρολογεί ένα μήνυμα, π.χ. “My message” και επιλέγει το πλήκτρο (button) με τίτλο “Create my window”.



Εικόνα 4.3.7 - Φόρμα HTML που δημιουργεί και ανοίγει ένα παράθυρο

Με την επιλογή του πλήκτρου ενεργοποιείται η συνάρτηση `MyWindow` με παράμετρο το μήνυμα “My message” που πληκτρολόγησε ο χρήστης (`MyWindow("My message")`).

Αρχικά η `MyWindow()` βρίσκει τη σημερινή ημερομηνία χρησιμοποιώντας τη συνάρτηση `Date()` και την περνάει σε μία μεταβλητή `now`. Στη συνέχεια χρησιμοποιεί τη μέθοδο `toLocaleString()` για να μετατρέψει την ημερομηνία σε κείμενο και να την εμφανίσει στο χρήστη.

Στην επόμενη εντολή:

```
contents='<body bgcolor="beige">' + '<h2>Hello</h2>' +
'Click on the message below to close this window <br>' +
'<A HREF = "javascript:window.close()" >'+ message + '</A>'
```

η συνάρτηση περνάει σε μία μεταβλητή `contents` μία συμβολοσειρά (`string`) με κώδικα HTML που προσδιορίζει τα περιεχόμενα του παραθύρου που θα εμφανιστεί μετά την επιλογή του πλήκτρου “Create my window”. Το `+` χρησιμοποιείται για να ενώσει διαφορετικές συμβολοσειρές (`string`). Η παραπάνω εντολή θα μπορούσε, επίσης, να γραφτεί ως:

```
contents='<body bgcolor="beige"><h2>Hello</h2>Click on the
message below to close this window <br><A HREF =
"javascript:window.close()" >'+ message + '</A>'
```

Το `message` δεν τοποθετείται μέσα σε εισαγωγικά, όπως το υπόλοιπο κείμενο, διότι είναι μία μεταβλητή. Στην εντολή αυτή έχει χρησιμοποιηθεί η μέθοδος παραθύρου

`window.close()`, η οποία κλείνει το παράθυρο όταν ο χρήστης επιλέξει το link με το μήνυμα (message) που πληκτρολόγησε.

Η τελευταία ακολουθία εντολών δημιουργεί και ανοίγει το παράθυρο που φαίνεται στην Εικόνα 4.3.8. Η μέθοδος παραθύρου `window.open ("", "mywindow", options)` ανοίγει ένα παράθυρο το οποίο εμφανίζεται αρχικά κενό, έχει όνομα "mywindow", και χαρακτηριστικά που προσδιορίζονται από την τιμή της μεταβλητής `options`.

Το αντικείμενο `document` του αντικειμένου `window`, αντιπροσωπεύει τα περιεχόμενα του παραθύρου που εμφανίζονται στην οθόνη. Η συνάρτηση τυπώνει το κείμενο που εμφανίζεται στο παράθυρο, αλλάζοντας την τιμή του `document`. Χρησιμοποιεί τις συναρτήσεις `writeln` και `write` για να γράψει το `document`.

Η `newwindow.document.writeln(LocalTime)` γράφει το κείμενο της ημερομηνίας σε μία γραμμή. Η `newwindow.document.write(contents)` γράφει τα περιεχόμενα της μεταβλητής `contents` στο `document`. Η διαφορά των δύο μεθόδων είναι ότι η δεύτερη μπορεί να τυπώσει κείμενο το οποίο ξεπερνάει τη μία γραμμή.

Η τελευταία μέθοδος που χρησιμοποιείται είναι η `document.close()` η οποία κλείνει το έγγραφο όταν ολοκληρωθούν οι αλλαγές του περιεχομένου του.



Εικόνα 4.3.8 - Παράθυρο που εμφανίζει στοιχεία της φόρμας από την οποία δημιουργήθηκε.

Κλήση HTML σελίδων και εμφάνισή τους σε νέο παράθυρο

Στο παράδειγμα που ακολουθεί εμφανίζεται μία σελίδα HTML με τρεις υπερσυνδέσμους. Για κάθε έναν από αυτούς τους υπερσυνδέσμους που επιλέγει ο χρήστης εμφανίζεται παράθυρο με διαφορετική σελίδα HTML.

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
<!--
```

```

function openWin(URL) {
  aWindow = window.open(URL, "thewindow",
    "toolbar=no,width=350,height=400,status=no,scrollbars=yes,
    resize=no, menubar=no");
}
//-->
</SCRIPT>
</HEAD>
<BODY>
<P><A HREF="javascript:openWin('Apples.html');">Apples</A></P>
<P><A HREF="javascript:openWin('Bananas.html');">Bananas</A></P>
<P><A HREF="javascript:openWin('Oranges.html');">Oranges</A></P>
</BODY>
</HTML>

```

Εντός των ετικετών `<script>` ορίζεται μία συνάρτηση, που ονομάζεται `openWin()`. Η συνάρτηση αυτή καλείται από τις ετικέτες άγκυρας, δηλαδή όταν ο χρήστης επιλέξει τους υπερσυνδέσμους που προσδιορίζονται μέσα σ' αυτές τις ετικέτες.

```

<A HREF="javascript:openWin('Bananas.html');">Bananas</A>

```

Στην παραπάνω γραμμή κώδικα η τιμή του `href` δεν καλεί κάποιο URL, αλλά μια συνάρτηση JavaScript. Η λέξη `javascript` ενημερώνει το φυλλομετρητή πως ακολουθεί ένας σύνδεσμος JavaScript. Στη συνέχεια καλείται η συνάρτηση `openWin()`, που έχει οριστεί στην κεφαλή (HEAD) του εγγράφου. Εφόσον η κλήση JavaScript είναι ένας σύνδεσμος, η συνάρτηση θα τρέξει όταν ο χρήστης πατήσει στον σύνδεσμο (δηλαδή, επιλέξει τη λέξη “Bananas”). Το περιεχόμενο στις παρενθέσεις—(‘Bananas.html’);—καθορίζει μια τιμή, που θα περαστεί στη συνάρτηση `openWin()`. Το υπόλοιπο της γραμμής είναι ένας κανονικός σύνδεσμος—το υπερ-κείμενο (Bananas) και το κλείσιμο της άγκυρας (``).

Το τμήμα κώδικα που ορίζει τη συνάρτηση `openWin()` είναι:

```

function openWin(URL) {
  aWindow = window.open(URL,"thewindow", "toolbar=no,width=350,
  height=400,status=no,scrollbars=yes,resize=no,menubar=no");
}

```

Η πρώτη γραμμή ορίζει μια νέα συνάρτηση με όνομα `openWin()`. Το ζεύγος των παρενθέσεων δηλώνει πως η συνάρτηση μπορεί να πάρει «ορίσματα» ή «παραμέτρους». Τα ορίσματα (arguments) είναι οι τιμές που επηρεάζουν την εκτέλεση και το αποτέλεσμα

της συνάρτησης. Η `openWin` έχει ως όρισμα ένα URL. Καλώντας τη συνάρτηση με ένα συγκεκριμένο URL, θα ανοίξει ένα νέο παράθυρο με αυτό το URL.

Μετά τη δήλωση της συνάρτησης υπάρχει μία αγκύλη (`{}`). Μια άλλη αγκύλη βρίσκεται στην τελευταία γραμμή. Ότι είναι μεταξύ αυτών είναι ο κώδικας της συνάρτησης, που θα εκτελεστεί. Οι δύο γραμμές στην ουσία είναι μια μεγάλη που εδώ λόγω περιορισμένου χώρου κόπηκε σε δυο γραμμές.

Η γραμμή ξεκινά δημιουργώντας μια νέα μεταβλητή `aWindow`. Στη μεταβλητή αυτή, τοποθετούμε την τιμή ενός παραθύρου το οποίο δημιουργείται από την μέθοδο `open()`. Η `window.open` ελέγχει και ανοίγει το παράθυρο, βασισμένη σε ορισμένες πληροφορίες για το παράθυρο αυτό. Υπάρχουν τρεις παράμετροι για την μέθοδο `window.open`: το URL που θα εμφανίσει το παράθυρο, το όνομα του παραθύρου και τα χαρακτηριστικά του παραθύρου.

Στη συνάρτηση δεν ορίσαμε ένα μοναδικό URL, αλλά ζητάμε να έχουμε ένα URL περασμένο στη συνάρτηση όταν αυτή εκτελείται. Π.χ. στην άγκυρα με το σύνδεσμο `Bananas` καλούμε την `openWin("Bananas.html")`. Δίνουμε, δηλαδή, την τιμή `Bananas.html` στο URL της συνάρτησης `openWin()`.

Η τιμή της δεύτερης παραμέτρου της `openWin`, δηλαδή το κείμενο `thewindow`, αντιστοιχεί στο όνομα του παραθύρου. Η τρίτη παράμετρος αντιστοιχεί στα χαρακτηριστικά του παραθύρου που είναι: χωρίς μπάρα εργαλείων (`toolbar=no`), πλάτος (`width`) 350 pixels, ύψος (`height`) 400 pixels, με ράβδους κύλισης (`scrollbars=yes`), χωρίς μπάρα καταστάσεως (`status=no`) και μπάρα μενού επιλογών (`menubar=no`), χωρίς μπορεί να μεγεθυνθεί από τον χρήστη (`resize=no`). Σημειώνουμε, ότι στο τελικό ζεύγος εισαγωγικών, δεν επιτρέπονται κενά ή αλλαγές γραμμής.

Όταν ο χρήστης πατήσει κάποιον σύνδεσμο, π.χ. τον σύνδεσμο `Bananas`, ο κώδικας `openWin()` εκτελείται, περνώντας το URL `Bananas.html` στη συνάρτηση, που ανοίγει ένα νέο παράθυρο 350x400 το οποίο εμφανίζει αυτό το URL. Όταν πατήσει το `Oranges`, εκτελείται η ίδια συνάρτηση, αλλά το URL παίρνει τιμή `Oranges.html` και η σελίδα αυτή εμφανίζεται στο παράθυρο.

Μηνύματα που φαίνονται στην μπάρα / γραμμή κατάστασης

Ένα απλό JavaScript που μπορεί κανείς να προσθέσει στην ιστοσελίδα του είναι ένα μήνυμα που θα εμφανιστεί στην μπάρα/γραμμή κατάστασης (την μπάρα στο κάτω μέρος που δείχνει το URL ή λέει "Document:Done"). Μπορούμε να χρησιμοποιήσουμε αυτή την μπάρα για να εμφανίσουμε ένα μήνυμα όταν ο χρήστης βάλει το δείκτη του ποντικιού του πάνω σε κάποιο σύνδεσμο. Για να επιτευχθεί αυτό, απλά προσθέτουμε ένα μικρό JavaScript σε μια κανονική άγκυρα. Δεν χρειάζεται να δηλώσουμε ένα script. Οι φυλλομετρητές που δεν υποστηρίζουν JavaScript, απλά θα αγνοήσουν τον κώδικα.

Το παρακάτω παράδειγμα εμφανίζει το κείμενο 'All kind of Apples' στην μπάρα κατάστασης, όταν ο χρήστης τοποθετήσει το δείκτη του ποντικιού πάνω από το σύνδεσμο `Apples`.

```
<A HREF="Apples.com" onMouseOver="window.status='All kind of Apples'; return true;">Apples</A>
```

Διαχείριση Πλαισίων

Η JavaScript έχει, επίσης τη δυνατότητα να φορτώσει περιεχόμενο σε πλαίσια. Μπορεί να αλλάξει πολλά διαφορετικά πλαίσια με ένα κλικ. Παρακάτω θα δούμε τον κώδικα που τροποποιεί τα περιεχόμενα ενός πλαισίου μπάρας εργαλείων και ενός κύριου πλαισίου με την επιλογή ενός συνδέσμου “Change Pages”. Υποθέτουμε ότι το πλαίσιο μπάρας εργαλείων έχει ονομαστεί *toolbar* και το κύριο πλαίσιο έχει ονομαστεί *main*.

```
function changePages (toolbarURL, mainURL) {  
    parent.toolbar.location.href=toolbarURL;  
    parent.main.location.href=mainURL;  
}
```

Η συνάρτηση αυτή θα μπορούσε να κληθεί από μία άγκυρα ως εξής:

```
<A HREF = "javascript:changePages ('toolbar_document2.html',  
'main_document2.html');" Change Pages</A>
```

Παράθυρα μηνυμάτων

Η JavaScript εμφανίζει παράθυρα μηνυμάτων με τις εντολές `alert`, `confirm` και `prompt`.

Alert: Η εντολή αυτή εμφανίζει ένα παράθυρο μηνύματος με το κείμενο που δίνεται ως όρισμά της.

```
<script>  
alert("Welcome, my friend!")  
</script>
```

Confirm: Η εντολή αυτή εμφανίζει ένα παράθυρο μηνύματος με το κείμενο που δίνεται ως όρισμά της και επιστρέφει μία τιμή ανάλογα με το αν η απάντηση του χρήστη είναι θετική ή αρνητική. Στο παράθυρο μηνύματος εμφανίζονται δύο πλήκτρα “OK” και “Cancel” που επιτρέπουν στο χρήστη να δηλώσει την απάντησή του.

```
<script>  
    var answer=confirm("Jump to Google?")  
    if (answer)  
        window.location=http://www.google.com  
</script>
```

Prompt: Η εντολή αυτή εμφανίζει ένα παράθυρο μηνύματος με το κείμενο που δίνεται ως όρισμά της καθώς και ένα πεδίο κειμένου στο οποίο μπορεί να γράψει ο χρήστης. Η συνάρτηση επιστρέφει ως τιμή το κείμενο που πληκτρολόγησε ο χρήστης.

```
<script>
```

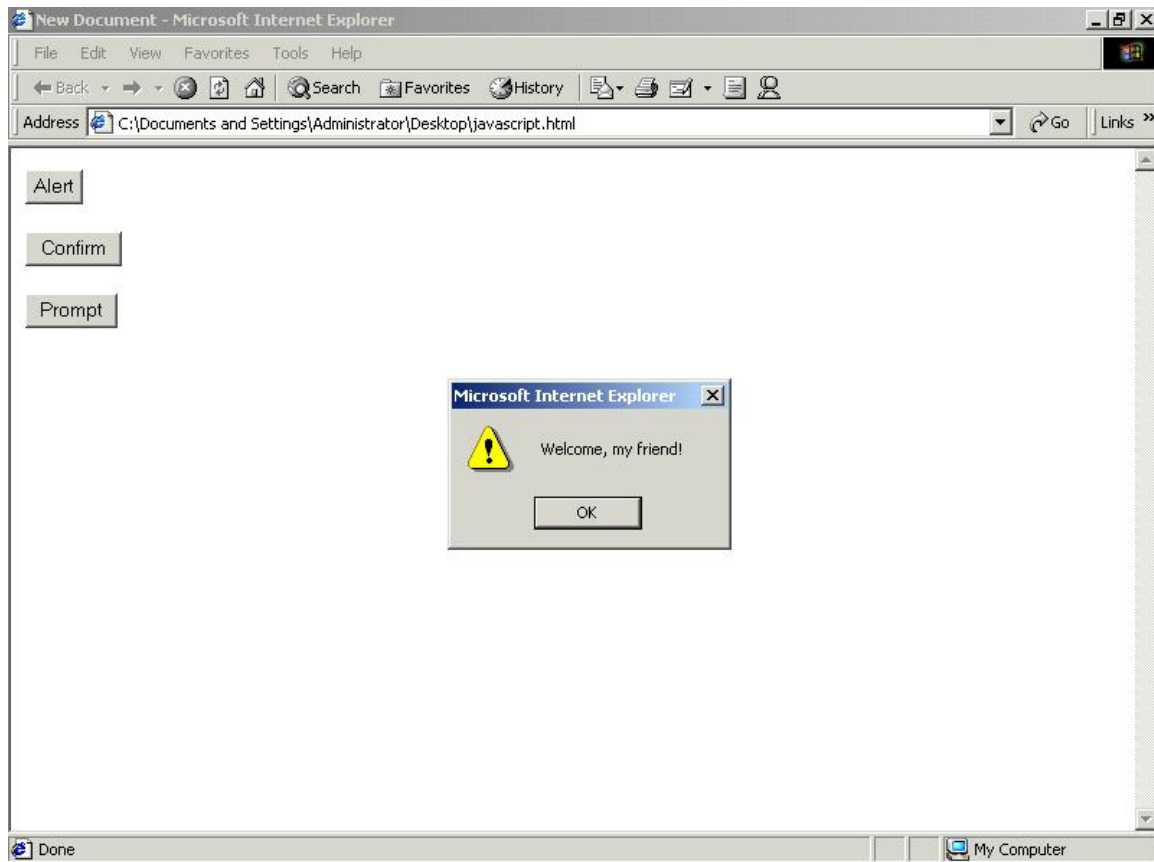
```
var answer=prompt("Please enter your name")
alert("Hello "+answer)
</script>
```

Στο παράδειγμα που ακολουθεί ορίζονται τρεις συναρτήσεις που περιέχουν τα παραπάνω σενάρια. Οι συναρτήσεις καλούνται όταν ο χρήστης επιλέξει ένα πλήκτρο (button) με την αντίστοιχη ονομασία.

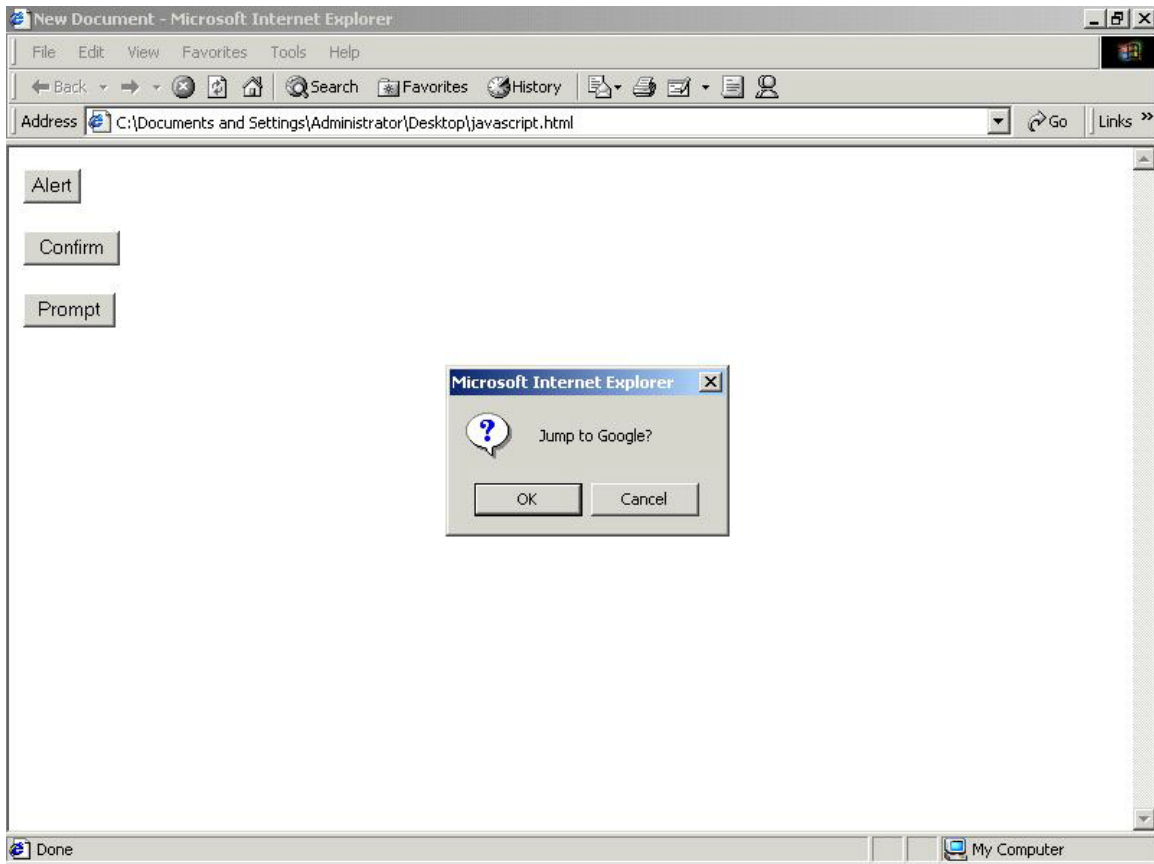
```
<HTML>
<HEAD>
  <SCRIPT>
  function box1(){
    alert("Welcome, my friend!")
  }
  function box2(){
    var answer=confirm("Jump to Google?")
    if (answer)
      window.location="http://www.google.com"
  }
  function box3(){
    var answer2=prompt("Please enter your name")
    alert("Hello "+answer2)
  }
  </SCRIPT>
</HEAD>
<BODY>
  <P><FONT face="Trebuchet MS"><INPUT onclick=box1()
  type=button value=Alert></FONT></P>
  <P><FONT face="Trebuchet MS"><INPUT onclick=box2()
  type=button value=Confirm></FONT></P>
  <P><FONT face="Trebuchet MS"><INPUT onclick=box3()
  type=button value=Prompt></FONT></P>
</BODY>
</HTML>
```

Τα αποτελέσματα του παραδείγματος φαίνονται στις Εικόνες 4.3.9, 4.3.10, 4.3.11 και 4.3.12:

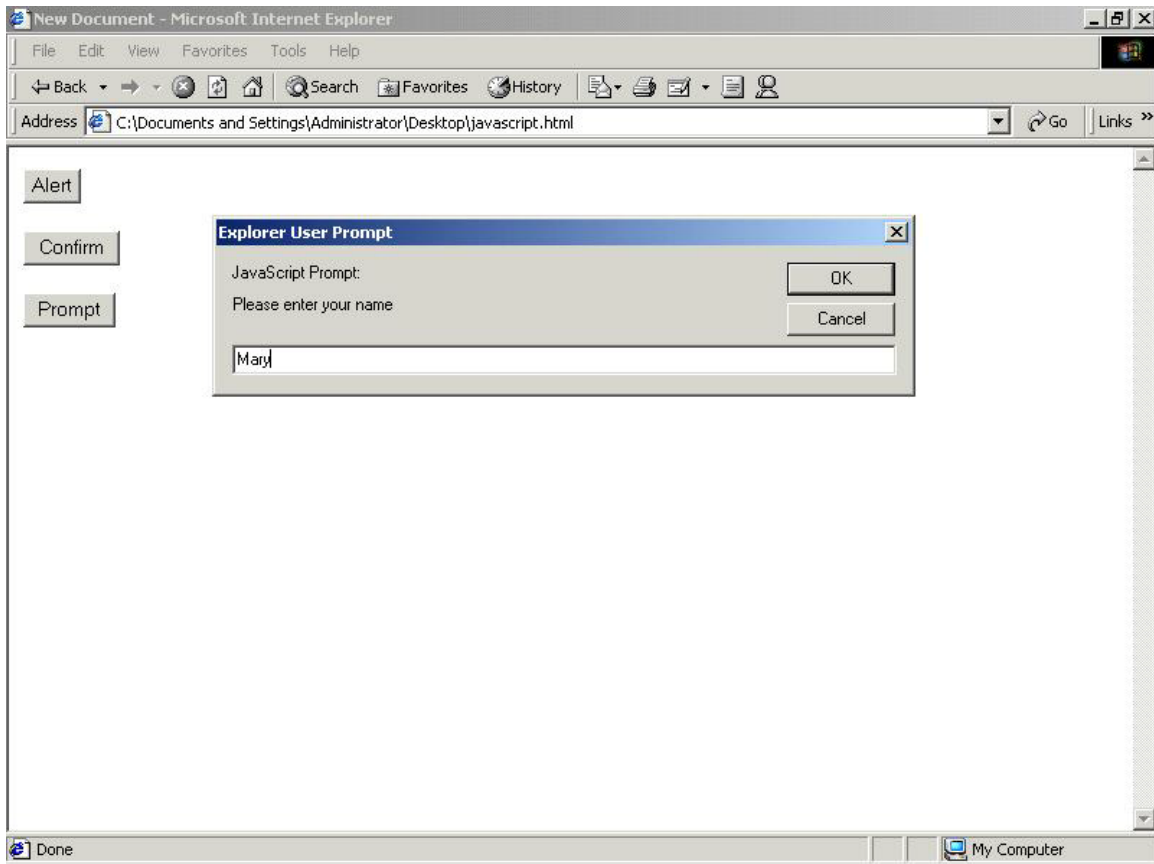
Όταν ο χρήστης επιλέγει το πλήκτρο alert εμφανίζεται το παράθυρο μηνύματος “Welcome, my friend!”



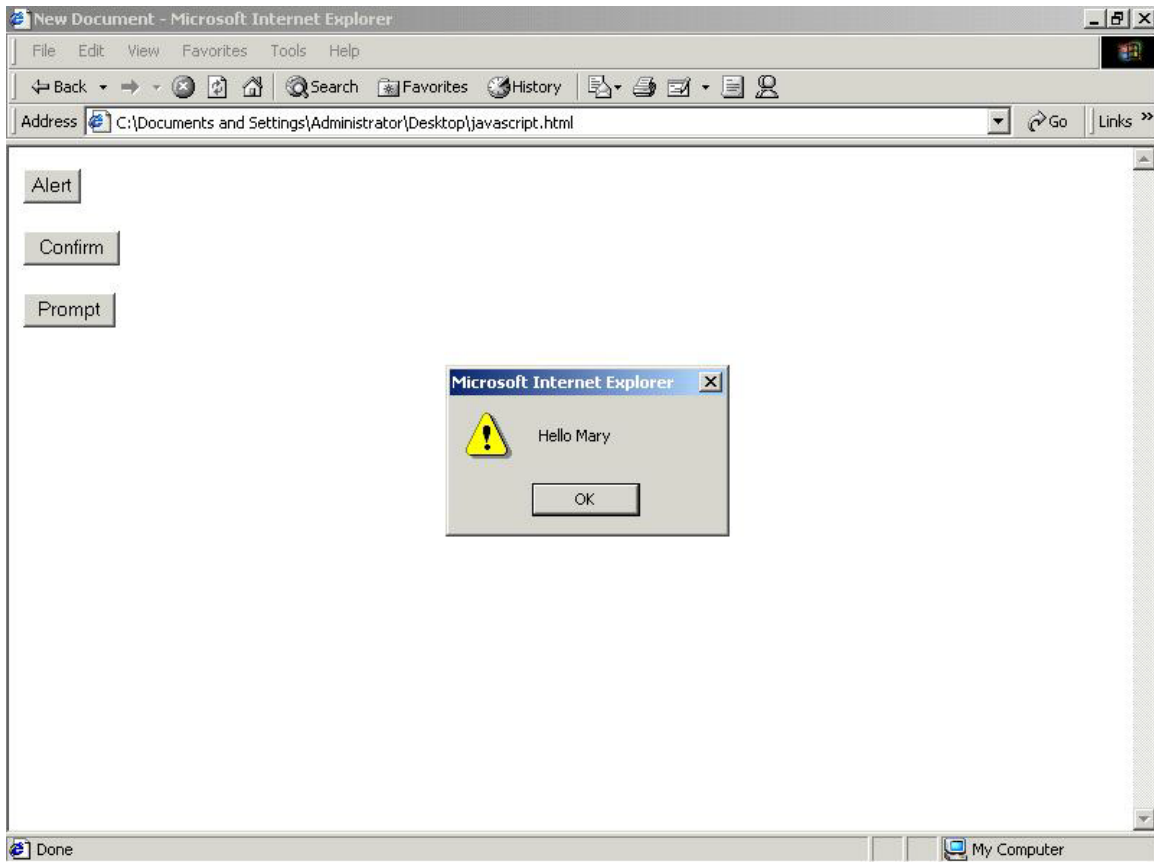
Εικόνα 4.3.9 - Παράδειγμα Alert



Εικόνα 4.3.10 - Παράδειγμα Confirm



Εικόνα 4.3.11 - Παράδειγμα Prompt



Εικόνα 4.3.12 - Παράδειγμα Prompt

4.3.15 Το αντικείμενο του εγγράφου

Το αντικείμενο του εγγράφου (document) προσδιορίζει το περιεχόμενο ενός παραθύρου, δηλαδή τη σελίδα HTML που εμφανίζεται στο παράθυρο. Κάποιες από τις ιδιότητες και μεθόδους του εγγράφου, παρουσιάστηκαν στα παραδείγματα των προηγούμενων ενοτήτων του κεφαλαίου. Σε αυτή την ενότητα θα αναφερθούμε συνοπτικά στις σημαντικότερες ιδιότητες και μεθόδους του εγγράφου. Κάποιες χρήσιμες ιδιότητες του εγγράφου παρουσιάζονται στον πίνακα 4.3.6:

Πίνακας 4.3.6 – Ιδιότητες εγγράφου

Ιδιότητες	Περιγραφή
bgColor	Το χρώμα που έχει το φόντο του εγγράφου
alinkColor	Το χρώμα των συνδέσμων
vlinkColor	Το χρώμα των συνδέσμων που έχει επισκεφθεί ένας χρήστης
referrer	Το όνομα του εγγράφου από το οποίο δημιουργήθηκε το έγγραφο
lastModified	Η ημερομηνία τελευταίας τροποποίησης του εγγράφου
title	Ο τίτλος του εγγράφου, δηλαδή το κείμενο της ετικέτας <TITLE> της HTML
URL	Η διεύθυνση URL της σελίδας HTML

Το αντικείμενο του εγγράφου έχει επίσης κάποιες μεθόδους όπως την `open()`, που ανοίγει ένα έγγραφο για να γραφτεί το περιεχόμενό του, την `close()`, που κλείνει το έγγραφο όταν ολοκληρωθούν οι αλλαγές του περιεχομένου του, καθώς και τις `write()` και `writeln()` που γράφουν το περιεχόμενό του. Η διαφορά των `write()` και `writeln()` είναι ότι η πρώτη μπορεί να γράψει κείμενο που ξεπερνάει τη μία γραμμή, ενώ η δεύτερη γράφει κείμενο μήκους μικρότερου ή ίσου με μία γραμμή.

Οι μέθοδοι εκτελούν ενέργειες και δε θα πρέπει να τις μπερδεύουμε με τις ιδιότητες που απλά αλλάζουν τιμές. Το `document.bgcolor` είναι μία ιδιότητα του εγγράφου η οποία μπορεί να πάρει διάφορες τιμές όπως `document.bgcolor= "red"`. Αντίθετα η `document.open()` είναι μία μέθοδος η οποία ανοίγει ένα νέο έγγραφο.

4.3.16 Το αντικείμενο της φόρμας

Οι φόρμες μπορούν να χρησιμοποιηθούν για την εισαγωγή κειμένου, την εμφάνιση αποτελεσμάτων και την εκτέλεση συναρτήσεων JavaScript. Οι φόρμες μίας σελίδας HTML είναι αντικείμενα, των οποίων τα στοιχεία (πεδία κειμένου (text), πλήκτρα (button), εικόνες (images), πεδία επιλογής (radio και checkbox) και μενού επιλογών) είναι επίσης αντικείμενα.

Το αντικείμενο της φόρμας έχει ιδιότητες όπως: `name` (όνομα φόρμας), `elements` (ένας πίνακας με τα στοιχεία της φόρμας), `method` (η μέθοδος (GET ή POST) με την οποία υποβάλλονται τα στοιχεία της) και `target` (το παράθυρο στο οποίο στοχεύει η υποβολή των στοιχείων της φόρμας).

Οι βασικές μέθοδοι του στοιχείου της φόρμας είναι οι `reset()` και `submit()`. Η πρώτη αρχικοποιεί τις τιμές των στοιχείων μιας φόρμας ενώ η δεύτερη υποβάλει τις τιμές τους.

Αντίστοιχα τα πιο σημαντικά γεγονότα σε μια φόρμα είναι το `onReset` και το `onFocus` τα οποία συμβαίνουν όταν πατηθούν τα πλήκτρα τύπου RESET και SUBMIT, δηλαδή όταν κληθούν οι μέθοδοι `reset()` και `submit()`.

Αντικείμενα μιας φόρμας

Όπως προαναφέρθηκε όλα τα στοιχεία μιας φόρμας, τα οποία ορίζονται στη σελίδα HTML με την ετικέτα `<INPUT>`, είναι αντικείμενα. Τα αντικείμενα της φόρμας έχουν ιδιότητες όπως `name` (το όνομα του πεδίου), `value` (η τιμή του πεδίου) και `defaultValue` (μια προκαθορισμένη τιμή για το πεδίο) καθώς και μεθόδους όπως `focus()`, `select()` και `click()`. Οι ιδιότητες και οι μέθοδοι μπορεί να διαφέρουν από αντικείμενο σε αντικείμενο, ανάλογα με το είδος του.

Στα αντικείμενα μιας φόρμας μπορούν να συμβούν γεγονότα `onFocus`, `onBlur` και `onChange`. Ένα αντικείμενο έχει το `FOCUS` όταν είναι ενεργό. Το γεγονός `onFocus` συμβαίνει όταν ο κέρσορας εισέρχεται στη συγκεκριμένη περιοχή κειμένου ή πεδίο κειμένου. Δεν έχει σημασία πού έχει τοποθετηθεί ο κέρσορας αρχικά. Μόλις ο δείκτης εισέλθει σε ένα πεδίο (είτε με επιλογή του χρήστη είτε με επιλογή του πλήκτρου TAB), θα συμβεί ένα γεγονός `onFocus`, που έχει προσαρτηθεί στο πεδίο αυτό.

Αντίθετα, το γεγονός `onBlur` συμβαίνει όταν ένα αντικείμενο παύει να είναι ενεργό. Όταν, δηλαδή ο κέρσορας μεταφερθεί σε άλλο αντικείμενο.

Ορισμένοι τύποι αντικειμένων φόρμας, όπως `checkboxes` και μενού επιλογών, παραμένουν συνεχώς ενεργά όσο είναι η φόρμα ενεργή, με αποτέλεσμα να μη μπορούμε να χρησιμοποιήσουμε τα γεγονότα `onFocus` και `onBlur` όταν κάνουμε τροποποιήσεις. Μπορούμε, όμως να χρησιμοποιήσουμε το γεγονός `onChange` για να καλέσουμε συναρτήσεις, όταν ο χρήστης επιλέξει ή αλλάξει την τιμή των αντικειμένων αυτών.

Άλλα γεγονότα που ενεργοποιούνται κατά την επιλογή ή τη μετακίνηση του δείκτη του ποντικιού στα αντικείμενα μίας φόρμας είναι τα ακόλουθα:

`onClick`: συμβαίνει όταν ο χρήστης επιλέξει με το ποντίκι ένα αντικείμενο.

`onmouseover`: συμβαίνει όταν ο δείκτης του ποντικιού βρεθεί πάνω από κάποιο αντικείμενο,

`onmouseout`: συμβαίνει όταν ο δείκτης του ποντικιού φύγει πάνω από κάποιο αντικείμενο.

Περισσότερες λεπτομέρειες για τις φόρμες και τα αντικείμενά τους δίνονται στις εφαρμογές και τα παραδείγματα που ακολουθούν:

Παράδειγμα απόκρισης σε γεγονός `onmouseover`

Το παρακάτω τμήμα κώδικα προσθέτει μία εικόνα με το logo του πανεπιστημίου σε μία σελίδα HTML. Όταν ο δείκτης του ποντικιού βρεθεί πάνω στην εικόνα θα ενεργοποιηθεί η εντολή `alert("hi")` και θα εμφανιστεί ένα κουτί μηνύματος με το κείμενο "hi".

