



ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ (Τ.Ε.Ι.) ΔΥΤΙΚΗΣ ΕΛΛΑΔΑΣ

ΣΧΟΛΗ ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΟΙΚΟΝΟΜΙΑΣ

Τμήμα Διοίκησης Επιχειρήσεων (Πάτρα)

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΑΝΑΛΥΣΗ ΚΑΙ ΣΧΕΔΙΑΣΜΟΣ ΕΦΑΡΜΟΓΩΝ ΣΕ HTML5

- ΔΡΙΜΗΣ ΑΝΑΣΤΑΣΙΟΣ
- ΒΕΛΛΙ ΓΚΕΣΙΑΝΑ
- ΜΠΕΣΗ ΑΦΡΟΥΛΑ

- ΕΠΟΠΤΕΥΩΝ ΚΑΘΗΓΗΤΗΣ: ΠΑΠΑΛΟΠΟΥΛΟΣ ΔΗΜΗΤΡΙΟΣ

ΠΑΤΡΑ - 2016

ΠΡΟΛΟΓΟΣ

Η συγκεκριμένη εργασία αποτέλεσε τον καρπό των κόπων και των τριών εκπονητών σε ένα αντικείμενο το οποίο αποτελούσε από τις πρώτες μέρες της ένταξης μας στο χώρο της τριτοβάθμιας εκπαίδευσης έναν διακαή πόθο τριβής. Πέρα από τις γνώσεις και τις κατευθύνσεις που μας παρείχε το ίδρυμα μας σε περισσότερο θεωρητικές κατευθύνσεις, το ενδιαφέρον μας προς περισσότερο πρακτικές μεθόδους όχι μόνο δεν εξανεμίστηκε, αλλά αντιθέτως η παρούσα εργασία πιστοποιεί ακριβώς το αντίθετό. Η ενασχόληση άλλωστε με αντικείμενα της σύγχρονης τεχνολογίας αποτελεί βασικό και αναπόσπαστο στοιχείο μιας επιτυχημένης ακαδημαϊκής αλλά και επαγγελματικής πορείας. Σε αυτό το σημείο θα θέλαμε να ευχαριστήσουμε τόσο τον επιβλέποντα μας για την καθοδήγηση που μας έδωσε, όσο και τους γονείς μας και τους φίλους μας που στάθηκαν στο πλάι μας κατά τη διάρκεια των φοιτητικών μας ετών.

ΠΕΡΙΛΗΨΗ

Στα πλαίσια της συγκεκριμένης πτυχιακής εργασίας, μελετήθηκε ο τομέας τη σχεδίασης ιστοσελίδων και επιμέρους εφαρμογών, τόσο ως προς το θεωρητικό κομμάτι όσο και ως προς το πρακτικό, εστιάζοντας κυρίως στην περίπτωση της εφαρμογής των ιδιαιτεροτήτων της τεχνολογίας HTML5. Η προσέγγιση που ακολουθήθηκε ήταν αρχικά σε θεωρητικό επίπεδο για τη μελέτη της εξέλιξης των δικτύων επικοινωνίας και των βασικών πρωτοκόλλων. Στη συνέχεια, μελετήθηκαν τα βασικά στοιχεία της HTML, συμπεριλαμβανομένων και των άλλων τεχνολογιών με τις οποίες αυτή αλληλοεπιδρά (CSS, JAVASCRIPT). Τέλος, παρουσιάστηκαν οι καινοτομίες της HTML5 και υλοποιήθηκε μία εφαρμογή βασισμένη σε αυτές.

ABSTRACT

In this work, the scientific domain of designing websites and applications was studied in depth. The two aspects that we examined was both the theoretical and the practical, and we mainly focused on the case of HTML5 technology and its properties. Firstly, the theoretical level consists of the study of the development of the communication networks and the basic protocols. To continue with, the basic components and the related theory of HTML5 were reviewed, including other technologies with which HTML5 interacts, such as CSS and JAVASCRIPT. Finally, the innovations of HTML5 were presented along with an illustrative implementation for highlighting its main differences with previous versions.

Περιεχόμενα

ΠΡΟΛΟΓΟΣ	3
ΠΕΡΙΛΗΨΗ	4
ABSTRACT.....	5
ΚΕΦΑΛΑΙΟ 1: Εξέλιξη Δικτύου και Διαδικτύου Κατεπέκταση Παγκοσμίου Ιστού.....	8
1.1 Εισαγωγή.....	8
1.2 Ιστορική Αναδρομή.....	9
1.3 Αρχιτεκτονικές Δικτύων	14
1.4 Πρωτόκολλα Δικτύων TCP/IPκαι OSI	21
1.5 World Wide Web & Client-Server Μοντέλο	25
1.6 Web Services.....	26
1.7 Web Application	27
ΚΕΦΑΛΑΙΟ 2:Παρουσίαση της γλώσσας HTML.....	30
2.1 Εισαγωγή	30
2.2 Η ιστορία της γλώσσας.....	30
2.3 Τα βασικά στοιχεία της HTML	31
2.4 Γλώσσα μορφοποίησης CSS.....	43
2.5 Η γλώσσα Javascript.....	46
2.6 Ανάλυση και ανάπτυξη Ιστοσελίδων.....	55
ΚΕΦΑΛΑΙΟ 3:Καινοτομίες της HTML5.....	58
3.1 Από την HTML στην HTML5	58
3.2 Καινούργια στοιχεία της HTML5.....	58
3.3 Canvas & SVG.....	61
3.4 Βίντεο και Ήχος.....	62
3.5 Αποθήκευση Πληροφοριών	63
3.6 Η CSS3.....	64
ΚΕΦΑΛΑΙΟ 4:Εφαρμογή της HTML5.....	68
4.1 Εισαγωγή	68
4.2 Η βασική δομή της εφαρμογής	68
4.3 Στοιχεία εισόδου	71
4.4 Γραφικά εφαρμογής.....	72
4.5 Ήχος.....	73
4.6 Μορφοποίηση εφαρμογής.....	74
4.7 JavaScript, JQuery & localStorage	77
Βιβλιογραφία	79
Παράρτημα.....	80

Πίνακας Εικόνων

Εικόνα 1:Το πρώτο πειραματικό δίκτυο	10
Εικόνα 2:Η εξάπλωση του Arpanet	10
Εικόνα 3:Το NSFNET	11
Εικόνα 4: Η διαχρονική εξέλιξη του NSFNET ως προς τον όγκο δεδομένων που μετέφερε. 12	
Εικόνα 5: Το διαδίκτυο σήμερα.....	13
Εικόνα 6: Παράδειγμα ακτινωτού δικτύου.....	15
Εικόνα 7: Παράδειγμα δικτύου βρόχου.....	16
Εικόνα 8: Παράδειγμα κομβικού δικτύου.	17
Εικόνα 9: Παράδειγμα δικτύου μεταγωγή πακέτων.....	18
Εικόνα 10: Παραδείγματα δικτύων ακρόασης.....	19
Εικόνα 11: Δομικές διαφορές HTML και HTML5	60
Εικόνα 12: Πρότυπο εικόνας για χρήση λογοτύπου.....	72
Εικόνα 13: Διεπαφή εφαρμογής Blog σε HTML5.....	78

ΚΕΦΑΛΑΙΟ 1: Εξέλιξη Δικτύου και Διαδικτύου Κατεπέκταση Παγκοσμίου Ιστού

1.1 Εισαγωγή

Τα τελευταία χρόνια η ανάπτυξη των Τηλεπικοινωνιών και της Πληροφορικής παγκοσμίως επιτυγχάνεται με ραγδαίους ρυθμούς. Ο κυριότερος παράγοντας που έχει συμβάλει προς αυτή την κατεύθυνση είναι οι σύγχρονες κοινωνικές απαιτήσεις, οι οποίες αυξάνονται συνέχεια με βασικό άξονα την υψηλή ταχύτητα και την αξιοπιστία σε τεχνικό επίπεδο.

Για την υλοποίηση όλων αυτών των υπηρεσιών που κατακλύζουν την καθημερινότητα μας και την εξυπηρέτηση των ανθρώπινων αναγκών και γενικότερα της κοινωνίας, πραγματοποιήθηκαν πολλές ενδιάμεσες βελτιώσεις και εφευρέσεις σταθμοί για την επιστημονική κοινότητα, επιτρέποντας πλέον στη σημερινή εποχή σε έναν απλό χρήστη, π.χ. οικιακό χρήστη, να εκτελεί τις περισσότερες υποχρεώσεις του με ελάχιστο κόπο, ύψιστη ασφάλεια και ταχύτητα από το χώρο εργασίας του ή/ και τον ίδιο τον οικιακό του χώρο. Η ενσωμάτωση άλλωστε των ψηφιακών υπηρεσιών από τις περισσότερες εταιρίες, τράπεζες, δημόσιους και εθνικούς φορείς, καθιστούν ορισμένες φορές την πρόσβαση σε αυτές υποχρεωτικές, ωθώντας με αυτές τις ενέργειες το κοινό να στραφεί προς τον νέο ψηφιακό κόσμο.

Το πιο απλό παράδειγμα ανάδειξης της αναγκαιότητας των δικτύων και της εξέλιξης των εκάστοτε τεχνολογιών σε όλο και καλύτερες διατάξεις αποτελεί το παράδειγμα του τηλεφωνικού δικτύου. Όταν ο Γράχαμ Μπελ έθεσε για πρώτη φορά σε πρακτική εφαρμογή το τηλέφωνο το 1876, συνομιλούσε με ένα και μόνο οικείο άτομο μέσω δύο τηλεφωνικών συσκευών και μίας απλής γραμμής. Όταν ο κοινωνικός του περίγυρος ενημερώθηκε για αυτή την πρωτοτυπία, αμέσως ζήτησε την παροχή αυτής της υπηρεσίας και για τους εαυτούς τους.

Κάπως έτσι όμως γεννήθηκε η ανάγκη για τη δημιουργία ενός Δικτύου(Jean, 1997). Η λύση του προβλήματος που προέκυψε πέρασε από αρκετά στάδια μέχρι να φτάσει σε μία πιο ολοκληρωμένη μορφή και αρκετά οικεία με τη σύγχρονη μορφή. Τα πρώτα τηλεφωνικά κέντρα συνδεόντουσαν ακτινωτά με μια αφιερωμένη γραμμή και μία συσκευή. Οι υπάλληλοι των κέντρων αυτών συνέδεαν τη γραμμή του καλούντος με την αντίστοιχη του καλούμενου μέσω κατάλληλα διαμορφωμένων βυσμάτων. Ουσιαστικά, αυτό το σύστημα αποτέλεσε την πρώτη μορφή τηλεφωνικού δικτύου για σήματα φωνής. Στα στάδια που ακολούθησαν, η μορφή των τηλεφωνικών κέντρων εξυγρονίστηκε και προόδευσε με την ανάπτυξη ηλεκτρομαγνητικών κέντρων, αλλά και τη χρήση της αυτόματης επιλογής. Παρόμοια, περνώντας οι δεκαετίες φτάσαμε στη σημερινή εποχή όπου ο κυρίαρχος ρόλος του Ηλεκτρονικού Υπολογιστή δε θα μπορούσε από το να αναπροσαρμόσει και το τηλεφωνικό δίκτυο, οδηγώντας σε υπερσύγχρονα τηλεφωνικά κέντρα για τις ενσύρματες ζεύξεις και δορυφορικά συστήματα για τις ασύρματες επικοινωνίες. Οι τρόποι μετάδοσης έχουν πλέον αυτοματοποιηθεί και πολλές νέα δυνατότητες παρέχονται ψηφιακά από το τηλεφωνικό δίκτυο.

Παρόμοια διαδρομή ακολούθησαν και άλλα δίκτυα, όπως το δίκτυο δεδομένων. Σήμερα, τα σύγχρονα δίκτυα δεν απαιτούν πολλαπλές συνδέσεις μεταξύ των συνδρομητών ενός δικτύου, αλλά η κάθε οντότητα συνδέεται με το πλησιέστερο τηλεπικοινωνιακό κέντρο ή

δίκτυο. Έχοντας σαν βασικό άξονα αυτή την συνδεσμολογία, δημιουργήθηκαν τα ευρέως γνωστά ιδιωτικά και δημόσια δίκτυα, όπως(Tanenbaum, 2003):

- το ISDN,
- το ίντερνετ,
- το Ethernet,
- το ασύρματο LAN: 802.11 και άλλα.

Οι μεταδόσεις των σημάτων ομιλίας, μουσικής, εικόνων, δεδομένων καθώς και η επικοινωνία υπολογιστικών συστημάτων πλέον ολοκληρώνονται μέσω ψηφιακών δικτύων υψηλών ταχυτήτων βασιζόμενα σε πρωτόκολλα όπως το ATM και το IP.

Οι ασύρματες επικοινωνίες επίσης διευκόλυναν την πρόσβαση πολλαπλών φορητών συσκευών σε δημόσια και ιδιωτικά δίκτυα, εξυπηρετώντας την ελευθερία κίνησης του χρήστη και την αποδέσμευση του από τον τοπικό παράγοντα ή το κριτήριο του χώρου. Ως αποτέλεσμα, τα δίκτυα δεδομένων να αντικαθιστούν πλήρως τις υπηρεσίες των κλασσικών τεχνολογιών, στη συντριπτική πλειοψηφία των σύγχρονων πόλεων.