```
<A HREF="#" onMouseOver="alert('hi')">  
  <IMG SRC="http://www.unipi.gr/logo.gif" BORDER=0 >  
</A>
```

Αλλαγή εικόνας με την κίνηση του δείκτη του ποντικιού

Το JavaScript που ακολουθεί δείχνει μια εικόνα που αλλάζει σε μια άλλη όταν ο δείκτης του ποντικού περάσει από πάνω της.

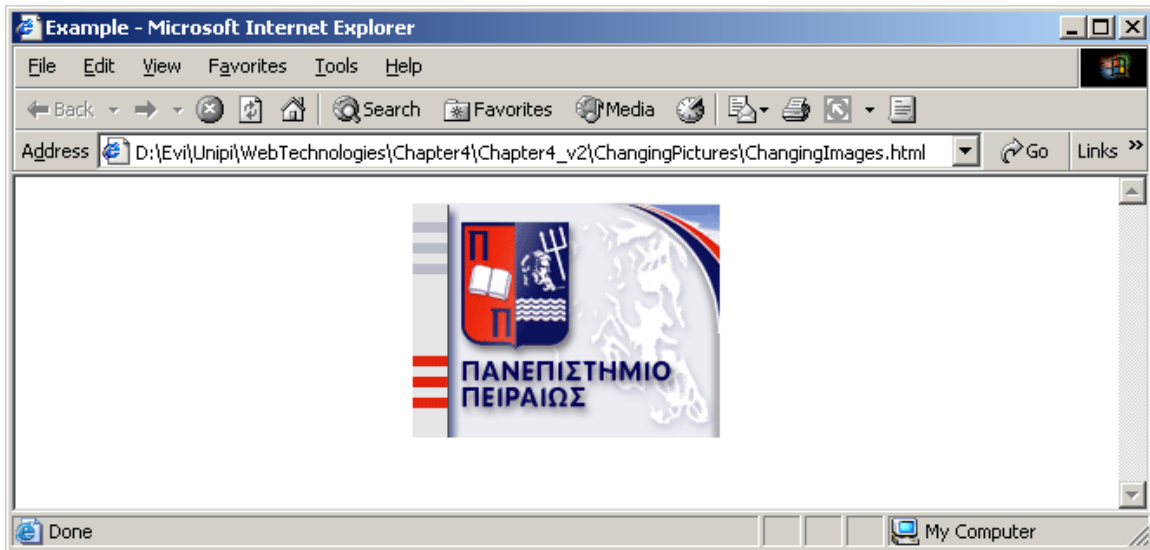
```
<HTML>  
<HEAD>  
<TITLE> Example </TITLE>  
</HEAD>  
<BODY>  
<DIV ALIGN="center">
```

```

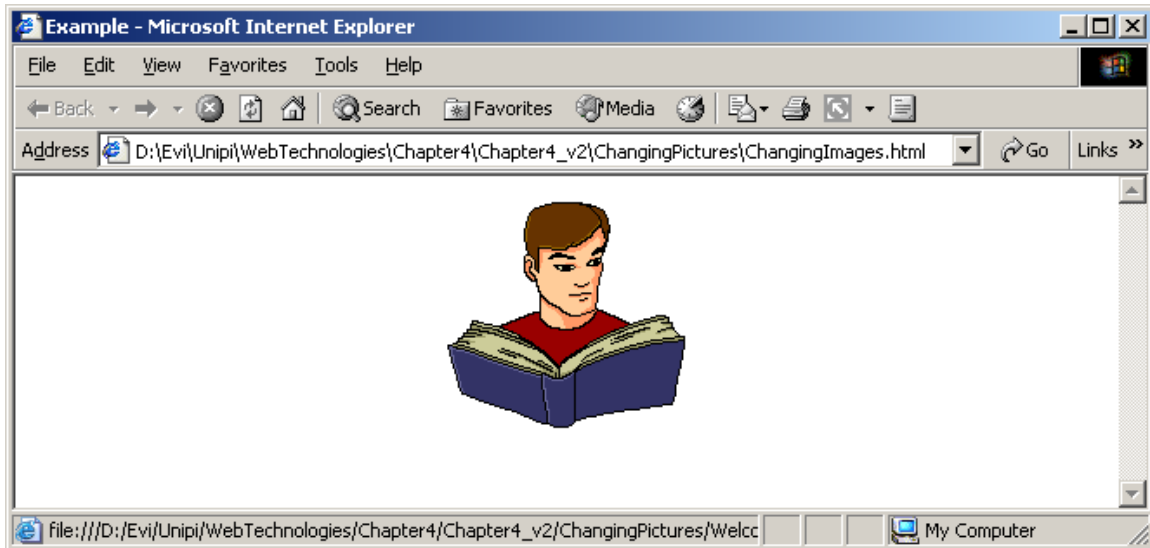
<a href="Welcome.htm"
onmouseover="if (document.images)
document.images.menu.src='unipi.gif'"
onmouseout="if (document.images)
document.images.menu.src=reader.gif'">
</a>
</DIV>
</BODY>
</HTML>

```

Τα αποτελέσματα του προγράμματος φαίνονται στις εικόνες 4.3.14 και 4.3.15. Όταν πρωτοεμφανίζεται η HTML σελίδα (Εικόνα 4.3.13) η εικόνα που φαίνεται είναι το logo του Πανεπιστημίου Πειραιά (unipi.gif). Όταν ο χρήστης μετακινήσει το δείκτη του ποντικιού πάνω στην εικόνα, η εικόνα θα αντικατασταθεί από αυτή ενός αναγνώστη (reader.gif) (Εικόνα 4.3.14).

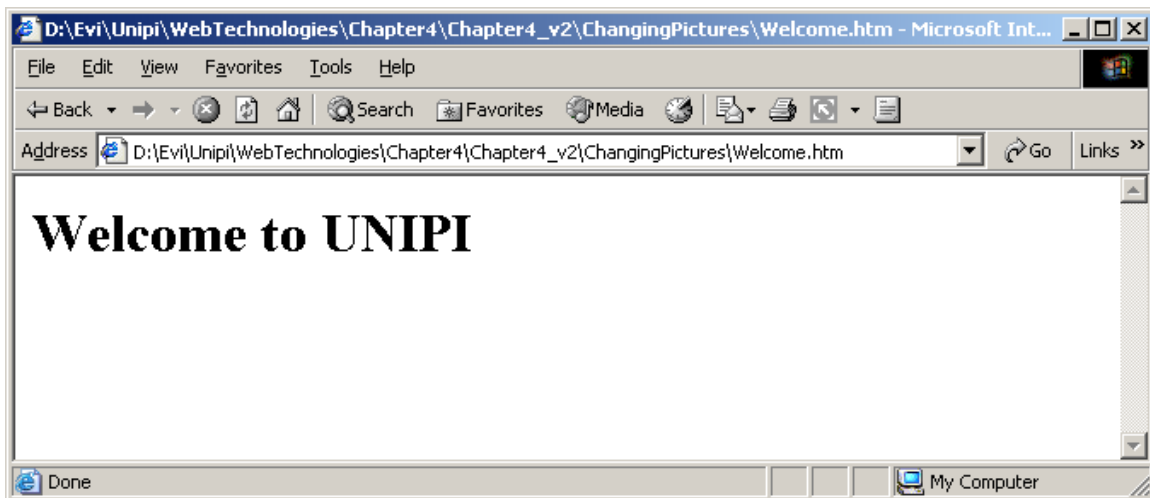


Εικόνα 4.3.13 - Αλλαγή εικόνων με την κίνηση του ποντικιού – Αρχική εικόνα



Εικόνα 4.3.14 - Αλλαγή εικόνων με την κίνηση του ποντικιού – Εικόνα που εμφανίζεται όταν ο δείκτης του ποντικιού μετακινηθεί

Εάν ο χρήστης επιλέξει με το δείκτη του ποντικιού την εικόνα, εμφανίζεται μία νέα σελίδα HTML (Welcome.htm) (Εικόνα 4.3-15).



Εικόνα 4.3.15 - Αλλαγή εικόνων με την κίνηση του ποντικιού – Εμφάνιση νέας σελίδας HTML.

Αντικείμενα φόρμας - Check Boxes

Κάθε checkbox έχει μια ιδιότητα που ονομάζεται checked η οποία είναι ίση με true ή false, ανάλογα με το αν το checkbox είναι επιλεγμένο ή όχι. Το παράδειγμα, που ακολουθεί εκτελεί μια συνάρτηση όταν ο χρήστης επιλέγει το checkbox. Η συνάρτηση εμφανίζει ένα μήνυμα όταν το checkbox είναι τσεκαρισμένο.

```
function flashbox(checked){  
if (checked==true) {alert(" Welcome! ")}
```

```
}  
<FORM>  
<INPUT TYPE=checkbox NAME="flash"  
OnClick="flashbox(form.flash.checked) "  
Go there <FORM>
```

Στην σελίδα HTML εμφανίζεται η ακόλουθη εικόνα

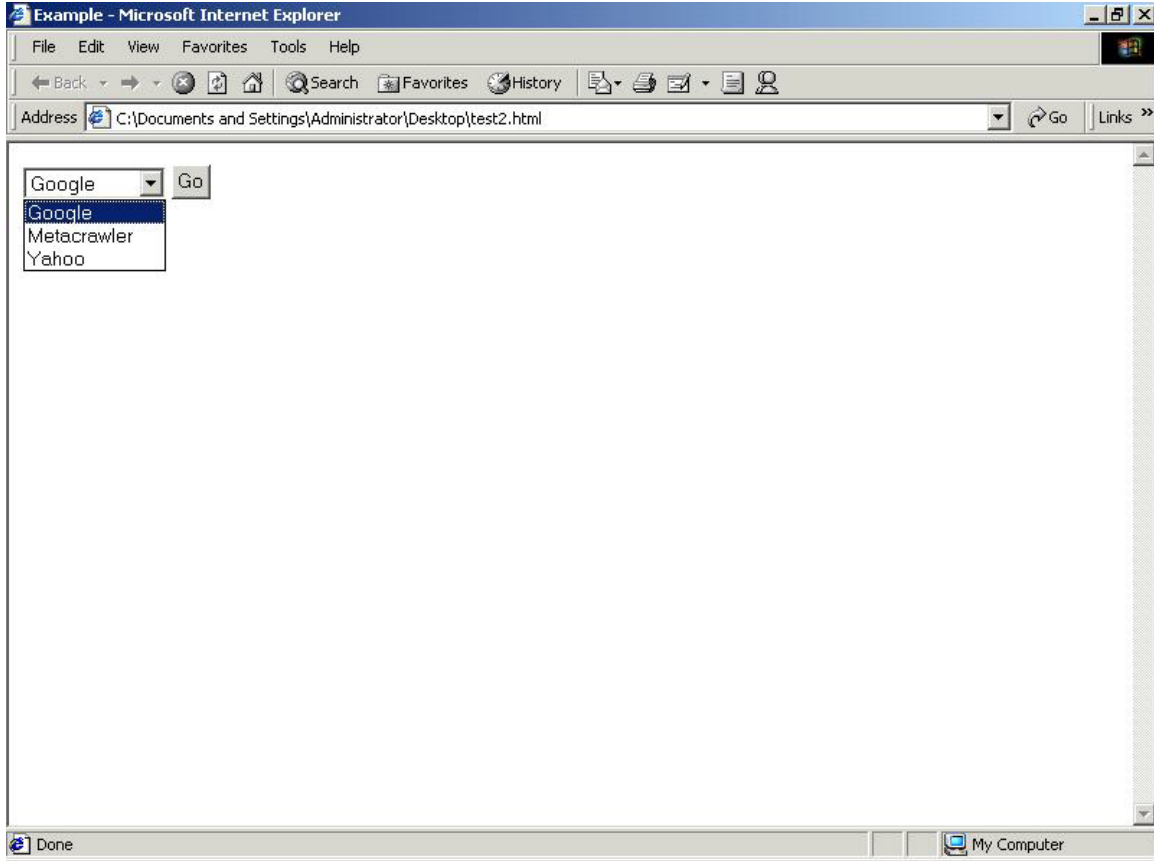
```
 Go there
```

Αντικείμενα φόρμας - Drop down menu box

Το παράδειγμα αυτό δημιουργεί ένα drop down menu box, το οποίο ανάλογα με την επιλογή του χρήστη παραπέμπει σε διαφορετική σελίδα HTML.

```
<HTML> they  
<HEAD>  
<TITLE> Example </TITLE>  
</HEAD>  
<BODY>  
<form name="c1">  
<p><select name="c2" size="1">  
  <option selected  
  value="http://www.google.com">Google</option>  
  <option  
  value="http://www.metacrawler.com">Metacrawler</option>  
  <option value="http://www.yahoo.com">Yahoo</option>  
</select>  
<input type="button" value="Go"  
onClick="location=document.c1.c2.options  
[document.c1.c2.selectedIndex].value"></p>  
</form>-  
</BODY>  
</HTML>
```

Το αποτέλεσμα του παραδείγματος φαίνεται στην Εικόνα 4.3.16. Ο χρήστης διαλέγει μία από τις τρεις επιλογές και στην οθόνη εμφανίζεται η ιστοσελίδα που αντιστοιχεί στην επιλογή του χρήστη.



Εικόνα 4.3.16 - Παράδειγμα ενός drop down box

Υποβολή στοιχείων φόρμας με επιλογή μιας εικόνας

Η JavaScript δεν είναι μόνο πρακτική αλλά βοηθάει και στη δημιουργία καλύτερων διεπαφών χρήσης. Τα πλήκτρα στις φόρμες HTML είναι αρκετά κακόγουστα και είναι δυνατόν να αντικατασταθούν από άλλα αντικείμενα όπως κάποια εικόνα. Με τη βοήθεια της JavaScript μπορούμε να χρησιμοποιήσουμε μία εικόνα της επιλογής μας στη θέση των πλήκτρων της φόρμας, για να εκτελέσει μια εργασία, όπως να υποβληθούν (submit) τα περιεχόμενα της φόρμας.

Έστω ότι έχουμε μία φόρμα με όνομα myform;

```
<form name="myform" >  
...  
</form>
```

Ο κώδικας με το συνηθισμένο πλήκτρο υποβολής φόρμας είναι:

```
<form name="myform" ...>
```

```
...  
<INPUT TYPE="Submit" NAME="Submit" VALUE="Send" >  
</form>
```

Για να αντικαταστήσουμε το συνηθισμένο πλήκτρο παράδοσης (<input>) με μία εικόνα γράφουμε το ακόλουθο:

```
<form name="myform" ...>  
...  
<a href="javascript:document.myform.submit()" >  
</a>  
</form>
```

Η γραμμή `javascript:document.myform.submit()`, λέει στην JavaScript να παραδώσει / υποβάλει τα στοιχεία της φόρμα με το όνομα `myform`, όταν πατηθεί / επιλεγεί η εικόνα.

4.3.17 Χειρισμός πολλαπλών φυλλομετρητών

Αντίθετα με τα CGI scripts, που τρέχουν στον εξυπηρετητή και δεν απαιτούν ιδιαίτερο λογισμικό από την πλευρά του φυλλομετρητή, ο κώδικας JavaScript είναι εντελώς εξαρτώμενος από την υποστήριξη του φυλλομετρητή. Αν τοποθετήσουμε κώδικα JavaScript σε μια σελίδα, οι φυλλομετρητές που δεν την καταλαβαίνουν δε θα γνωρίζουν τι να την κάνουν. Αυτοί οι φυλλομετρητές θα μεταγλωττίσουν τον κώδικα αυτόν ως απλό κείμενο και το αποτέλεσμα θα είναι κάθε άλλο παρά επιθυμητό.

Έλεγχος για φυλλομετρητές

Οι φυλλομετρητές που δεν υποστηρίζουν JavaScript, εμφανίζουν ολόκληρο τον κώδικα πάνω στην σελίδα HTML. Για να αποφευχθεί αυτό, αμέσως μετά την ετικέτα <script> χρησιμοποιούμε τις ετικέτες σχολίων (“<!--” και “//-->”) της HTML

```
<script type="text/javascript">  
<!--  
    κάποιες εντολές  
//-->  
</script>
```

Οι δύο πλάγιες γραμμές “//” πριν την ετικέτα του σχολίου “- ->” εμποδίζουν το μεταγλωττιστή (compiler) κώδικα JavaScript να εκτελέσει τη γραμμή αυτή.

Αν έχουμε κάποιο script, για το οποίο γνωρίζουμε ότι δουλεύει σε κάποιον φυλλομετρητή, αλλά δεν δουλεύει σε κάποιον άλλον, μπορούμε να ελέγξουμε πρώτα την έκδοση και τον τύπο του φυλλομετρητή και να δώσουμε στους χρήστες που χρησιμοποιούν τον κατάλληλο φυλλομετρητή το script που έχουμε και σε όλους τους υπόλοιπους κάποιο απλό κείμενο HTML. Το πρώτο βήμα είναι να ελέγξουμε το όνομα

και την έκδοση του φυλλομετρητή και να αποθηκεύσουμε αυτή την πληροφορία σε μια μεταβλητή:

```
<HTML>
<HEAD>
<TITLE>A Page</TITLE>
<SCRIPT LANGUAGE = "JavaScript">
<!--
    bName = navigator.appName;
    bVer = parseInt(navigator.appVersion);
    if (bName == "Netscape" && bVer >= 4) br = "n4";
    else if (bName == "Netscape" && bVer == 3) br = "n3";
    else if (bName == "Netscape" && bVer == 2) br = "n2";
    else if (bName=="Microsoft Internet Explorer" && bVer >= 4)
br = "e4";
    else if (bName == "Microsoft Internet Explorer") br = "e3";
    else br = "n2";
    ...

```

Αυτός ο κώδικας βάζει το όνομα του φυλλομετρητή σε μια μεταβλητή που καλείται bName και τον αριθμό της έκδοσης σε μια μεταβλητή bVer. Η μεταβλητή br παίρνει μια τιμή που συμβολίζει τους διαφορετικούς φυλλομετρητές και που εξαρτάται από το όνομα και τον αριθμό των παραπάνω μεταβλητών. Έτσι, αν ο φυλλομετρητής είναι Netscape 4, η br παίρνει την τιμή n4, αν ο φυλλομετρητής είναι IE 4, η br είναι e4. Στη συνέχεια χρησιμοποιούμε τις δηλώσεις if/else για να ελέγξουμε την τιμή της μεταβλητής και για να διαχωρίσουμε τον κώδικα που είναι κατάλληλος για κάθε φυλλομετρητή.

```
    if br=n4 {
//Netscape 4-specific JavaScript goes here
    }
    else if br=e4 {
//IE-4 specific code goes here
    }
    //-->
</SCRIPT>
</HEAD>

```

Στον κώδικα αυτό, η πρώτη δήλωση if ελέγχει αν ο φυλλομετρητής είναι Netscape 4. Αν είναι, τότε εκτελεί τον ειδικό κώδικα για Netscape-4. Αν δεν είναι, τότε ελέγχει αν πρόκειται για IE4 (Internet Explorer). Αν είναι IE4, τότε εκτελεί τον κατάλληλο κώδικα. Στην περίπτωση, που δεν είναι ούτε ο ένας ούτε ο άλλος, δεν τρέχει κάποιο script αλλά εμφανίζεται απευθείας το σώμα του εγγράφου HTML.

```
<BODY>
//Standard HTML code goes here
</BODY>
</HTML>

```

4.4 Εισαγωγή στη VBScript

Η VBScript υλοποιήθηκε από τη Microsoft και αποτελεί ένα υποσύνολο, ή μία περιορισμένη έκδοση της Visual Basic. Παρόλο που δεν έχει τις ίδιες δυνατότητες με τη Visual Basic, αποτελεί ένα εύχρηστο και εύκολο στην εκμάθηση εργαλείο, που μπορεί να χρησιμοποιηθεί για να προσθέσει λειτουργικότητα στις σελίδες HTML.

Ο κώδικας της VBScript ενσωματώνεται στον κώδικα της HTML σελίδας, όπως ακριβώς γίνεται και με την JavaScript. Ο κώδικας της VBScript φορτώνεται στον φυλλομετρητή του χρήστη μαζί με αυτόν της σελίδας HTML. Οι εντολές VBScript μπορούν να εκτελεστούν είτε μόλις η σελίδα εμφανιστεί στο φυλλομετρητή, είτε όταν συμβούν κάποια γεγονότα τα οποία τις καλούν. Η VBScript είναι μία διερμηνευτική γλώσσα προγραμματισμού όπως η JavaScript. Ο κώδικάς της δε χρειάζεται να μεταγλωττιστεί σε εκτελέσιμη μορφή προκειμένου να τρέξει, αλλά οι εντολές της ερμηνεύονται από τους φυλλομετρητές τη στιγμή που φορτώνονται.

Ένα από τα μειονεκτήματα της VBScript είναι ότι δεν είναι κατανοητή από όλους τους φυλλομετρητές. Ο Internet Explorer (έκδοση 3 και πάνω) υποστηρίζει VBScript, ενώ ο Netscape χρειάζεται το ScriptActive plug-in από την Ncompass Labs.

4.4.1 Προσθήκη VBScript σε HTML σελίδα

Η VBScript ενσωματώνεται στην σελίδα HTML σαν επέκταση του κώδικα HTML. Ο φυλλομετρητής λαμβάνει τα scripts μαζί με το έγγραφο της ιστοσελίδας, τα αναλύει και τα επεξεργάζεται ξεχωριστά.

Ο VBScript κώδικας τοποθετείται ανάμεσα στις ετικέτες <SCRIPT>..</SCRIPT>. Η ετικέτα <SCRIPT> ορίζει την αρχή του κώδικα ενώ η </SCRIPT> το τέλος του.

Παράδειγμα – Εμφάνιση παράθυρου μηνύματος:

Το παράδειγμα αυτό εμφανίζει ένα κουτί μηνύματος με κείμενο “Your first VBScript message”.

```
<HTML>
<HEAD>
<TITLE>Working With VBScript</TITLE>
<SCRIPT LANGUAGE="VBScript">
    MsgBox "Your first VBScript message"
</SCRIPT>
</HEAD>
</HTML>
```

Στην πρώτη ετικέτα <SCRIPT> υπάρχει η ιδιότητα language, στην οποία προσδιορίζεται η γλώσσα σεναρίων που θα χρησιμοποιηθεί. Εάν δηλαδή θα είναι η JavaScript ή η VBScript ή κάποια άλλη γλώσσα.

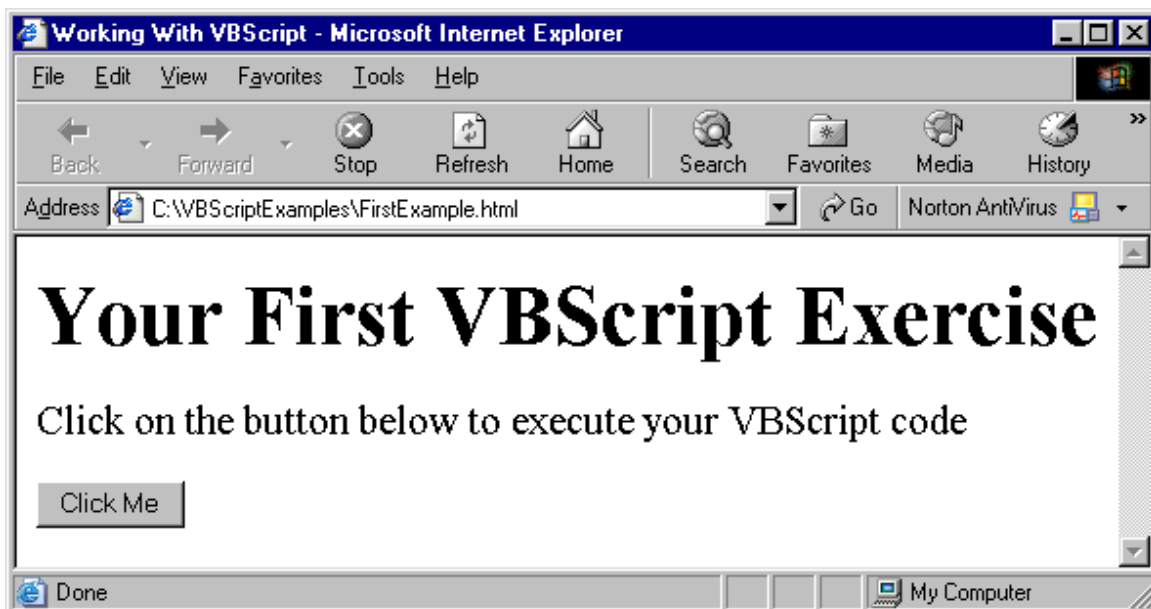
Παράδειγμα – Εμφάνιση παράθυρου μηνύματος μέσα από φόρμα:

Στο παράδειγμα αυτό δημιουργούμε ένα σενάριο το οποίο εκτελείται όταν ο χρήστης επιλέξει ένα πλήκτρο με όνομα "MyButton".

Αρχικά δημιουργούμε μία φόρμα HTML με το πλήκτρο αυτό, χωρίς να προσθέσουμε το VBScript κώδικα στη σελίδα.

```
<HTML>
<HEAD>
  <TITLE>Working With VBScript</TITLE>
</HEAD>
<BODY>
  <H1>Your First VBScript Exercise </H1>
  <P> Click on the button below to execute your VBScript
  code </P>
  <FORM NAME="form1">
    <INPUT TYPE="Button" NAME="MyButton" VALUE="Click Me">
  </FORM>
</BODY>
</HTML>
```

Όταν αυτός ο κώδικας φορτωθεί στον φυλλομετρητή θα εμφανισθεί η Εικόνα 4.4.1:

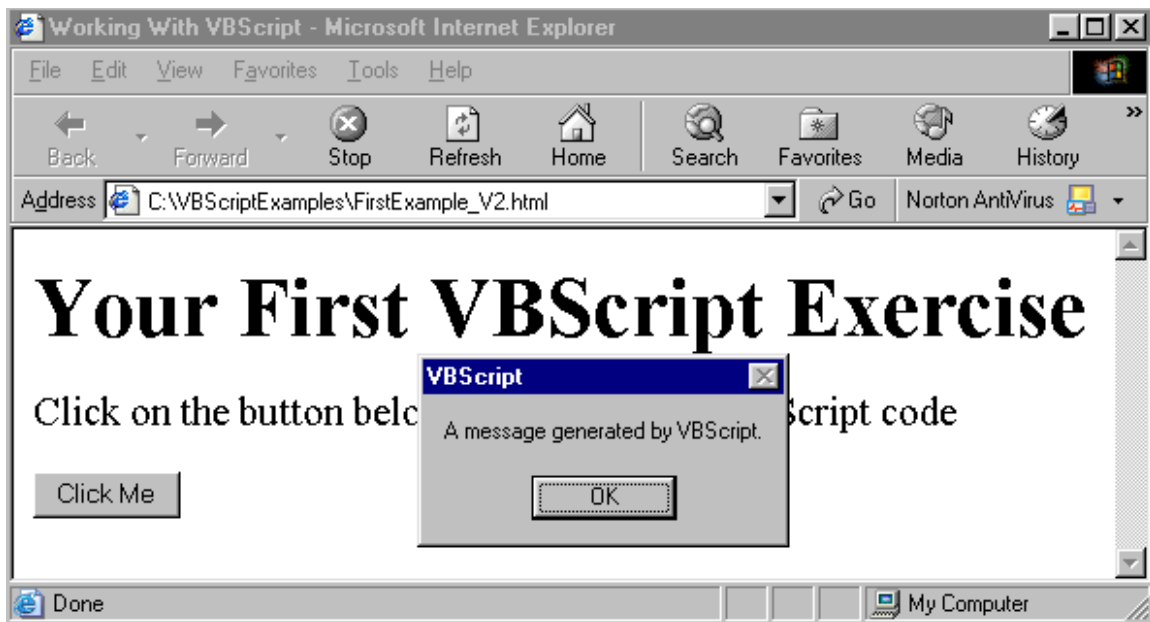


Εικόνα 4.4.1 - Παράδειγμα ενός απλού VBScript κώδικα

Όταν ο χρήστης επιλέξει το πλήκτρο “Click me” δεν εκτελείται τίποτα διότι δεν έχει ακόμα προστεθεί ο κώδικας VBScript. Με την προσθήκη του Script ο κώδικας της σελίδας μετατρέπεται ως εξής:

```
<HTML>
<HEAD>
<TITLE> Working With VBScript </TITLE>
</HEAD>
<BODY>
  <H1> Your First VBScript Exercise </H1>
  <P> Click on the button below to execute your VBScript
code </P>
  <FORM NAME="form1">
    <INPUT TYPE="Button" NAME="MyButton" VALUE="Click Me">
    <SCRIPT FOR="MyButton" EVENT="onClick" LANGUAGE="VBScript">
      MsgBox "A message generated by VBScript."
    </SCRIPT>
  </FORM>
</BODY>
</HTML>
```

Όταν ο παραπάνω κώδικας φορτωθεί στο φυλλομετρητή εμφανίζεται η σελίδα του προηγούμενου σχήματος. Όταν, όμως, ο χρήστης επιλέξει το πλήκτρο “Click me” εμφανίζεται ένα παράθυρο μηνύματος (Εικόνα 4.4.2) με το κείμενο "A message generated by VBScript."



Εικόνα 4.4.2 - Παράθυρο μηνύματος VBScript

Στην HTML σελίδα που παρουσιάστηκε στο παραπάνω παράδειγμα προσθέσαμε τον ακόλουθο κώδικα:

```
<SCRIPT FOR="MyButton" EVENT="onClick" LANGUAGE="VBScript">
    MsgBox "A message generated by VBScript."
</SCRIPT>
```

Μέσα στην ετικέτα SCRIPT έχουμε τα ακόλουθα στοιχεία:

Το FOR δηλώνει σε ποιο αντικείμενο αναφέρεται το σενάριο. Σε αυτό το παράδειγμα το σενάριο αναφέρεται στο αντικείμενο με όνομα “MyButton” το οποίο δηλώθηκε μέσα στην ετικέτα INPUT.

Το EVENT δηλώνει ποιο γεγονός στο συγκεκριμένο αντικείμενο θα ενεργοποιήσει το script. Η τιμή “onClick” δηλώνει ότι το script θα εκτελεστεί όταν επιλεγεί / πατηθεί το πλήκτρο “myButton”.

Τέλος το LANGUAGE δηλώνει τη γλώσσα στην οποία βασίζεται το script.

Η δεύτερη γραμμή, είναι καθαρός κώδικας VBScript. Χρησιμοποιείται η εντολή MsgBox, η οποία εμφανίζει ένα παράθυρο μηνύματος με το κείμενο που προσδιορίζεται αμέσως μετά την εντολή.

Η τελευταία γραμμή, δηλαδή η ετικέτα </SCRIPT> δηλώνει το τέλος του κώδικα VBScript.

4.4.2 Διαχείριση φυλλομετρητών

Στη VBScript παρουσιάζονται περισσότερα προβλήματα ασυμβατότητας με τους φυλλομετρητές απ’ ότι στη JavaScript. Όπως προαναφέραμε, η VBScript υποστηρίζεται κυρίως από τον Internet Explorer και για να εκτελεστεί σε Netscape χρειάζεται το ScriptActive plug-in της Ncompass Labs, το οποίο είναι διαθέσιμο στη διεύθυνση

www.netscape.com. Όλα τα σενάρια VBScript πρέπει να μεταγλωτιστούν σε ένα αρχείο, που ονομάζεται (AXS file) το οποίο φορτώνεται στο plug-in μέσω ενός σεναρίου JavaScript, το οποίο ελέγχει εάν πρόκειται για ένα φυλλομετρητή Netscape, ώστε να προσαρτήσει το AXS αρχείο στο plug-in:

```
<SCRIPT LANGUAGE="JavaScript">
<!--
// NCompass ScriptActive(TM) -- do not remove this comment
ua = navigator.appName;
ua = ua.toLowerCase();
if ((ua.indexOf('netscape',0)) != -1)
{document.writeln('<EMBED SRC=\"AwesomeScript.AXS\"
WIDTH=1 HEIGHT=1 LANGUAGE=\"VBScript\">');
}
//-->
</SCRIPT>
```

Οι φυλλομετρητές που δεν καταλαβαίνουν τον κώδικα VBScript παρουσιάζουν όλο το κείμενο του κώδικα πάνω στην ιστοσελίδα. Για να αποφευχθεί αυτό ο κώδικας τοποθετείται πάντα ανάμεσα σε ετικέτες σχολίων (<!--και -->).

Χρησιμοποιώντας τις ετικέτες σχολίων, το παράδειγμα 1 της προηγούμενης παραγράφου παίρνει την ακόλουθη μορφή:

```
<HTML>
<HEAD>
<TITLE>Working With VBScript</TITLE>
<SCRIPT LANGUAGE="VBScript">
<!--
  MsgBox "Your first VBScript message"
-->
</SCRIPT>
</HEAD>
</HTML>
```

Ένας φυλλομετρητής, επομένως, που δεν υποστηρίζει JavaScript, θα αντιμετωπίσει τον παραπάνω κώδικα VBScript σαν ένα σχόλιο και θα τον αγνοήσει.

Παράδειγμα – Χρήση ετικετών <!-- και -->:

Οι ετικέτες <!--και --> θα μπορούσαν, επίσης να προστεθούν στο παράδειγμα της προηγούμενης ενότητας, ώστε η εντολή MsgBox να αγνοηθεί από τους μη συμβατούς φυλλομετρητές.

```
<SCRIPT FOR="MyButton" EVENT="onClick" LANGUAGE="VBScript">
<!--
  MsgBox "A message generated by VBScript."
-->
</SCRIPT>
```


Η παραπάνω ενέργεια, δηλαδή το άνοιγμα του παραθύρου μηνύματος, που ενεργοποιείται με το πάτημα του πλήκτρου “MyButton”, θα μπορούσε να οριστεί και μέσα σε μία μέθοδο. Ο τρόπος αυτός είναι προτιμότερος και ο κώδικας καλύτερα δομημένος, καθώς οι εντολές VBScript είναι συγκεντρωμένες σε ένα μέρος και δεν είναι διασκορπισμένες σε διάφορα σημεία του κώδικα HTML.

```
<HTML>

  <HEAD>
  <TITLE>Working With VBScript</TITLE>
  <SCRIPT LANGUAGE="VBScript">
  <!-- Non-IE browsers skip the following lines.
    Sub MyButton_OnClick
      MsgBox "A message generated by VBScript."
    End Sub
  -->
</SCRIPT>
</HEAD>
<BODY>
  <H1>Your First VBScript Exercise</H1>
  <P> Click on the button below to execute your VBScript
code </P>
  <FORM NAME="form1">
    <INPUT TYPE="Button" NAME="MyButton" VALUE="Click Me">
  </FORM>
</BODY>
</HTML>
```

Το αποτέλεσμα του παραδείγματος αυτού είναι ακριβώς το ίδιο με αυτό του Παραδείγματος της προηγούμενης ενότητας. Τα ορίσματα FOR και EVENT δεν έχουν συμπεριληφθεί στην ετικέτα SCRIPT, αλλά έχουν αντικατασταθεί από το όνομα της μεθόδου “MyButton_OnClick”. Στο όνομα της μεθόδου δηλώνεται το όνομα του αντικείμενου (“MyButton”) με το οποίο σχετίζεται η μέθοδος αυτή, καθώς και το γεγονός που πρέπει να συμβεί στο αντικείμενο ώστε να την ενεργοποιήσει. Άρα η MyButton_OnClick εκτελείται κάθε φορά που ο χρήστης επιλέγει το πλήκτρο MyButton. Αυτού του είδους οι μέθοδοι ονομάζονται μέθοδοι γεγονότων. Στην περίπτωση αυτή το γεγονός είναι ο χρήστης που επιλέγει ένα πλήκτρο.

4.4.3 Οι μεταβλητές στη VBScript

Οι μεταβλητές αποθηκεύουν τιμές κατά τη διάρκεια της εκτέλεσης των σεναρίων. Οι μεταβλητές χρησιμοποιούνται για να:

- Αποθηκεύουν τιμές που εισάγει ο χρήστης στην σελίδα HTML.
- Αποθηκεύουν τιμές που επιστρέφουν οι συναρτήσεις.
- Παίρνουν ως τιμή το αποτέλεσμα υπολογισμών.

Η VBScript υποστηρίζει ένα μοναδικό τύπο δεδομένων που ονομάζεται variant. Οι μεταβλητές τύπου variant μπορούν να πάρουν οποιοδήποτε τύπο δεδομένων τους δοθεί, π.χ. είτε κάποιον αριθμό είτε αλφαριθμητικό. Οι τύποι των δεδομένων που μπορούν να αποθηκευτούν σε μία μεταβλητή variant ονομάζονται υποτύποι (subtypes). Οι υποτύποι που υποστηρίζονται από τη VBScript περιγράφονται στον πίνακα που ακολουθεί.

Πίνακας 4.4.1 – Περιγραφή υποτύπων της VBScript

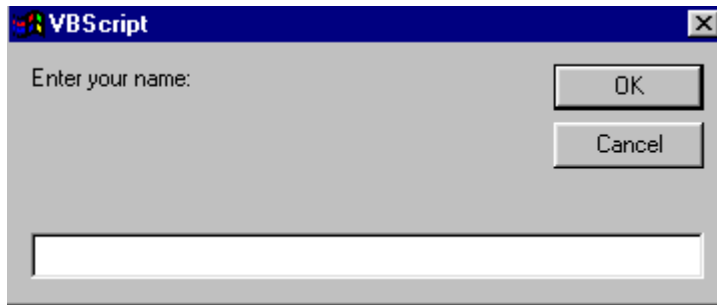
Υποτύποι	Περιγραφή Χρήσης κάθε Υποτύπου
Byte	Ακέραιος αριθμός από 0 έως 255
Boolean	<i>True</i> ή <i>False</i>
Currency	Νομίσματα
Date	Ημερομηνία και ώρα
Double	Πολύ μεγάλοι αριθμοί με δεκαδικά ψηφία
Empty	Η τιμή που παίρνει μια μεταβλητή προτού χρησιμοποιηθεί
Error	Ένας αριθμός που προσδιορίζει το είδος του λάθους
Integer	Ακέραιοι μεταξύ του -32,768 και του 32,767
Long	Πολλοί μεγάλοι ακέραιοι (-2,147,483,648 και 2,147,483,647)
Object	Αντικείμενα
Single	Αριθμοί με δεκαδικά ψηφία
String	Αλφαριθμητικό

Τα δύο παραδείγματα που ακολουθούν δείχνουν πώς δηλώνονται και πώς χρησιμοποιούνται οι μεταβλητές.

Παράδειγμα χρήσης μεταβλητών

```
Sub cmdVariables_OnClick
    Dim Name
    Name = InputBox("Enter your name: ")
    MsgBox "Your name is " & Name
End Sub
```

Η παραπάνω συνάρτηση εκτελείται όταν ο χρήστης επιλέξει το αντικείμενο (πλήκτρο) cmdVariables. Στη συνάρτηση χρησιμοποιείται μία εντολή Dim για να ορίσει μία μεταβλητή Name. Η Name παίρνει την τιμή που επιστρέφεται από τη συνάρτηση InputBox. Η InputBox ανοίγει ένα παράθυρο μηνύματος με ένα πεδίο κειμένου, το οποίο μπορεί να συμπληρώσει ο χρήστης. Στο συγκεκριμένο παράδειγμα εμφανίζεται ένα παράθυρο μηνύματος που ζητά από το χρήστη να συμπληρώσει το όνομά του (Εικόνα 4.4.3).



Εικόνα 4.4.3 - Παράθυρο για εισαγωγή μηνύματος

Η InputBox επιστρέφει ως αποτέλεσμα το όνομα που εισάγει ο χρήστης. Το αποτέλεσμα αυτό αποθηκεύεται στη μεταβλητή Name. Στη συνέχεια χρησιμοποιείται η συνάρτηση MsgBox για να εμφανίσει το όνομα του χρήστη (Εικόνα 4.4.4). Το "&" χρησιμοποιείται για να ενώσει δύο αλφαριθμητικά. Εάν για παράδειγμα η Name έχει την τιμή "John", στο παράθυρο μηνύματος θα εμφανιστεί το κείμενο "Your name is John".



Εικόνα 4.4.4 - Παράθυρο μηνύματος VBScript

Παράδειγμα αλλαγής τιμής μεταβλητής

Στο παράδειγμα αυτό ορίζεται μία συνάρτηση η οποία αλλάζει δύο φορές την τιμή μιας μεταβλητής (myvar) και εκτυπώνει κάθε φορά τη μεταβλητή στην σελίδα HTML.

```
<HTML>
  <HEAD>
    <SCRIPT LANGUAGE="VBScript">
      <!-- Non-IE browsers skip the following lines.
        Sub VariableDemo
```

```

    Dim myvar
    myvar=25.34
    document.write("myvar is " & CStr(myvar) & "<BR>")
    myvar="hello"
    document.write("myvar is " & myvar & "<BR>")
    myvar=FALSE
    document.write("myvar is " & CStr(myvar) & "<BR>")
End Sub
-->
</SCRIPT>
</HEAD>
<BODY>
    <H1>Your First Example of variables</H1>
    <HR>
    <SCRIPT>
        <!--
        VariableDemo
        -->
    </SCRIPT>
</BODY>
</HTML>

```

Παρατηρούμε ότι μία μεταβλητή μπορεί να πάρει διαφορετικά είδη τιμών (είτε αριθμό, είτε αλφαριθμητικό, είτε μία τιμή τύπου Boolean), σε αντίθεση με άλλες γλώσσες προγραμματισμού στις οποίες κάθε μεταβλητή παίρνει ένα συγκεκριμένο είδος τιμών. Στη συνάρτηση VariableDemo η μεταβλητή myvar παίρνει αρχικά την τιμή 25.34 η οποία τυπώνεται στην σελίδα HTML με την εντολή document.write. Η myvar μετατρέπεται σε αλφαριθμητικό με τη συνάρτηση CStr και ενώνεται με την υπόλοιπη πρόταση που θα τυπωθεί στη σελίδα μέσω του τελεστή "&". Η CStr χρησιμοποιείται επίσης, όταν η myvar παίρνει τιμή τύπου Boolean, δηλαδή όταν η myvar γίνεται false. Η μετατροπή της τιμής της myvar μέσω της CStr δε χρειάζεται όταν η myvar παίρνει την τιμή "hello", όταν δηλαδή έχει είδη την τιμή αλφαριθμητικού.

Η συνάρτηση VariableDemo, καλείται μέσα από το κύριο τμήμα (Body) της HTML σελίδας. Η κλήση της συνάρτησης γίνεται με μία απλή αναφορά στο όνομά της, η οποία, όμως θα πρέπει να τοποθετηθεί ανάμεσα σε ετικέτες <SCRIPT> και </SCRIPT>.

Δήλωση Μεταβλητών

Υπάρχουν δύο τρόποι με τους οποίους μπορεί κάποιος να ορίσει μεταβλητές, ο έμμεσος και ο άμεσος. Χρησιμοποιώντας την εντολή Dim, δηλώνουμε κανονικά τις μεταβλητές. Όπως:

```
Dim Name
```

Θα μπορούσαμε, επίσης, να δηλώσουμε περισσότερες από μία μεταβλητές στην ίδια γραμμή, χωρίζοντάς τις με κόμματα.

```
Dim Name, Address, myvar
```

Οι μεταβλητές μπορούν επίσης να δηλωθούν με έμμεσο τρόπο, απλά με τη χρήση των ονομάτων τους μέσα στο κώδικα. Αυτός, όμως, ο τρόπος δεν ενδείκνυται γιατί οδηγεί σε λάθη.

Τα ονόματα που δίνονται στις μεταβλητές πρέπει να τηρούν τους ακόλουθους κανόνες:

- πρέπει να ξεκινούν με ένα χαρακτήρα αλφαριθμητικού (γράμμα ή αριθμό)
- δε πρέπει να περιλαμβάνουν τελείες
- πρέπει να είναι μοναδικές στο χώρο που έχουν ισχύ
- δεν πρέπει να ξεπερνούν τους 255 χαρακτήρες

Οι μεταβλητές που δηλώνονται μέσα στις συναρτήσεις / ρουτίνες ονομάζονται μεταβλητές επιπέδου ρουτίνας (procedure level variable) και έχουν ισχύ μόνο στις συναρτήσεις αυτές. Για να μπορέσουν να χρησιμοποιηθούν και από άλλες συναρτήσεις θα πρέπει να δηλωθούν εκτός των συναρτήσεων. Ονομάζονται τότε μεταβλητές επιπέδου σεναρίου και έχουν ισχύ για όλες τις συναρτήσεις του σεναρίου.

Στο παράδειγμα κώδικα που ακολουθεί η μεταβλητή counter μπορεί να χρησιμοποιηθεί και από τις δύο ρουτίνες, ενώ η temp μπορεί να χρησιμοποιηθεί μόνο από τη συνάρτηση Button1_OnClick.

```
<SCRIPT>
  Dim counter
  Sub Button1_onClick
    Dim temp
  End Sub
  Sub Button2_onClick
    ...
  End Sub
</SCRIPT>
```

Ανάθεση τιμών σε μεταβλητές

Όπως φάνηκε στα προηγούμενα παραδείγματα, για να δώσουμε τιμές στις μεταβλητές γράφουμε απλά το όνομα της μεταβλητής και την τιμή που θα της αναθέσουμε.

```
My_Variable = value
```

Τα ακόλουθα παραδείγματα αναθέτουν διάφορες τιμές στη μεταβλητή My_Variable

```
My_Variable = "John"  
My_Variable = 25.20  
My_Variable = True
```

Στην περίπτωση που η τιμή που δίνουμε στη μεταβλητή είναι τύπου αλφαριθμητικό, θα πρέπει να μπει ανάμεσα σε διπλά εισαγωγικά.

4.4.4 Οι πίνακες στη VBScript

Οι πίνακες (arrays) στη VBScript δεν είναι αντικείμενα όπως στη JavaScript αλλά μεταβλητές. Μπορούμε να δηλώσουμε ένα πίνακα (array) χρησιμοποιώντας την εντολή Dim, όπως ακριβώς γίνεται και με τις υπόλοιπες μεταβλητές:

```
Dim Names (5)
```

Ο παραπάνω πίνακας είναι ένας πίνακας με 6 στοιχεία. Ο πίνακας ξεκινάει από τη θέση 0, ενώ η θέση του τελευταίου του ψηφίου είναι ο αριθμός που δίνεται στη δήλωση του πίνακα (δηλαδή το 5).

Η ανάθεση τιμών στα στοιχεία των πινάκων γίνεται όπως η ανάθεση τιμών στις μεταβλητές.:

```
Names(0) = "John"  
Names(3) = "Mary"
```

Η VBScript υποστηρίζει πίνακες πολλών διαστάσεων. Μπορούμε να δηλώσουμε πίνακες που έχουν μέχρι 60 διαστάσεις. Στο παράδειγμα που ακολουθεί δηλώνουμε έναν πίνακα 3 διαστάσεων, που αποθηκεύει στοιχεία μεταπτυχιακών φοιτητών, όπως το όνομα και το επίθετο τους. Επίσης, ταξινομεί του φοιτητές ανά κατεύθυνση, η οποία προσδιορίζεται από την τρίτη διάσταση του πίνακα. Εάν έχουμε 4 κατευθύνσεις και 10 φοιτητές σε κάθε κατεύθυνση, ο πίνακας θα πρέπει να δηλωθεί ως εξής:

```
Dim Postgraduate(9,1,3)
```

Για να δώσουμε τιμή σε κάποιο στοιχείο του πίνακα θα πρέπει να προσδιορίσουμε τη θέση του ως προς όλες τις διαστάσεις του πίνακα. Έστω ότι έχουμε τον John Smith που είναι ο όγδοος κατά αλφαβητική σειρά στην πρώτη κατεύθυνση και τη Mary Richard, έκτη στην τρίτη κατεύθυνση. Οι φοιτητές αυτοί αποθηκεύονται στον πίνακα με τον ακόλουθο τρόπο:

```
Postgraduate(7,0,0)="John"  
  
Postgraduate(7,1,0)="Smith"  
  
Postgraduate(5,0,2)="Mary"  
  
Postgraduate(5,1,2)="Richard"
```

Η VBScript υποστηρίζει, επίσης, δυναμικούς πίνακες. Δηλαδή πίνακες των οποίων οι διαστάσεις δε δηλώνονται εξ αρχής και που μπορούν να μεταβληθούν κατά τη διάρκεια του script.

Για να δηλώσουμε έναν δυναμικό πίνακα, απλά γράφουμε:

```
Dim Customers()
```

Ενώ όταν θέλουμε να δηλώσουμε τη διάστασή του γράφουμε:

```
ReDim Customers(100)
```

4.4.5 Τελεστές στη VBScript

Η VBScript υποστηρίζει τους συνηθισμένους μαθηματικούς τελεστές (+, -, /, *), οι οποίοι λειτουργούν όπως και σε έναν υπολογιστή χειρός. Εκτός από τους τέσσερις γνωστούς τελεστές, η VBScript υποστηρίζει άλλους τρεις μαθηματικούς τελεστές (\, Mod, ^), επεξηγήσεις των οποίων δίνονται στο Παράρτημα 3.

Στη VBScript υπάρχουν οι ακόλουθοι τελεστές σύγκρισης (=, <>, <=<, >=>, >=) καθώς και οι λογικοί τελεστές: And, Or, Xor, Not, Eqv και Imp.

4.4.6 Συναρτήσεις στη VBScript

Υπάρχουν δύο είδη συναρτήσεων (procedures) στη VBScript. Οι συναρτήσεις που επιστρέφουν τιμή και αυτές που απλώς εκτελούν κάποιες ενέργειες χωρίς να επιστρέφουν τιμή. Η πρώτη κατηγορία συναρτήσεων ορίζονται με τη δήλωση Function ενώ η δεύτερη με τη δήλωση Sub.

Οι συναρτήσεις τύπου Sub αποτελούνται από μία σειρά εντολών οι οποίες περιβάλλονται από τις δηλώσεις Sub και End Sub. Οι συναρτήσεις αυτές δεν επιστρέφουν τιμές. Μπορούν να περιλαμβάνουν κάποια ορίσματα (π.χ. procedure_name(argument1, argument2)), τα οποία λαμβάνουν τιμές κατά την κλήση της συνάρτησης. Οι συναρτήσεις που δεν περιλαμβάνουν ορίσματα, ορίζονται μόνο με ένα ζεύγος παρενθέσεων (π.χ. procedure_name()).

```

Sub procedure_name()
    some statements
End Sub

or

Sub procedure_name(argument1,argument2,argument3)
    some statements
End Sub

```

Στο παράδειγμα που ακολουθεί η συνάρτηση PrintName δέχεται ως ορίσματα το όνομα και το επίθετο ενός χρήστη και τα εκτυπώνει στην σελίδα HTML.

```

Sub PrintName(name, surname)
    document.write ("Your name is: " & name & " <BR>")
    document.write ("Your surname is: " & surname)
End Sub

```

Οι συναρτήσεις που επιστρέφουν τιμή ορίζονται με τη δήλωση Function. Επιστρέφουν την τιμή που ανατίθεται στο όνομά τους. Αυτού του είδους οι συναρτήσεις μπορούν, επίσης, να περιλαμβάνουν ορίσματα (π.χ. function_name (argument1, argument2)). Όταν δε έχουν ορίσματα δηλώνονται με ένα απλό ζεύγος παρενθέσεων (π.χ. function_name ()).

```

Function function_name()
    some statements
    function_name=returnvalue
End Function

or

Function function_name (argument1,argument2)
    some statements
    function_name=returnvalue
End Function

```

Στο παράδειγμα που ακολουθεί γράφουμε μία συνάρτηση η οποία υπολογίζει και επιστρέφει το γινόμενο δύο αριθμών, αφού αυξήσει τον πρώτο κατά 10 και μειώσει το δεύτερο κατά 2.

```

Function multiply_numbers(number1, number2)
    number1=number1+10
    number2=number2-2
    Multiply_numbers= number1*number2
End Function

```


Κλήση Συναρτήσεων

Για να καλέσουμε μία συνάρτηση η οποία δεν επιστρέφει τιμή χρησιμοποιούμε είτε τη δήλωση Call, είτε αναφερόμαστε απευθείας στο όνομα της συνάρτησης. Εάν η συνάρτηση έχει ορίσματα, τους δίνουμε τιμή κατά την κλήση της συνάρτησης.

Έστω ότι έχουμε τη συνάρτηση με όνομα MyProcedure με δύο ορίσματα arg1 και arg2. Για να καλέσουμε τη συνάρτηση γράφουμε:

```
Call MyProcedure (arg1, arg2)
```

είτε

```
MyProcedure arg1, arg2
```

Για να καλέσουμε μία συνάρτηση που επιστρέφει τιμή, απλά αναφερόμαστε στο όνομά της δίνοντας τιμή στα ορίσματά της. Για να μπορέσουμε όμως να χρησιμοποιήσουμε την τιμή που επιστρέφει η συνάρτηση, θα πρέπει είτε να την αποθηκεύσουμε σε μία μεταβλητή είτε να την καλέσουμε μέσα από κάποια εντολή, όπως φαίνεται στο παραδείγμα που ακολουθεί.

```
name = findname()
```

```
msgbox "Your name is " & findname()
```

Στο παράδειγμα αυτό η συνάρτηση findname επιστρέφει μία τιμή η οποία εμφανίζεται στο κουτί μηνύματος.

Συναρτήσεις της VBScript

Η VBScript, υποστηρίζει δικές της συναρτήσεις που εκτελούν συγκεκριμένες λειτουργίες. Υπάρχουν συναρτήσεις που ελέγχουν το είδος των μεταβλητών, όπως οι IsArray, IsDate, Is Numeric και IsEmpty. Άλλες μετατρέπουν μεταβλητές όπως οι Cbyte, CInt, CDouble, CStr, CDate, DateValue και TimeValue. Υπάρχουν συγκεκριμένες συναρτήσεις για πίνακες (Arrays) (π.χ. LBound, UBound), για αντικείμενα (objects) (π.χ. IsObject), αλφαριθμητικά (strings) (π.χ. Lcase, Ucase, InStr, Left, Mid, Right, StrComp, Chr, Trim) και στοιχεία Ημερομηνίας και Ώρας (π.χ. Date, Time, Month, Weekly, Minute). Τέλος η VBScript υποστηρίζει μαθηματικές συναρτήσεις (π.χ. Cos, Sin, Tan, Exp) καθώς και συναρτήσεις για την εμφάνιση παραθύρων μηνυμάτων, όπως η MsgBox και InputBox, που είδαμε σε προηγούμενα παραδείγματα.

4.4.7 Εντολές υπό συνθήκη

Χρησιμοποιώντας εντολές υπό συνθήκη μπορούμε να αναπτύξουμε scripts τα οποία ελέγχουν και συγκρίνουν δεδομένα και χρησιμοποιούν κριτήρια για να προσδιορίσουν τις ενέργειες που θα ακολουθήσουν.

Η VBScript υποστηρίζει δύο μορφές εντολών υπό συνθήκη:

- If...Then...Else
- Select...Case

If..Then..Else

Η if...then...else εντολή χρησιμοποιείται για να ελέγξει εάν μία συνθήκη είναι αληθής ή ψευδής. Εάν η συνθήκη είναι αληθής θα εκτελεστούν οι εντολές που βρίσκονται κάτω από την εντολή if. Εάν όμως η συνθήκη είναι ψευδής θα εκτελεστούν οι εντολές που βρίσκονται κάτω από την εντολή else.

Η πιο απλή μορφή αυτής της υπό συνθήκη εντολής περιλαμβάνει μόνο μία πρόταση if. Ελέγχει εάν μία συνθήκη είναι αληθής και εάν είναι εκτελεί την εντολή που ακολουθεί διαφορετικά προχωράει στις επόμενες γραμμές κώδικα.