Με την εμφάνιση οποιουδήποτε νέου μέσου επικοινωνίας, η κοινωνία και οι δομές της αναμφισβήτητα επηρεάζονται(Jean, 1997). Η επίδραση αυτή πηγάζει κυρίως από την τεχνολογία του νέου μέσου. Είναι πλέον εμφανές πως το διαδίκτυο, ως ένα από τα πλέον πιο σημαντικά μέσα επικοινωνίας στον σύγχρονο κόσμο έχει διεισδύσει σε κάθε γωνιά της υφηλίου, δημιουργώντας εικονικούς "χώρους" καθώς και εικονικές "κοινότητες", όπου παύουν να υφίστανται οι γεωγραφικές, κοινωνικές και πολιτιστικές διαχωριστικές γραμμές που υπάρχουν στον πραγματικό κόσμο και που τα παραδοσιακά μέσα επικοινωνίας αδυνατούν να ξεπεράσουν εύκολα. Με την δημιουργία του διαδικτύου, η επικοινωνία έγινε άμεση και αμφίδρομη. Οι αποστάσεις εκμηδενίστηκαν και δόθηκε η δυνατότητα σε κάθε χρήστη ηλεκτρονικού υπολογιστή συνδεδεμένου στο Διαδίκτυο, να πληροφορηθεί αλλά και να πληροφορήσει ανταλλάσσοντας απόψεις μέσω ενός συμμετοχικού διαύλου επικοινωνίας. Οι χρήστες αποκτούν ολοένα και περισσότερο την ιδιότητα του παγκόσμιου πολίτη. Είναι εμφανείς λοιπόν οι λόγοι για τους οποίους το διαδίκτυο θεωρείται ένα από τα πιο δημοκρατικά σύγχρονα μέσα μαζικής επικοινωνίας καθώς δίνει στον μέσο άνθρωπο τη δυνατότητα πρόσβασης σε μεγάλο όγκο πληροφοριών συγκεντρωμένων σε ένα "χώρο" και την δυνατότητα της προσωπικής επιλογής των πληροφοριών αυτών.

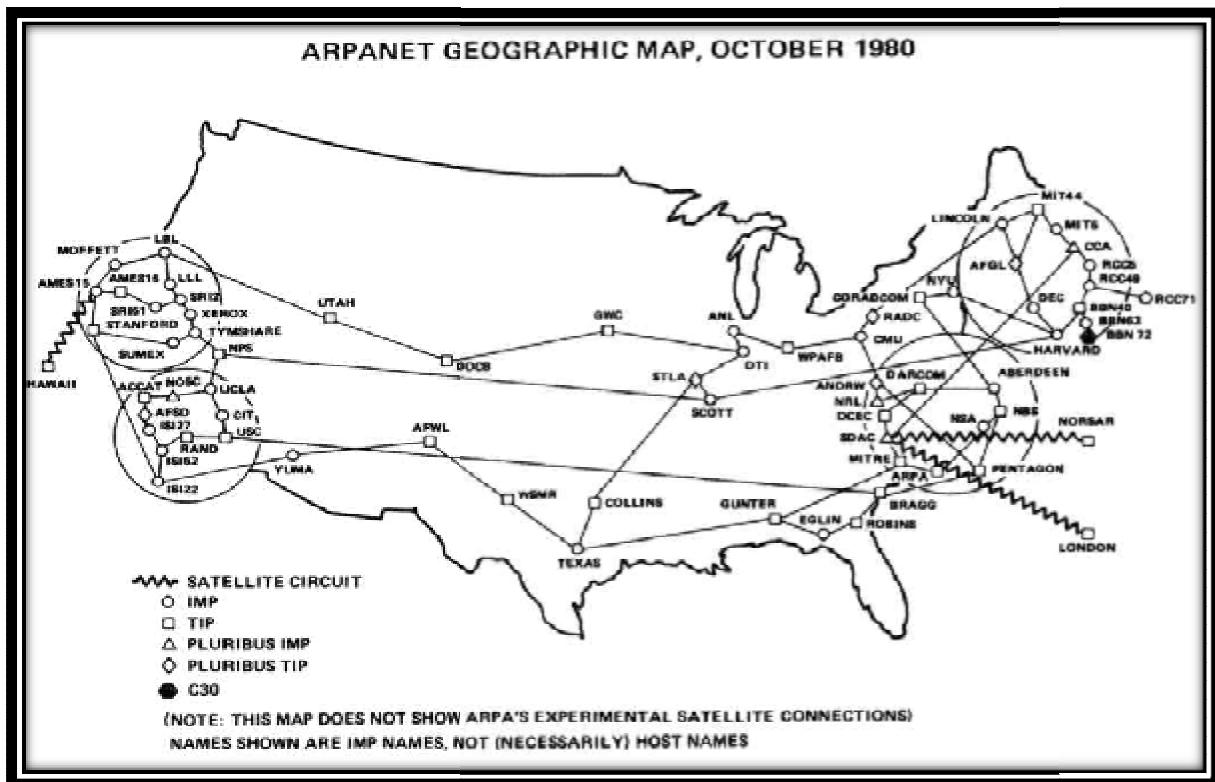
1.2 Ιστορική Αναδρομή

Το διαδίκτυο ξεκίνησε ως ένα πείραμα στα τέλη του 1960 από την ARPA (Advanced Research Projects Agency) του Υπουργείου Άμυνας των Ηνωμένων Πολιτειών. Το Δεκέμβρη του 1969 λειτούργησε για πρώτη φορά ένα πειραματικό δίκτυο, με την ονομασία ARPANET, με τη σύνδεση 4 κόμβων μέσω γραμμών ταχύτητας 56kpbs. Η νέα αυτή τεχνολογία αποδείχτηκε πολύ επιτυχής και οδήγησε στη δημιουργία δύο άλλων παρόμοιων στρατιωτικών δικτύων, του MILNET στις Η.Π.Α και του MINET στην Ευρώπη. Χιλιάδες υπολογιστές υπηρεσίας και χρήστες σύνδεσαν τα δίκτυα τους στο ARPANET δημιουργώντας έτσι το ARPA-Ίντερνετ. Στις εικόνες 1 και 2 βλέπουμε το ARPANET τις πρώτες μέρες λειτουργίας του, από την εγκαινίαση του το 1969 έως την εξάπλωσή του το 1976 (Jean, 1997).

Ο συνδυασμός ακαδημαϊκών και κυβερνητικών δικτύων με τον κορμό του ARPANET ήταν η αρχή αυτού που σήμερα γνωρίζουμε ως Ίντερνετ, και στο εξής θα αναφέρεται ως διαδίκτυο. Σύντομα η αυξανόμενη χρήση του ARPANET λόγω της χρηστικότητας του οδήγησε σε προβλήματα κλιμάκωσης με πιο προφανές την υπερφόρτωση των γραμμών, παρόλο που η χρήση του για εμπορικούς λόγους ήταν απαγορευμένη. Σαν αποτέλεσμα η National Science Foundation (NSF) ξεκίνησε την ανάπτυξη του NSFNET. Το ARPANET αποσύρθηκε το 1989.

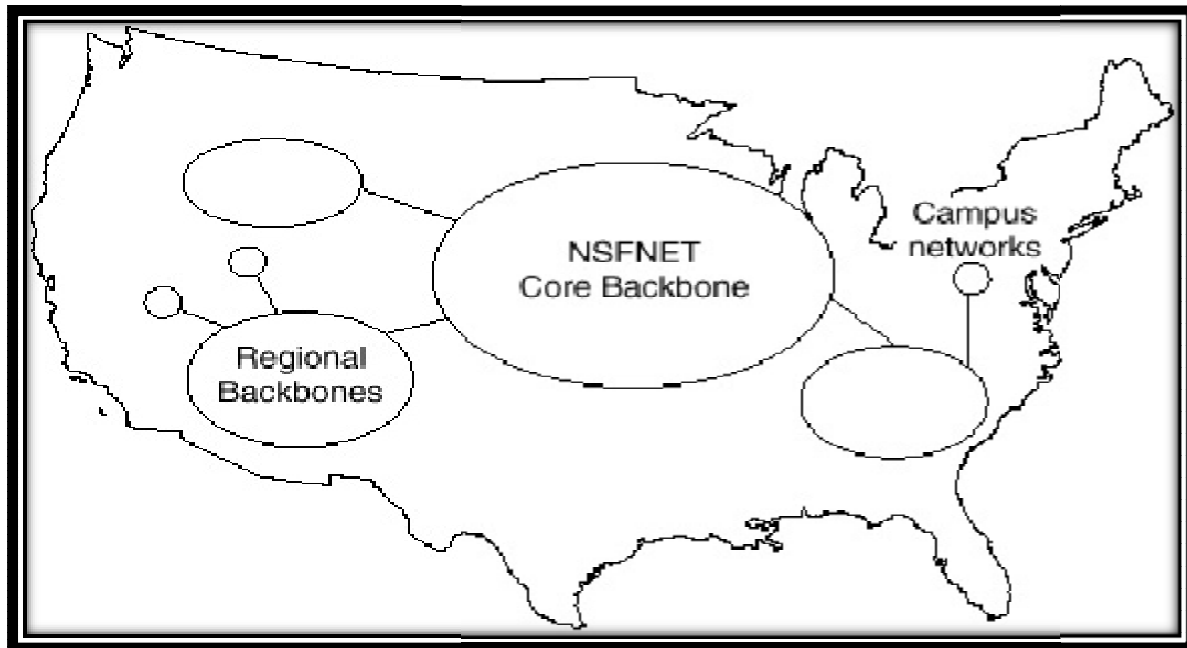


Εικόνα 1: Το πρώτο πειραματικό δίκτυο



Εικόνα 2: Η εξάπλωση του Arpanet

Το NSFNET, στην πρώτη του μορφή το 1986, εισήγαγε μία πιο κατανομημένη αρχιτεκτονική τριών επιπέδων. Η αρχιτεκτονική αυτή συνέδεε πανεπιστήμια, ερευνητικούς οργανισμούς και δίκτυα ομότιμων κόμβων σε ένα κύριο κορμό ο οποίος συνέδεε έξι κέντρα υπερ-υπολογιστών τα οποία χρηματοδοτούνταν από το κράτος. Οι αρχικές ζεύξεις των 56kbps αναβαθμίστηκαν το 1988 σε ταχύτερες ζεύξεις τύπου T1 με ταχύτητα 1.544 Mbps. Ο κορμός NSFNET T1 συνέδεε επτά διαδικτυακούς χώρους, οι οποίες ήταν τα BARRNET, MidNet, Westnet, NorthWestNet, SESQUINET, SURAnet, NCAR (National Center for Atmospheric Research), καθώς και πέντε υπερυπολογιστικά κέντρα(Tanenbaum, 2003) Στην εικόνα 3.φαίνεται το αυθεντικό NSFNET.

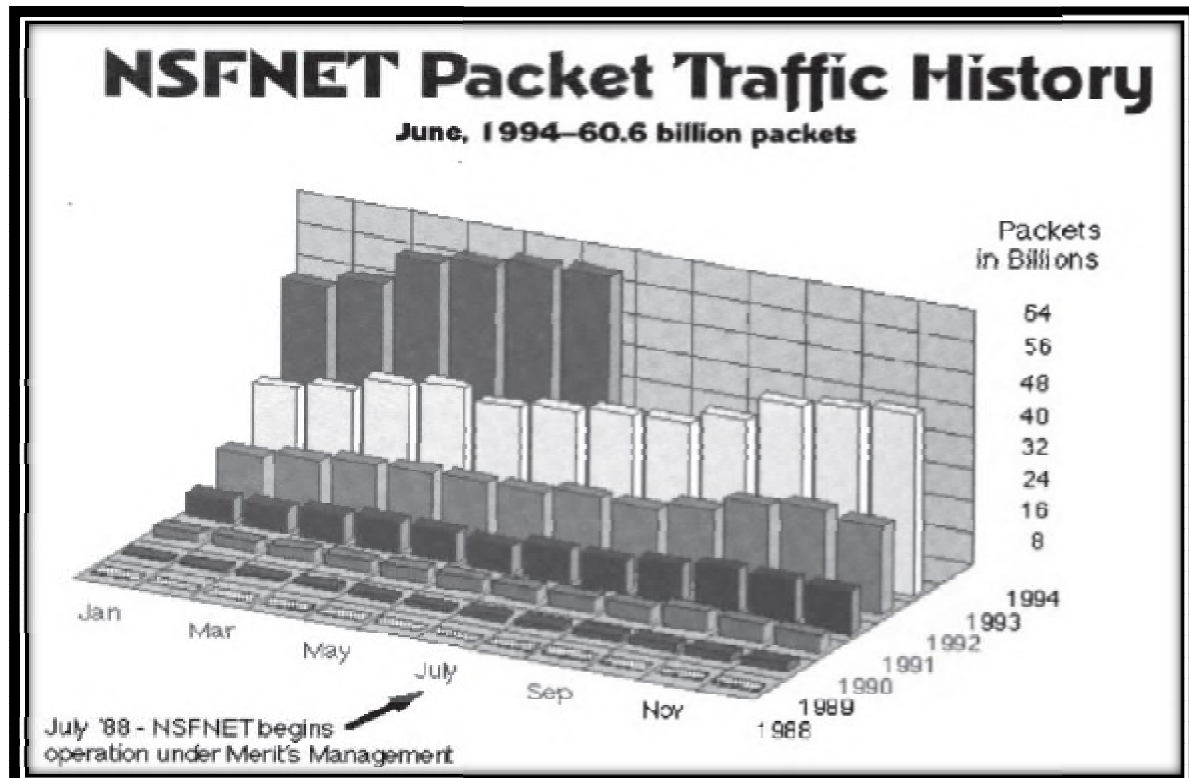


Εικόνα 3:Το NSFNET

Το 1990 οι εταιρείες Merit, IBM και MCI δημιούργησαν ένα νέο οργανισμό γνωστό ως Advanced Network and Services (ANS). Το τμήμα του Internet engineering της Merit παρείχετο 1990 μία βάση δεδομένων με χρήσιμα δεδομένα για την υλοποίηση: α) πολιτικών και κανόνων δρομολόγησης και β) υπηρεσιών διαχείρισης του NSFNET. Κάνοντας χρήση των υποδομών αυτών, ο ANS ήταν σε θέση να λειτουργήσει τους δρομολογητές κορμού σε ένα Network Operation Center (NOC). Μέχρι το 1991 το φορτίο του δικτύου είχε αυξηθεί τρομερά οπότε το δίκτυο κορμού αναβαθμίστηκε με χρήση ζεύξεων τύπου T3 ταχύτητας 45 Mbps(Jean, 1997).

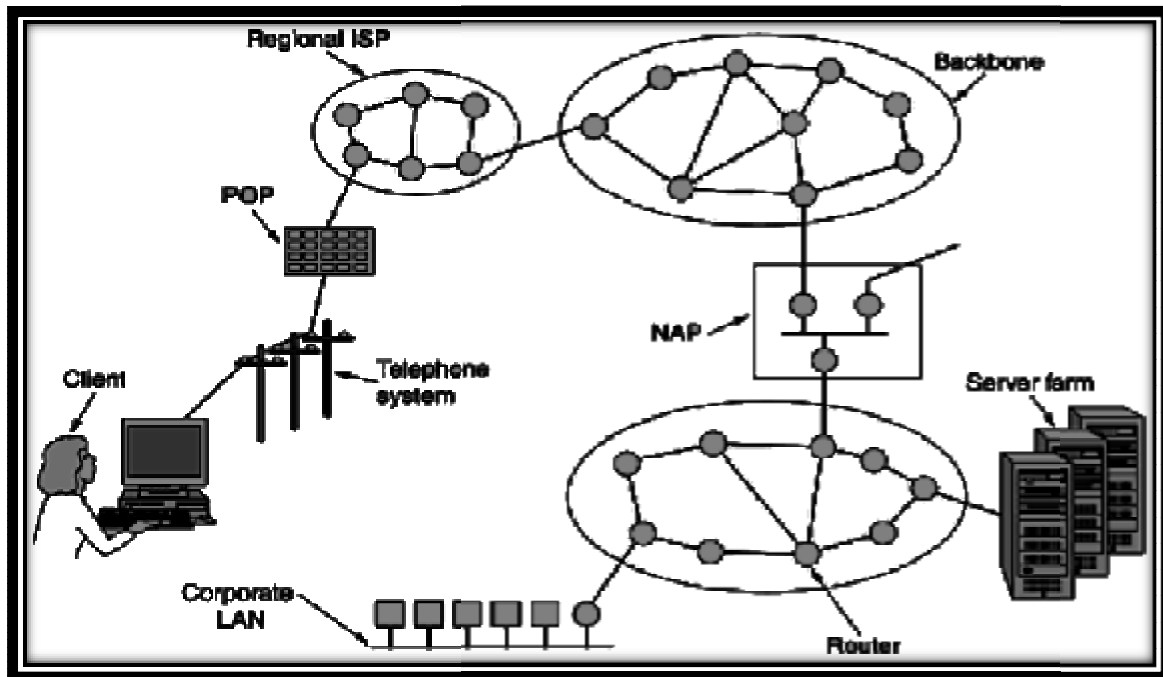
Η ραγδαία ανάπτυξη του NSFNET και το αυξανόμενο πλήθος των χρηστών του τράβηξε το ενδιαφέρον πολλών κερδοσκοπικών οργανισμών. Ωστόσο η πολιτική χρήσης του επέτρεπε στους κερδοσκοπικούς οργανισμούς να επενδύσουν μόνο στην έρευνα και την εκπαίδευση μέσω του διαδικτύου και απέκλειε τον διαμοιρασμό προσωπικών δεδομένων, τις δωρεές και τις διαφημίσεις. Αυτό είχε σαν αποτέλεσμα να δημιουργηθούν κάποιοι ανεξάρτητοι Πάροχοι Υπηρεσιών Διαδικτύου (ISP) οι οποίοι είχαν εμπορικό προσανατολισμό αλλά είχαν και τη δυνατότητα να συνδέονται με το κορμό του NSFNET αλλά και μεταξύ τους για να ανταλλάσσουν δεδομένα απαλλαγμένα από τους όρους χρήσης του NSFNET.

Όπως φαίνεται από την εικόνα 4, η τεράστια επιτυχία του NSFNET οδήγησε στο συμπέρασμα ότι ο κρατικά επιδοτούμενος κορμός θα έπρεπε να δώσει τη θέση του σε ιδιωτικά συνδρομητικά δίκτυα στα οποία οι χρήστες θα είχαν πρόσβαση καταβάλλοντας συγκεκριμένες μηνιαίες συνδρομές βάσει συμβολαίων, όπως ακριβώς γίνεται με όλα τα συνδρομητικά δίκτυα (π.χ. τηλεφωνία)(Γεώργιος Αλεξόπουλος, 2015).



Εικόνα 4: Η διαχρονική εξέλιξη του NSFNET ως προς τον όγκο δεδομένων που μετέφερε.

Στην εικόνα 5 δείχνεται μία απλουστευμένη δομή του διαδικτύου όπως είναι σήμερα, στην οποία η πρόσβαση του οικιακού πελάτη γίνεται μέσω κλασικής αναλογικής γραμμής που τερματίζει σε ADSL Modems(Tanenbaum, 2003). Το τερματικό του οικιακού πελάτη παράγει ψηφιακά σήματα τα οποία αφού διαμορφωθούν κατάλληλα μεταφέρονται μέσω της τηλεφωνικής γραμμής σε ένα σημείο συγκέντρωσης. Τα σημεία αυτά αναφέρονται σαν σημεία παρουσίας (Point of Presence POP), όπως για παράδειγμα είναι οι πολυπλέκτες DSLAM που χρησιμοποιούνται σήμερα στο Ελληνικό δίκτυο. Το POP είναι αποτελεί ουσιαστικά το σημείο πρόσβασης του κάθε οικιακού χώρου (site) στο διαδίκτυο. Το κάθε POP έχει τη δική του IP διεύθυνση(Γεώργιος Αλεξόπουλος, 2015). Ο κάθε ISP διαθέτει διάσπαρτα τέτοια POPs σε όλο το γεωγραφικό χώρο στον οποίο εκτείνονται οι οικιακοί πελάτες του (στην ουσία ο αριθμός των POPs που διαθέτει ο ISP καθορίζει και το επιχειρησιακό μέγεθός του).



Εικόνα 5: Το διαδίκτυο σήμερα

Όλα τα POPs όλων των ISP συνδέονται σε κεντρικά ψηφιακά δίκτυα κορμού που διαθέτουν Κεντρικούς Κόμβους Δρομολόγησης του παραγόμενου φορτίου. Επειδή το φορτίο μεταφέρεται χρησιμοποιώντας το payload των IP πακέτων, η δρομολόγηση στους κόμβους γίνεται με τη χρήση δικτύων μεταγωγής πακέτου. Το ψηφιακό δίκτυο κορμού δρομολογεί την κίνηση που παράγεται από τα POPs του κάθε ανεξάρτητου ISP μιας ευρύτερης περιοχής (π.χ. μία πόλη) σε ένα συγκεκριμένο σύνολο κόμβων δρομολόγησης που διαθέτει ο συγκεκριμένος ISP στην περιοχή αυτή για να εξυπηρετήσει τους πελάτες του. Οι κόμβοι αυτοί αποτελούν το Περιφερειακό Δίκτυο του ISP. Τα πακέτα της περιοχής που δημιουργούνται και απευθύνονται σε πελάτες του ίδιου ISP εξυπηρετούνται αποκλειστικά και μόνο από το Περιφερειακό Δίκτυο του ISP. Στην περίπτωση όπου οι πελάτες ανήκουν σε διαφορετικούς ISPs, οι Κεντρικοί Κόμβοι Δρομολόγησης αναλαμβάνουν (σε ένα ανώτερο επίπεδο δρομολόγησης) να μετάγουν τα πακέτα στα Περιφερειακά Δίκτυα των εμπλεκόμενων ISPs.

Στην κορυφή αυτής της αλυσίδας μεταγωγής βρίσκονται μεγάλες εξειδικευμένες εταιρείες που διαχειρίζονται τα πολύ μεγάλα διεθνή δίκτυα κορμού με χιλιάδες δρομολογητές που συνδέονται με οπτικές ίνες υψηλού εύρους ζώνης (Tanenbaum, 2003). Οι ιστότοποι όλων των μεγάλων εταιρειών που διαθέτουν τεράστιο αριθμό χρηστών, αλλά και οι ιστοσελίδες των μικρότερων εταιρειών γενικότερα στεγάζονται στις εγκαταστάσεις συγκεκριμένων εταιρειών που δρουν ως Πάροχοι Φιλοξενίας Ιστοτόπων και Υπηρεσιών (Host Service Providers) των οποίων οι διακομιστές συνδέονται κατευθείαν στα ανωτέρω διεθνή δίκτυα κορμού (Γεώργιος Αλεξόπουλος, 2015).

Τέλος, επειδή υπάρχουν πολλά διαφορετικά δίκτυα κορμού στον κόσμο ένα πακέτο μπορεί να χρειαστεί να κινηθεί μέσω ενός ανταγωνιστικού δικτύου και για αυτό όλα τα μεγάλα δίκτυα κορμού συνδέονται στα NAP (Network Access Point). Κατά βάση, το NAP είναι ένα δωμάτιο (Room) με τουλάχιστον ένα δρομολογητή για κάθε δίκτυο κορμού. Ένα LAN μέσα στο δωμάτιο συνδέει όλους τους δρομολογητές, έτσι ώστε τα πακέτα να μπορούν να προωθηθούν από οποιοδήποτε δίκτυο κορμού σε οποιοδήποτε άλλο. Εκτός από τη

διασύνδεση στα NAP, τα δίκτυα κορμού έχουν και πολλές άμεσες συνδέσεις, μία τεχνική που είναι γνωστή ως «ιδιωτική ομότιμη σύνδεση» (Jean, 1997).

Το παγκόσμιο δίκτυο για όλους είναι πλέον γεγονός. Όλο και περισσότερες χώρες συνδέονται στο NSFNet, μεταξύ των οποίων και η Ελλάδα το 1990. Το 1993 το εργαστήριο CERN στην Ελβετία παρουσιάζει το World Wide Web (WWW), ή διαφορετικά τον Παγκόσμιο Ιστό, που αναπτύχθηκε από τον Tim Berners-Lee. Πρόκειται για ένα σύστημα διασύνδεσης πληροφοριών σε μορφή πολυμέσων (multimedia) που βρίσκονται αποθηκευμένες σε χιλιάδες υπολογιστές του Internet σε ολόκληρο τον κόσμο και παρουσίασης τους σε ηλεκτρονικές σελίδες, στις οποίες μπορεί να περιηγηθεί κανείς χρησιμοποιώντας το ποντίκι. Το γραφικό αυτό περιβάλλον έκανε την εξερεύνηση του Internet προσιτή στον απλό χρήστη. Παράλληλα, εμφανίζονται στο Internet διάφορα εμπορικά δίκτυα που ανήκουν σε εταιρείες παροχής υπηρεσιών Internet (Internet Service Providers – ISP) και προσφέρουν πρόσβαση στο Internet για όλους. Η ανακάλυψη του WWW σε συνδυασμό με την ευκολία απόκτησης πρόσβασης στο Internet προσέελκυσε έναν μεγάλο αριθμό καινούργιων χρηστών και έφερε την “έκρηξη” που παρακολούθησαμε τα τελευταία χρόνια.