```
If (x>8.5) then document.write("Your grade is Excellent")
```

```
If ((x>6.5) and (x<8.5)) then document.write("Your grade is Very Good")
```

Κάτω από την πρόταση if μπορούν, επίσης, να ακολουθούν περισσότερες από μία εντολές. Στην περίπτωση αυτή το τέλος της εντολής υπό συνθήκη δηλώνεται με την εντολή end if.

```
If (5<x<6.5) then
    document.write("Your grade is Good <BR>")
    document.write("You have to try harder")
end if
```

Η πρόταση else επιτρέπει τη δήλωση εντολών που θα εκτελεστούν στην περίπτωση που η συνθήκη if αποδειχθεί ψευδής.

```
if (x>5) then
    document.write("You have passed the exams")
else
    document.write("You have failed")
```

Παράδειγμα

Το παράδειγμα αυτό παρουσιάζει ένα τμήμα από κάποιο σενάριο το οποίο υπολογίζει την τιμή ενός εισιτηρίου με βάση την ηλικία του χρήστη καθώς και με το αν ο χρήστης είναι φοιτητής. Το script υπολογίζει επίσης το συνολικό αριθμό των φοιτητών και τον αποθηκεύει σε μία μεταβλητή count.

```
if (age<18) then
    price=10
elseif (age<=25) then
    if (student=TRUE) then
        price=12
        count=count+1
```

```

else
    price=15
end if
elseif (age<=28) then
    if (student=TRUE) then
        price=17
        count=count+1
    else
        price=20
    end if
else
    price=25
end if

```

Select Case

Εκτός όμως από την *if...then* εντολή υπάρχει και η *Select Case* εντολή η οποία χρησιμοποιείται στην περίπτωση που πρέπει να ελεγχθούν πολλές συνθήκες για μία μεταβλητή. Οι ίδιοι έλεγχοι μπορούν να πραγματοποιηθούν και με την *if...then...else* εντολή, όμως ο κώδικας είναι πιο πολύπλοκος. Όπως και στην περίπτωση της *if* εντολής η δομή της *Select Case* ελέγχει μία συνθήκη και εάν αυτή είναι αληθής εκτελεί μια σειρά από εντολές.

Η *Select case* εντολή έχει την ακόλουθη σύνταξη:

```

Select Case condition

    Case value

    Case value

    ...

    Case Else

End Select

```

Εάν για παράδειγμα, θέλουμε να προσδιορίσουμε την τιμή ενός εισιτηρίου με βάση την κατηγορία στην οποία ανήκει ο χρήστης, γράφουμε την ακόλουθη εντολή *select...case*.

```

Select Case Document.frmOrder.txtStatus.Value

    Case "Child"

        TicketPrice= 10

    Case "Student"

        TicketPrice = 15

```

```

Case "Elderly"
    TicketPrice= 20
Case Else
    TicketPrice=25
End Select

```

Η Select Case ελέγχει κάθε μία από τις συνθήκες case έως ότου βρει μία που να είναι αληθής. Εάν καμία από τις συνθήκες δε βρεθεί αληθής τότε εκτελείται η εντολή που βρίσκεται μέσα στην case else.

Παρόλο που η case else δεν είναι υποχρεωτικό να υπάρχει σε μία select case εντολή, είναι καλό να χρησιμοποιείται για να επεξεργάζεται συνθήκες που δεν είχαν ληφθεί υπόψη εξ αρχής.

Οι παραπάνω εντολές θα μπορούσαν επίσης, να γραφούν με τη χρήση της εντολή if...then...else. Στην περίπτωση όμως αυτή, όπως φαίνεται παρακάτω, ο κώδικας είναι πιο πολύπλοκος.

```

Status= Document.frmOrder.txtStatus.Value
if status="child" then
    TicketPrice=10
elseif status="Student" then
    TicketPrice = 15
elseif status="Elderly" then
    TicketPrice= 20
else
    TicketPrice=25
End if

```

4.4.8 Εντολές Βρόχων

Οι βρόχοι επιτρέπουν την επανάληψη μίας συγκεκριμένης ακολουθίας ενεργειών και εντολών περισσότερες από μία φορές.

Η VBScript υποστηρίζει τις ακόλουθες εντολές βρόχων:

- For...Next
- For Each...Next
- Do...While
- Do...Until
- While...Wend

Τις εντολές αυτές μπορούμε να τις χωρίσουμε σε δύο κατηγορίες: Στις εντολές For (For...Next και For Each...Next) και στις εντολές Do ή While. Η πρώτη κατηγορία εντολών χρησιμοποιείται όταν η εκτέλεση ορισμένων ενεργειών επαναλαμβάνεται για συγκεκριμένο αριθμό φορών, ενώ η δεύτερη κατηγορία χρησιμοποιείται όταν οι ενέργειες ή οι εντολές επαναλαμβάνονται έως ότου μία συνθήκη γίνει αληθής ή ψευδής.

For..Next

Η εντολή *For..Next* χρησιμοποιείται όταν θέλουμε να επαναλάβουμε κάποιες ενέργειες ένα συγκεκριμένο αριθμό φορών. Λειτουργεί σαν ένας μετρητής ο οποίος αυξάνεται ή μειώνεται κάθε φορά που επαναλαμβάνεται ο βρόχος. Το παράδειγμα που ακολουθεί παρουσιάζει ένα βρόχο For...Next ο οποίος επαναλαμβάνεται 5 φορές και εκτυπώνει τον αριθμό των φορών που εκτελείται ο βρόχος:

```
For counter = 1 To 5
    document.write("Number of times in the loop: " &
        CStr(counter) & "<BR>")
Next counter
```

Στο παράδειγμα αυτό ο μετρητής (counter) ξεκινάει από την τιμή 1 και φτάνει τελικά στην τιμή 5. Η εντολή Next counter που βρίσκεται στο τέλος του βρόχου αυξάνει την τιμή του μετρητή. Η αύξηση που γίνεται βασίζεται στο βήμα αύξησης που δίνεται στην πρόταση for. Όταν δεν έχει προσδιοριστεί κάποιο βήμα αύξησης, όπως στο παραπάνω παράδειγμα, ο μετρητής αυξάνεται πάντα κατά 1.

Το βήμα αύξησης μπορεί να προσδιοριστεί με τη δήλωση Step όπως φαίνεται στον κώδικα που ακολουθεί:

```
For counter = 1 To 8 Step 2
    result = 2 * counter
    MsgBox CStr(counter) & " times 2 is " & CStr(result)
Next counter
```

Η VBScript υποστηρίζει, επίσης, τη δυνατότητα να ορίσουμε αρνητικό βήμα. Σ' αυτήν την περίπτωση, όμως, ο αρχικός αριθμός που παίρνει ο μετρητής πρέπει να είναι μεγαλύτερος από τον τελικό. Δηλαδή:

```
For counter = 8 To 1 Step -1
    ...
Next counter
```

For Each..Next

Ο βρόχος *For Each..Next* είναι παρόμοιος με το βρόχο *For..Next*, μόνο που αντί να επαναλαμβάνεται έναν ορισμένο αριθμό φορών, επαναλαμβάνεται για κάθε στοιχείο μιας συγκεκριμένης συλλογής.

```
Dim colors(3)
colors(0)= "red"
colors(1)= "blue"
colors(2)= "green"
colors(3)= "yellow"

For each color in colors
    Document.write("The color is: " & color & "<BR>")
Next
```

Η συνάρτηση αυτή εκτυπώνει τις ακόλουθες προτάσεις στην σελίδα HTML:

```
The color is: red
The color is: blue
The color is: green
The color is: yellow
```

Do..Loop

Η εντολή βρόχου *Do..Loop* επαναλαμβάνει ένα σύνολο εντολών έως ότου πάψει να ισχύει μία συνθήκη. Συνήθως η συνθήκη που ελέγχουμε είναι αποτέλεσμα λειτουργιών που εκτελούνται εντός του βρόχου. Υπάρχουν δύο εκδόσεις ενός τέτοιου βρόχου, ο βρόχος *Do... While* και ο βρόχος *Do... until*.

Do..While

Ο βρόχος *Do...Loop* που περιλαμβάνει τη δήλωση *while*, επαναλαμβάνει την εκτέλεση μίας σειράς εντολών όσο εξακολουθεί να ισχύει μία συνθήκη. Όπως φαίνεται παρακάτω, ο έλεγχος για τη συνθήκη μπορεί να γίνεται είτε στην αρχή είτε στο τέλος του βρόχου.

```
Do While condition

    statement

    statement

    ...

Loop
```

Παράδειγμα:

```
counter=0
do while(counter<=10)
    document.write("Counter is : " &counter & "<BR>")
    counter=counter+2
loop
```

Η ίδια εντολή μπορεί επίσης να γραφτεί με την ακόλουθη μορφή:

```
Do
    statement
    statement
    ...
Loop While condition
```

Η διαφορά ανάμεσα στους δύο τρόπους σύνταξης του βρόχου είναι ότι στην πρώτη περίπτωση οι εντολές μπορεί να μην εκτελεστούν καθόλου, ενώ στη δεύτερη θα εκτελεστούν τουλάχιστον μία φορά.

Do..Until

Ένας βρόχος *Do* ο οποίος περιλαμβάνει μία εντολή *Until* επαναλαμβάνει μία σειρά από εντολές όσο μία συνθήκη είναι ψευδής. Όταν η συνθήκη γίνει αληθής ο βρόχος σταματάει. Όπως και στην περίπτωση της εντολής *Do..While*, υπάρχει δυνατότητα ο έλεγχος για την συνθήκη να γίνεται είτε στην αρχή είτε στο τέλος του βρόχου:

```
Do Until condition
    statement
    statement
    ...
Loop
```

Μπορεί επίσης να γραφτεί ως εξής:

```
Do
    statement
    statement
    ...
Loop Until
```

Η διαφορά των δύο αυτών τρόπων, είναι ότι στην πρώτη περίπτωση οι εντολές του βρόχου μπορεί να μην εκτελεστούν καθόλου, ενώ στη δεύτερη εκτελούνται τουλάχιστον μία φορά.

Μία *Do..Loop* εντολή μπορεί να χρησιμοποιηθεί όπως φαίνεται στο παρακάτω παράδειγμα:

```
password = InputBox("Enter your password:")
Do Until password = "adefthlk"
    MsgBox "Invalid password - please try again."
    Password = InputBox("Enter you password:")
Loop
```

Στο παράδειγμα αυτό εμφανίζεται ένα παράθυρο εισαγωγής κειμένου στο οποίο ο χρήστης εισάγει τον κωδικό πρόσβασης του (password). Εάν ο κωδικός πρόσβασης είναι "adefthlk", η συνθήκη password = "adefthlk" είναι αληθής και οι εντολές του βρόχου δε θα εκτελεστούν καθόλου. Εάν, όμως, το password δεν είναι "adefthlk", η συνθήκη password = "adefthlk" είναι ψευδής και αρχίζουν να εκτελούνται οι εντολές του βρόχου. Αρχικά εμφανίζεται ένα μήνυμα λάθους στο χρήστη και στη συνέχεια εμφανίζεται πάλι το παράθυρο για την εισαγωγή του password. Οι ενέργειες αυτές επαναλαμβάνονται μέχρι ο χρήστης να πληκτρολογήσει το σωστό password, μέχρι, δηλαδή, η συνθήκη να γίνει αληθής.

While..Wend

Η τελευταία εντολή βρόχου που υποστηρίζει η VBScript είναι η *While..Wend*. Η οποία επαναλαμβάνει τις εντολές που υπάρχουν μέσα στο βρόχο όσο μία συνθήκη είναι αληθής. Η εντολή αυτή βρόχου μοιάζει με την εντολή *Do...Loop*, μόνο που δεσμεύει ο έλεγχος της συνθήκης να γίνεται πάντα στην αρχή.

Η δομή για τη εντολή *While...Wend* είναι:

```
While condition
    statement
    statement
    ...
Wend
```


Το προηγούμενο παράδειγμα της εισαγωγής του password θα μπορούσε επίσης να υλοποιηθεί με τη χρήση της εντολής while...wend ως εξής:

```
password = InputBox("Enter your password:")
While password <> "adefthlk"
    MsgBox "Invalid password - please try again."
    Password = InputBox("Enter you password:")
Wend
```

Στο παράδειγμα αυτό όσο το password που δίνει ο χρήστης είναι διαφορετικό από το "adefthlk", εμφανίζεται ένα μήνυμα λάθους και δίνεται ξανά η ευκαιρία εισαγωγής του password στο χρήστη. Όταν η συνθήκη password <> "adefthlk" γίνει ψευδής, δηλαδή όταν ο χρήστης πληκτρολογήσει το σωστό password τότε ο βρόχος παύει να εκτελείται.

4.4.9 Φόρμες HTML και VBScript

Όπως όλες οι γλώσσες σεναρίων, η VBScript μπορεί να χρησιμοποιηθεί για τον έλεγχο των στοιχείων μίας φόρμας, προτού αυτά αποσταλούν στον εξυπηρετητή. Στην ενότητα αυτή θα δοθούν παραδείγματα για το πώς η VBScript μπορεί να ελέγξει τα στοιχεία μίας φόρμας και να εμφανίζει μήνυμα λάθους όταν αυτά λείπουν ή δεν έχουν πληκτρολογηθεί σωστά. Αφού ολοκληρωθεί ο έλεγχος των στοιχείων που εισήγαγε ο χρήστης η φόρμα μπορεί να υποβληθεί και τα στοιχεία της να αποσταλούν στον εξυπηρετητή μέσω της εντολής submit. Στην ενότητα αυτή θα ασχοληθούμε τόσο με την ενέργεια ελέγχου των στοιχείων μιας φόρμας όσο και με την ενέργεια της υποβολής της φόρμας.

Έλεγχος στοιχείων φόρμας

Ο έλεγχος των στοιχείων μιας φόρμας περιλαμβάνει:

- Έλεγχος για το εάν έχουν δοθεί όλα τα στοιχεία και,
- Έλεγχος για το αν αυτά είναι σωστά.

Για τον έλεγχο των στοιχείων μιας φόρμας χρησιμοποιούνται οι εντολές που επεξηγήθηκαν στις προηγούμενες παραγράφους. Ο τρόπος που γίνεται ο έλεγχος των στοιχείων που εισάγει ένας χρήστης φαίνεται στο ακόλουθο παράδειγμα.

Παράδειγμα

Το παράδειγμα αυτό παρουσιάζει μία πολύ απλή φόρμα στην οποία ο χρήστης παραγγέλνει κάποιο προϊόν. Υπάρχει ένα πεδίο κειμένου στο οποίο ο χρήστης εισάγει την ποσότητα που θέλει να παραγγείλει. Το πρόγραμμα ελέγχει εάν ο χρήστης έχει συμπληρώσει σωστά αυτό το πεδίο και εάν όχι εμφανίζει μήνυμα λάθους στο χρήστη. Διαφορετικά, προωθεί την παραγγελία.

```

<HTML>
<HEAD>
<SCRIPT LANGUAGE="VBScript">
<!--Instruct non-IE browsers to skip over VBScript modules.
Sub SubmitItems_OnClick
` Check if the user has filled in the quantity's textbox.
  If (Len(document.frmOrder.txtQuantity.value) = 0) Then
    MsgBox "You must type the number of items you want to order."
    Exit Sub
  End If
` Check if the user has entered a number in the textbox.
  If (Not(IsNumeric(document.frmOrder.txtQuantity.value))) Then
    MsgBox "You must enter a number to specify how many products you
    want to order."
    Exit Sub
  End If
` Check to see if the number specified is valid.
  If (document.frmOrder.txtQuantity.value < 0) then
    MsgBox "The number of items you entered is invalid."
    Exit Sub
  End If
` If data looks valid submit it.
  MsgBox "You have ordered" &
  CStr(document.frmOrder.txtQuantity.value) & "items. <BR> Thank you
  for choosing our product"
  Document.frmOrder.submit
End Sub
-->
</SCRIPT>
</HEAD>
<BODY>
<H1>A VBScript Example on Checking User's Input</H1>
<P> This example demonstrates validation techniques in VBScript. </P>
<FORM NAME="frmOrder">

```

```

<TABLE>
  <TR>
    <TD>Enter the number of items:</TD>
    <TD><INPUT TYPE="Text" NAME="txtQuantity" SIZE="3">
  <TR>
    <TD><INPUT TYPE="Button" NAME="SubmitItems" VALUE="Submit"></TD>
    <TD></TD>
  </TR>
</TABLE>
</FORM>
</BODY>
</HTML>

```

Παρατηρούμε ότι έχουμε μία φόρμα η οποία περιλαμβάνει μόνο ένα πεδίο κειμένου

```
<INPUT TYPE="Text" NAME="txtQuantity" SIZE="3">
```

και ένα πλήκτρο υποβολής των στοιχείων της φόρμας.

```
<INPUT TYPE="Button" NAME="SubmitItems" VALUE="Submit">
```

Ο χρήστης πληκτρολογεί στο πεδίο κειμένου τον αριθμό των προϊόντων που θέλει να παραγγείλει και στη συνέχεια επιλέγει το πλήκτρο με όνομα "SubmitItems" και όνομα για να προχωρήσει η παραγγελία. Με την επιλογή του πλήκτρου "SubmitItems" ενεργοποιείται η συνάρτηση SubmitItems_OnClick η οποία προωθεί την παραγγελία, αφού όμως πρώτα ελέγξει εάν ο χρήστης συμπλήρωσε σωστά τον αριθμό των προϊόντων που επιθυμεί να παραγγείλει.

Αρχικά η συνάρτηση ελέγχει εάν το πεδίο έχει συμπληρωθεί:

```

If (Len(document.frmOrder.txtQuantity.value) = 0) Then
  MsgBox "You must type the number of items you want to order."
  Exit Sub
End If

```

Για να το διαπιστώσει αυτό χρησιμοποιεί τη συνάρτηση Len και υπολογίζει το μήκος του κειμένου που έχει πληκτρολογήσει ο χρήστης, δηλαδή το κείμενο που βρίσκεται μέσα στο πεδίο κειμένου txtQuantity (document.frmOrder.txtQuantity.value). Εάν το πεδίο βρεθεί κενό η συνάρτηση εμφανίζει μήνυμα λάθους λέγοντας στο χρήστη πως πρέπει να συμπληρώσει αυτό το πεδίο και στη συνέχεια διακόπτει την παραγγελία.

Στη συνέχεια η συνάρτηση ελέγχει εάν ο χρήστης έχει πληκτρολογήσει έναν αριθμό.

```

If (Not(IsNumeric(document.frmOrder.txtQuantity.value))) Then
    MsgBox "You must enter a number to specify how many
products you want to order."
    Exit Sub
End If

```

Ο έλεγχος αυτός γίνεται με την μέθοδο IsNumeric(). Εάν το IsNumeric(document.frmOrder.txtQuantity.value) επιστρέψει false το Not(IsNumeric(document.frmOrder.txtQuantity.value)) θα είναι true με αποτέλεσμα να εκτελεστούν οι εντολές μέσα στη δήλωση If...End If. Σε αυτή την περίπτωση η συνάρτηση εμφανίζει ένα μήνυμα λάθους στο χρήστη ενημερώνοντάς τον πως πρέπει να πληκτρολογήσει ένα νούμερο. Η συνάρτηση διακόπτεται (Exit Sub) χωρίς να πραγματοποιήσει την παραγγελία.

Ο τελευταίος έλεγχος που γίνεται είναι για να διαπιστωθεί εάν ο αριθμός που πληκτρολόγησε ο χρήστης είναι έγκυρος. Ελέγχει εάν ο αριθμός είναι αρνητικός, οπότε και εμφανίζει μήνυμα λάθους και διακόπτει την παραγγελία.

```

If (document.frmOrder.txtQuantity.value < 0) then
    MsgBox "The number of items you entered is invalid."
    Exit Sub
End If

```

Εάν και οι τρεις αυτοί έλεγχοι αποτύχουν, η τιμή που πληκτρολόγησε ο χρήστης θεωρείται έγκυρη. Τότε εμφανίζεται σχετικό μήνυμα στο χρήστη και τα στοιχεία της φόρμας υποβάλλονται στον εξυπηρετητή (document.frmOrder.submit).

Στο τρίτο κεφάλαιο είδαμε πως για να υποβάλλουμε τα στοιχεία μιας φόρμας στον εξυπηρετητή θα πρέπει να έχουμε ένα αντικείμενο τύπου SUBMIT. Στο συγκεκριμένο παράδειγμα διαπιστώσαμε πως η υποβολή μιας φόρμας μπορεί επίσης να πραγματοποιηθεί μέσα από μία συνάρτηση VBScript με τη μέθοδο submit, του αντικειμένου της φόρμας.

4.5 Εισαγωγή στα Java Applets

4.5.1 Γενικά χαρακτηριστικά της Java

Όπως προαναφέρθηκε στην αρχή του κεφαλαίου η Java έχει πολλές διαφορές με τη JavaScript. Οι γλώσσες σεναρίων επιτρέπουν τη διασύνδεση εντολών με αντικείμενα της σελίδας, αλλά δεν έχουν απόλυτο έλεγχο πάνω στην εμφάνιση της σελίδας. Ο κώδικάς τους είναι ενσωματωμένος σε αυτόν της HTML σελίδας και ερμηνεύεται και εκτελείται την ώρα που φορτώνεται στο φυλλομετρητή. Καμιά φορά, όμως, χρειάζεται μια λύση που δεν εξαρτάται από το συγκεκριμένο φυλλομετρητή. Αυτή η λύση δίνεται από τη Java που επιτρέπει ολόκληρες εφαρμογές, που ονομάζονται applets, να κατεβαίνουν και να τρέχουν στο μηχάνημα του χρήστη.

Η Java, όμως, δε χρησιμοποιείται μόνο για ανάπτυξη εφαρμογών στο Διαδίκτυο. Αναπτύχθηκε κυρίως για να επιλύσει το πρόβλημα της ασυμβατότητας και της μη μεταφερισιμότητας των προγραμμάτων σε διαφορετικά μηχανήματα και επεξεργαστές.

Στις περισσότερες γλώσσες προγραμματισμού πρέπει να προαποφασιστεί σε ποιον επεξεργαστή και σε ποιο λειτουργικό σύστημα πρόκειται να τρέξει το πρόγραμμα, ώστε να ενσωματωθούν βιβλιοθήκες που σχετίζονται με αυτό το λειτουργικό σύστημα. Όταν το πρόγραμμα είναι έτοιμο προς εκτέλεση στέλνεται σε έναν μεταγλωττιστή, ο οποίος το μετατρέπει σ' ένα σύνολο από εντολές, που ο συγκεκριμένος επεξεργαστής καταλαβαίνει και μπορεί να εκτελέσει.

Ο προγραμματισμός σε Java είναι διαφορετικός διότι δε χρειάζεται να ενσωματωθούν κλήσεις σε βιβλιοθήκες του συγκεκριμένου λειτουργικού συστήματος (π.χ. Windows, Mac OS). Η Java έχει δικές της βιβλιοθήκες, που ονομάζονται πακέτα (packages). Επίσης, τα προγράμματά της δεν παρουσιάζουν διαφορά στην εκτέλεσή τους από διαφορετικούς επεξεργαστές. Ο μεταγλωττιστής της Java δεν μεταφράζει το πρόγραμμα σε πρωτογενείς εντολές, όπως συμβαίνει στις άλλες γλώσσες προγραμματισμού, αλλά σε bytecode που εκτελούνται από τη JVM (Java Virtual Machine). Η JVM δεν είναι μία πραγματική μηχανή, αλλά ένα λογισμικό το οποίο έχει υλοποιηθεί από την εταιρεία Sun, σε διάφορες εκδόσεις, για τις πιο συνηθισμένες πλατφόρμες. Έτσι όταν το αρχείο, που έχει μορφή bytecode (δηλαδή το αρχείο class), φορτώνεται, σε μία μηχανή και εκτελείται από τη JVM της μηχανής αυτής. Η JVM διαβάζει το αρχείο class και εκτελεί τις εντολές που έχουν προσδιοριστεί στο αρχείο Java.

Όπως αναφέρθηκε η Java είναι μία αντικειμενοστραφής γλώσσα προγραμματισμού και οι εφαρμογές σε Java αντιμετωπίζονται σαν ένα σύνολο από κλάσεις που σχετίζονται μεταξύ τους. Οι κλάσεις περιλαμβάνουν μεθόδους, οι οποίες είτε έχουν υλοποιηθεί και υποστηρίζονται από τη Java είτε μπορούν να οριστούν και να δημιουργηθούν από προγραμματιστές.

Άλλο ένα χαρακτηριστικό της Java είναι ότι υποστηρίζει το μοντέλο πελάτη-εξυπηρετητή. Όπως συμβαίνει και στη C++ η Java επέκτεινε τις βιβλιοθήκες για να υποστηρίξουν δικτυακή επικοινωνία. Για παράδειγμα η Java επιτρέπει να πραγματοποιηθεί μία σύνδεση με μία σελίδα ή μία εφαρμογή Διαδικτύου και να γράψει ή να διαβάσει δεδομένα όπως η C++ γράφει και διαβάζει δεδομένα από ένα τοπικό τερματικό.

4.5.2 Εφαρμογές Java και Java Applets

Το κύριο πλεονέκτημα της Java είναι ότι επιτρέπει την ανάπτυξη μεταφέρσιμων (portable) εφαρμογών. Δηλαδή εφαρμογών που μπορούν να εκτελεστούν χωρίς ιδιαίτερες διαφορές σε διαφορετικά μηχανήματα και επεξεργαστές. Όπως αναφέρθηκε στην προηγούμενη παράγραφο, στις περισσότερες γλώσσες προγραμματισμού οι εφαρμογές / τα προγράμματα αφού υλοποιηθούν μετατρέπονται σε ένα εκτελέσιμο αρχείο με κατάληξη (exe), το οποίο αποτελείται από εντολές σε δυαδική μορφή. Στη Java, οι εφαρμογές μετατρέπονται σε αρχεία με κατάληξη class, τα οποία έχουν μορφή bytecodes και μπορούν να εκτελούνται από τη JVM (Java Virtual Machine).

Εκτός από τις αυτόνομες (stand alone) εφαρμογές, η Java υποστηρίζει, επίσης, εφαρμογές που εκτελούνται μέσα από φυλλομετρητές. Οι εφαρμογές αυτές ονομάζονται

applets και δημιουργήθηκαν για να εκμεταλλευτούν τη μεγάλη διάδοση και χρήση του Διαδικτύου. Για να μπορέσουν να εκτελεστούν τα applets θα πρέπει να μετατραπούν σε μορφή bytecode, δηλαδή σε αρχεία με κατάληξη class. Τα αρχεία αυτά καλούνται από τη σελίδα HTML μέσα από την ετικέτα <APPLET>. Όταν η σελίδα HTML κληθεί από ένα φυλλομετρητή το applet μεταφέρεται ολόκληρο στο μηχάνημα του χρήστη και εκτελείται από το φυλλομετρητή.

Μέσω των applets δίνεται η δυνατότητα στους φυλλομετρητές να εκτελούν και να εμφανίζουν ολόκληρες εφαρμογές. Οι σελίδες HTML μπορούν έτσι να γίνονται εξαιρετικά πλούσιες και δυναμικές. Οι δυνατότητες που παρέχουν τα Java Applets ξεπερνούν αυτές των γλωσσών σεναρίων, οι οποίες συμβάλλουν στη δημιουργία δυναμικών σελίδων χωρίς όμως να έχουν απόλυτο έλεγχο στην εμφάνιση και τα αντικείμενα της διεπαφής χρήστη. Ένα άλλο πλεονέκτημα των Java applets είναι ότι επιτρέπουν τη σύνδεση εφαρμογών του μηχανήματος του χρήστη με εφαρμογές του εξυπηρετητή, επιτρέποντας έτσι την ανάπτυξη εφαρμογών πελάτη-εξυπηρετητή.

Υπάρχουν, όμως, και μειονεκτήματα όπως το μέγεθος των applets, το οποίο μεγαλώνει ανάλογα με τη λειτουργικότητα τους, ο χρόνος που χρειάζονται για να φορτωθούν στο μηχάνημα του χρήστη, η υπολογιστική ισχύς και οι πόροι που καταναλώνουν. Άλλο ένα μειονέκτημα των applets είναι ότι δε μπορούν να εκτελεστούν πάντα από όλους τους φυλλομετρητές. Η μεταφερσιμότητά τους εξαρτάται από την έκδοση της Java που υποστηρίζεται ή που έχει εγκατασταθεί στο συγκεκριμένο φυλλομετρητή.

4.5.3 Βασικά στοιχεία της Java

Τύποι δεδομένων

Η Java είναι πολύ πιο αυστηρή στη δήλωση μεταβλητών από τις JavaScript και τη VBScript. Ο τύπος της μεταβλητής θα πρέπει να δηλώνεται εξαρχής και δεν μπορεί να αλλάζει κατά τη διάρκεια του προγράμματος. Αντίθετα στη VBScript οι προγραμματιστές δεν είναι υποχρεωμένοι να δηλώνουν τον τύπο των μεταβλητών από την αρχή και οι μεταβλητές αλλάζουν τύπο ανάλογα με την τιμή που τους ανατίθεται.

Οι τύποι μεταβλητών που υποστηρίζει η Java παρουσιάζονται στον πίνακα 4.5.1:

Πίνακας 4.5.1 – Τύποι μεταβλητών της Java

Boolean	Παίρνει τις τιμές true ή false
Char	Ένας Unicode χαρακτήρας (καταλαμβάνει 16 bit)
Byte	Ακέραιος αριθμός (-127 έως 127) (καταλαμβάνει 8 bit)
Short	Ακέραιος αριθμός (-32768 έως 32767) (καταλαμβάνει 16 bit)
Int	Ακέραιος αριθμός (καταλαμβάνει 32 bit)
Long	Μεγάλος ακέραιος αριθμός (καταλαμβάνει 64 bit)
Float	Πραγματικός αριθμός (καταλαμβάνει 32 bit)
Double	Μεγάλος πραγματικός αριθμός (καταλαμβάνει 64 bit)

Τελεστές στη Java

Με τη βοήθεια των τελεστών μπορούμε να χειριστούμε τις μεταβλητές στη Java. Οι τελεστές της Java αναλύονται στο παράρτημα III. Σε αυτήν την παράγραφο υπάρχουν παραδείγματα των τελεστών που χρησιμοποιούνται πιο συχνά και που διαφοροποιούνται από αυτές άλλων γλωσσών προγραμματισμού.

Η Java υποστηρίζει τελεστές αυτόματης αύξησης (++) και αυτόματης μείωσης (--) της τιμής μιας μεταβλητής. Π.χ. έστω ότι έχουμε τις ακόλουθες εντολές

```
j=1;  
j++;
```

Η πρώτη εντολή αναθέτει στο j την τιμή 1 ενώ η δεύτερη αυξάνει την τιμή του j κατά 1 και η τελική τιμή του j γίνεται 2. Το j++ είναι επομένως ισοδύναμο με την εντολή (j=j+1).

Όπως δείξαμε η ανάθεση τιμής γίνεται με το (=). Ο έλεγχος για την ισότητα δύο τιμών γίνεται με το (==) ενώ ο έλεγχος για την ανισότητά τους γίνεται με τον τελεστή (!=).

Η Java υποστηρίζει επίσης τους λογικούς τελεστές (&&, ||, !) οι οποίοι συμβολίζουν το AND, OR και NOT αντίστοιχα. Η χρήση των τελεστών αυτών φαίνεται στα παρακάτω παραδείγματα:

```
if (k == j) && (j != 5) {  
}  
if (k != j) || (k == 5) {  
}  
if !(k == 5) {  
}
```

Η πρώτη συνθήκη ισχύει εάν η μεταβλητή k είναι ίση με τη j και εάν η μεταβλητή j είναι διαφορετική από το 5. Η δεύτερη ισχύει όταν η k είναι διαφορετική από τη j, είτε όταν η k είναι ίση με 5. Τέλος, η τρίτη ισχύει όταν η συνθήκη (k==5) είναι ψευδής.

Η Java υποστηρίζει, επίσης, κάποιους τελεστές που συνδυάζουν μία πράξη με την ανάθεση της τιμής της. Οι τελεστές αυτοί παρουσιάζονται στον Πίνακα 4.5.2:

Πίνακας 4.5.2 – Τελεστές που συνδυάζουν πράξη με ανάθεση τιμής

Τελεστής	Χρήση	Αποτέλεσμα
+=	op1 += op2	op1 = op1 + op2
-=	op1 -= op2	op1 = op1 - op2
*=	op1 *= op2	op1 = op1 * op2
/=	op1 /= op2	op1 = op1 / op2
%=	op1 %= op2	op1 = op1 % op2

Πίνακες στη Java

Για τη δήλωση ενός πίνακα χρησιμοποιούμε τους τελεστές της αγκύλης “[]”. Όπως και στη JavaScript οι πίνακες στη Java δεν είναι απλοί τύποι δεδομένων αλλά κλάσεις. Για να δημιουργήσουμε ένα νέο πίνακα θα πρέπει πρώτα να του δώσουμε ένα όνομα (π.χ. arrayOfFloats) και να δηλώσουμε τι τύπου είναι (π.χ. float[] arrayOfFloats) και στη συνέχεια να ορίσουμε ένα συγκεκριμένο αντικείμενο (π.χ. arrayOfFloats= new float[10]). Η δήλωση ενός πίνακα μπορεί να γραφτεί και σε μία γραμμή δηλαδή:

```
Float[] arrayOfFloats = new float[10];
```

Για να αναφερθούμε σε ένα στοιχείο πίνακα της Java χρησιμοποιούμε το όνομα του πίνακα και τη θέση του στοιχείου (π.χ. arrayOfFloats[6]). Το πρώτο στοιχείο των πινάκων της Java έχει πάντα τη θέση 0. Άρα το arrayOfFloats[6] είναι το 7^ο στοιχείο του πίνακα. Στο παράδειγμα που ακολουθεί φαίνεται η δήλωση ενός πίνακα καθώς και η ανάθεση τιμών στα στοιχεία του.