1.3 Αρχιτεκτονικές Δικτύων

Όπως έχει ήδη αναφερθεί, ο όρος δίκτυα αποτελεί μία πολύ γενικευμένη έννοια, με ποικίλη χρήση και πολλές δυνατές ερμηνείες. Στις επόμενες παραγράφους θα αναφερθούν και άλλες πτυχές αυτών που θα προσδιορίσουν με μεγαλύτερη σαφήνεια τα δίκτυα πληροφορικής που αποτελούν το κύριο κομμάτι της εργασίας. Σε αυτό το χωρίο θα δοθούν ορισμένα γενικότερα κριτήρια διαχωρισμού αυτών με βάση το γεωγραφικό κριτήριο, το τοπολογικό και την τεχνική προώθησης (Tanenbaum, 2003). Επομένως έχουμε τις εξής κατηγορίες:

1. Γεωγραφικό κριτήριο

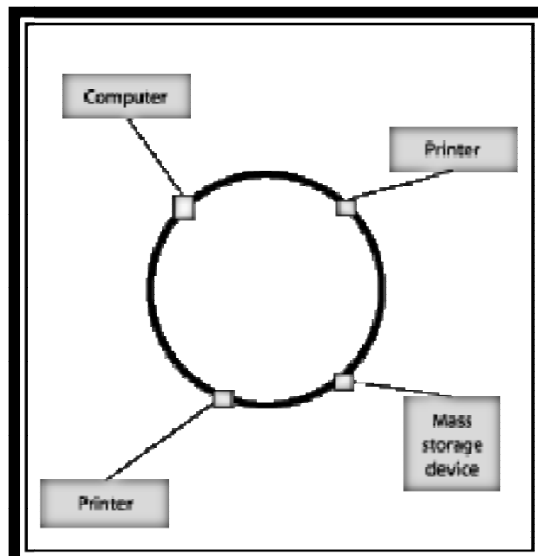
- **Δίκτυα Ευρείας Περιοχής (Wide Area Network – WAN)** – αποτελούνται από ένα σύνολο υπολογιστών, τερματικών, τηλεπικοινωνιακών συσκευών, τηλεπικοινωνιακών γραμμών και συνδέσεων, τα οποία εκτείνονται σε ευρεία γεωγραφική περιοχή, αστική και υπεραστική, φεύγοντας από τα στενά όρια ενός συγκεκριμένου χώρου. Χαρακτηριστικά παραδείγματα αποτελούν τα τραπεζικά δίκτυα, αυτά των αεροπορικών εταιριών, το ίντερνετ κτλ.
- **Τοπικά Δίκτυα (Local Area Networks – LAN)** – αποτελούνται από υπολογιστές, επικοινωνιακές συσκευές, διασυνδέσεις και γραμμές που βρίσκονται στον ίδιο γεωγραφικό χώρο. Το κυριότερο παράδειγμα με το οποίο οι περισσότεροι έχουμε αλληλοεπιδράσει είναι το Ethernet. Η κύρια διαφορά μεταξύ των LAN, τα οποία είναι σε γενικές γραμμές ταχύτατα, και των WAN δικτύων είναι οι διαφορετικές τεχνικές λειτουργίας. Στο LAN δίκτυο οι υπολογιστές είναι συνδεδεμένοι σε ένα κοινό μέσο μετάδοσης και χρησιμοποιούν διαφορετικά πρωτόκολλα επικοινωνίας, διαφορετικούς τύπους διασυνδέσεων (interfaces), και διαφορετικές ταχύτητες, μιας και κυμαίνονται στα πλαίσια των 10 bps – 1000Mbps, σε αντίθεση με τα WAN τα οποία

κυμαίνονται στα 900 bps – 2Mbps. Άλλωστε, τα δίκτυα τύπου WAN υποστηρίζουν κατά βάση επικοινωνίες; Μεταξύ τοπικών δικτύων.

- **Αστικά ή Μητροπολιτικά Δίκτυα (Metropolitan Area Networks – MAN)** – αποτελούν μία υποδιαίρεση δικτύων που πρωτοεμφανίστηκε το 1990 και τα οποία δε ξεπερνούν τα σύνορα μίας πόλης. Σε σχέση με τα τοπικά δίκτυα ξεπερνούν τους περιορισμούς σε ταχύτητα και ζώης έκτασης και χρησιμοποιούνται κυρίως για να διασυνδέουν αποδοτικά τοπικά δίκτυα εντός μίας πόλης. Σαν μέσο μετάδοσης χρησιμοποιείται η οπτική ίνα, επιτυγχάνοντας ρυθμούς μετάδοσης της τάξης των εκατοντάδων Mbps. Ορισμένα πρωτόκολλα που υποστηρίζουν δίκτυα αυτής της μορφής είναι τα FDDI, ATM, Metro Ethernet και MPLS.

2. Τοπολογικό κριτήριο

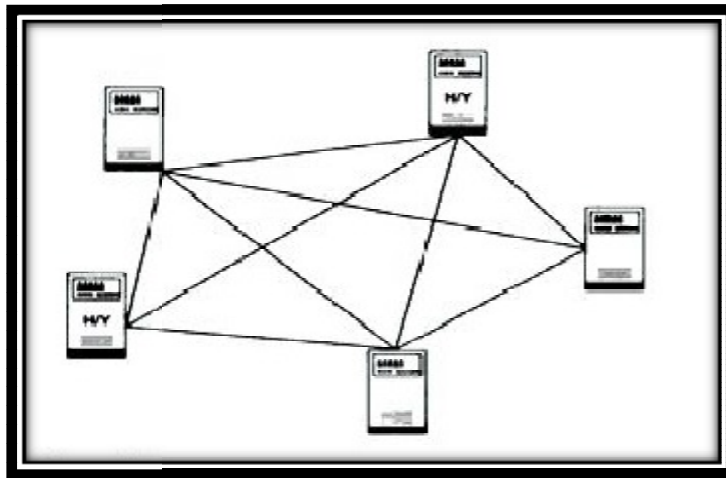
- **Ακτινωτό** – είναι τα δίκτυα στα οποία όλες οι περιφερειακές τερματικές συσκευές συνδέονται ακτινωτά με ένα κεντρικό σημείο. Οι επιμέρους συνδέσεις, όπως αυτές φαίνονται και στην παρακάτω εικόνα, ονομάζονται και point-to-point συνδέσεις. Τα πλεονεκτήματα αυτών των δικτύων είναι η εύκολη σχεδίαση και υποστήριξη, ο μικρός χρόνος απόκρισης αλλά και η εξαιρετική αξιοπιστία του. Από την άλλη πλευρά, το κυριότερο αρνητικό στοιχείο αποτελεί η απαραίτητη διέλευση όλων των επιμέρους διασυνδέσεων από τον κεντρικό κόμβο, γεγονός που προκαλεί συχνά υπερφόρτωση του δικτύου για μεγάλο πλήθος συμμετεχόντων. Για αυτόν ακριβώς το λόγο, αυτού του είδους τα δίκτυα συναντώνται κατεξοχήν σε μικρές και μεσαίες εφαρμογές, όπως στη σύνδεση τερματικών σταθμών με υπολογιστές, ή αποτελούν τμήματα μεγαλύτερων δικτύων (Γεώργιος Αλεξόπουλος, 2015).



Εικόνα 6: Παράδειγμα ακτινωτού δικτύου.

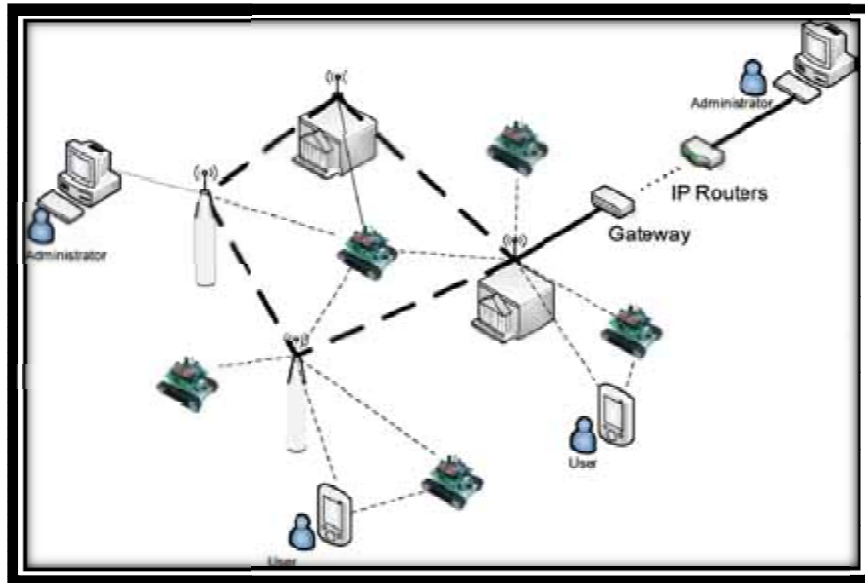
- **Δίκτυα Βρόχου (Mesh)** – τα δίκτυα βρόχου ή Mesh networks χαρακτηρίζονται από την ιδιότητα πως κάθε σταθμός είναι άμεσα

συνδεδεμένος με τους υπολοίπους με τοθλάχιστον δύο διαφορετικούς δρόμους προς αυτούς έτσι ώστε να κλείνουν βρόγχους. Η κύρια χρήση τους παρατηρείται στη διασύνδεση τηλεπικοινωνιακών κόμβων μεταξύ τους. Φυσικά η πολυπλοκότητα του δικτύου σύνδεσης αυξάνει σε αυτήν την περίπτωση, μιας και για ενωθούν πλήρως N σημεία, απαιτούνται $N*(N-1)/2$ συνδέσεις. Τα πλεονεκτήματα πάντως είναι μεγάλου ενδιαφέροντος, ειδικά για πρακτικές εφαρμογές, μιας και υπάρχει η δυνατότητα εναλλακτικής δρομολόγησης των μηνυμάτων σε περίπτωση διακοπής ή υπεφόρτωσης μίας σύνδεσης. Φυσικά, η χρήση του παρόντος δικτύου απαιτεί την ύπαρξη κόμβων που δύνανται να αποθηκεύσουν δεδομένα δρομολόγησης και αναδρομολόγησης για την ορθή λειτουργία εντός του δικτύου. Η παρακάτω εικόνα απεικονίζει ένα τυπικό δίκτυο βρόχου:



Εικόνα 7: Παράδειγμα δικτύου βρόχου.

- **Κομβικά Δίκτυα (Nodal)** – τα κομβικά δίκτυα, τα οποία ονομάζονται και ιεραρχικά, δημιουργούνται από τη σύνθεση πολλών ακτινωτών δικτύων σε ένα δίκτυο κορμού με κόμβους κατάλληλους για την δρομολόγηση μηνυμάτων. Λόγω μάλιστα αυτής της ύπαρξης έξυπνων κόμβων, πραγματοποιείται καλύτερη χρήση των γραμμών και παρέχονται λύσεις σε περιπτώσεις διακοπής κεντρικών γραμμών. Επιπλέον, υποβοηθάτε η λύση μετάδοσης χωρίς λαθή, μειώνοντας παράλληλα το κόστος των μισθωμένων τηλεπικοινωνιακών κυκλωμάτων. Στην εικόνα που ακολουθεί, βλέπουμε ένα κομβικό δίκτυο, του οποίου ο κορμός βασίζεται σε τοπολογία βρόχου με περιφερειακούς κόμβους, ενώ τα ακραία σημεία χρησιμοποιούν είτε ακτινωτά δίκτυα είτε σειριακά:

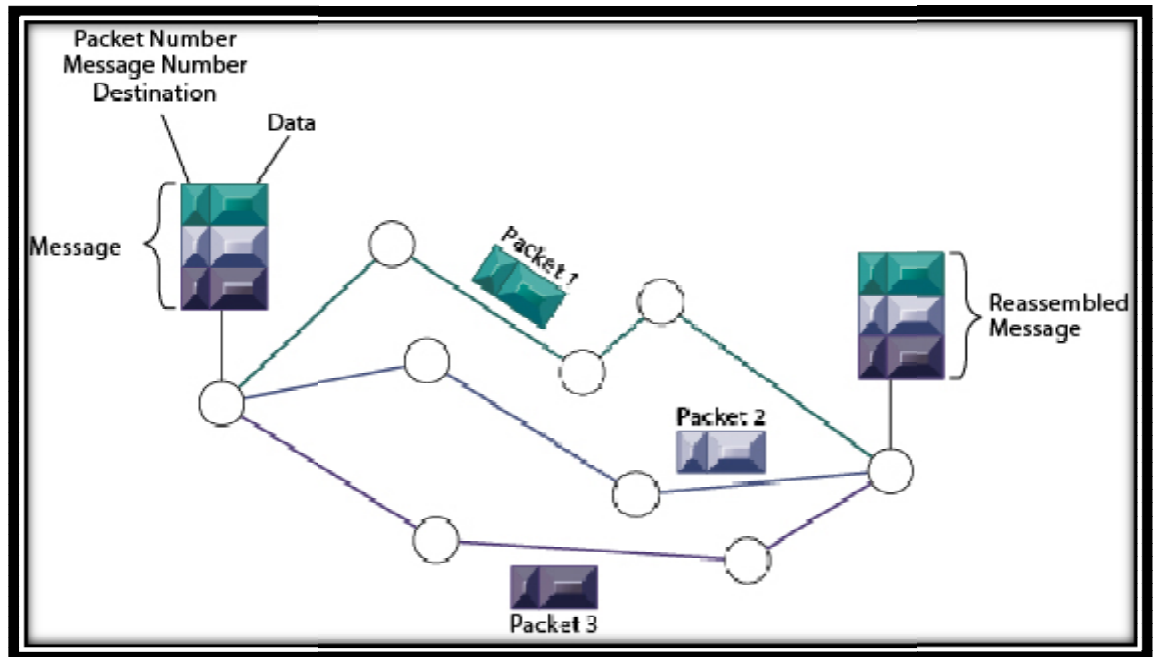


Εικόνα 8: Παράδειγμα κομβικού δικτύου.