```
public class ArrayExample {
    public static void main(String[] args) {

        // declaration of an array of integers
        int[] anArray;

        // create an array of integers
        anArray = new int[5];

        // assign a value to each array element
        for (int i = 0; i < anArray.length; i++) {
            anArray[i] = i+2;
        }
    }
}
```

Συμβολοσειρές στη Java

Η συμβολοσειρά (String) στη Java είναι μία κλάση και όχι ένας απλός τύπος δεδομένων. Η κλάση της συμβολοσειράς απεικονίζει μία σειρά από χαρακτήρες. Όταν ορίσουμε μία νέα συμβολοσειρά και τις δώσουμε μία τιμή, δημιουργούμε ένα αντίγραφο (instance) της κλάσης. Η τιμή που θα δοθεί σε μία συμβολοσειρά δε μπορεί να αλλαχτεί κατά τη διάρκεια του προγράμματος. Μπορούμε να ορίσουμε και να αναθέσουμε μία τιμή σε μία συμβολοσειρά με τους ακόλουθους δύο ισοδύναμους τρόπους:

```
String str = "sentence";
```

ή

```
char data[] = {'s', 'e', 'n', 't', 'e', 'n', 'c', 'e'};
String str = new String(data);
```

Κάποιες μέθοδοι της κλάσης String παρουσιάζονται στο παρακάτω παράδειγμα:


```

String str1 = "Hello ";
String str2 = "World!";
System.out.println(str1 + str2);
String all=str1.concat(str2);
System.out.println(all);
int len=all.length();
System.out.println(len);
int num=all.lastIndexOf('l');
System.out.println(num);
String part = all.substring(2,num);
System.out.println(part);

```

Για να εκτελεστεί ο παραπάνω κώδικας θα πρέπει να ενσωματωθεί σε κάποια εφαρμογή ή σε ένα Applet. Κατά την εκτέλεση του γίνονται οι ακόλουθες ενέργειες:

Ορίζονται δύο στιγμιότυπα (instances) της κλάσης String με ονόματα str1 και str2, στα οποία δίνονται τιμές “Hello ” και “World!” αντίστοιχα. Οι δύο αυτές συμβολοσειρές ενώνονται σε μία συμβολοσειρά με τη βοήθεια του τελεστή ‘+’. Το αποτέλεσμα του (str1+str2) είναι η συμβολοσειρά “Hello World!” και τυπώνεται μέσω της εντολής System.out.println() σε ένα παράθυρο εξόδου της Java (Java Console). Η εντολή αυτή διευκολύνει τους προγραμματιστές στην παρακολούθηση και στην εύρεση λαθών κατά την εκτέλεση του προγράμματος.

Στη συνέχεια ορίζεται ένα νέο στιγμιότυπο της κλάσης String με όνομα all και τιμή το αποτέλεσμα της εντολής str1.concat(str2). Η μέθοδος concat χρησιμοποιείται για να προσαρτηθεί σε μία συμβολοσειρά μία άλλη συμβολοσειρά που δίνεται ως argument στην concat. Στο συγκεκριμένο παράδειγμα η str2 θα προσαρτηθεί στη str1 δηλαδή η str1.concat(str2) θα έχει αποτέλεσμα ίσο με (str1+str2).

Η μέθοδος length επιστρέφει το μήκος μιας συμβολοσειράς. Το all.length() επομένως έχει τιμή 12. Η μέθοδος lastIndexOf παίρνει ως όρισμα ένα χαρακτήρα και επιστρέφει την τελευταία θέση στην οποία εμφανίζεται αυτός ο χαρακτήρας στη συμβολοσειρά. Το all.lastIndexOf('l') έχει την τιμή 9 (η Java ξεκινάει την αρίθμηση από το 0) διότι η τελευταία φορά που εμφανίζεται το l στη συμβολοσειρά Hello World! είναι στη λέξη World.

Η substring(int i, int j) επιστρέφει ένα υποσύνολο της συμβολοσειράς, με πρώτο χαρακτήρα αυτόν που βρίσκεται στη θέση i της συμβολοσειράς και τελευταίο αυτόν που βρίσκεται στη θέση j-1. Επομένως το all.substring(2,9) θα έχει ως αποτέλεσμα “llo Wor” διότι το πρώτο “l” είναι στη θέση 2 της συμβολοσειράς all και το “r” στη θέση 8.

Κατά την εκτέλεση των εντολών στο παράθυρο εξόδου της Java (java console) θα τυπωθούν οι ακόλουθες εντολές.

```

Hello World!
Hello World!
12
9
llo Wor

```

Υπό συνθήκη εντολές στη Java

Η Java υποστηρίζει την υπό συνθήκη εντολή (conditional statement) if...else, της οποίας η σύνταξη φαίνεται στο ακόλουθο παράδειγμα:

```
String remark;
if (grade>=5) {
    remark="You passed the exams";
} else {
    remark="You have failed";
}
```

Οι εντολές κάτω από το else εκτελούνται όταν η συνθήκη (grade>5) είναι αληθής. Η Java υποστηρίζει, επίσης, την εντολή else if, η οποία επιτρέπει να προσαρτηθεί μία συνθήκη στη δήλωση else.

```
String remark;
if (grade>=8.5) {
    remark="Excellent";
} else if (grade>=6.5) && (grade<8.5){
    remark="Very Good";
} else if (grade>=5) && (grade<6.5){
    remark="Good";
} else {
    remark="Fail";
}
```

Τέλος στη Java υπάρχει και η εντολή select...case η οποία χρησιμοποιείται στην περίπτωση που πρέπει να ελεγχθούν πολλές συνθήκες για μία μεταβλητή.

```
int weekday = 4;
switch (weekday) {
    case 1:
        System.out.println("Sunday");
        break;
    case 2:
        System.out.println("Monday");
        break;
    case 3:
        System.out.println("Tuesday");
        break;
    case 4:
        System.out.println("Wednesday");
        break;
    case 5:
        System.out.println("Thursday");
        break;
    case 6:
        System.out.println("Friday");
        break;
    case 7:
        System.out.println("Saturday");
        break;
}
```

```
}

```

Με την εντολή `break` το πρόγραμμα αγνοεί τις επόμενες `case` δηλώσεις και βγαίνει εκτός της `select` εντολής. Στο παραπάνω παράδειγμα το `case 1`, `case 2` και `case 3` είναι `false`. Το `case 4` είναι `true` και εκτελούνται οι εντολές

```
System.out.println("Wednesday");
break;
```

Το `break` δείχνει πως δε θα εξεταστούν τα υπόλοιπα `case` και το πρόγραμμα μεταφέρεται αυτόματα στο τέλος της `select`.

Εντολές βρόχου στη Java

Η `java` υποστηρίζει τρεις εντολές βρόχου των οποίων η σύνταξη φαίνεται στον Πίνακα 4.5.3:

Πίνακας 4.5.3 – Εντολές βρόχου στη Java

<code>for</code>	<code>for (k=0; k++; k<=5) { //...εντολές... }</code>
<code>while</code>	<code>while (k<6) { //...εντολές και αλλαγή της τιμής του k... }</code>
<code>do...while</code>	<code>do { //...εντολές και αλλαγή της τιμής του k... } while (k<6)</code>

Η διαφορά των δύο τελευταίων εντολών είναι ότι στην πρώτη περίπτωση ο βρόχος μπορεί να μην εκτελεστεί καθόλου ενώ στη δεύτερη εκτελείται τουλάχιστον μία φορά.

Σχόλια στη Java

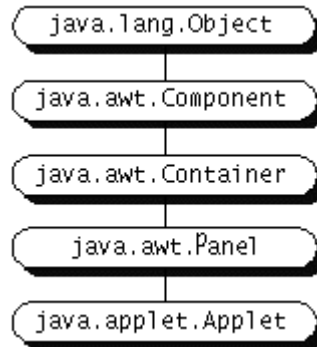
Η `Java` υποστηρίζει τα είδη σχολίων που παρουσιάζονται στον Πίνακα 4.5.4:

Πίνακας 4.5.4 – Σχόλια στη Java

Μορφή Σχολίου	Επεξήγηση
<code>/*...*/</code>	Σχόλια που ξεπερνούν τη μία γραμμή
<code>// ...</code>	Σχόλια που δε ξεπερνούν τη μία γραμμή
<code>/** ... */</code>	Σχόλια που τοποθετούνται στην επικεφαλίδα του προγράμματος και χρησιμοποιούνται κυρίως για λόγους τεκμηρίωσης.

4.5.4 Δημιουργία και εκτέλεση ενός Applet

Κάθε Applet είναι ένα αντικείμενο και δημιουργείται από μία ευρύτερη κλάση Applet από την οποία κληρονομεί τις ιδιότητες και τις μεθόδους του. Η κλάση Applet προκύπτει από άλλες κλάσεις των οποίων η ιεραρχία φαίνεται στην Εικόνα 4.5.1:



Εικόνα 4.5.1 - Ιεραρχία αντικειμένων στη Java. Πηγή Java Tutorial (<http://java.sun.com>)

Η ευρύτερη και πιο γενική κλάση της Java είναι η Object η οποία ανήκει στο πακέτο (package) java.lang . Η κλάση αυτή περιλαμβάνει τα γενικά χαρακτηριστικά όλων των αντικειμένων.

Για να δημιουργήσετε το πρώτο σας Applet θα πρέπει να έχετε εγκαταστήσει το JDK, το οποίο παρέχεται δωρεάν στο site της Sun: <http://java.sun.com>.

Για την ανάπτυξη και την εκτέλεση ενός Java Applet ακολουθούνται τα παρακάτω βήματα:

- Συγγραφή του κώδικα του Applet σε ένα αρχείο κειμένου (text file).
- Μεταγλώττιση του κώδικα του Applet με τη χρήση της εντολής javac.
- Συγγραφή του κώδικα μιας HTML σελίδας που καλεί το Applet.
- Κλήση της HTML σελίδας από ένα φυλλομετρητή.

Παράδειγμα ενός Applet

Το παράδειγμα που ακολουθεί παρουσιάζει ένα Applet το οποίο ζωγραφίζει την πρόταση "Your First Applet!", πάνω στην HTML σελίδα. Για να δείτε τα αποτελέσματα αυτού του applet, τοποθετήστε τον ακόλουθο κώδικα σε ένα αρχείο κειμένου και αποθηκεύστε το αρχείο με όνομα FirstApplet.java.

```
import java.applet.Applet;
import java.awt.Graphics;

public class FirstApplet extends Applet {

public void paint(Graphics MyGraphics) {
    MyGraphics.drawString("Your First Applet!", 50, 25);
}
}
```

Παρατηρούμε ότι στις δύο πρώτες γραμμές γίνεται κλήση των πακέτων (packages) της Java τα οποία θα χρησιμοποιήσουμε για την υλοποίηση του Applet. Το πρώτο πακέτο αφορά την κλάση Applet από την οποία θα δημιουργηθεί το “FirstApplet” και το δεύτερο αφορά την κλάση Graphics η οποία μας επιτρέπει να σχεδιάσουμε μέσα στο Applet. Τα προθέματα java.applet και java.awt ενημερώνουν τον μεταγλωττιστή για το ποια πακέτα περιλαμβάνουν τις κλάσεις. Και τα δύο αυτά πακέτα ανήκουν στο κεντρικό API της Java. Το πακέτο java.applet περιλαμβάνει όλες τις κλάσεις που είναι απαραίτητες για την υλοποίηση των applets, ενώ το java.awt περιέχει τις κλάσεις που χρησιμοποιούνται στο Abstract Window Toolkit (AWT). Το AWT παρέχει όλες τις κλάσεις που δημιουργούν το γραφικό περιβάλλον της Java (GUI).

```
import java.applet.Applet;  
import java.awt.Graphics;
```

Το Applet εκτελείται ακόμα και εάν παραλείψουμε τις δύο πρώτες γραμμές. Στην περίπτωση, όμως, αυτή ο κώδικας θα πρέπει να γραφτεί ως εξής:

```
public class FirstApplet extends java.applet.Applet {  
  
    public void paint(java.awt.Graphics MyGraphics) {  
        MyGraphics.drawString("Your First Applet!", 50, 25);  
    }  
}
```

Στην περίπτωση, δηλαδή, που δεν έχουμε καλέσει τις κλάσεις Applet και Graphics από την αρχή, θα πρέπει, στον κώδικα του Applet, να αναφερόμαστε σε αυτές χρησιμοποιώντας τα προθέματα που δείχνουν τα πακέτα στα οποία βρίσκονται.

Για να αναπτύξουμε ένα applet μπορεί να χρειαζόμαστε πολλές κλάσεις από ένα πακέτο της java. Στην περίπτωση αυτή, δεν είναι ανάγκη να καλούμε μία μία τις διαφορετικές κλάσεις, αλλά μπορούμε να καλέσουμε ολόκληρο το πακέτο, όπως φαίνεται στο ακόλουθο παράδειγμα:

```
import java.applet.*;  
import java.awt.*;
```

Μετά την κλήση των βιβλιοθηκών ακολουθεί ο ορισμός της κλάσης FirstApplet που θα δημιουργήσουμε. Η κλάση FirstApplet βασίζεται σε μία ευρύτερη κλάση Applet η οποία υπάρχει στο πακέτο java.applet της Java. Για να δηλώσουμε πως η κλάση FirstApplet παίρνει στοιχεία από την “Applet” γράφουμε “class FirstApplet extends java.applet.Applet”.

Παρατηρούμε, επίσης, ότι για τον ορισμό της FirstApplet χρησιμοποιήσαμε τη δήλωση “public”. Η δήλωση αυτή δείχνει το επίπεδο πρόσβασης στο Applet. Η Java υποστηρίζει τα επίπεδα πρόσβασης public, protected και private. Στη Java εξ’ ορισμού (δηλαδή εάν δεν προσδιοριστεί συγκεκριμένο επίπεδο πρόσβασης) όλες οι κλάσεις μπορούν να χρησιμοποιηθούν μόνο από κλάσεις του ίδιου πακέτου. Με τη δήλωση “public” η κλάση που ορίζουμε μπορεί να χρησιμοποιηθεί και από κλάσεις που

βρίσκονται σε διαφορετικά πακέτα. Ένα applet το ορίζουμε public, ώστε το περιβάλλον της java να μπορεί να το προσπελάσει και να εκτελέσει της μεθόδους του.

Η κλάση FirstApplet, που ορίσαμε, υλοποιεί μία μέθοδο paint. Η μέθοδος paint() κληρονομείται από την ευρύτερη κλάση "Applet" και χρησιμοποιείται για να ζωγραφίσει το περιεχόμενο του applet. Το applet σχεδιάζεται σ' ένα παράθυρο (Frame) ή μια συγκεκριμένη περιοχή του φυλλομετρητή. Όποτε η περιοχή αυτή καλύπτεται από κάποιο άλλο παράθυρο ή εξαφανίζεται, λόγω σμίκρυνσης του παραθύρου, η περιοχή του applet γίνεται ανενεργή. Όταν η περιοχή αυτή επανεμφανιστεί το applet θα ξανασχεδιαστεί με τη βοήθεια της εντολής repaint(). Η repaint() είναι μία μέθοδος εσωτερική στο Applet η οποία όταν καλείται χρησιμοποιεί και εκτελεί τη μέθοδο paint().

Ο γραφικός χώρος που δεσμεύει ο φυλλομετρητής για την παρουσίαση ενός applet αντιπροσωπεύεται από μία κλάση Graphics. Στο πρόγραμμά μας ορίζουμε ένα αντικείμενο της κλάσης αυτής με όνομα MyGraphics. Αυτό το αντικείμενο καλείται από την paint(), η οποία το χρησιμοποιεί για να ζωγραφίσει το applet μέσα σ' αυτό. Η κλάση Graphics υλοποιεί πολλές μεθόδους σχεδιασμού όπως τις setColor(), drawLine(), drawRect(), fillRect(), drawOval(), fillOval() και πολλές άλλες. Παραδείγματα και επεξηγήσεις αυτών των μεθόδων δίνονται στο Παράρτημα IV. Στο applet που δημιουργήσαμε χρησιμοποιήσαμε τη μέθοδο drawString(). Η drawString() έχει 3 ορίσματα: τη συμβολοσειρά που θα ζωγραφιστεί και τις συντεταγμένες x και y της θέσης στην οποία θα ξεκινήσει να ζωγραφίζεται. Οι συντεταγμένες μετράνε από την πάνω αριστερή γωνία της περιοχής της ιστοσελίδας που δεσμεύεται για το applet.

Για να μπορέσει το applet να φορτωθεί και να τρέξει σε μία HTML σελίδα, θα πρέπει πρώτα να μεταγλωττιστεί σε bytecode. Θα πρέπει δηλαδή να δημιουργηθεί ένα αρχείο class. Η μεταγλώττιση του κώδικα java γίνεται με την εντολή javac η οποία βρίσκεται μέσα στον υποφάκελο bin του φακέλου που δημιουργεί η εγκατάσταση του JDK. Καλώντας την ακόλουθη εντολή από το παράθυρο εντολών (command prompt) του DOS, ο κώδικας μεταγλωττίζεται και εάν δε βρεθούν λάθη δημιουργείται το class αρχείο με όνομα ίδιο με αυτό του java αρχείου (δηλαδή FirstApplet.class).

```
javac FirstApplet.java
```

Σημείωση: Για να μπορέσει να εκτελεστεί η παραπάνω εντολή θα πρέπει το αρχείο FirstApplet.java να βρίσκεται στον ίδιο φάκελο (directory) με την εντολή javac (δηλαδή στο \jdk1.4\bin). Για να μπορούμε να τρέξουμε τις εντολές του JDK από οποιοδήποτε φάκελο θα πρέπει να προσθέσουμε το φάκελο bin του JDK στη variable path που βρίσκεται στις Environment variables του System Properties.

Κλήση και εκτέλεση ενός applet

Ένα applet μπορεί να κληθεί και να εκτελεστεί μέσα από ένα περιβάλλον φυλλομετρητή. Για να ενσωματώσουμε ένα applet σε μία σελίδα HTML χρησιμοποιούμε την ετικέτα <APPLET>, όπως φαίνεται στον ακόλουθο κώδικα HTML:

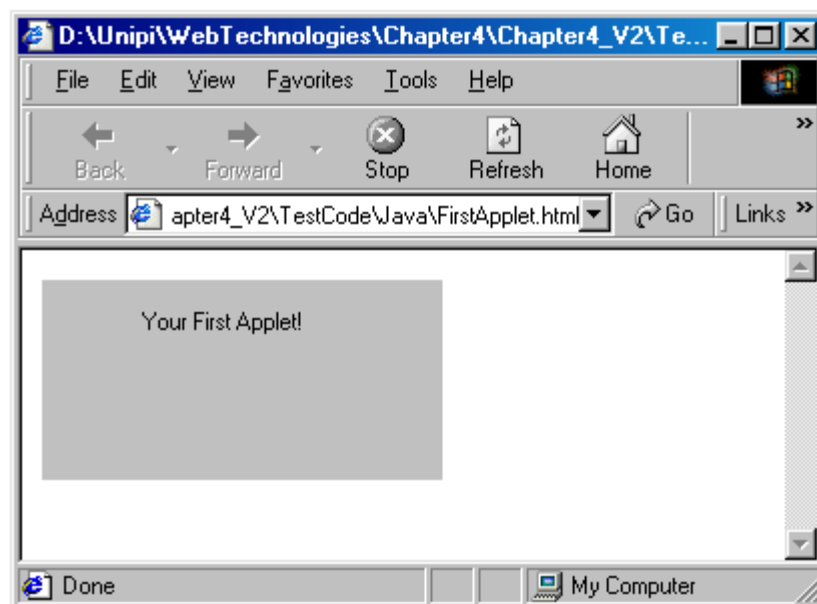
```
<HTML>
<BODY>
<APPLET CODE="FirstApplet.class" WIDTH=200 HEIGHT=100>
</APPLET>
</BODY>
</HTML>
```

Μέσα στην ετικέτα APPLET έχουμε τα χαρακτηριστικά CODE, WIDTH και HEIGHT. Στο πρώτο προσδιορίζουμε το όνομα του αρχείου που θα φορτωθεί και θα εκτελεστεί από το φυλλομετρητή. Στην περίπτωση που το αρχείο αυτό δε βρίσκεται στον ίδιο φάκελο με το HTML αρχείο, τότε μέσα στην ετικέτα APPLET θα πρέπει, επίσης, να προσδιορίσουμε το φάκελο στον οποίο βρίσκεται το αρχείο. Ο προσδιορισμός του φακέλου γίνεται σε ένα χαρακτηριστικό (attribute) CODEBASE. Εάν για παράδειγμα το αρχείο "First Applet.class" βρίσκεται στο φάκελο "C:\My Documents\MyApplets", τότε η ετικέτα APPLET θα περιλαμβάνει τα ακόλουθα:

```
<APPLET CODEBASE="C:\My Documents\My Applets"
CODE="FirstApplet.class" WIDTH=200 HEIGHT=100>
</APPLET>
```

Τα χαρακτηριστικά WIDTH και HEIGHT δείχνουν τον αριθμό των pixels της σελίδας που καταλαμβάνουν το πλάτος και το μήκος του Applet αντίστοιχα.

Όταν η HTML σελίδα ανοιχτεί από έναν φυλλομετρητή, το applet θα εμφανιστεί όπως δείχνει η Εικόνα 4.5.2.



Εικόνα 4.5.2 – Αποτέλεσμα εκτέλεσης Applet σε μία σελίδα HTML

Ένα applet μπορεί επίσης να εκτελεστεί μέσα από το παράθυρο εντολών του DOS με την εντολή “AppletViewer” η οποία ακολουθείται από το όνομα του αρχείου (AppletViewer FirstApplet.html). Η εκτέλεση του Applet μέσω του AppletViewer έχει το πλεονέκτημα ότι εμφανίζει τα λάθη που προκύπτουν στο παράθυρο εντολών (command prompt) του DOS. Εάν δε βρεθούν λάθη εμφανίζεται το παράθυρο της Εικόνας 4.5.3.



Εικόνα 4.5.3 – Αποτέλεσμα εκτέλεσης Applet στο AppletViewer

Για την κλήση του παραπάνω applet στην ετικέτα APPLET της HTML σελίδα; δόθηκαν τιμές στα χαρακτηριστικά CODE, WIDTH και HEIGHT. Εκτός, όμως από τα χαρακτηριστικά αυτά που είναι υποχρεωτικά, η ετικέτα APPLET περιλαμβάνει και τα ακόλουθα προαιρετικά χαρακτηριστικά:

```
<APPLET CODEBASE = codebaseURL
  ARCHIVE = archiveList
  CODE = appletFile ...or...
  ALT = alternateText
  NAME = appletInstanceName
  WIDTH = pixels
  HEIGHT = pixels
  ALIGN = alignment >
  <PARAM NAME = appletAttribute1 VALUE = value>
  <PARAM NAME = appletAttribute2 VALUE = value>
</APPLET>
```

Το χαρακτηριστικό ARCHIVE χρησιμοποιείται για την κλήση αρχείων τα οποία περιλαμβάνουν πληροφορίες και βιβλιοθήκες απαραίτητες για την εκτέλεση του applet. Το ALT χαρακτηριστικό περιλαμβάνει ένα κείμενο που εμφανίζεται σε περίπτωση που το applet δε μπορεί να εκτελεστεί από το φυλλομετρητή. Το ALIGN χρησιμοποιείται για να προσδιορίσει τη στοίχιση του applet ως προς τη σελίδα. Τέλος οι ετικέτα PARAM χρησιμοποιείται για να εισάγει τιμές σε παραμέτρους του applet. Το χαρακτηριστικό NAME προσδιορίζει το όνομα της παραμέτρου ενώ το VALUE προσδιορίζει την τιμή της.

4.5.5 Χρόνος ζωής ενός Applet

Όταν φορτώνεται ένα applet ακολουθούνται τα παρακάτω βήματα:

- Δημιουργείται ένα στιγμιότυπο (instance) της κλάσης Applet.

- Το applet αρχικοποιείται (παίρνει τις τιμές που χρειάζεται για να ξεκινήσει να τρέχει).
- Το applet ξεκινάει να τρέχει.

Από τη στιγμή που καλείται το applet και μέχρι να ξεκινήσει εμφανίζεται μήνυμα "initializing... starting...", το οποίο δηλώνει πως το applet αρχικοποιείται και ξεκινάει.

Όταν ο χρήστης εγκαταλείπει την ιστοσελίδα για να πάει σε μια άλλη σελίδα, το applet σταματάει να τρέχει. Όταν ο χρήστης επιστρέφει στη σελίδα το applet ξεκινάει πάλι. Το ίδιο συμβαίνει όταν ο χρήστης ελαχιστοποιήσει (minimizes) το παράθυρο του φυλλομετρητή και μετά το ανοίξει ξανά.

Επίσης όταν ο χρήστης κλείσει την εφαρμογή του φυλλομετρητή, το applet σταματάει να τρέχει και ελευθερώνει τους πόρους του μηχανήματος του χρήστη που είχε χρησιμοποιήσει.

Για να εκτελεστούν οι προαναφερόμενες ενέργειες καλούνται κάποιες μέθοδοι οι οποίες αποτελούν το σκελετό του applet.

Σκελετός ενός applet

Τα applet κληρονομούν από την κλάση Applet τις μεθόδους στις οποίες στηρίζονται οι βασικές λειτουργίες ενός applet. Αυτές οι μέθοδοι φαίνονται στο ακόλουθο παράδειγμα κώδικα:

```
import java.applet.*;
import java.awt.*;
public class AppletSkel extends Applet {

    public void init() {
        // initialization
    }
    public void start() {
        // start or resume execution
    }
    public void stop() {
        // suspends execution
    }
    public void paint (Graphics g) {
        // redisplay contents of window
    }
    public void destroy (Graphics g) {
        // perform shutdown activities
    }
}
```

Όταν ο φυλλομετρητής καλέσει ένα applet, το περιβάλλον της Java δημιουργεί ένα στιγμιότυπο (instance) της κλάσης Applet και καλεί την μέθοδο init(). Η μέθοδος αυτή χρησιμοποιείται για να αρχικοποιήσει κάποιες παραμέτρους του applet. Σε αυτή τη μέθοδο τοποθετείται κώδικας ο οποίος εκτελείται μόνο μία φορά κατά τη διάρκεια ζωής του applet.

Αφού αρχικοποιηθεί ένα applet , το περιβάλλον της Java καλεί τη μέθοδο start() για να ξεκινήσει η εκτέλεση του applet. Όταν ο χρήστης αλλάξει, μικρύνει ή κρύψει την ιστοσελίδα εκτελείται αυτόματα η μέθοδος stop() η οποία αναστέλλει τη λειτουργία του applet. Η start() καλείται ξανά όταν επανεμφανιστεί το παράθυρο του applet.

Το περιβάλλον της Java καλεί την paint() κάθε φορά που θεωρεί πως το παράθυρο του applet έχει κρυφτεί. Επομένως, η paint() εκτελείται πολύ περισσότερες φορές από αυτές που θεωρεί ο χρήστης.

Τέλος η μέθοδος destroy εκτελείται μόνο όταν ο χρήστης κλείσει το φυλλομετρητή. Όσο ο φυλλομετρητής παραμένει ανοιχτός το applet διατηρείται σε μία σταματημένη μορφή. Λίγο πριν κλείσει ο φυλλομετρητής καλείται η destroy() και οι πόροι που χρησιμοποιούσε το applet απελευθερώνονται

4.5.6 Παραδείγματα Applet

Παράδειγμα Applet που δέχεται παραμέτρους από τη σελίδα HTML

Στο παράδειγμα αυτό δημιουργείται ένα applet το οποίο παίρνει τιμή για κάποια παράμετρο από την σελίδα HTML. Για να μπορούμε να δώσουμε τιμές σε παραμέτρους μέσα από τη σελίδα HTML χρησιμοποιούμε μία ετικέτα <PARAM > με χαρακτηριστικά NAME (το όνομα της παραμέτρου) και VALUE (την τιμή της παραμέτρου).

Στον κώδικα HTML που ακολουθεί καλείται ένα applet με όνομα WelcomeApplet και του δίνεται μία παράμετρος με όνομα WelcomeParam και τιμή "Welcome to".