3. Κριτήριο ως προς τις τεχνικές δικτύων που χρησιμοποιούνται

- **Δίκτυα Μεταγωγής (Switching)** – σε αυτό το δίκτυο δεσπόζει η δυνατότητα του κάθε συνδρομητή να καλεί επιλογικά τον ανταποκριτή του. Τα δίκτυα μεταγωγής αποτελούνται από συνδεδεμένους μεταξύ τους κόμβους, οι οποίοι αναλαμβάνουν τη δρομολόγηση της εκπεμπόμενης πληροφορίας από τον εκάστοτε αποστολέα πληροφορίας. Δηλαδή, τα δεδομένα που εισέρχονται από κάποιον τερματικό σταθμό δρομολογούνται από κόμβο σε κόμβο μέχρι τον προκαθορισμένο δέκτη. Μερικοί κόμβοι φυσικά δεν συνδέονται άμεσα με τερματικούς σταθμούς. Συνεπώς αυτοί, εκτελούν αποκλειστικά το ρόλο του δρομολογητή του εκάστοτε πακέτου πληροφορίας. Για λόγους αύξησης της αξιοπιστίας των δικτύων αυτών, οι συνδέσεις των κόμβων υλοποιούνται με σκοπό την ύπαρξη εναλλακτικών δρόμων μεταξύ των τερματικών σημείων. Και εδώ, η κύρια εφαρμογή βρίσκεται στα τηλεπικοινωνιακά δίκτυα. Μάλιστα, ένας περαιτέρω διαχωρισμός όσον αφορά τη σύνδεση δύο τερματικών σταθμών στα δίκτυα μεταγωγής, οδηγεί στην ύπαρξη των παρακάτω μεθόδων:
 - i. Μεταγωγή κυκλώματος (circuit switching)
 - ii. Μεταγωγή μηνυμάτων (message switching)
 - iii. Μεταγωγή πακέτων (Packet switching)

Ακολουθεί χαρακτηριστική εικόνα:



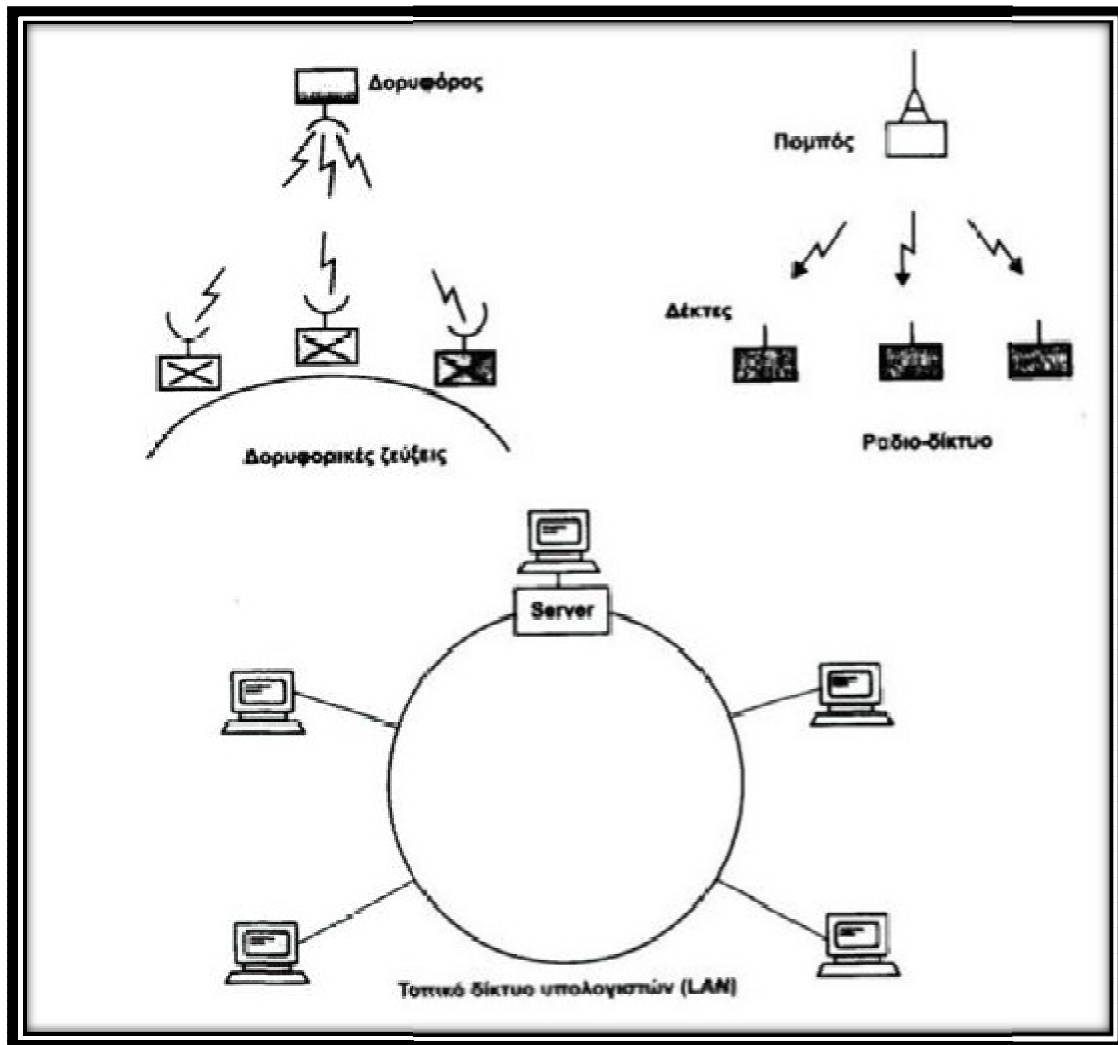
Εικόνα 9: Παράδειγμα δικτύου μεταγωγή πακέτων.

- **Δίκτυα Ακρόασης (Broadcasting)** – τα δίκτυα ακρόασης από την πλευρά τους διαφέρουν αισθητά με τα δίκτυα μεταγωγής, μιας και χρησιμοποιούν διαφορετική προσέγγιση. Τα χαρακτηριστικότερα παραδείγματα αποτελούν το ραδιοφωνικό και το τηλεοπτικό δίκτυο, όπου οι τερματικοί σταθμοί λαμβάνουν το ίδιο σήμα. Τα σημαντικότερα χαρακτηριστικά τους είναι τα επόμενα:
 - i. Το μέσο επικοινωνίας είναι συνήθως κοινό για όλους,
 - ii. Δεν υπάρχουν κόμβοι μεταγωγής,
 - iii. Κάθε μήνυμα που εκπέμπεται από έναν σταθμό λαμβάνεται από όλους,
 - iv. Μπορεί να υπάρχουν πολλοί πομποί που μοιράζονται χρονικά το ίδιο μέσο,
 - v. Ύπαρξη ειδικών τεχνικών ελέγχου προσπέλασης προς το κοινό μέσο επικοινωνίας.

Τα σημαντικότερα δίκτυα που ακολουθούν τους παραπάνω κανόνες και υπάγονται στα δίκτυα ακρόασης είναι τα εξής:

- i. Packet radio δίκτυα,
- ii. Δορυφορικά δίκτυα,
- iii. Τοπικά δίκτυα (LAN),
- iv. Τηλεοπτικά και ραδιοφωνικά δίκτυα.

Ακολουθεί χαρακτηριστική απεικόνιση σχετικών δικτύων:



Εικόνα 10: Παραδείγματα δικτύων ακρόασης.

Περισσότερες πληροφορίες σε σχέση με το διαχωρισμό των άλλων κριτηρίων μπορούν να βρεθούν στη βιβλιογραφία (Jean, 1997).

Στη συνέχεια θα δοθούν ορισμένοι πολύ καίριοι ορισμοί όρων που συχνά συναντώνται σε κείμενα σχετικά με δίκτυα:

- **10BASE-T** - Baseband δίκτυο Ethernet με ταχύτητα 10 Mbps, που χρησιμοποιεί στρεπτό ζεύγος καλωδίων.
- **ALOHA** - Μέθοδος προσπέλασης σε κοινό επικοινωνιακό μέσο, όπου ένας σταθμός μεταδίδει όποτε έχει δεδομένα. Αν δεν λάβει μήνυμα αναγνώρισης, επαναλαμβάνει τη μετάδοση.
- **ARPANET** - Ο πρόδρομος του Internet που δημιουργήθηκε από το Υπουργείο Άμυνας των ΗΠΑ και καταργήθηκε το 1990.
- **Backbone δίκτυο** - Ένα κεντρικό δίκτυο, πάνω στο οποίο συνδέονται άλλα δίκτυα.
- **Baseband** - Τεχνική μετάδοσης δεδομένων σύμφωνα με την οποία τα δεδομένα μεταδίδονται μέσα από το επικοινωνιακό μέσο με τη μορφή διακριτών παλμών ηλεκτρικού ρεύματος ή φωτός.

- **Broadband** - Τεχνική μετάδοσης δεδομένων σύμφωνα με την οποία τα δεδομένα μεταδίδονται μέσα από το επικοινωνιακό μέσο με τη μορφή ηλεκτρομαγνητικών κυμάτων που διακρίνονται από το πλάτος, τη συχνότητα και τη φάση τους.
- **Bus** - Τοπολογία δικτύου, σύμφωνα με την οποία όλοι οι σταθμοί συνδέονται απευθείας στο κοινό επικοινωνιακό μέσο.
- **Γέφυρα (Bridge)** - Μια συσκευή που συνδέει δύο ομογενή δίκτυα. Δέχεται όλα τα πακέτα που απευθύνονται από το ένα δίκτυο στο άλλο, τα αποθηκεύει και τα αναμεταδίδει στον προορισμό τους.
- **Δρομολογητής (Router)** - Συσκευή για τη σύνδεση δικτύων. Τα πακέτα κατευθύνονται στον δρομολογητή, ο οποίος αναλαμβάνει τη δρομολόγησή τους.
- **Επαναλήπτες (Repeaters)** - Συσκευές που χρησιμοποιούνται για τη διασύνδεση μεταξύ όμοιων δικτύων σ' ένα μεγαλύτερο δίκτυο. Δέχονται δεδομένα από το ένα δίκτυο και τα μεταφέρουν bit προς bit στο άλλο.
- **Ethernet** - Τοπικό δίκτυο τοπολογίας bus που χρησιμοποιεί προσπέλαση τύπου CSMA/CD και ακολουθεί το πρότυπο 802.3 της IEEE.
- **Internet** - Ένα παγκόσμιο σύνολο δικτύων, που περιλαμβάνει το ARPANET, τοπικά δίκτυα πανεπιστημίων, στρατιωτικά και εμπορικά δίκτυα και ακολουθεί το πρωτόκολλο TCP/IP.
- **ISDN (Integrated Services Digital Network)** - Ένα ψηφιακό δίκτυο, το οποίο επιτρέπει μετάδοση ήχου και δεδομένων.
- **Πακέτο (Packet)** - Σύνολο από bits που περιέχουν δεδομένα και μια ή δύο διευθύνσεις και χρησιμοποιούνται για τη μεταφορά δεδομένων.
- **Πρωτόκολλο (Protocol)** - Το σύνολο των κανόνων που ισχύουν κατά την επικοινωνία και ορίζουν τη μορφή των δεδομένων, καθώς και τις διαδικασίες που πρέπει να ακολουθούνται για τη μεταφορά τους.
- **Πύλη (Gate)** - Συσκευή για τη σύνδεση δικτύων που υποστηρίζουν διαφορετικά πρωτόκολλα. Κάνει όλες τις απαραίτητες μετατροπές από το ένα πρωτόκολλο στο άλλο.
- **Ring** - Τοπολογία δικτύου, με την οποία οι σταθμοί συνδέονται στο δίκτυο μέσω επαναληπτών που ενώνονται μεταξύ τους, δημιουργώντας έναν κλειστό δακτύλιο.
- **Star** - Τοπολογία δικτύου, σύμφωνα με την οποία υπάρχει ένας κεντρικός ελεγκτής (hub), πάνω στον οποίο συνδέονται απευθείας οι σταθμοί εργασίας.
- **Σταθμός Εργασίας (WorkStation)** - Οι διάφοροι υπολογιστές που είναι προσαρτημένοι στο δίκτυο, από τους οποίους οι χρήστες μπορούν να εκτελούν τις εφαρμογές τους και να έχουν πρόσβαση σε δίσκους και εκτυπωτές.
- **Token** - Ένα ειδικό πλαίσιο (κουπόνι, σκυτάλη), που κινείται μέσα στο δίκτυο. Κάθε σταθμός που έχει το token αποκτά αυτόματα και το δικαίωμα να μεταδώσει δεδομένα μέσα στο κανάλι.
- **Token-Bus** - Τοπικό δίκτυο που ακολουθεί το πρότυπο 802.4, χρησιμοποιεί τοπολογία bus, ενώ από λογική άποψη οι σταθμοί οργανώνονται σ' ένα δακτύλιο (ring). Ένας σταθμός μπορεί να μεταδώσει μόνο όταν έχει το token.
- **Token-Ring** - Τοπικό δίκτυο που ακολουθεί το πρότυπο 802.5 και χρησιμοποιεί τοπολογία ring. Ένας σταθμός μπορεί να μεταδώσει μόνο όταν έχει το token.

1.4 Πρωτόκολλα Δικτύων TCP/IP και OSI

Το TCP/IP είναι το πρωτόκολλο επικοινωνίας που χρησιμοποιείται στο Internet. Ένα πρωτόκολλο επικοινωνίας υπολογιστών (computercommunicationprotocol) είναι μια περιγραφή των κανόνων που πρέπει να ακολουθούν οι υπολογιστές για να μπορούν να επικοινωνούν μεταξύ τους. Συνεπώς, το TCP/IP είναι το πρωτόκολλο επικοινωνίας (communicationprotocol) για την επικοινωνία ανάμεσα σε υπολογιστές που είναι συνδεδεμένοι στο παγκόσμιο δίκτυο υπολογιστών Internet (Διαδίκτυο). Τα αρχικά TCP/IP αναφέρονται σε δύο από τα σημαντικότερα πρωτόκολλα που χρησιμοποιούνται στο Internet, δηλ. στο TCP και στο IP, και όχι σ' όλα τα πρωτόκολλα του Internet (Tanenbaum, 2003).

Το ακρωνύμιο TCP/IP ορίζεται από το TransmissionControlProtocol/InternetProtocol, δηλαδή Πρωτόκολλο Ελέγχου Εκπομπής/Πρωτόκολλο του Internet. Το πρότυπο (standard) ορίζει το πώς οι ηλεκτρονικές συσκευές, όπως είναι οι υπολογιστές, θα πρέπει να συνδέονται στο Internet και πώς θα πρέπει να μεταδίδονται τα δεδομένα ανάμεσά τους. Μέσα στο TCP/IP υπάρχουν πολλά πρωτόκολλα για τον χειρισμό της επικοινωνίας των δεδομένων (datacommunication):

- Το TCP (Transmission Control Protocol), για επικοινωνία ανάμεσα σε εφαρμογές (applications),
- Το UDP (User Datagram Protocol), για απλή επικοινωνία ανάμεσα σε εφαρμογές (applications),
- Το IP (Internet Protocol), για επικοινωνία ανάμεσα σε υπολογιστές,
- Το ICMP (Internet Control Message Protocol), για λάθη και στατιστικές,
- Το DHCP (Dynamic Host Configuration Protocol), για δυναμική διεύθυνση (dynamic addressing).

Πιο συγκεκριμένα, το πρωτόκολλο TCP προορίζεται για επικοινωνία ανάμεσα σε εφαρμογές (applications). Όταν μια εφαρμογή θελήσει να επικοινωνήσει με μια άλλη εφαρμογή μέσω του TCP, στέλνει μια αίτηση επικοινωνίας (communicationrequest). Αυτή η αίτηση θα πρέπει να σταλεί σε μια συγκεκριμένη διεύθυνση. Αφού καθιερωθεί μια χειραψία (handshake) ανάμεσα στις δύο εφαρμογές, το TCP θα καθιερώσει μια ταυτόχρονη αμφίπλευρη (full-duplex) επικοινωνία ανάμεσα στις δύο εφαρμογές. Η ταυτόχρονη αμφίπλευρη (full-duplex) επικοινωνία θα καταλάβει τη γραμμή επικοινωνίας ανάμεσα στους δύο υπολογιστές μέχρι αυτή να κλείσει από μια από τις δύο εφαρμογές. Το πρωτόκολλο UDP είναι πολύ παρόμοιο με το TCP με τη διαφορά ότι είναι πιο απλό και λιγότερο αξιόπιστο (reliable).

Αντιθέτως, το πρωτόκολλο IP προορίζεται για επικοινωνία ανάμεσα σε υπολογιστές. Το IP είναι ένα connection-less (χωρίς σύνδεση) πρωτόκολλο επικοινωνίας (communicationprotocol). Αυτό σημαίνει ότι δεν καταλαμβάνει τη γραμμή επικοινωνίας ανάμεσα σε δύο επικοινωνούντες υπολογιστές. Μ' αυτόν τον τρόπο το IP ελαττώνει την ανάγκη για γραμμές δικτύωσης. Έτσι, η κάθε γραμμή θα μπορεί να χρησιμοποιηθεί για επικοινωνία ανάμεσα σε πολλούς διαφορετικούς υπολογιστές την ίδια στιγμή. Επιπλέον, με το IP, τα μηνύματα (ή και άλλα δεδομένα) διασπώνται σε μικρά ανεξάρτητα πακέτα (packets) και στέλνονται ανάμεσα στους υπολογιστές μέσω του Internet (Tanenbaum, 2003). Το IP είναι υπεύθυνο για τη δρομολόγηση (routing) του κάθε πακέτου μέχρι αυτό να φθάσει στον τελικό του προορισμό. Η επικοινωνία μέσω του IP είναι σαν να στέλνουμε ένα μακροσκελές γράμμα σαν ένα μεγάλο πλήθος από μικρότερα μηνύματα, όπου το καθένα θα βρει τον δικό του, συχνά διαφορετικό, δρόμο μέχρι τον παραλήπτη.

Σημαντικότερο ρόλο σε όλη αυτή τη διαδρομή παίζουν οι ενδιάμεσοι δρομολογητές. Ουσιαστικά αυτοί αποτελούν τον εγκέφαλο όλου του δικτύου. Όταν ένα πακέτο του IP στέλνεται από έναν υπολογιστή, αυτό φθάνει σ' έναν IP δρομολογητή (router). Ο IP δρομολογητής είναι υπεύθυνος για τη δρομολόγηση (routing) του πακέτου μέχρι τον τελικό του προορισμό, είτε απευθείας είτε μέσω ενός άλλου δρομολογητή. Η διαδρομή (path) που θα ακολουθήσει ένα πακέτο μπορεί να είναι διαφορετική από αυτήν που θα ακολουθήσουν άλλα πακέτα του ίδιου μηνύματος. Ο δρομολογητής είναι υπεύθυνος για τη σωστή διευθυνσιοδότηση των πακέτων (addressing) ανάλογα και με τον όγκο της κυκλοφορίας, τα τυχόν λάθη που θα λάβουν χώρα στο δίκτυο ή και από άλλες παραμέτρους (Γεώργιος Αλεξόπουλος, 2015).

Συμπεραίνει κανείς από τα παραπάνω πως το πρωτόκολλο TCP/IP αποτελείται από τα πρωτόκολλα TCP και το IP, τα οποία συνεργάζονται μεταξύ τους. Οι ρόλοι του καθενός είναι ευδιάκριτοι και η αρμονική συνεργασία τους οδηγεί στην σωστή, άμεση και χωρίς προβλήματα επικοινωνία. Συνοπτικά, το TCP φροντίζει για την επικοινωνία ανάμεσα στο λογισμικό μιας εφαρμογής, δηλαδή το φυλλομετρητή, και το λογισμικό του δικτύου μας. Από την άλλη πλευρά, το IP φροντίζει για την επικοινωνία με τους άλλους υπολογιστές. Το TCP είναι υπεύθυνο για τη διάσπαση των δεδομένων σε IP πακέτα πριν αυτά αποσταλούν καθώς και για την αντίστοιχη συναρμολόγησή τους όταν αυτά φθάσουν στον προορισμό τους. Το IP είναι υπεύθυνο για την αποστολή των πακέτων στον παραλήπτη (receiver). Το TCP/IP χρησιμοποιεί 32 bits ή 4 αριθμούς με τιμές ανάμεσα στο 0 και τα 255 για να αποδώσει μια διεύθυνση (address) σ' έναν υπολογιστή.

Σε ότι έχει να κάνει με τις IP διευθύνσεις, αυτές παίρνουν τον ρόλο της αστυνομικής ταυτότητας σε σχέση με την ανθρώπινη κοινωνία. Κάθε υπολογιστής θα πρέπει να διαθέτει μια IP διεύθυνση πριν μπορέσει να συνδεθεί στο Internet και το κάθε IP πακέτο (packet) θα πρέπει να διαθέτει μια διεύθυνση (address) πριν μπορέσει να αποσταλεί σ' έναν άλλον υπολογιστή. Για παράδειγμα, η παρακάτω IP διεύθυνση 62.38.6.153 ανήκει στο sitegoogle.gr. Ένα άλλο παράδειγμα IP διεύθυνσης είναι το εξής : 83.235.160.51. Το TCP/IP χρησιμοποιεί 4 ακέραιους αριθμούς για να αποδώσει μια διεύθυνση σ' έναν υπολογιστή. Έτσι, ο κάθε υπολογιστής θα πρέπει να διαθέτει μια μοναδική διεύθυνση με 4 ακέραιους αριθμούς. Οι αριθμοί αυτοί έχουν πάντα τιμές ανάμεσα στο 0 και το 255. Οι διευθύνσεις γράφονται με τους 4 αριθμούς χωρισμένους με τελείες, ως εξής : 193.106.1.51. Το TCP/IP χρησιμοποιεί διευθυνσιοδότηση (addressing) των 32 bits. Ένα byte στους υπολογιστές είναι ίσο με 8 bits. Έτσι, το TCP/IP χρησιμοποιεί 4 bytes. Ένα byte μπορεί να περιέχει 256 διαφορετικές τιμές, που προκύπτουν από τις επαναληπτικές διατάξεις των 2 πραγμάτων ανά 8, δηλ. συνολικό πλήθος τιμών $2^8 = 256$. Οι αριθμοί αυτοί είναι οι εξής :

```
00000000
00000001
00000010
00000011
00000100
00000101
00000110
00000111
00001000
...
11111111
```

Επειδή είναι πολύ δύσκολο να θυμόμαστε τους αριθμούς και η χρήση των ονομάτων είναι σαφώς ευκολότερη, τα ονόματα που χρησιμοποιούνται και που αντιστοιχούν στις διευθύνσεις TCP/IP αποκαλούνται domainnames (ονόματα χώρου ή ονόματα περιοχής). Για παράδειγμα, το in.gr είναι ένα domainname (όνομα χώρου). Όταν αποκτάμε πρόσβαση σ' έναν δικτυακό τόπο (Website), όπως είναι το <http://www.google.gr>, για παράδειγμα, το όνομα αυτό μεταφράζεται (αντιστοιχείται) σε μια σειρά από αριθμούς από μια διαδικασία που αποκαλείται DNS (DomainNameServer) ή σύστημα ονοματοδοσίας. Σε όλο τον κόσμο, υπάρχει ένας μεγάλος αριθμός από DNSservers που είναι συνδεδεμένοι στο Internet. Οι DNSservers είναι υπεύθυνοι για τη μετάφραση των domainnames σε αντίστοιχες διευθύνσεις του TCP/IP καθώς και για την αλληλοενημέρωσή τους με τις καινούργιες καταχωρήσεις ή και τις τροποποιήσεις και διαγραφές των domainnames. Όταν καταχωρείται στο μητρώο (registered) ένα καινούργιο domainname μαζί με μια διεύθυνση TCP/IP, οι DNSservers σ' όλο τον κόσμο ενημερώνονται με τα καινούργια στοιχεία.

Τέλος, μία άλλη προσέγγιση του TCP/IP αποτελεί η αντιμετώπιση του ως μια μεγάλη συλλογή από πολλά διαφορετικά πρωτόκολλα επικοινωνίας, τα οποία βασίζονται στα δύο σημαντικότερα και αρχικά πρωτόκολλα, το TCP και το IP. Παρακάτω αυτά αναφέρονται με μια σύντομη περιγραφή:

- TCP – TransmissionControlProtocol

Το πρωτόκολλο TCP χρησιμοποιείται για τη μετάδοση των δεδομένων από μια εφαρμογή (application) στο δίκτυο. Το TCP είναι υπεύθυνο για τη διάσπαση των δεδομένων σε μικρότερα IP πακέτα (packets) πριν αυτά αποσταλούν καθώς και για την αντίστοιχη συναρμολόγηση των πακέτων όταν αυτά φθάσουν στον προορισμό τους.

- IP – InternetProtocol

Το πρωτόκολλο IP φροντίζει για την επικοινωνία με τους άλλους υπολογιστές και είναι υπεύθυνο για την αποστολή και τη λήψη (δρομολόγηση) των πακέτων δεδομένων μέσω του Internet.

- HTTP – Hyper Text Transfer Protocol

Το πρωτόκολλο HTTP φροντίζει για την επικοινωνία ανάμεσα σ' έναν Webserver και έναν Webbrowser (φυλλομετρητή). Χρησιμοποιείται για την αποστολή των αιτήσεων (requests) από έναν Webclient (browser) σ' έναν Webserver καθώς και για την επιστροφή του περιεχομένου (Webcontent), δηλ. των ιστοσελίδων (Webpages), από τον server πίσω στον χρήστη (client).

- HTTPS – SecureHTTP

Το πρωτόκολλο HTTPS φροντίζει για την ασφαλή επικοινωνία ανάμεσα σ' έναν Webserver και έναν Webbrowser. Χειρίζεται τυπικά τις συναλλαγές που γίνονται με πιστωτικές κάρτες (creditcardtransactions) καθώς και μ' άλλα ευαίσθητα δεδομένα.

- SSL – SecureSocketsLayer

Το πρωτόκολλο SSL χρησιμοποιείται για την κωδικοποίηση (κρυπτογράφηση – encryption) των δεδομένων για να είναι έτσι ασφαλής η μεταφορά τους.

- SMTP – SimpleMailTransferProtocol

Το πρωτόκολλο SMTP χρησιμοποιείται για την αποστολή των μηνυμάτων ηλεκτρονικού ταχυδρομείου (e-mails) από τον χρήστη σ' έναν e-mailserver.

- IMAP – InternetMessageAccessProtocol
Το πρωτόκολλο IMAP χρησιμοποιείται για την αποθήκευση (storing) και την ανάκτηση (retrieving) των μηνυμάτων ηλεκτρονικού ταχυδρομείου (e-mails).
- POP – PostOfficeProtocol
Το πρωτόκολλο POP χρησιμοποιείται για το κατέβασμα (downloading) των μηνυμάτων ηλεκτρονικού ταχυδρομείου (e-mails) από έναν e-mailserver στον προσωπικό υπολογιστή (personalcomputer) του χρήστη.
- FTP – FileTransferProtocol
Το πρωτόκολλο FTP φροντίζει για την αποστολή των αρχείων (ανέβασμα – upload και κατέβασμα – download) ανάμεσα σε υπολογιστές και κυρίως ανάμεσα σ' έναν FTPserver και έναν απλό προσωπικό υπολογιστή.
- NTP – Network Time Protocol
Το πρωτόκολλο NTP χρησιμοποιείται για τον συγχρονισμό της ώρας (του ρολογιού) ανάμεσα στους υπολογιστές.
- DHCP – DynamicHostConfigurationProtocol
Το πρωτόκολλο DHCP χρησιμοποιείται για την κατανομή των δυναμικών IP διευθύνσεων στους υπολογιστές ενός δικτύου (network).
- SNMP – SimpleNetworkManagementProtocol
Το πρωτόκολλο SNMP χρησιμοποιείται για τη διοίκηση (administration) των δικτύων υπολογιστών (computernetworks).
- LDAP – LightweightDirectoryAccessProtocol
Το πρωτόκολλο LDAP χρησιμοποιείται για τη συλλογή πληροφοριών σχετικά με χρήστες (users) και διευθύνσεις e-mail από το Internet.
- ICMP – Internet Control Message Protocol
Το πρωτόκολλο ICMP φροντίζει για την αντιμετώπιση των λαθών (errorhandling) στο δίκτυο (network).
- ARP – AddressResolutionProtocol
Το πρωτόκολλο ARP χρησιμοποιείται από το IP για να βρεθεί η διεύθυνση υλικού (hardwareaddress) μιας κάρτας δικτύου υπολογιστή που βασίζεται στην IP διεύθυνση.
- RARP – Reverse Address Resolution Protocol
Το πρωτόκολλο RARP χρησιμοποιείται από το IP για να βρεθεί η IP διεύθυνση που βασίζεται στη διεύθυνση υλικού (hardwareaddress) μιας κάρτας δικτύου υπολογιστή.
- BOOTP – BootProtocol
Το πρωτόκολλο BOOTP χρησιμοποιείται για το ξεκίνημα των υπολογιστών από το δίκτυο (network).
- PPTP – Point to Point Tunneling Protocol
Το πρωτόκολλο PPTP χρησιμοποιείται για την εγκαθίδρυση μιας σύνδεσης (tunnel) ανάμεσα σε ιδιωτικά δίκτυα (privatenetworks).