```
<HTML>
<BODY>
WelcomeApplet will appear below in a Java enabled browser.
<BR>
< APPLET
    CODEBASE = "classes"
    CODE = "WelcomeApplet.class"
    WIDTH = 400
    HEIGHT = 100 >
    <PARAM NAME = WelcomeParam VALUE = "Welcome to">
</APPLET>
</BODY>
</HTML>
```

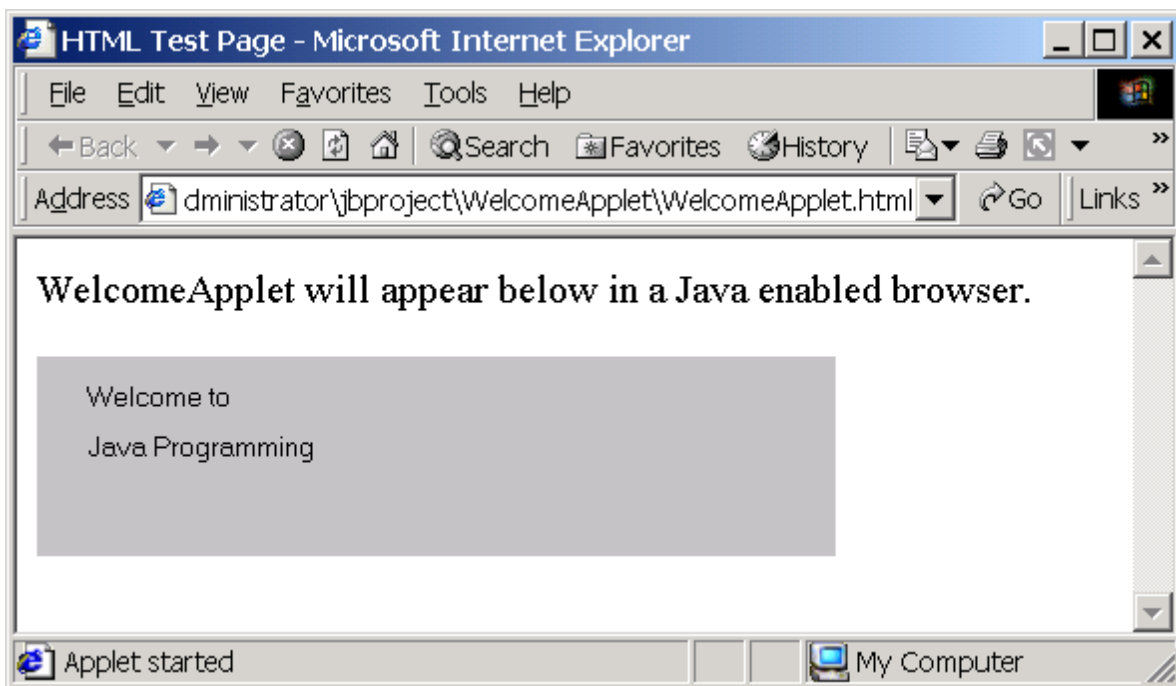
Όπως φαίνεται στον κώδικα που ακολουθεί το applet παίρνει την τιμή της παραμέτρου χρησιμοποιώντας την εντολή getparameter με όρισμα το όνομα της παραμέτρου (WelcomeParam) που δόθηκε από τη σελίδα HTML. Η τιμή της

getParameter("WelcomeParam") αποθηκεύεται σε μία μεταβλητή για να μπορέσει να χρησιμοποιηθεί από τις υπόλοιπες μεθόδους του applet.

```
import java.applet.Applet;
import java.awt.Graphics;
public class WelcomeApplet extends Applet {
    String WelcomeString;

    public void init() {
        WelcomeString=" ";
    }
    public void start() {
        WelcomeString=getParameter("WelcomeParam");
    }
    public void paint(Graphics g) {
        g.drawString(WelcomeString, 25, 25);
        g.drawString("Java Programming", 25, 40);
    }
}
```

Όταν το applet φορτωθεί στο φυλλομετρητή εμφανίζεται η Εικόνα 4.5.4.



Εικόνα 4.5.4 – Applet που δέχεται παραμέτρους από τη σελίδα HTML

Παράδειγμα Applet που περιλαμβάνει σχήματα και χρώματα

Στη Java με τη βοήθεια της κλάσης Graphics μπορούμε, επίσης, να σχεδιάσουμε και να χρωματίσουμε διάφορα σχήματα χρησιμοποιώντας έτοιμες εντολές. Στο

παράδειγμα που ακολουθεί δημιουργείται ένα applet στο οποίο εμφανίζεται ένα πράσινο ορθογώνιο, μία κόκκινη έλλειψη, έναν μπλε κύκλο και έναν κόκκινο ρόμβο.

```
import java.applet.Applet;
import java.awt.Graphics;
import java.awt.Color;

public class ShapeDemo extends Applet {
public void paint(Graphics MyGraphics) {

    int xCoords[]={150,100, 150, 200};
    int yCoords[]={50, 100, 150, 100};

    MyGraphics.setColor(Color.blue);
    MyGraphics.drawString("Your First Applet using Colors and
Shapes!", 50, 25);
    MyGraphics.setColor(Color.green);
    MyGraphics.drawRect(50,50, 200,100);
    MyGraphics.setColor(Color.red);
    MyGraphics.drawOval(50,50, 200,100);
    MyGraphics.setColor(Color.blue);
    MyGraphics.fillOval(100,50, 100,100);
    MyGraphics.setColor(Color.red);
    MyGraphics.fillPolygon(xCoords, yCoords, 4);
}
}
```

Στην Java τα χρώματα αλλάζουν με τη μέθοδο `setColor` της κλάσης `Graphics`. Όλα τα χρώματα ορίζονται ως χαρακτηριστικά (attributes) της κλάσης `Color`. Άρα για να ορίσουμε ένα χρώμα στη `Graphics` γράφουμε `MyGraphics.setColor(Color.red)`, όπου “`MyGraphics`” το όνομα του αντικειμένου που προέρχεται από την `Graphics`.

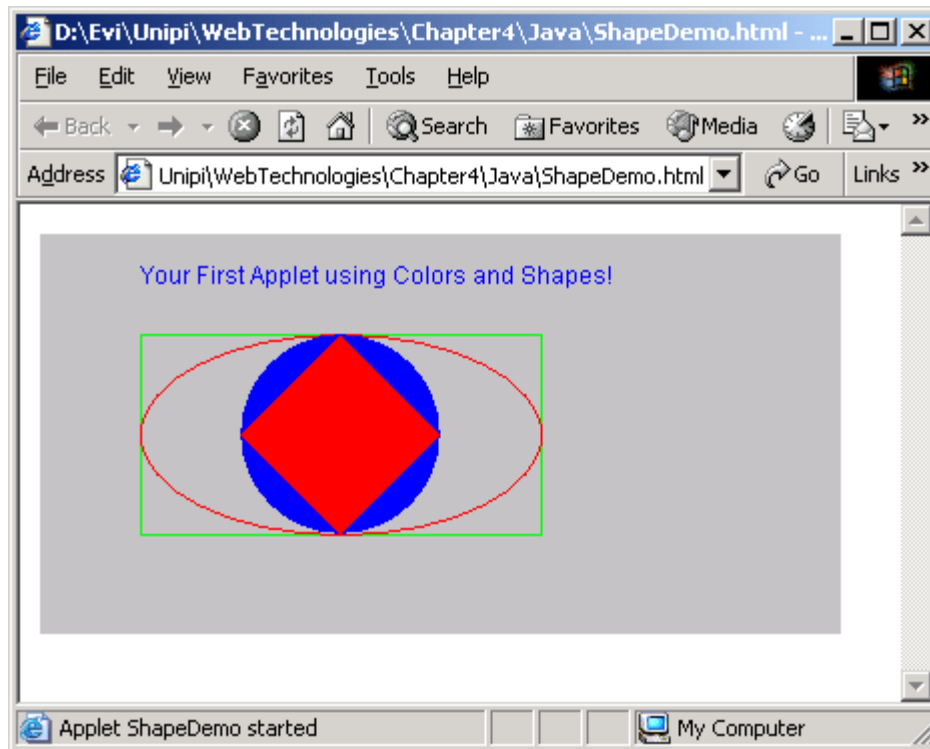
Στο παραπάνω παράδειγμα αλλάζουμε το χρώμα κάθε φορά που θέλουμε να ζωγραφίσουμε ένα καινούριο σχήμα. Αρχικά σχεδιάζουμε ένα ορθογώνιο με την εντολή `drawRect(int x, int y, int width, int height)`. Η εντολή αυτή παίρνει 4 ορίσματα τύπου ακέραιο (`integer`). Τα δύο πρώτα δηλώνουν τη θέση της πάνω αριστερής γωνίας του σχήματος, δηλαδή τον αριθμό των pixels από το αριστερό και από το πάνω όριο του παραθύρου του applet αντίστοιχα. Τα δύο επόμενα ορίσματα δηλώνουν το πλάτος και το ύψος του σχήματος σε αριθμό pixels. Η `drawRect` σχεδιάζει το περίγραμμα του ορθογώνιου. Το ορθογώνιο θα μπορούσε, επίσης, να σχεδιαστεί χρωματισμένο με την εντολή `FillRect`.

Η `drawOval(50,50,200,100)` σχεδιάζει μία έλλειψη που βρίσκεται εγγεγραμμένη σε ένα ορθογώνιο που σχεδιάστηκε με τη `drawRect` με τα ίδια ορίσματα. Επομένως η έλλειψη που θα σχεδιαστεί θα βρίσκεται μέσα στο ορθογώνιο που δημιουργήθηκε από την `drawRect(50,50,200,100)`.

Η `drawOval` μπορεί, επίσης, να σχεδιάσει έναν κύκλο όταν τα ορίσματα του πλάτους και του ύψους είναι ίδια. Στο παράδειγμα αυτό χρησιμοποιήθηκε η `fillOval` για να σχεδιαστεί ένας χρωματισμένος κύκλος. Οι συντεταγμένες του κύκλου υπολογίστηκαν με τέτοιο τρόπο ώστε ο κύκλος να τοποθετηθεί στο κέντρο της έλλειψης.

Τέλος χρησιμοποιήθηκε η `fillPolygon` για να σχεδιάσει ένα πολύγωνο. Η `fillPolygon` έχει ως ορίσματα ένα `array` με τις τετμημένες των γωνιών του (θέση των γωνιών ως προς τον άξονα x), ένα `array` με τις τεταγμένες των γωνιών του (θέση των γωνιών ως προς τον άξονα y) και ένα νούμερο που δείχνει τον αριθμό των πλευρών του πολυγώνου. Για να σχεδιάσουμε το περίγραμμα ενός πολυγώνου χρησιμοποιούμε κατά αναλογία την `drawPolygon`.

Όταν μεταγλωττιστεί και εκτελεστεί το applet εμφανίζεται η Εικόνα 4.5.5.



Εικόνα 4.5.5 – Applet που εμφανίζει σχήματα και χρώματα.

Παράδειγμα Applet που επιτρέπει αλληλεπίδραση με το χρήστη.

Το παράδειγμα αυτό εμφανίζει ένα applet που επιτρέπει αλληλεπίδραση με το χρήστη. Στο applet εμφανίζεται ένα πεδίο κειμένου (`TextField`) με όνομα “`SentenceField`”, δύο πλήκτρα (`Button`) με ονόματα “`AddButton`” και “`EraseButton`” και μία λίστα επιλογής κειμένου (`Choice`).

Ο χρήστης μπορεί να εισάγει στο πεδίο κειμένου μία λέξη ή μία πρόταση. Με την επιλογή του πλήκτρου “`AddButton`” η πρόταση προστίθεται στη λίστα επιλογής κειμένου, ενώ με την επιλογή του “`EraseButton`” όλες οι λέξεις ή οι προτάσεις διαγράφονται από τη λίστα. Οι προαναφερόμενες λειτουργίες επιτυγχάνονται με τον κώδικα που ακολουθεί:

```
import java.applet.Applet;
import java.awt.Panel;
import java.awt.GridLayout;
import java.awt.Label;
import java.awt.Button;
```

```

import java.awt.Font;
import java.awt.TextField;
import java.awt.Choice;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

public class AddSentence extends Applet {
Panel MyPanel;
Font labelFont;
Label SentenceLabel;
Button AddButton;
Button EraseButton;
TextField SentenceField;
Choice SentenceChoice;

public void init() {
SentenceLabel=new Label("Write a sentence");
labelFont = new Font ("Helvetica", Font.BOLD, 14);
AddButton = new Button("Add sentence");
AddButton.setBackground(java.awt.Color.green);
EraseButton= new Button("Erase all");
EraseButton.setBackground(java.awt.Color.red);
SentenceField=new TextField();
SentenceChoice=new Choice();
add("Center", MyPanel=new Panel());
MyPanel.setLayout(new GridLayout(3,2));
MyPanel.add(SentenceLabel).setFont(labelFont);
MyPanel.add(SentenceField);
MyPanel.add(EraseButton);
MyPanel.add(AddButton);
MyPanel.add(new Label("All sentences"));
MyPanel.add(SentenceChoice);

AddButton.addActionListener(new ActionListener() {
public void actionPerformed (ActionEvent e) {
if (SentenceField.getText().trim().length()>=1)
SentenceChoice.addItem(SentenceField.getText().trim());
SentenceField.setText("");
}
});
EraseButton.addActionListener(new ActionListener() {
public void actionPerformed (ActionEvent e) {
SentenceChoice.removeAll();
}
});
}
}

```

Όλα τα αντικείμενα που χρησιμοποιούνται στο applet ανήκουν σε βιβλιοθήκες της Java οι οποίες εισάγονται με την εντολή import. Τα αντικείμενα πλήκτρου (Button),

ετικέτας (Label), γραμματοσειράς (Font), πεδίου κειμένου (TextField) και επιλογής κειμένου (Choice) ανήκουν στο AWT της Java και μπορούν να κληθούν είτε ένα ένα :

```
import java.awt.Label;  
import java.awt.Button;  
import java.awt.Font;  
import java.awt.TextField;  
import java.awt.Choice;
```

είτε όλα μαζί με την ακόλουθη εντολή:

```
import java.awt.*
```

Όλα τα αντικείμενα τοποθετούνται και ζωγραφίζονται πάνω σε ένα πλαίσιο σχεδιασμού (Panel), με όνομα myPanel, το οποίο ανήκει, επίσης, στο AWT της Java.

```
Panel MyPanel;
```

Στο Panel αυτό θα τοποθετηθούν 6 αντικείμενα (2 Labels, 2 Buttons, 1 TextField και 1 Choice). Αρχικά ορίζουμε και δίνουμε κάποιες τιμές στα αντικείμενα αυτά με τις ακόλουθες εντολές:

```
SentenceLabel=new Label("Write a sentence");  
labelFont = new Font ("Helvetica", Font.BOLD, 14);  
AddButton = new Button("Add sentence");  
AddButton.setBackground( java.awt.Color.green);  
EraseButton= new Button("Erase all");  
EraseButton.setBackground( java.awt.Color.red);  
SentenceField=new TextField();  
SentenceChoice=new Choice();
```

Για να εμφανίσουμε τα αντικείμενα στοιχισμένα και όχι τυχαία το ένα πίσω από το άλλο, προσθέτουμε στο Panel ένα πλέγμα διάταξης GridLayout, με την εντολή setLayout:

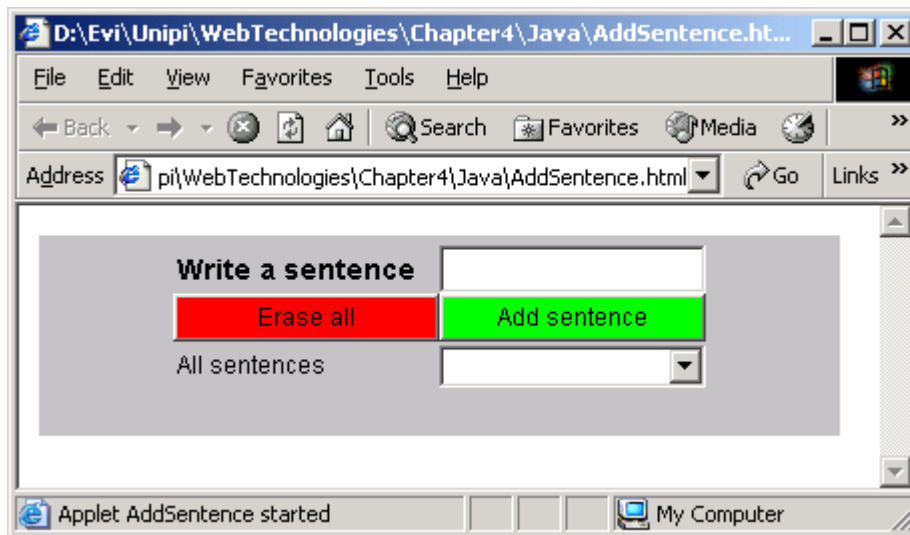
```
MyPanel.setLayout(new GridLayout(3,2));
```

Η new GridLayout(3,2) δημιουργεί ένα πλέγμα διάταξης με διαστάσεις (3,2), δηλαδή με 3 γραμμές και 2 στήλες. Κάθε φορά, λοιπόν, που ένα νέο αντικείμενο προστίθεται στο πλέγμα, τοποθετείται στην επόμενη κενή θέση του πλέγματος.

Τα αντικείμενα τοποθετούνται στο πλέγμα διάταξης με τις ακόλουθες εντολές:

```
MyPanel.add(SentenceLabel).setFont(labelFont);  
MyPanel.add(SentenceField);  
MyPanel.add(EraseButton);  
MyPanel.add(AddButton);  
MyPanel.add(new Label("All sentences"));  
MyPanel.add(SentenceChoice);
```

Με την τοποθέτηση των αντικειμένων στο πλέγμα διάταξης επιτυγχάνεται το αποτέλεσμα τη Εικόνας 4.5.6.



Εικόνα 4.5.6 – Applet που επιτρέπει αλληλεπίδραση με το χρήστη.

Η δυνατότητα των αντικειμένων να αλληλεπιδρούν με το χρήστη δίνεται από μία κλάση ActionListener. Ένα αντικείμενο τύπου ActionListener προστίθεται στα πλήκτρα AddButton και EraseButton με τον ακόλουθο κώδικα:

```
AddButton.addActionListener(new ActionListener() {
    public void actionPerformed (ActionEvent e) {
        if (SentenceField.getText().trim().length()>=1)
            SentenceChoice.addItem(SentenceField.getText().trim());
        SentenceField.setText("");
    }
});
EraseButton.addActionListener(new ActionListener() {
    public void actionPerformed (ActionEvent e) {
        SentenceChoice.removeAll();
    }
});
```

Το αντικείμενο ActionListener ενεργοποιείται όταν συμβεί κάποιο γεγονός πάνω στο συγκεκριμένο αντικείμενο.

Όταν για παράδειγμα ο χρήστης επιλέξει το πλήκτρο AddButton, η ActionListener ενεργοποιείται και εκτελεί τις ακόλουθες ενέργειες:

Ελέγχει εάν η πρόταση δεν είναι κενή. Για να το κάνει αυτό σβήνει τα κενά (SentenceField.getText().trim()) και ελέγχει εάν το μήκος του κειμένου χωρίς κενά είναι μεγαλύτερο ή ίσο του ένα (SentenceField.getText().trim()>=1).

Εάν η πρόταση δεν είναι κενή προστίθεται στη λίστα SentenceChoice με την εντολή (SentenceChoice.addItem(SentenceField.getText().trim())) και το κείμενο σβήνεται από το πεδίο κειμένου (SentenceField.setText("")).

Όταν ο χρήστης επιλέξει το πλήκτρο `EraseButton` εκτελείται η εντολή `(SentenceChoice.removeAll())` η οποία σβήνει τα περιεχόμενα της λίστας.

4.6 DHMTL

Η απλή HTML επιτρέπει τη δημιουργία στατικών σελίδων, των οποίων το περιεχόμενο μπορεί να ξανασημασιάζεται ή να αλλάζει μόνο μέσω της εντολής ανανέωσης (Refresh) του φυλλομετρητή. Όπως προαναφέρθηκε στις προηγούμενες ενότητες, η προσθήκη κίνησης ή η δυναμική αλλαγή περιεχομένου στις σελίδες HTML, επιτυγχάνεται με την ενσωμάτωση γλωσσών σεναρίων ή με την προσθήκη αυτόνομων προγραμμάτων όπως Java Applets και ActiveX Controls.

Από την έκδοση 4 και πάνω οι φυλλομετρητές υποστηρίζουν την **Dynamic HTML (DHTML)** η οποία αντιστοιχεί σε μία πιο δυναμική μορφή της HTML. Με τη χρήση της DHTML μπορούν να δημιουργηθούν δυναμικές ιστοσελίδες, δηλαδή σελίδες που κινούνται, αλλάζουν μορφή ή αλληλεπιδρούν με τον χρήστη, αφού φορτωθούν στο φυλλομετρητή. Για παράδειγμα, μία παράγραφος θα μπορούσε να γίνει μπλε όταν ο δείκτης του ποντικιού κινείται πάνω της, ή μια επικεφαλίδα θα μπορούσε να κυλήσει επί της οθόνης. Επομένως, οι χρήστες μπορούν να έχουν μια περισσότερο ευχάριστη και αλληλεπιδραστική εμπειρία με τον Ιστό, χωρίς να απαιτούνται σταθερές κλήσεις προς έναν εξυπηρετητή ή φόρτωση νέων σελίδων ή μεγάλων προγραμμάτων (applets).

Η DHTML δεν είναι ένα πρότυπο ορισμένο από το World Wide Web Consortium (W3C) αλλά ένας όρος που χρησιμοποιείται από τη Microsoft και τη Netscape για να περιγράψουν ένα σύνολο από τεχνολογίες, οι οποίες συμβάλλουν στην ανάπτυξη δυναμικών ιστοσελίδων. Η DHTML συνδυάζει τις ακόλουθες τεχνολογίες:

- HTML 4.0
- JavaScript
- Cascading Style Sheets (CSS): καθορίζουν τα χαρακτηριστικά και τη μορφή των στοιχείων και των αντικειμένων της σελίδας HTML
- Document Object Model (DOM): είναι το μέσο που επιτρέπει την προσπέλαση και διαχείριση των αντικείμενων της σελίδας HTML.

Παρόλο που οι σελίδες HTML, που χρησιμοποιούν έστω και μία από αυτές τις τεχνολογίες, θεωρούνται «δυναμικές», ο όρος DHTML γενικά αναφέρεται στην χρήση όλων μαζί των παραπάνω τεχνολογιών.

Εκτιμώντας τη δυσκολία όλων των τεχνικών σχεδιασμού ιστοσελίδων, η DHTML βρίσκεται κάπου στη μέση. Δεν είναι τόσο δύσκολη όσο η Java ή ο προγραμματισμός σε Perl, αλλά είναι πιο δύσκολη από την απλή HTML. Υπάρχουν, όμως, οπτικά εργαλεία ανάπτυξης DHTML, όπως τα Dreamweaver και Fusion, που υποστηρίζουν και διευκολύνουν την υλοποίηση ιστοσελίδων DHTML.

4.6.1 Πλεονεκτήματα χρήσης της DHTML

Τα πλεονεκτήματα χρήσης της DHTML είναι τα ακόλουθα:

- *Μικρό μέγεθος αρχείων:* Τα αρχεία DHTML είναι πιο μικρά σε σύγκριση με τα αρχεία που δημιουργούνται από άλλα αλληλεπιδραστικά μέσα, όπως το Flash ή το Shockwave. Για το λόγο αυτό απαιτούν μικρότερο χρόνο φόρτωσης και καταλαμβάνουν λιγότερο εύρος ζώνης.

- Η DHTML στηρίζεται σε πρότυπα (standards) που ορίστηκαν από το World Wide Web Consortium (W3C)
- Δεν απαιτείται η εγκατάσταση πρόσθετου λογισμικού: Για την εμφάνιση μιας σελίδας DHTML δεν απαιτούνται επιπρόσθετα εργαλεία ή προγράμματα. Η DHTML υποστηρίζεται από τους φυλλομετρητές Netscape και Internet Explorer έκδοση 4 και νεώτερες και για να εμφανιστεί μία DHTML σελίδα απαιτείται μόνο ένας κατάλληλος φυλλομετρητής. Ο χρήστης δεν χρειάζεται να κατεβάσει κάποιο εργαλείο ή αρχείο για να δει τη σελίδα.
- Δεν απαιτείται μια Java Virtual Machine (JVM). Η DHTML δε στηρίζεται σε τεχνολογία Java. Η DHTML παρέχει πολλές λειτουργίες που θα μπορούσαν να επιτευχθούν μέσω της Java. Οι σελίδες, που περιέχουν προγράμματα σε Java (applets) χρησιμοποιούν μεγάλο εύρος ζώνης και απαιτούν χρόνο ώστε να ξεκινήσει η JVM και να φορτωθεί ο κώδικας Java Παρ' όλο που η Java ενδείκνυται για κάποιες εφαρμογές, η DHTML μπορεί να αποτελέσει μια ελκυστική εναλλακτική λύση για δημιουργία δυναμικών ιστοσελίδων.

4.6.2 Μειονεκτήματα

Οι εφαρμογές DHTML έχουν τα ακόλουθα μειονεκτήματα.

- Η DHTML δεν υποστηρίζεται από παλαιότερες εκδόσεις φυλλομετρητών. Η DHTML υποστηρίζεται μόνο από τις εκδόσεις 4.0 και νεώτερες των Netscape και Internet Explorer. Υπάρχουν όμως κάποιοι χρήστες παλαιότερων εκδόσεων, οι οποίοι δεν μπορούν να δουν εφαρμογές DHTML.
- Παρουσιάζονται διαφορές και ασυμβατότητες ως προς την υποστήριξη της DHTML από τη Microsoft και τη Netscape. Ακόμα και οι πιο πρόσφατες εκδόσεις φυλλομετρητών δεν υποστηρίζουν την ίδια σύνταξη και τις ίδιες εντολές της DHTML.
- Η DHTML προϋποθέτει καλή γνώση τεχνολογιών όπως HTML, JavaScript, CSS, και DOM και απαιτείται προσοχή στο σχεδιασμό και στον τρόπο που συνδυάζονται οι τεχνολογίες αυτές. Υπάρχουν, όμως, και εργαλεία DHTML που προσπαθούν να εξαλείψουν αυτό το πρόβλημα.
- Ο πηγαίος κώδικας δεν είναι προστατευμένος. Είναι ενσωματωμένος σε αυτόν της σελίδας HTML και εμφανίζεται στο χρήστη όταν αυτός επιλέξει 'View/Source' από την μπάρα εργαλείων του φυλλομετρητή του.

4.6.3 Διαφορές φυλλομετρητών

Προκειμένου να δημιουργήσουμε περιεχόμενο DHTML διαθέσιμο και προσβάσιμο στο μεγαλύτερο μέρος του κοινού, απαιτείται η κατανόηση των διαφορών μεταξύ της Microsoft και της Netscape, ως προς την υποστήριξη της DHTML. Όπως φαίνεται στον Πίνακα 4.6.1 οι φυλλομετρητές Netscape Navigator και Internet Explorer δεν υποστηρίζουν τις ίδιες τεχνολογίες, πράγμα που δυσχεραίνει την ανάπτυξη ιστοσελίδων σε DHTML.

Πίνακας 4.6.1 – Διαφορές φυλλομετρητών στις Τεχνολογίες που υποστηρίζουν και στην υλοποίηση της DHTML. Πηγή http://www.w3schools.com/dhtml/dhtml_intro.asp.

Netscape Navigator 4.0 (NN4)	DHTML που υποστηρίζεται από IE4 και NN4	Internet Explorer 4.0 (IE4)
<ul style="list-style-type: none"> ▪ JSS (JavaScript Style Sheets) (επιτρέπει τον έλεγχο της εμφάνισης διαφορετικών στοιχείων της HTML) ▪ Layers (επιτρέπει τον έλεγχο της θέσης και της ορατότητας των στοιχείων της HTML) 	<ul style="list-style-type: none"> ▪ CSS1 ▪ CSS2 (επιτρέπει τον έλεγχο της εμφάνισης διαφορετικών στοιχείων της HTML) ▪ CSS-P (Positioning) (επιτρέπει τον έλεγχο της θέσης ενός αντικειμένου) ▪ JavaScript 	<ul style="list-style-type: none"> ▪ Visual Filters (επιτρέπει την εφαρμογή οπτικών εφέ σε κείμενο και γραφικά). ▪ Dynamic CSS (επιτρέπει τον έλεγχο της θέσης και της ορατότητας των στοιχείων σε μία σελίδα HTML).

Η έκδοση 4.0 και των δύο υποστηρίζει τα πρότυπα CSS1, CSS2 και CSS-P (Cascading Style Sheet Positioning) που αναπτύχθηκαν από το W3C. Τα CSS1 και CSS2 καθορίζουν τη μορφή και την εμφάνιση των αντικειμένων σε ένα έγγραφο HTML. Ενώ το CSS-P καθορίζει τη θέση των αντικειμένων στη σελίδα. Με βάση τα χαρακτηριστικά που προσδιορίζονται από το CSS-P, τα αντικείμενα της σελίδας μπορούν να κρυφτούν, να εμφανισθούν και να τοποθετηθούν σε σειρά.

Παρόλο που και οι δύο φυλλομετρητές χρησιμοποιούν το CSS-P, η Microsoft και η Netscape το έχουν εφαρμόσει με διαφορετικό τρόπο στα αντικείμενα HTML. Τα δυναμικά αντικείμενα στον Internet Explorer αναφέρονται ως *styles*. Τα δυναμικά αντικείμενα σε έναν φυλλομετρητή Netscape έκδοσης 4 και πάνω υλοποιούνται με τη βοήθεια των στρωμάτων ή επιπέδων (*layers*). Η Netscape έχει δημιουργήσει τις ετικέτες <LAYER> και <ILAYER> για να προσδιορίσει τη διδιάστατη και τρισδιάστατη θέση των στοιχείων και αντικειμένων σε ένα έγγραφο HTML. Αυτές, όμως, οι ετικέτες απορρίφθηκαν από το W3C και έτσι η Netscape υποστήριξε και το πρότυπο CSS-P. Τα αντικείμενα που τοποθετούνται στη σελίδα, είτε με τη χρήση του προτύπου CSS-P είτε με τη χρήση Layers, μπορούν να προσπελαστούν, κατά τον ίδιο τρόπο, με τη βοήθεια της JavaScript.

Στην ενότητα αυτή θα δοθούν παραδείγματα προσδιορισμού της εμφάνισης στοιχείων ιστοσελίδων με τη χρήση του προτύπου CSS. Θα περιγραφεί το CSS-P και τέλος θα γίνει μία σύντομη αναφορά στο Document Object Model της DHTML. Ενώ, όπως προαναφέρθηκε, υπάρχουν αρκετές διαφορές στην υλοποίηση της DHTML από τους δύο επικρατέστερους φυλλομετρητές, στην ενότητα αυτή θα δοθεί έμφαση σε πρότυπα και στοιχεία της DHTML που υποστηρίζονται και από τους δύο φυλλομετρητές.

4.6.4 Cascading Style Sheets (CSS)

CSS είναι τα αρχικά του Cascading Style Sheets, τα οποία αναπτύχθηκαν στην HTML 4.0 για να ορίσουν την εμφάνιση και τη μορφή των στοιχείων ενός εγγράφου HTML. Αρχικά οι ετικέτες της HTML είχαν αναπτυχθεί για να ορίσουν το περιεχόμενο

μιας ιστοσελίδας. Σιγά σιγά, όμως, νέες ετικέτες και στοιχεία άρχισαν να προστίθενται στους δύο επικρατέστερους φυλλομετρητές (Netscape και Internet Explorer), για να βελτιώσουν την εμφάνιση των σελίδων. Έτσι, εκτός από ετικέτες, όπως οι , και <I> προστέθηκαν και ιδιότητες, όπως η color. Όμως, ο προσδιορισμός της εμφάνισης των στοιχείων κάθε φορά που αυτά δηλώνονται στον κώδικα της ιστοσελίδας, είχε ως αποτέλεσμα την αύξηση της πολυπλοκότητας στην ανάπτυξη και τη δημιουργία των σελίδων HTML. Για το λόγο αυτό η μη κερδοσκοπική διεθνής εταιρική συνεργασία του Παγκοσμίου Ιστού (World Wide Web Consortium), ανέπτυξε το πρότυπο του CSS μαζί με την HTML 4.

Τα στυλ (styles) στην HTML 4 ορίζουν τον τρόπο εμφάνισης των στοιχείων της σελίδας. Με τη χρήση των CSS το περιεχόμενο της σελίδας διαχωρίζεται από τον τρόπο εμφάνισής του. Οι προγραμματιστές μπορούν να ορίσουν μία μόνο φορά την εμφάνιση των στοιχείων και όλα τα στοιχεία του ίδιου τύπου εμφανίζονται με τον ίδιο τρόπο στη σελίδα. Τα στυλ που προσδιορίζονται μπορούν, επίσης, να αποθηκευθούν σε ένα εξωτερικό αρχείο με κατάληξη .css. Το αρχείο αυτό καλείται από τη σελίδα HTML και οι κανόνες μορφοποίησης που περιέχει εφαρμόζονται σε όλα τα στοιχεία της σελίδας. Σε μία σελίδα HTML μπορούμε να καλέσουμε περισσότερα του ενός εξωτερικά αρχεία.

Η δυνατότητα αυτή των CSS λύνει ένα από τα βασικότερα προβλήματα υλοποίησης δικτυακού τόπου, που σχετίζεται με την ομοιόμορφη παρουσίαση όλων των σελίδων που ανήκουν σε αυτόν. Υλοποιώντας εξωτερικά αρχεία μορφοποίησης (style sheets) και καλώντας τα από τις σελίδες HTML, οι προγραμματιστές επιτυγχάνουν την ομοιόμορφη παρουσίαση των στοιχείων όλων των σελίδων. Επομένως, ο έλεγχος και η αλλαγή της εμφάνισης πολλών ιστοσελίδων, μπορεί να γίνει αυτόματα μέσω της ενημέρωσης ενός μόνο αρχείου.

Το πρότυπο CSS, παρέχει κάποια ευελιξία ως προς τον προσδιορισμό των στυλ των διαφορετικών στοιχείων της ιστοσελίδας. Τα στυλ για τον προσδιορισμό του τρόπου εμφάνισης του συγκεκριμένου στοιχείου μπορούν να οριστούν είτε σε ένα εξωτερικό αρχείο, είτε μέσα στο <HEAD> της σελίδας HTML, είτε μέσα σε μια ετικέτα. Εάν έχουν προσδιοριστεί περισσότερα του ενός στυλ στη σελίδα, η σειρά με την οποία αυτά θα εφαρμοστούν είναι η ακόλουθη:

1. Τα στυλ που ορίζει ο φυλλομετρητής
2. Εξωτερικά φύλλα μορφοποίησης (External Style Sheets)
3. Στυλ που προσδιορίζονται μέσα στο HEAD της σελίδας HTML
4. Στυλ που προσδιορίζονται μέσα στην ετικέτα ενός στοιχείου HTML

Η προτεραιότητα των παραπάνω στυλ αυξάνεται από πάνω προς τα κάτω. Δηλαδή τα στυλ που προσδιορίζονται μέσα στην ετικέτα έχουν μεγαλύτερη προτεραιότητα από (δηλαδή αντικαθιστούν) αυτά που ορίζονται μέσα στο HEAD, είτε σε ένα εξωτερικό αρχείο.

Σύνταξη

Ο προσδιορισμός του στυλ ενός στοιχείου είναι αρκετά απλός. Αρχικά ορίζεται το στοιχείο της HTML και στη συνέχεια μέσα σε άγκιστρα { } προσδιορίζονται οι ιδιότητές του και οι τιμές τους, όπως φαίνεται στην παρακάτω γραμμή:

```
{ Τύπος_στοιχείου {ιδιότητα1: τιμή1[; ιδιότητα2: τιμή2;...]}
```

Το τμήμα της γραμμής που βρίσκεται μέσα σε αγκύλες [] μπορεί να παραληφθεί. Όπως φαίνεται παραπάνω οι ιδιότητες διαχωρίζονται με κόμματα. Για καλύτερη παρουσίαση του κώδικα οι ιδιότητες μπορούν, επίσης, να τοποθετηθούν σε διαφορετικές γραμμές, όπως φαίνεται παρακάτω:

```
τύπος_στοιχείου {ιδιότητα1: τιμή1[;  
                 ιδιότητα2: τιμή2;  
                 ιδιότητα3: τιμή3 ...]}
```

Εάν για παράδειγμα θέλουμε τα στοιχεία τύπου h1 (heading 1) να εμφανίζονται με μέγεθος 18pt και χρώμα κόκκινο προσδιορίζουμε ένα στυλ ως εξής:

```
h1 {color: red; font-size= 18pt}
```

Ο παραπάνω προσδιορισμός μπορεί, επίσης, να γραφτεί ως εξής:

```
h1 {color: red;  
    font-size= 18pt}
```

Ομαδοποίηση

Το πρότυπο CSS επιτρέπει να δώσουμε το ίδιο στυλ σε διαφορετικά στοιχεία της HTML. Αυτό επιτυγχάνεται ορίζοντας όλα τα στοιχεία μαζί και χωρίζοντάς τα με κόμματα, όπως φαίνεται παρακάτω:

```
h1, h2, h3  
{ color: blue;  
  text-align: center;  
  font-size: 10pt;  
  font-style: italic  
}
```

Τα headings αυτά όταν εμφανιστούν στην ιστοσελίδα, θα έχουν στοίχιση στο κέντρο, μέγεθος 10 και θα έχουν μορφή πλάγιων γραμμμάτων.

Κλάσεις

Ένα στοιχείο του ίδιου τύπου (π. χ. μία επικεφαλίδα), μπορεί, επίσης, να εμφανιστεί με δύο ή περισσότερους διαφορετικούς τρόπους μέσα στην ιστοσελίδα. Αυτό επιτυγχάνεται με τον προσδιορισμό διαφορετικών κλάσεων για το στοιχείο αυτό. Στο παράδειγμα που ακολουθεί παρουσιάζεται η δυνατότητα προσδιορισμού διαφορετικών κλάσεων και στυλ του ίδιου στοιχείου HTML, καθώς και η εμφάνιση του στοιχείου με διαφορετικά στυλ μέσα στη σελίδα.

```

<HTML>
<HEAD>
<STYLE type="text/css">
h1.center {
color: blue;
    text-align: center;
    font-size: 10pt;
    font-style: italic
}
h1.right{
color: red;
    text-align: right;
    font-size: 10pt;
    font-style: italic
}
h1.large {
    text-align: right;
    font-size: 16pt
}
</STYLE>
</HEAD>
<BODY>
<h1 class="large"> My first large heading 1</h1>
<h1 class="right"> A smaller heading 1</h1>
<h1 class="center"> Another heading 1</h1>
<h1> This is a default heading 1 </h1>
</BODY>
</HTML>

```

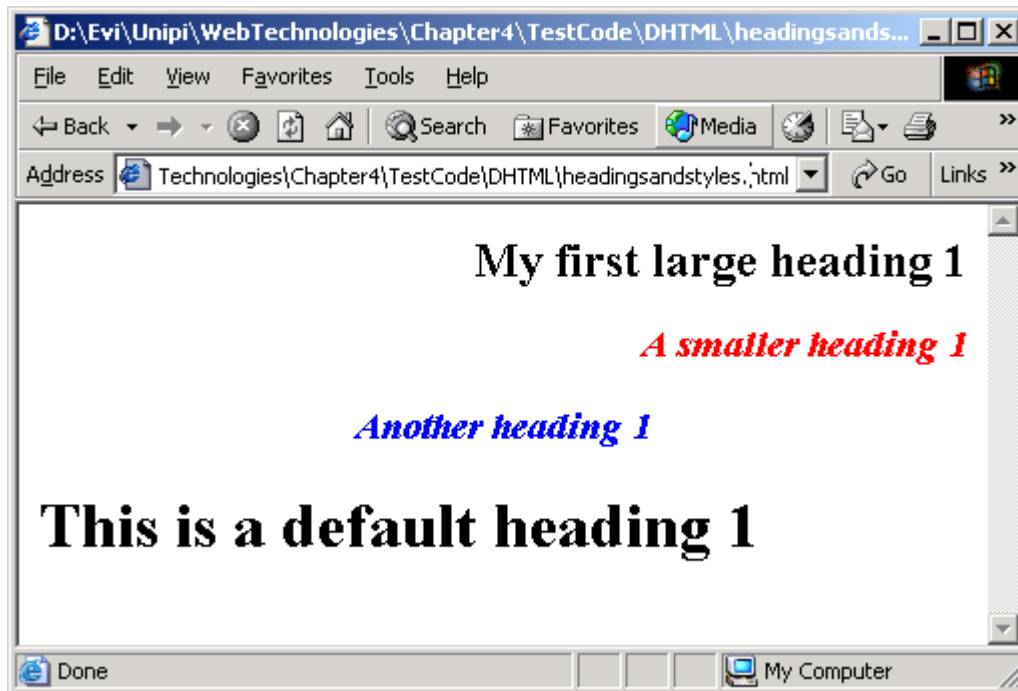
Για να προσδιορισθούν διαφορετικά στυλ για ένα στοιχείο HTML, ορίζονται διαφορετικές κλάσεις για το στοιχείο αυτό. Η κλάση ορίζεται ως attribute με όνομα class μέσα στην ετικέτα του στοιχείου:

```
<h1 class= "center">...</h1>
```

Όταν ο φυλλομετρητής συναντήσει μία τέτοιου είδους ετικέτα αναζητά κάποιο στυλ που έχει οριστεί για το στοιχείο αυτής της κλάσης. Αναζητά δηλαδή στυλ της μορφής:

```
h1.center {attribute1:value1;...}
```

Όταν ο παραπάνω κώδικας HTML κληθεί από κάποιο φυλλομετρητή, θα εμφανισθεί η ιστοσελίδα της Εικόνας 4.6.1.



Εικόνα 4.6.1 – Εφαρμογή διαφορετικών στυλ σε στοιχείο τύπου επικεφαλίδας 1

Για κάθε τύπο στοιχείου μπορεί να προσδιοριστεί μόνο μία κλάση. Επομένως η γραμμή κώδικα που ακολουθεί είναι λάθος διότι μπερδεύει το φυλλομετρητή.

```
<h1 class= "right" class= "center">...</h1>
```

Το πρότυπο CSS δίνει, επίσης τη δυνατότητα του προσδιορισμού μιας κλάσης που μπορεί να χρησιμοποιηθεί από οποιοδήποτε στοιχείο της HTML και να εμφανιστεί με βάση τους κανόνες μορφοποίησης που προσδιορίζει η κλάση. Έστω ότι για παράδειγμα ορίζουμε μία κλάση:

```
.left {text-align: left}
```

Στον κώδικα που ακολουθεί τα στοιχεία h1, h2 και p, ορίζονται όλα ως στοιχεία κλάσης "center" και σχεδιάζονται όλα στο κέντρο της σελίδας:

```
<h1 class="left"> This heading is left-aligned </h1>
<h2 class="left"> This heading is also left-aligned </h2>
<p class="left"> This paragraph is left-aligned.</p>
```

Χαρακτηριστικό (id) στοιχείων

Ένας άλλος τρόπος να ορίσουμε στυλ σε συγκεκριμένο τύπο στοιχείου HTML χωρίς να επηρεάσουμε όλα τα στοιχεία της ιστοσελίδας, είναι μέσω του προσδιορισμού ενός χαρακτηριστικού id. Η διαφορά του χαρακτηριστικού αυτού από τις κλάσεις που αναφέρθηκαν στην προηγούμενη ενότητα είναι ότι μία κλάση μπορεί να χρησιμοποιηθεί από πολλούς τύπους στοιχείων ενώ ένα χαρακτηριστικό id μπορεί να χρησιμοποιηθεί μόνο από ένα στοιχείο.

Το στυλ που ακολουθεί μπορεί να χρησιμοποιηθεί από ένα στοιχείο τύπου “p” με τιμή χαρακτηριστικού id ίση με “paragraph”. Ενώ δεν μπορεί να χρησιμοποιηθεί από ένα χαρακτηριστικό τύπου h3 ακόμα και εάν αυτό έχει id ίσο με paragraph.

```
p#paragraph
{
    text-style: bold;
    color: red
}
```

Ο φυλλομετρητής εφαρμόζει το παραπάνω στυλ σε μία παράγραφο που ορίζεται ως εξής:

```
<p id= "paragraph"> This is a paragraph </p>
```

Εισαγωγή Style Sheets σε ένα αρχείο HTML

Όπως προαναφέρθηκε υπάρχουν τρεις τρόποι εισαγωγής style sheets σε μία σελίδα HTML. Ο πρώτος τρόπος είναι να δημιουργηθούν ένα ή περισσότερα εξωτερικά αρχεία μέσα στα οποία προσδιορίζονται τα διαφορετικά στυλ των στοιχείων της σελίδας. Τα αρχεία αυτά καλούνται μέσα από το HEAD της σελίδας και τα διαφορετικά στυλ που προσδιορίζουν εφαρμόζονται στα στοιχεία της σελίδας. Το πλεονέκτημα της δημιουργίας εξωτερικών αρχείων στυλ, είναι ότι μπορούν να κληθούν από πολλές σελίδες HTML, επιτυγχάνοντας την ομοιογενή εμφάνιση όλων των σελίδων.

Για παράδειγμα, για να δημιουργηθεί ένα σύνολο από σελίδες HTML με τα ακόλουθα χαρακτηριστικά:

- έχουν φόντο μια εικόνα (picture1.gif),
 - οι επικεφαλίδες τύπου “heading 1” και “heading 2” εμφανίζονται με πλάγια γράμματα και χρώματα μπλε και κόκκινο αντίστοιχα,
 - οι παράγραφοι ξεκινούν 30 pixels δεξιά από το αριστερό περιθώριο της σελίδας και τα γράμματα των παραγράφων είναι τύπου “verdana”
- θα πρέπει να δημιουργηθεί ένα αρχείο με το ακόλουθο περιεχόμενο:

```
body {background-image: url("images/picture1.gif")}
h1{ color:blue; font-style: italic}
h2 {color:red; font-style:italic}
p { margin-left: 20px; font-family=verdana}
```

Όταν το αρχείο κληθεί από τις σελίδες, τα στυλ που προσδιορίζει εφαρμόζονται στα στοιχεία των σελίδων αυτών. Εάν το όνομα του αρχείου είναι “mystyles.css”, η σύνταξη για την κλήση του αρχείου στην επικεφαλίδα μιας σελίδας HTML είναι η ακόλουθη:

```
<head>
<link rel="stylesheet" type="text/css" href="mystyles.css" />
</head>
```

Καλώντας το αρχείο από πολλές σελίδες, τα στυλ που προσδιορίζει εφαρμόζονται αυτόματα σε όλες τις σελίδες. Με αυτόν τον τρόπο διευκολύνεται η δημιουργία ομοιόμορφων δικτυακών τόπων. Ένα άλλο πλεονέκτημα της χρήσης εξωτερικών αρχείων είναι η δυνατότητα γρήγορης αλλαγής του τρόπου εμφάνισης των σελίδων, εφόσον με μία απλή ενημέρωση του αρχείου αλλάζει αυτόματα το στυλ σε όλες τις σελίδες που το καλούν.

Ο δεύτερος τρόπος εισαγωγής φύλλων στυλ σε μία σελίδα HTML, είναι η τοποθέτησή τους μέσα σε μία ετικέτα style στο head του κώδικα της σελίδας, όπως φαίνεται στο παράδειγμα που ακολουθεί:

```
<head>
  <style type="text/css">
    body {background-image: url("images/picture1.gif")}
    h1{ color:blue; font-style: italic}
    h2 {color:red; font-style:italic}
    p { margin-left: 20px; font-family=verdana}
  </style>
</head>
```

Ο φυλλομετρητής διαβάσει τα περιεχόμενα της ετικέτας style και ανάλογα αλλάζει την εμφάνιση των στοιχείων της σελίδας. Αυτός ο τρόπος εισαγωγής στυλ είναι κατάλληλος στην περίπτωση που θέλουμε να προσδιορίσουμε το στυλ μίας μόνο σελίδας HTML, ενώ θα πρέπει να αποφεύγεται όταν έχουμε πολλές ομοιόμορφες σελίδες.

Ο τρίτος τρόπος εισαγωγής ενός στυλ είναι μέσα στην ετικέτα ενός στοιχείου της σελίδας. Ο προσδιορισμός του στυλ γίνεται μέσω της χαρακτηριστικού style, όπως φαίνεται στο παράδειγμα που ακολουθεί:

```
<h1 style="color:green; font-style: italic"> This is a green heading
</h1>
```

Αυτός ο τρόπος εισαγωγής στυλ δεν έχει τα πλεονεκτήματα των δύο προηγούμενων τρόπων, εφόσον αναμειγνύει το περιεχόμενο του στοιχείου με την εμφάνισή του. Δίνει, όμως, τη δυνατότητα προσδιορισμού διαφορετικού τρόπου εμφάνισης για ένα στοιχείο συγκεκριμένου περιεχομένου. Όταν δηλαδή ο φυλλομετρητής διαβάσει την παραπάνω γραμμή κώδικα, θα εμφανίσει την επικεφαλίδα "This is a green heading", με πράσινο χρώμα και πλάγια γράμματα, ανεξάρτητα από το εάν έχουν προηγουμένως προσδιοριστεί διαφορετικά στυλ για τις επικεφαλίδες τύπου 1.

Όπως προαναφέρθηκε στην αρχή της ενότητας αυτής, σε μία σελίδα μπορούν να προσδιοριστούν περισσότερα του ενός στυλ. Οι φυλλομετρητές εφαρμόζουν τα διαφορετικά στυλ ως εξής: Τα στυλ που προσδιορίζονται στο εξωτερικό αρχείο αντικαθίσταται από αυτά που έχουν εισαχθεί στον κώδικα της σελίδας HTML. Ενώ τα στυλ που ορίζονται στο head της σελίδας αντικαθίστανται από αυτά που εισάγονται στις ετικέτες των στοιχείων.

Ο τρόπος εφαρμογής των στυλ από τους φυλλομετρητές φαίνεται στο παράδειγμα που ακολουθεί.

Παράδειγμα

Ένα αρχείο με όνομα external.css περιλαμβάνει τα στυλ που φαίνονται στο παρακάτω πλαίσιο.

```
body {background-color: yellow }
h1 {color:blue; font-size:20pt}
h2 {font-family: courier;
    font-size: 130%;
    color: "red"
}
hr {color:navy}
p {font-size:11pt; margin-left: 15px}
a:link      {color:green}
a:visited   {color:gray}
a:active    {color:blue}
a:hover     {color:black}
```

Όταν το αρχείο κληθεί από μία σελίδα HTML θα μετατρέψει το χρώμα του φόντου σε κίτρινο και θα αλλάξει τη μορφή των επικεφαλίδων τύπου 1 και 2. Οι επικεφαλίδες τύπου 1 θα έχουν μέγεθος γραμμάτων 20 pt και χρώμα μπλε, ενώ οι επικεφαλίδες τύπου 2 θα έχουν γράμματα τύπου “courier”, με μέγεθος το 130% αυτού που προσδιορίζεται από το φυλλομετρητή και χρώμα κόκκινο. Μια γραμμή τύπου hr στη σελίδα θα έχει χρώμα μπλε σκούρο και μία παράγραφος θα έχει μέγεθος γραμμάτων 11pt και αριστερό περιθώριο 15 px. Τέλος ένας σύνδεσμος θα εμφανίζεται με χρώμα πράσινο, όταν επιλεγεί και είναι ενεργός θα γίνεται μπλε, αφού επιλεγεί θα γίνεται γκρι και κάθε φορά που ο δείκτης του ποντικιού θα δείχνει πάνω του, θα γίνεται μαύρος.

Το αρχείο αυτό καλείται από μία σελίδα HTML με την εντολή

```
<link rel="stylesheet" type="text/css" href="external.css" />
```

Στη σελίδα αυτή εκτός από το εξωτερικό αρχείο εισάγονται επιπλέον στυλ, μέσα στο head της σελίδας καθώς και μέσα στις ετικέτες HTML.

```
<html>
<head>
<link rel="stylesheet" type="text/css"
href="external.css" />
<style type="text/css">
h1 {font-size: 30pt; font-style=italic}
</style>
</head>

<body>

<h1>This is a header 1</h1>
<h1 style= "font-style: normal">This is a modified header 1</h1>
<h2> This is a header 2 </h2>
<h2 style= "color:green"> This is a modified header 2 </h2>
<hr>

<p>You can see that the style
sheet formats the text</p>
```

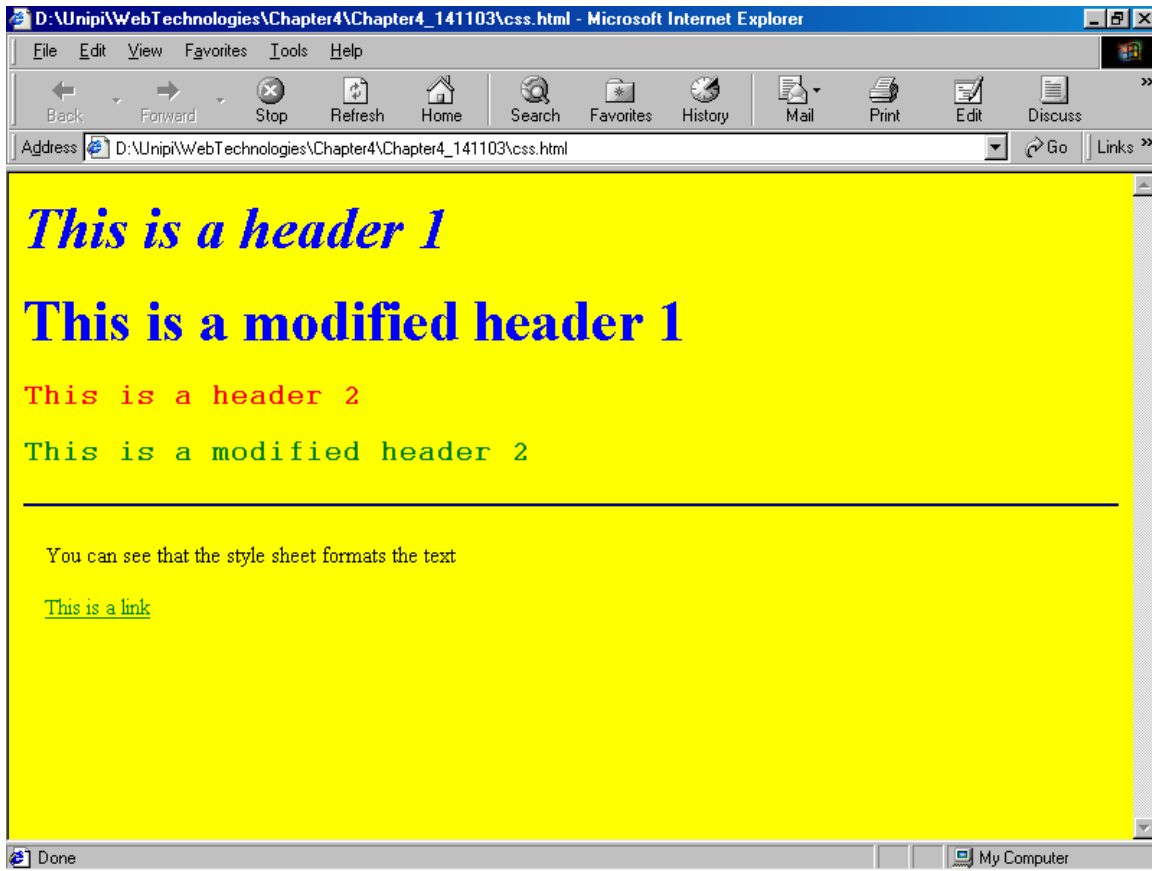
```
<p><a href="http://www.unipi.gr" target="_blank">This is a link</a></p>  
</body>  
</html>
```

Στο head της σελίδας HTML προσδιορίζεται ένα νέο στυλ για την επικεφαλίδα τύπου h1, σύμφωνα με το οποίο η επικεφαλίδα θα έχει πλάγια γράμματα και μέγεθος γραμμάτων 30pt. Το νέο μέγεθος θα αντικαταστήσει αυτό που είχε προσδιοριστεί στο εξωτερικό αρχείο, δηλαδή το 20pt, το χρώμα θα παραμείνει ίδιο δηλαδή μπλε, ενώ θα προστεθεί άλλη μία ιδιότητα και τα γράμματα θα είναι πλάγια. Άρα η πρώτη επικεφαλίδα με περιεχόμενο “This is a header” θα εμφανιστεί με πλάγια γράμματα, μεγέθους 30 και χρώματος μπλε.

Στην επόμενη επικεφαλίδα με περιεχόμενο “This is a modified header 1” έχει προστεθεί, στην ετικέτα, η ιδιότητα `font-style:normal`. Η ιδιότητα αυτή θα αντικαταστήσει τη `font-style:italic` που ορίστηκε στο HEAD και η επικεφαλίδα αυτή θα εμφανιστεί με τα χαρακτηριστικά της πρώτης, μόνο που τα γράμματα δε θα είναι πλάγια, αλλά κανονικά.

Η τρίτη επικεφαλίδα με περιεχόμενο “This is a header 2”, θα εμφανιστεί όπως ορίστηκε στο εξωτερικό αρχείο, ενώ η τέταρτη θα εμφανιστεί με τον ίδιο ακριβώς τρόπο, μόνο που θα έχει χρώμα πράσινο. Η ιδιότητα επομένως `color:green`, που ορίστηκε στην ετικέτα της επικεφαλίδας θα αντικαταστήσει την αντίστοιχη ιδιότητα (`color:red`) που είχε οριστεί στο εξωτερικό αρχείο.

Το αποτέλεσμα του κώδικα φαίνεται στην Εικόνα 4.6.2.



Εικόνα 4.6.2 – Εφαρμογή στυλ σε στοιχεία της σελίδας HTML

4.6.5 Χρήση του προτύπου CSS-P

Το πρότυπο CSS-P προσδιορίζει τη θέση και την εμφάνιση των αντικειμένων ή των στοιχείων σε μία σελίδα HTML.

Θέση αντικειμένου

Η θέση ενός αντικειμένου καθορίζεται από τα χαρακτηριστικά `left` και `top`. Όπου το `left` προσδιορίζει τη θέση του αντικειμένου από την αριστερή πλευρά της σελίδας ενώ το `top` προσδιορίζει τη θέση του από την πάνω πλευρά της σελίδας. Υπάρχουν δύο είδη προσδιορισμού θέσεων του αντικειμένου: η σχετική θέση του αντικειμένου (`position:relative`) και η απόλυτη θέση το υ αντικείμενο υ (`position:absolute`).

position:relative

Η ιδιότητα αυτή τοποθετεί ένα αντικείμενο σε μία νέα θέση σχετική με την προηγούμενη του θέση. Στο παράδειγμα που ακολουθεί η παράγραφος τοποθετείται 20 pixel πιο δεξιά και 10 pixel πιο κάτω από τη θέση που τοποθετείται συνήθως.