- TCP/IP Email

Το email αποτελεί έναν από τους σημαντικότερους χρήστες του TCP/IP. Όταν γράφουμε ένα email, δεν χρησιμοποιούμε το TCP/IP, αλλά ένα πρόγραμμα email όπως είναι το LotusNotes ή το Outlook ή το OutlookExpress της Microsoft ή το NetscapeMessenger ή το NetscapeCommunicator ή το ThunderBird της Mozilla ή και το Eudora. Ένα email πρόγραμμα χρησιμοποιεί τα εξής διαφορετικά πρωτόκολλα του TCP/IP:

1. Στέλνει τα μηνύματά μας (emails) με το πρωτόκολλο SMTP.
2. Κατεβάζει (download) τα μηνύματά μας (emails) από έναν email server με το πρωτόκολλο POP.
3. Μπορεί να συνδεθεί σ' έναν email server με το πρωτόκολλο IMAP.

- SMTP – Simple Mail Transfer Protocol

Το πρωτόκολλο SMTP χρησιμοποιείται για την αποστολή των e-mails. Στην ουσία φροντίζει για την αποστολή των μηνυμάτων μας σ' έναν άλλον υπολογιστή. Κανονικά το email μας στέλνεται σ' έναν emailserver (SMTPserver) και από εκεί σ' έναν άλλον server ή και σ' άλλους servers, μέχρι να φθάσει στον τελικό του προορισμό. Το SMTP μπορεί μόνο να στείλει απλό κείμενο. Δεν μπορεί να μεταδώσει εικόνα (picture) ή ήχο (sound) ή κινούμενη εικόνα (movie).

1.5 World Wide Web & Client-Server Μοντέλο

Η απλούστερη περιγραφή που θα μπορούσε να δώσει κανείς για το Internet, η οποία συνοψίζει ταυτόχρονα και την ουσία της ύπαρξης και της λειτουργίας του, είναι ότι αυτό αποτελεί έναν “χώρο” όπου αποθηκεύεται η πληροφορία. Από την ίδια τη φύση της όμως η πληροφορία δεν μπορεί να είναι στατική, διότι τότε χάνει την έννοιά της. Επεκτείνοντας λοιπόν την αρχική περιγραφή, ένας πιο ακριβής ορισμός θα έλεγε ότι το Internet είναι ένας χώρος αποθήκευσης αλλά ταυτόχρονα και ένα μέσο μετάδοσης, επεξεργασίας και μετεξέλιξης της πληροφορίας (Tanenbaum, 2003).

Κατά τη διάρκεια της εξέλιξης του δικτύου, εμφανίστηκαν και δοκιμάστηκαν πολλά εργαλεία, μέθοδοι και τρόποι αποθήκευσης και μετάδοσης της πληροφορίας, αρκετοί από τους οποίους πέτυχαν και κατάφεραν τελικά να καθιερωθούν. Ωστόσο το αποφασιστικό βήμα για την ολοκλήρωση (integration) της αναζήτησης της πληροφορίας στο Internet έγινε με την εισαγωγή μίας νέας υπηρεσίας που ονομάζεται World Wide Web (WWW) ή αλλιώς WEB. Ο Παγκοσμίως Ευρύς Ιστός, με απλά λόγια το Web, είναι ένα δίκτυο υπολογιστών βασισμένο στο Internet, που επιτρέπει στους χρήστες ενός υπολογιστή να έχουν πρόσβαση σε πληροφορίες που είναι αποθηκευμένες σε κάποιο άλλο υπολογιστή μέσω αυτού του παγκόσμιου δικτύου.

Πιο συγκεκριμένα, έστω ότι έχουμε πολλούς υπολογιστές διασυνδεδεμένους μέσω καλωδιώσεων σε ένα είδος δικτύου ευρείας περιοχής WAN (Wide Area Network), που είναι ουσιαστικά η ίδια ανάπτυξη που διακρίνουμε σε όλο το Internet. Κάθε ένας από αυτούς τους υπολογιστές καλείται εξυπηρετητής ή διανομέας Web (Web server) και περιέχει ένα σύνολο από δεδομένα υπερμέσων, τα λεγόμενα Web Documents, τα οποία δεικτοδοτούνται μεταξύ τους μέσω των διευθύνσεων των υπολογιστών στους οποίους περιέχονται συν των σχετικών τους εσωτερικών διευθύνσεων εντός των υπολογιστών αυτών. Πρόκειται, στην ουσία για ένα αλληλεπιδραστικό multimedia περιβάλλον που προσφέρει τη δυνατότητα στον χρήστη να

δημιουργήσει εντυπωσιακά ηλεκτρονικά έγγραφα και να τα “δημοσιεύσει” στον Web, καθιστώντας τα έτσι προσβάσιμα για οποιονδήποτε ενδιαφερόμενο (Jean, 1997).

Το Web είναι δομημένο σε μία αρχιτεκτονική client / server. Σε μία απλουστευμένη προσέγγιση, ο server όπως είδαμε περιέχει τα λεγόμενα Web Documents, μέσα στα οποία βρίσκονται οι διάφοροι σύνδεσμοι και “διοχετεύει” τις πληροφορίες προς τον client, χειριζόμενος ταυτόχρονα και τις αιτήσεις για επικοινωνία και διασύνδεση με άλλους απομακρυσμένους υπολογιστές (hosts). Στο Web ο client αποκαλείται Web Browser. Ένας Web browser (φυλλομετρητής ιστοσελίδων, πλοηγός Web, πρόγραμμα περιήγησης Web ή περιηγητής Ιστού) είναι ένα λογισμικό που επιτρέπει στον χρήστη του να προβάλλει και να αλληλεπιδρά με όλα τα Web Documents, που είναι συνήθως αναρτημένα σε μια ιστοσελίδα ενός ιστοτόπου στον Παγκόσμιο Ιστό ή σε ένα τοπικό δίκτυο. Το πρωτόκολλο που χρησιμοποιείται από τον Web για την εκτέλεση όλων αυτών των διεργασιών είναι το HTTP (Πρωτόκολλο Μεταφοράς Υπερκειμένου), ενώ η HTML ((Γλώσσα Επισήμανσης Υπερκειμένου) αποτελεί την γλώσσα κατασκευής των Web Documents, της πληροφορίας που ρέει στο Web.

1.6 Web Services

Τα web services εμφανίστηκαν το 2001 όταν άρχισε να γίνεται λόγος για αυτό-περιγραφόμενες και αυτό-προσδιοριζόμενες εφαρμογές οι οποίες μπορούν να ανακαλυφθούν και να προσπελαστούν μέσω διαδικτύου από άλλες εφαρμογές. Οι υπηρεσίες WEB κατάφεραν όχι μόνο να παραμείνουν στο προσκήνιο των εξελίξεων στον χώρο της τεχνολογίας όλη τη δεκαετία, αλλά να αποτελέσουν ένα από τα πιο σημαντικά κεφάλαια της υψηλής τεχνολογίας του Διαδικτύου (wikipedia, n.d.).

Web service είναι οποιοδήποτε τμήμα λογισμικού ή συστατικό εφαρμογής που είναι διαθέσιμο μέσω διαδικτύου και το οποίο μπορεί να ολοκληρώνει εργασίες και να λύνει προβλήματα. Ωστόσο αν θέλουμε να προσδιορίσουμε πιο σωστά την έννοια των web services, θα μπορούσαμε να χρησιμοποιήσουμε τον ορισμό που έρχεται από την IBM :

Ορισμός: Τα web services είναι μια τεχνολογία που επιτρέπει στις εφαρμογές να επικοινωνούν μεταξύ τους ανεξαρτήτως πλατφόρμας και γλώσσας προγραμματισμού. Ένα web service αποτελεί μια διεπαφή λογισμικού (software interface) που περιγράφει μια συλλογή από λειτουργίες οι οποίες μπορούν να προσεγγιστούν από το δίκτυο μέσω πρότυπων μηνυμάτων XML. Χρησιμοποιεί πρότυπα βασισμένα στη γλώσσα XML για να περιγράψει μια λειτουργία προς εκτέλεση και τα δεδομένα προς ανταλλαγή με κάποια άλλη εφαρμογή. Μια ομάδα από web services οι οποίες αλληλεπιδρούν μεταξύ τους καθορίζει μια εφαρμογή web services.

Με απλά λόγια μια Web service μπορούμε να την αντιληφθούμε σαν ένα λογισμικό που γνωρίζει με ποιόν τρόπο πρέπει να ‘μιλήσει’ σε διαφορετικού τύπου λογισμικά μέσω δικτύου. Πιο συγκεκριμένα μια υπηρεσία Web πραγματοποιεί τις ακόλουθες λειτουργίες:

1. Παρέχει την δυνατότητα σε άλλες εφαρμογές να κατανοήσουν την λειτουργικότητά της, εκθέτοντας και περιγράφοντας την παρουσία της σε αυτές με τη χρήση ενός πρότυπου δικτυακού πρωτοκόλλου.
2. Μπορεί να γίνει εύκολα αντιληπτή από άλλες εφαρμογές μέσω ενός καταλόγου στον οποίο βρίσκετε καταχωρημένα.

3. Μπορεί να προβληθεί ή να χρησιμοποιηθεί από την εφαρμογή που την χρειάζεται μέσω απλών πρωτοκόλλων.

Η δυνατότητα που δίνεται σε οποιαδήποτε σύγχρονη επιχείρηση να παρέχει online συναλλαγές με τους προμηθευτές της αλλά και με τους πελάτες της μέσω της συγκεκριμένης υπηρεσίας, είναι μόλις μια από τις πολλές εφαρμογές της υπηρεσίας Web. Δεν είναι τυχαία άλλωστε η ραγδαία εξάπλωση της παρά τις αρνητικές πιθανότητες των αρχών της δεκαετίας του 2000, όπου πολλές άλλες προσανατολισμένες εφαρμογές της τεχνολογίας προς το Διαδίκτυο, εξαφανίστηκαν σχεδόν όσο γρήγορα εμφανίστηκαν.

1.7 Web Application

Διαδικτυακή εφαρμογή (web application ή web app) ονομάζεται κάθε εφαρμογή η οποία είναι διαθέσιμη στους χρήστες της μέσω του Διαδικτύου ή του ενδοδικτύου(Intranet) μιας εταιρίας και ο χρήστης χρησιμοποιεί μόνο τον περιηγητή του για να την χρησιμοποιήσει. Οι εφαρμογές αυτές συνήθως εκτελούνται σε ισχυρές υπολογιστικές μηχανές οι οποίες έχουν τον ρόλο του σταθμού εξυπηρέτησης και παρέχουν τις υπηρεσίες τους σε περισσότερους του ενός χρήστη(wikipedia, n.d.).

Βασικά συστατικά μέρη μιας διαδικτυακής εφαρμογής αποτελούν όλες οι προαναφερθείσες τεχνολογίες οι οποίες παρουσιάστηκαν σε προηγούμενες παραγράφους καθώς επίσης και οι διάφορες γλώσσες προγραμματισμού της, με βασικότερη την γλώσσα υπερκειμένου HTML, η οποία θα παρουσιαστεί εκτενώς στα επόμενα κεφάλαια που θα ακολουθήσουν.