```
p { position:relative;
    left:20;
```

```
top:10
```

position:absolute

Η ιδιότητα αυτή τοποθετεί ένα αντικείμενο σε μία συγκεκριμένη θέση που προσδιορίζεται από τα χαρακτηριστικά `top` και `left`. Στο παράδειγμα που ακολουθεί η παράγραφος τοποθετείται 20 pixel δεξιότερα από το αριστερό άκρο της σελίδας και 10 pixel πιο κάτω από το πάνω άκρο της σελίδας.

```
p { position:absolute;
    left:20;
    top:10
}
```

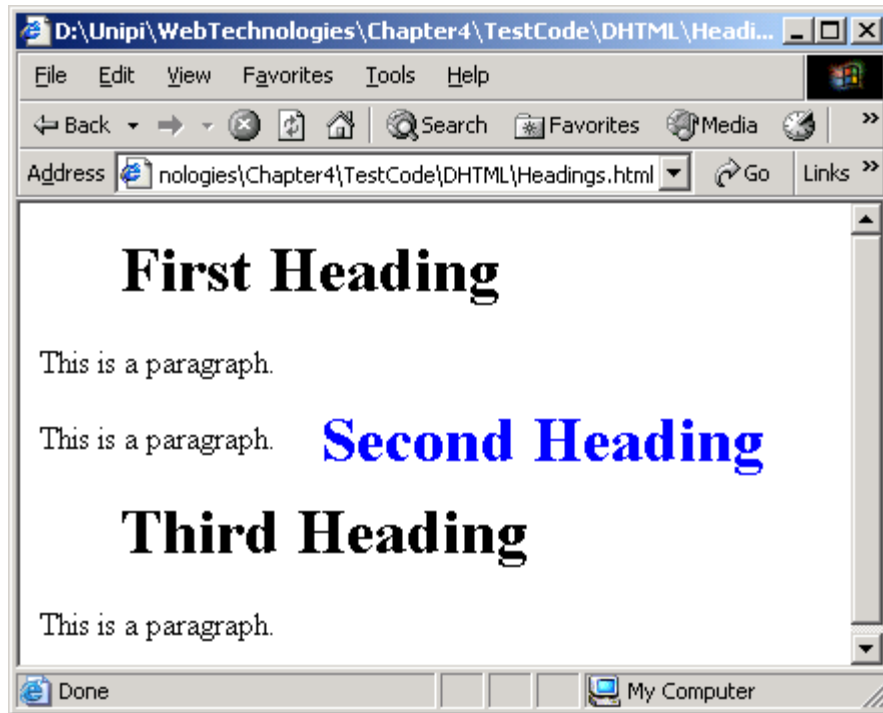
Παράδειγμα προσδιορισμού θέσης αντικειμένου

Στο παράδειγμα που ακολουθεί φαίνεται η διαφορά των ιδιοτήτων `position:relative` και `position:absolute`.

```
<html>
<head>
<style>
h1
{
position:relative;
left:40px;
}
h1.x
{
position:absolute;
left:150px;
top:100px;
color:blue;
}
</style>
</head>
<body>
<h1>First Heading</h1>
<p>This is a paragraph.</p>
<h1 class=x>Second Heading</h1>
<p>This is a paragraph.</p>
<h1>Third Heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

Το αποτέλεσμα του παραπάνω κώδικα σε ένα φυλλομετρητή φαίνεται στην Εικόνα 4.6.3. Το δεύτερο heading που σχεδιάστηκε με μπλε χρώμα έχει απόλυτη θέση

(position:absolute) 150 pixel από αριστερά και 100 pixel από πάνω. Οι υπόλοιπες επικεφαλίδες έχουν θέση 40 pixel πιο αριστερά από τη θέση στην οποία θα σχεδιάζονταν κανονικά. Εάν δεν είχε προσδιοριστεί τιμή για το left, το αριστερό άκρο των επικεφαλίδων θα ήταν στην ίδια ευθεία με το αριστερό άκρο των παραγράφων.



Εικόνα 4.6.3 – Παράδειγμα προσδιορισμού θέσης αντικειμένου

Ορατότητα

Στο Netscape, αλλά και στον Internet Explorer, η ιδιότητα της ορατότητας (visibility) ενός αντικειμένου μπορεί να πάρει τις τιμές: visible και hidden. Η visible δηλώνει πως το αντικείμενο θα εμφανίζεται, ενώ η hidden πως θα παραμείνει κρυφό. Στο παράδειγμα που ακολουθεί οι επικεφαλίδες τύπου h1 δεν φαίνονται ενώ οι επικεφαλίδες τύπου h2 εμφανίζονται κανονικά.

```
h1 { visibility:hidden; }  
h2 { visibility:visible; }
```

Εάν δεν δοθεί κάποια τιμή για την ορατότητα (visibility) του αντικειμένου, αυτή θεωρείται εξ'ορισμού visible.

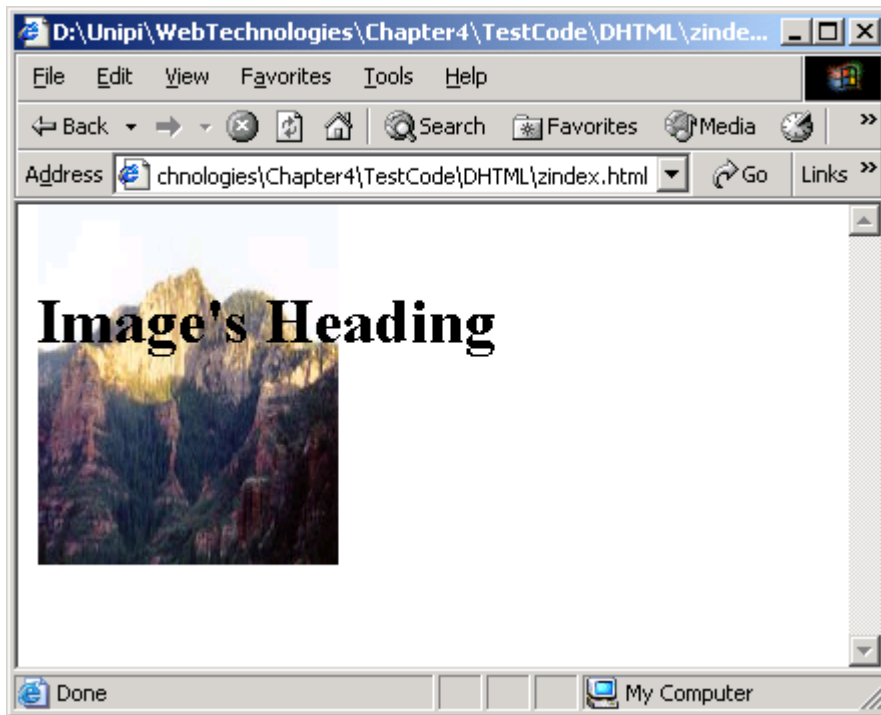
Stacking order (z-index)

Η ιδιότητα z-index καθορίζει τη θέση του αντικειμένου σε μία στοίβα από αντικείμενα και χρησιμοποιείται για να προσδιορίσει πότε ένα αντικείμενο βρίσκεται πίσω από κάποιο άλλο. Η σειρά που έχει δοθεί στη στοίβα προσδιορίζεται από έναν ακέραιο. Εξ'ορισμού το z-index έχει τιμή 0, ενώ η μικρότερη τιμή που μπορεί να πάρει είναι η -1. Κατά συνέπεια ένα αντικείμενο με z-index= -1 τοποθετείται κάτω από τα υπόλοιπα αντικείμενα. Στο παράδειγμα που ακολουθεί η εικόνα έχει z-index=-1 και τοποθετείται κάτω από το heading.

```
<html>
<head>
<style type="text/css">
img.x
{
position:absolute;
left:10px;
top:0px;
z-index:-1;
}
h1.y
{
position:absolute;
left:10px;
top:40px;
}
</style>
</head>
<body>

<h1 class="y">Image's Heading</h1>
</body>
</html>
```

Όταν φορτωθεί η παραπάνω σελίδα σε φυλλομετρητή Internet Explorer θα εμφανιστεί το αποτέλεσμα της Εικόνας 4.6.4:



Εικόνα 4.6.4 – Χρήση z-index για την εμφάνιση εικόνας πίσω από κείμενο

Ο προσδιορισμός της θέσης του αντικειμένου με τη χρήση του z-index δε λειτουργεί πάντα σε φυλλομετρητή Netscape.

Φίλτρα

Ο Internet Explorer υποστηρίζει, επίσης, κάποια φίλτρα που επιτρέπουν τη μετατροπή της εμφάνισης αντικειμένων της σελίδας HTML. Τα φίλτρα ορίζονται με την ιδιότητα filter η οποία μπορεί να πάρει τιμές όπως: filter:blur (κάνει ένα αντικείμενο θαμπό), filter: glow (αυξάνει τη λάμψη ενός αντικειμένου) και filter:gray (μετατρέπει ένα αντικείμενο σε απρόμαυρο). Η ιδιότητα filter παίρνει και άλλες τιμές για τις οποίες όμως δε γίνεται αναφορά, καθώς τα φίλτρα υποστηρίζονται μόνο από τον Internet Explorer.

4.6.6 Document Object Model (DOM)

Το Document Object Model (DOM) της DHTML παρέχει ένα ιεραρχικό μοντέλο για την απεικόνιση των αντικειμένων της σελίδας και επιτρέπει την πρόσβαση στα αντικείμενα καθώς και την αλλαγή των ιδιοτήτων τους μέσω της JavaScript.

Το DOM ξεκινάει από ένα βασικό αντικείμενο που ονομάζεται “**window**”, που αναφέρεται στο ίδιο το παράθυρο του φυλλομετρητή. Εντός του αντικειμένου του παραθύρου, βρίσκεται το αντικείμενο του εγγράφου “**document**” το οποίο αναφέρεται στη σελίδα HTML. Κάτω από το αντικείμενο του εγγράφου σε ιεραρχική δομή (δομή δέντρου) τοποθετούνται όλα τα αντικείμενα της σελίδας, όπως οι εικόνες, τα πεδία κειμένου, τα πλήκτρα και οι φόρμες. Κάθε αντικείμενο έχει ιδιότητες. Οι ιδιότητες είναι

τα τυπικά χαρακτηριστικά του αντικειμένου όπως χρώμα, μέγεθος, θέση κτλ. και διαφέρουν για κάθε αντικείμενο.

Όπως προαναφέρθηκε στην ενότητα 4.3 χρησιμοποιώντας τη JavaScript μπορούμε να αναφερθούμε στα αντικείμενα της σελίδας (κλαδιά), καλώντας κάθε αντικείμενο, από την ρίζα έως το κλαδί και χωρίζοντας τα ονόματά τους με τελείες. Για παράδειγμα έστω ότι έχουμε μία εικόνα με όνομα "start". Ο κώδικας HTML, που δίνει σε μια εικόνα το όνομά της είναι:

```
<IMG SRC="start.gif" NAME="start">
```

Για να αναφερθούμε σε μία εικόνα χρησιμοποιώντας τη JavaScript γράφουμε:

```
document.images["image_name"]
```

Για να αναφερθούμε στην εικόνα που ορίστηκε παραπάνω γράφουμε

```
document.images["start"]
```

Δύο άλλα αντικείμενα που υποστηρίζει το DOM είναι τα αντικείμενα του "navigator" το οποίο περιλαμβάνει πληροφορίες για το φυλλομετρητή του χρήστη και το αντικείμενο "event" το οποίο περιλαμβάνει πληροφορίες για τα γεγονότα που συμβαίνουν σε ένα έγγραφο HTML.

Στις περισσότερες περιπτώσεις, ο Netscape και ο Internet Explorer χρησιμοποιούν παρόμοιο DOM. Όμως, στις πιο πολύπλοκες ιστοσελίδες, το DOM του Netscape και το DOM του Internet Explorer διαφέρουν αρκετά.

Προσπέλαση αντικειμένων

Όταν ένα στοιχείο έχει ένα χαρακτηριστικό id μπορεί να προσπελαστεί με την εντολή getElementById, όπως φαίνεται στο παράδειγμα που ακολουθεί:

```
<html>
<body>
<h1 id="myheader">This is a header</h1>
<script type="text/javascript">
document.getElementById('myheader').style.color="red"
</script>
</body>
</html>
```

Στην σελίδα HTML που θα εμφανιστεί η επικεφαλίδα θα παρουσιαστεί με κόκκινο χρώμα.

4.6.7 Διαχείριση Γεγονότων

Κάποια νέα events που προστέθηκαν στην έκδοση 4.0 της HTML φαίνονται στον Πίνακα 4.6.2:

Πίνακας 4.6.2 – Γεγονότα που προστέθηκαν στην έκδοση 4.0 της HTML

onDbClick	Διπλό κλικ στο ποντίκι
onKeyDown	Όταν πατάμε κάποιο κουμπί στο πληκτρολόγιο
onKeyPress	Όταν πατάμε και αφήνουμε κάποιο κουμπί στο πληκτρολόγιο
onKeyUp	Όταν ελευθερώνουμε ένα πατημένο κουμπί
onMouseDown	Όταν είναι πατημένο το ποντίκι
onMouseMove	Όταν κινείται το ποντίκι
onMouseUp	Όταν ελευθερώσουμε το ποντίκι
onResize	Όταν μεγενθύνουμε το παράθυρο

Τα γεγονότα που συμβαίνουν σε μία φόρμα μπορούν να προκαλέσουν την εκτέλεση συναρτήσεων καθώς και αλλαγές στις ιδιότητες αντικειμένων.

Στο παράδειγμα που ακολουθεί χρησιμοποιούνται τα γεγονότα onmouseover και onmouseout. Η επικεφαλίδα εμφανίζεται αρχικά μπλε, όταν ο δείκτης του ποντικιού περάσει πάνω από την επικεφαλίδα, το χρώμα της γίνεται κόκκινο. Ενώ το χρώμα γίνεται ξανά μπλε, όταν ο δείκτης του ποντικιού απομακρυνθεί.

```
<html>
<body>

<h1 style="color='blue' "
onmouseover="style.color='red' "
onmouseout="style.color='blue' ">
Mouse over this text</h1>

</body>
</html>
```

Στο παρακάτω παράδειγμα παρουσιάζονται τα γεγονότα onmousemove και onclick. Όταν ο δείκτης του ποντικιού μετακινείται πάνω στη σελίδα συμβαίνει το γεγονός onmousemove και ενεργοποιείται η συνάρτηση movetext(), η οποία μεταρρέπει τη θέση του στοιχείου σε σχέση με την κανονική του θέση (position: relative). Η μεταβλητή i αυξάνεται κατά ένα (i++) και το στοιχείο μετακινείται i pixels αριστερά από το άκρο της σελίδας.

Με το πάτημα του πλήκτρου του ποντικιού συμβαίνει ένα γεγονός onclick και ενεργοποιείται η συνάρτηση changecolor() η οποία θέτει το i =1 και αλλάζει τη θέση της επικεφαλίδας σε ένα pixel δεξιά από το αριστερό άκρο της σελίδας. Επίσης, μεταρρέπει το χρώμα της επικεφαλίδας σε κόκκινο.

```
<html>
<head>
<script type="text/javascript">
var i=1
function movetext()
{
document.getElementById('header').style.position="relative"
document.getElementById('header').style.left=i
i++
}
function changecolor()
{
i=1;
document.getElementById('header').style.color="red"
}
</script>
</head>
<body onmousemove="movetext()" onclick="changecolor()">
<h1 id="header">Moving Text</h1>
</body>
</html>
```

4.7 Ερωτήσεις – Ασκήσεις – Θέματα για ανάπτυξη

Ερωτήσεις

Ερωτήσεις στη JavaScript	
Ανάμεσα σε ποια ετικέτα τοποθετείται ο κώδικας της JavaScript ;	<script> <js> <javascript> <jscript>
Ποια είναι η σωστή σύνταξη για να γράψουμε σε μία σελίδα “Welcome to JavaScript”;	(“Welcome to JavaScript”) document.write(“Welcome to JavaScript”) write(“Welcome to JavaScript”) response.write(“Welcome to JavaScript”)
Πού τοποθετείται ο κώδικας της JavaScript;	Πριν το <head> της HTML σελίδας Μέσα στην περιοχή του <head> Μέσα στην περιοχή του <body> Είτε στο <body> είτε στο <head>
Πώς γράφουμε ένα μήνυμα “Hello World” μέσα σ’ ένα κουτί μηνύματος alert;	alertmessage(“Hello World”) alert “Hello World” alertBox= “HelloWorld” alert(“Hello World”)
Πώς ορίζουμε / δημιουργούμε μία νέα συνάρτηση;	var function newfunction() function=newfunction() function newfunction() var newfunction()
Πώς καλούμε την παραπάνω συνάρτηση για να εκτελεστεί;	Run newfunction() call newfunction() newfunction newfunction()
Πώς γράφουμε μία εντολή υπό συνθήκη: if “j” είναι ίσο με 10;	if j=10 then if j = = 10 if (j = = 10) if (j=10)
Πώς γράφουμε μία εντολή υπό συνθήκη: if “i” διαφορετικό από 10;	If (j<>10) if j!=10 then if (j!=10) if j<>10
Πόσοι τύποι βρόχων υπάρχουν στη JavaScript;	Οι βρόχοι “for” και “while” Οι βρόχοι “for”, “while” και “do...while” Οι βρόχοι “while” και “do...while” Ο βρόχος “for”
Πώς συντάσσεται ένας βρόχος for;	for (i=1 to 10) for (i<=10; i++) for i=1 until i=10 for (i=0; i<=10; i++)

Πώς προσθέτουμε ένα σχόλιο στη JavaScript;	*This is a comment //This is a comment 'This is a comment' *This is a comment
Ποια είναι η σωστή σύνταξη για να δηλώσουμε ένα καινούριο array;	var temp= new Array(5, 8, 10) var temp =new Array[5, 8, 10] var temp=new array "5, 8, 10" temp=new array(5, 8, 10)
Πώς βρίσκουμε τον μικρότερο από τους αριθμούς 5 και 8;	math.min(5,8) math.less(5,8) min(5,8) less(5,8)
Πώς ανοίγουμε ένα παράθυρο με όνομα mywindow που να δείχνει στη διεύθυνση http://www.unipi.gr ;	window("http://www.unipi.gr", "mywindow") window.open("http://www.unipi.gr", "mywindow") window.new("http://www.unipi.gr", "mywindow") new mywindow=window("http://www.unipi.gr")
Πώς μπορούμε να βρούμε το είδος του φυλλομετρητή στον οποίο θα εκτελεστεί ο κώδικας JavaScript;	client.navigator application.navName browser.name navigator.appName
Ερωτήσεις στη VBScript	
Ανάμεσα σε ποιες ετικέτες περιλαμβάνονται τα scripts;	<script>...</script> <%...%> <&>...</&> <%>....</%>
Πώς μπορούμε με τη VBScript να γράψουμε "Hello World" σε μία HTML σελίδα;	Document.Write("Hello World") Document.println("Hello World") "Hello World" Response.Write("Hello World")
Πόσα στοιχεία έχει ένας πίνακας που δηλώνεται ως εξής: Dim colors(10);	9 10 11
Ποια ανάθεση τιμής είναι σωστή;	Colors[1].setvalue= "red" colors(1).value= "red" colors(1) = "red"
Πώς δηλώνεται μία συνάρτηση που επιστρέφει τιμή;	Dim calculate() Sub calculate() Procedure calculate() Function calculate()
Ποια είναι η σωστή εντολή για την εμφάνιση ενός κουτιού μηνύματος;	MsgBox ("hello") Alert "hello" MsgBox "hello" Alert("hello")
Ερωτήσεις στη Java	

Ποια από τις εντολές είναι λάθος;	<pre>if (j<=7) {...} if (j= =7) {...} if (j=7) {...}</pre>
Ποιο είναι το σωστό;	<pre>if (k<>=8) {...} if (k<>8) {...} if (k!=8) {...}</pre>
Ποιο είναι το σωστό;	<pre>if (j==1) and (j<k) {...} if (j==1) & (j<k) {...} if (j==1) && (j<k) {...} if (j==1) AND (j<k) {...}</pre>
Πώς δημιουργούμε μία κλάση applet με όνομα MyApplet;	<pre>Public applet MyApplet {...} Public class MyApplet extends Applet {...} Public void MyApplet extends Applet {...} Public class MyApplet from Applet {...}</pre>
Ποιο από τα σχόλια είναι λάθος;	<pre>/** This is a comment that exceeds one line */ // This is a one line comment // This is a comment that exceeds one line / /* This is a comment that exceeds one line */</pre>
Με ποιο τρόπο αλλάζουμε το χρώμα στο αντικείμενο graphics1 της κλάσης Graphics;	<pre>graphics1.setColor("red"); graphics1.Color(color.red); graphics1.setColor(Color.red); graphics1.color= "red";</pre>
Ερωτήσεις στη DHTML και στο πρότυπο CSS	
Σε ποιο τμήμα της σελίδας HTML καλούμε ένα εξωτερικό αρχείο (style sheet);	<pre>Στο <head> της σελίδας Στο <body> της σελίδας Μέσα στην ετικέτα html Πριν την ετικέτα <html></pre>
Ποια ιδιότητα χρησιμοποιείται για τον προσδιορισμό των στυλ μέσα στην ετικέτα ενός στοιχείου;	<pre>style css font-style styles</pre>
Ποια σύνταξη είναι σωστή	<pre>body {color: black} {body:color=black} {body;color:black} body:color=black</pre>
Πώς μπορεί να προσδιοριστεί το χρώμα του φόντου μιας σελίδας HTML;	<pre>body {background-color: yellow } body {bgcolor: yellow } <body style="bgcolor:yellow"> document {background-color: yellow }</pre>

Με ποια εντολή μπορούμε να ορίσουμε πως όλες οι επικεφαλίδες τύπου 1 θα εμφανιστούν με πλάγια γράμματα;	<pre><h1 style="font-style:italic"> h1 {font-style:italic} <h1 font="style:italic"> h1.all {font-style:italic} h1 {font-type:italic}</pre>
Με ποιο τρόπο μπορούμε να μετατρέψουμε τα γράμματα όλων των παραγράφων σε bold;	<pre><p style="font-size:bold"> <p style="text-size:bold"> p {font-weight:bold} p {text-size:bold}</pre>
Με ποια ιδιότητα μπορούμε να ορίσουμε το μέγεθος του κειμένου;	<pre>FontSize text-size font-size text-style</pre>

Θέματα για ανάπτυξη

1. Ποια τα πλεονεκτήματα και ποια τα μειονεκτήματα των εφαρμογών που εκτελούνται στον πελάτη;
2. Ποιες οι διαφορές μεταξύ της Java και της JavaScript;
3. Ποιες δυνατότητες μας δίνει η JavaScript στην ανάπτυξη ιστοσελίδων;
4. Συγκρίνετε τη JavaScript και τη VBScript.
5. Ποιες οι διαφορές της VBScript και της JavaScript στην υλοποίηση των πινάκων (arrays).
6. Ποιες οι διαφορές της VBScript και της JavaScript στη δήλωση και την κλήση συναρτήσεων.
7. Ποιες οι διαφορές στην εκτέλεση ενός προγράμματος Java, με την εκτέλεση προγραμμάτων που έχουν υλοποιηθεί σε άλλες γλώσσες προγραμματισμού;
8. Ποια τα πλεονεκτήματα που δίνει η χρήση της JVM (Java Virtual Machine) στη Java.
9. Τι είναι ένα Java Applet; Ποιες δυνατότητες δίνει στην ανάπτυξη εφαρμογών Διαδικτύου; Μπορούν οι δυνατότητες που παρέχει το applet να υλοποιηθούν και από μία γλώσσα σεναρίων (π.χ. VBScript ή JavaScript);
10. Ποια είναι τα μειονεκτήματα των Applets;
11. Τι γνωρίζετε για το χρόνο ζωής ενός Applet;
12. Ποιες είναι οι βασικές μέθοδοι που υλοποιούν ένα Applet;
13. Τι σημαίνει το ακρόνυμο DHTML. Από ποιες τεχνολογίες αποτελείται;
14. Αναφέρετε τα πλεονεκτήματα και τα μειονεκτήματα της ανάπτυξης ιστοσελίδων με τη χρήση της DHTML.
15. Τι είναι το πρότυπο CSS και ποια πλεονεκτήματα παρέχει στην ανάπτυξη κόμβων ιστοσελίδων (web sites);
16. Με ποιους τρόπους μπορούμε να εισάγουμε στυλ σε μια σελίδα HTML; Ποια τα πλεονεκτήματα του κάθε τρόπου εισαγωγής στυλ;
17. Πώς μπορούμε να ορίσουμε τη θέση ενός αντικειμένου με βάση το CSS-P;
18. Τι γνωρίζεται για το DOM της DHTML; Πώς μπορούμε να προσπελάσουμε και να αλλάξουμε την ιδιότητα ενός αντικειμένου με τη βοήθεια της JavaScript;

Ασκήσεις:

1. Δημιουργήστε ένα σενάριο JavaScript το οποίο εκτυπώνει τα στοιχεία σας (Όνομα, Επίθετο, Ημερομηνία Γέννησης και Διεύθυνση) σε μία κενή σελίδα HTML. Το Όνομα και το Επίθετο να εκτυπωθούν στην ίδια γραμμή με γράμματα μεγέθους 5, ενώ τα υπόλοιπα στοιχεία να εκτυπωθούν σε διαφορετικές σειρές με γράμματα μπλε χρώματος.

2. Να δημιουργηθεί μία σελίδα HTML στην οποία με χρήση Javascript θα:

- ζητούνται από το χρήστη τέσσερις αριθμοί (a, b, c, d αντίστοιχα).
- θα εμφανίζεται στη σελίδα ο 2x2 πίνακας

a	b
c	d

- θα υπολογίζεται και εμφανίζεται η ορίζουσα D.
Η ορίζουσα δίνεται από τον τύπο $D=ad-bc$

3. Φτιάξτε μία φόρμα HTML στην οποία ο χρήστης μπορεί να εισάγει δύο αριθμούς. Στην φόρμα αυτή υπάρχουν δύο πλήκτρα, όπου το μεν ένα επαναφέρει τις αρχικές τιμές της φόρμας και το άλλο ενεργοποιεί μία συνάρτηση. Η συνάρτηση αυτή υπολογίζει το άθροισμα και τη διαφορά των αριθμών που εισήγαγε ο χρήστης και εμφανίζει τα αποτελέσματα σε ένα κουτί μηνύματος.

- Η άσκηση να υλοποιηθεί και σε JavaScript και σε VBScript.
- Να γίνεται έλεγχος των στοιχείων που πληκτρολόγησε ο χρήστης και εάν αυτά δεν είναι ακέραιοι αριθμοί να εμφανίζεται μήνυμα λάθους.

4. Φτιάξτε μία σελίδα HTML στην οποία, ανάλογα με το τί ημέρα είναι, να εμφανίζεται διαφορετικό μήνυμα. Ο έλεγχος της ημέρας και ο προσδιορισμός του μηνύματος που θα εμφανίζεται να γίνεται με τη VBScript.

5. Φτιάξτε μία φόρμα εγγραφής στην οποία ο χρήστης εισάγει τα στοιχεία του (Όνομα, Επίθετο, Επάγγελμα και Ηλικία). Στη φόρμα να υπάρχουν ένα πλήκτρο υποβολής (submit) και ένα πλήκτρο ακύρωσης (reset), για την επαναφορά των αρχικών τιμών στα πεδία της φόρμας. Όταν ο χρήστης επιλέξει το πλήκτρο “submit” να εκτελούνται οι ακόλουθες ενέργειες:

- Να γίνεται έλεγχος εάν ο χρήστης έχει συμπληρώσει όλα τα πεδία.
- Να γίνεται έλεγχος στο πεδίο της ηλικίας
- Σε περίπτωση που κάτι δεν είναι σωστό να εμφανίζεται σχετικό μήνυμα λάθους σε ένα παράθυρο μηνύματος.
- Εάν όλα είναι σωστά να εμφανίζεται μία νέα HTML σελίδα στην οποία επιβεβαιώνεται η εγγραφή του χρήστη και τυπώνονται τα στοιχεία που έδωσε ο χρήστης.

6. Δημιουργήστε ένα Applet το οποίο παίρνει ως παραμέτρους από τη σελίδα HTML τις συμβολοσειρές: “A simple Applet” και “that prints text”. Το Applet εκτυπώνει την πρώτη συμβολοσειρά με χρώμα μπλε και τη δεύτερη με χρώμα πράσινο.
7. Δημιουργήστε ένα applet στο οποίο παρουσιάζεται ένα ορθογώνιο με διαστάσεις 100x150. Στο applet να υπάρχει ένα πλήκτρο με τίτλο “Fill Rectangle” τα οποία, όταν επιλεγεί, χρωματίζει το εσωτερικό του ορθογωνίου. Όταν το ορθογώνιο χρωματιστεί το πλήκτρο “Fill Rectangle” γίνεται ανενεργό.
8. Στην κεφαλή μιας σελίδας HTML συμπληρώστε τα ακόλουθα στυλ:
 - οι επικεφαλίδες τύπου 1 (heading 1) να εμφανίζονται με κόκκινο χρώμα
 - οι επικεφαλίδες τύπου 2 (heading 2) υπογραμμισμένες και
 - οι παράγραφοι να εμφανίζονται με πλάγια γράμματα και στοιχισμένες στο κέντρο.Συμπληρώστε κάποιες επικεφαλίδες και παραγράφους στο έγγραφο HTML και παρατηρήστε τον τρόπο εμφάνισής τους
9. Δημιουργήστε ένα εξωτερικό φύλλο μορφοποίησης το οποίο όταν κληθεί από μία σελίδα HTML θα μετατρέπει το χρώμα του φόντου σε γκρι και θα αλλάζει τη μορφή των επικεφαλίδων τύπου 1 και 2. Οι επικεφαλίδες τύπου 1 θα έχουν χρώμα μπλε και μέγεθος γραμμάτων το 120%, αυτού που προσδιορίζεται από το φυλλομετρητή, ενώ οι επικεφαλίδες τύπου 2 θα έχουν γράμματα τύπου “courier”, με μέγεθος 25pt και χρώμα πράσινο. Μια γραμμή τύπου hr στη σελίδα θα έχει πάχος 5pt, χωρίς σκιά και χρώμα κόκκινο. Τέλος μία παράγραφος θα έχει μέγεθος γραμμάτων 10pt και αριστερό περιθώριο 20 px.