Οι τοπικές εφαρμογές υπολογιστών παραμένουν μέχρι και σήμερα αρκετά δημοφιλείς και αυτό οφείλεται εν μέρη στο ότι οι τοπικές εφαρμογές ήταν και οι πρώτες εφαρμογές που δημιουργήθηκαν(Γεώργιος Αλεξόπουλος, 2015). Με την πάροδο του χρόνου όμως όλο και περισσότερες διαδικτυακές εφαρμογές κάνουν την εμφάνιση τους. Συγκρίνοντας τις δυο κατηγορίες εφαρμογών οι χρήστες θα εντοπίσουν τόσο θετικά όσο και αρνητικά στοιχεία και στις δυο και σε συνάρτηση με τις εργασίες που θέλουν η εφαρμογή τους να πραγματοποιεί. Παρακάτω παρατίθενται μερικά από τα πλεονεκτήματα μιας διαδικτυακής εφαρμογής:

- **Άμεση πρόσβαση από οποιαδήποτε συσκευή:** Οι χρήστες των διαδικτυακών εφαρμογών έχουν άμεση προσβασιμότητα στις εφαρμογές που θέλουν να χρησιμοποιήσουν από οποιονδήποτε υπολογιστή ή άλλη συσκευή έχει internet χωρίς την εγκατάσταση κάποιου επιπρόσθετου λογισμικού. Η μόνη απαραίτητη εφαρμογή είναι ο περιηγητής διαδικτύου ο οποίος είναι προεγκατεστημένος σε όλα τα λειτουργικά συστήματα ακόμα και στις φορητές συσκευές αλλά και στα κινητά τηλέφωνα. Η ιδιότητα αυτή των διαδικτυακών εφαρμογών είναι ιδιαίτερα σημαντική για μεγάλες επιχειρήσεις με πολλούς χρήστες που στην περίπτωση της τοπικής εφαρμογής θα έπρεπε να εγκατασταθεί η εφαρμογή σε κάθε ένα υπολογιστή ξεχωριστά.
- **Δυνατότητα χρήσης ανεξαρτήτου τοποθεσίας:** Ως συνέχεια του παραπάνω οι χρήστες των διαδικτυακών εφαρμογών μπορούν να τις χρησιμοποιούν ακόμα και αν δεν βρίσκονται στον χώρο εργασίας τους. Η δυνατότητα αυτή δίνει ευελιξία στους χρήστες ώστε να χρησιμοποιούν τις εφαρμογές οπουδήποτε αυτοί επιθυμούν

επιτρέποντας τους ακόμα και να εργάζονται από απομακρυσμένες περιοχές ή και από το σπίτι τους.

- **Συμβατές με όλα τα λειτουργικά συστήματα:** Ένα ακόμα πλεονέκτημα των διαδικτυακών εφαρμογών είναι ότι είναι συμβατές με όλα τα λειτουργικά συστήματα. Καθώς η εφαρμογή εκτελείτε μέσω του περιηγητή του διαδικτύου και όχι στον υπολογιστή του χρήστη την κάνει ικανή να εκτελείται σε όλα τα λειτουργικά συστήματα. Η ιδιότητα αυτή οφείλεται επίσης και στην προτυποποίηση των γλωσσών προγραμματισμού τις οποίες χρησιμοποιεί η εφαρμογή.
- **Δεν καταναλώνουν πόρους:** Ως συνέχεια του παραπάνω και εφόσον οι διαδικτυακές εφαρμογές δεν εκτελούνται στον υπολογιστή του χρήστη δεν καταναλώνουν και πόρους από το σύστημα. Για τον λόγο αυτό οι εφαρμογές διαδικτύου είναι ιδιαίτερα ελαφριές για την υπολογιστική μονάδα.
- **Δεν καταλαμβάνουν χώρο:** Ακολουθώντας την ίδια λογική με νωρίτερα οι εφαρμογές αυτές δεν καταλαμβάνουν καθόλου ή σχεδόν καθόλου χώρο στον δίσκο του χρήστη αφού το σύνολο της εφαρμογής είναι αποθηκευμένο στον εξυπηρετητή και μόνο κατά την χρήση της εφαρμογής μπορεί να υπάρχει μεταφορά δεδομένων προς την υπολογιστική μονάδα του χρήστη και μόνο στην περίπτωση που ο χρήστης το επιθυμεί.
- **Γρήγορη αναβάθμιση:** Σημαντικό πλεονέκτημα συγκριτικά με τις τοπικές εφαρμογές εμφανίζεται στις περιπτώσεις που η εφαρμογή χρειάζεται κάποια αναβάθμιση. Σε μια κλασική τοπική εφαρμογή η αναβάθμιση του συστήματος θα πρέπει να γίνει σε κάθε ένα υπολογιστή ξεχωριστά πράγμα που απαιτεί χρόνο και χρήμα. Αντίθετα σε μια διαδικτυακή εφαρμογή η αναβάθμιση πραγματοποιείται μόνο στον εξυπηρετητή που φιλοξενεί την εφαρμογή και ταυτόχρονα το αναβαθμισμένο πρόγραμμα είναι διαθέσιμο σε όλους τους χρήστες. Με τον τρόπο αυτό εξοικονομείτε χρόνος ο οποίος είναι ιδιαίτερα πολύτιμος κυρίως για τις μεγάλες επιχειρήσεις ενώ ως συνέπεια του παραπάνω σημαντικά μειωμένο είναι και το κόστος της αναβάθμισης μιας και απαιτείται λιγότερο εργατικό δυναμικό για την διεκπεραίωση της αναβάθμισης.
- **Νέο, βελτιωμένο περιβάλλον:** Ένα ακόμα πλεονέκτημα των διαδικτυακών εφαρμογών είναι ότι πλέον με την εμφάνιση της HTML5 είναι δυνατό ο δημιουργός της εφαρμογής να την εμπλουτίσει έτσι ώστε να είναι πιο φιλική, εύχρηστη και ευχάριστη προς τον χρήστη με εύκολο τρόπο. Παλαιότερα οι εφαρμογές αυτές υστερούσαν στην εμφάνιση ωστόσο πλέον είναι ιδιαίτερα εύκολα να εμπλουτιστούν.
- **Δυνατότητα χρήσης και εκτός διαδικτύου:** Ένα ακόμα πλεονέκτημα των σύγχρονων διαδικτυακών εφαρμογών(εφαρμογές με χρήση HTML5) είναι η δυνατότητα της εκτός διαδικτύου χρήσης μιας διαδικτυακής εφαρμογής με την προϋπόθεση ότι η εφαρμογή έχει κατασκευαστεί με ανάλογο τρόπο. Για παράδειγμα αν για κάποιο λόγο η σύνδεση στο διαδίκτυο διακοπή αυτό δεν επηρεάζει τον χρήστη ο οποίος συνεχίζει να χρησιμοποιεί την εφαρμογή κανονικά. Αυτό επιτυγχάνεται από τον περιηγητή ο οποίος κρατάει ένα αντίγραφο από τα αρχεία τα οποία είναι

απαραίτητα για την εκτός δικτύου χρήση της εφαρμογής ,στον υπολογιστή του χρήστη ,και τα χρησιμοποιεί όταν αυτό κριθεί απαραίτητο. Η συγκεκριμένη δυνατότητα δεν είναι διαθέσιμη σε όλες τις εφαρμογές που χρησιμοποιούν HTML5 αλλά μόνο σε αυτές που έχει υπάρξει πρόβλεψη για χρήση της εφαρμογής και εκτός διαδικτύου.

ΚΕΦΑΛΑΙΟ 2: Παρουσίαση της γλώσσας HTML

2.1 Εισαγωγή

Στη ρίζα της επιτυχίας του Ιστού βρίσκεται μια απλή, βασισμένη σε κείμενο γλώσσα σήμανσης που είναι εύκολη στην εκμάθηση και την οποία μπορεί να διαβάσει οποιαδήποτε συσκευή με έναν απλό φυλλομετρητή ιστού: η HTML. Όλες οι ιστοσελίδες απαιτούν τουλάχιστον κάποιον κώδικα HTML, χωρίς αυτόν δεν θα ήταν ιστοσελίδες.

Το όνομα της το πήρε από τα αρχικά της λέξης HTML δηλαδή HyperText Markup Language δηλαδή Γλώσσα Σήμανσης Υπερκειμένου. Όπως θα δούμε και παρακάτω η HTML χρησιμοποιείται για τον καθορισμό της εμφάνισης και του περιεχομένου της ιστοσελίδας. Πιο συγκεκριμένα, προκειμένου να δώσουμε μια πιο καθορισμένη μορφή στις ιστοσελίδες χρησιμοποιούμε τα αρχεία CSS(φύλλα στυλ) τα οποία χρησιμοποιούν την δική τους σύνταξη που δεν απέχει πολύ από την ίδια την HTML και φυσικά είναι άρρηκτα συνδεδεμένα με αυτή. Τόσο οι σελίδες HTML όσο και τα αρχεία CSS είναι αρχεία κειμένου, γεγονός που κάνει εύκολη την επεξεργασία τους(w3schools, n.d.).

Την HTML και CSS έρχεται με τη σειρά της να πλαισιώσει μια ακόμα γλώσσα του διαδικτύου η JavaScript. Η γλώσσα αυτή έχει γίνει πλέον αναπόσπαστο κομμάτι των σημερινών ιστοσελίδων αφού μας δίνει την δυνατότητα να προσθέσουμε λογική στις σελίδες μας αλλά και δυναμικότητα, φέρνοντας το χρήστη ένα βήμα πιο κοντά στην σελίδα και βοηθώντας τον να αλληλεπιδράσει καλύτερα με αυτή.

Στο κεφάλαιο αυτό θα γίνει μια βασική παρουσίαση των τριών αυτών δομικών στοιχείων των Ιστοσελίδων. Αξίζει να σημειωθεί ωστόσο, ότι στο διαδίκτυο του σήμερα έχουν αναπτυχθεί και χρησιμοποιούνται πολλές άλλες τεχνολογίες και γλώσσες προγραμματισμού, κάποιες από τις οποίες θα παρουσιάσουμε επιγραμματικά στα επόμενα κεφάλαια, και αφορούν κυρίως το server-side κομμάτι μιας ιστοσελίδας.

2.2 Η ιστορία της γλώσσας

Η ιστορία της γλώσσας HTML ξεκινάει το 1980 από τον Λονδρέζο φυσικό και προγραμματιστή Τίμοθι Τζον Μπέρνερς Λι κατά την περίοδο που εργαζόταν στο ευρωπαϊκό ερευνητικό κέντρο CERN ως Σύμβουλος Μηχανικός Προγραμματισμού. Ιδέα για την δημιουργία της γλώσσας αποτέλεσε το βασικό πρόβλημα του διαμοιρασμού των διαφόρων εγγράφων μεταξύ των ερευνητών του ιδρύματος.

Το 1993 με την υποστήριξη του IETF (Internet Engineering Task Force), οργανισμού που δημιουργεί και προωθεί standards για το διαδίκτυο, εκδίδεται το πρώτο πρότυπο της HTML. Το 1994 ο IETF δημιουργεί μια ομάδα με σκοπό την ανάπτυξη της γλώσσας. Ακολούθως, το Νοέμβριο του 1995 έχουμε την έκδοση της HTML 2.0. Στη συνέχεια αναλαμβάνει δράση το W3C. Τον Ιανουάριο του 1997 εκδίδει την HTML 3.2 και τον Δεκέμβριο του 1997 την 4.0, με την βελτιωμένη έκδοση 4.01 να κάνει την εμφάνιση της τον Μάιο του 2000. Πάνω από 25 χρόνια εξέλιξης έχουν από τότε που κυκλοφόρησε για πρώτη φορά η HTML, με revisions της να προσθέτουν μορφοποιήσεις κειμένου μέσω της CSS καθώς και πολλές άλλες δυνατότητες όπως προσθήκη εικόνων. Μετά από την έκδοση 4.0 κυκλοφορεί η βελτιωμένη

XHTML με όχι και τόσο συνταρακτικές διαφορές από τις προηγούμενες εκδόσεις. Ο περιορισμένος όγκος δεδομένων που μπορούσαν να επεξεργαστούν οι ηλεκτρονικοί υπολογιστές, οι πολύ αργές διαδικτυακές συνδέσεις καθώς και το ότι το ίδιο το Διαδίκτυο δεν ήταν και τόσο ευρέως διαδεδομένο, ήταν οι κύριοι λόγοι που οδήγησαν σε στασιμότητα.

Το διαδίκτυο όμως δεν έμεινε στάσιμο. Η ταχύτητά του μετριέται πλέον σε Mbps. Οι υπολογιστές είναι πιο γρήγοροι από όσο θα μπορούσε να φανταστεί κανείς και εμείς χρησιμοποιούμε ακόμη και τα κινητά μας για να περιηγηθούμε στο διαδίκτυο, το οποίο έχει γίνει πλέον αναπόσπαστο κομμάτι της καθημερινότητάς μας. Όλα αυτά κατέστησαν την αντικατάσταση της HTML αναγκαία. Το W3C (Web Standards Project), ξεκίνησε το 2007 την ανάπτυξη της HTML5. Ο βασικός στόχος του βελτιωμένου αυτού προτύπου είναι να μπορεί να χειριστεί οποιαδήποτε μορφή περιεχομένου και αλληλεπίδρασης μέσα από τον φυλλομετρητή του χρήστη.

2.3 Τα βασικά στοιχεία της HTML

Η HTML δεν ορίζεται ακριβώς ως γλώσσα προγραμματισμού αλλά ως γλώσσα Σήμανσης (Markup Language). Είναι δηλαδή μία γλώσσα η οποία βοηθά στην αναπαράσταση ενός συνόλου εικόνων, κειμένου κ.α. στον φυλλομετρητή μας.

Τα κύρια στοιχεία της HTML τα οποία δομούν μια ιστοσελίδα ονομάζονται ετικέτες. Οι HTML ετικέτες (tags) είναι λέξεις κλειδιά που περικλείονται μέσα σε σύμβολα «μεγαλύτερο από» και «μικρότερο από» (για παράδειγμα <head>), μέσα στο περιεχόμενο της ιστοσελίδας. Οι ετικέτες HTML συνήθως λειτουργούν ανά ζεύγη (για παράδειγμα <div> και </div>), με την πρώτη να ονομάζεται ετικέτα έναρξης και τη δεύτερη ετικέτα λήξης (ή σε άλλες περιπτώσεις ετικέτα ανοίγματος και ετικέτα κλεισίματος αντίστοιχα). Ανάμεσα στις ετικέτες, οι σχεδιαστές ιστοσελίδων μπορούν να τοποθετήσουν κείμενο, πίνακες, εικόνες κλπ. Στη συνέχεια θα παρουσιάσουμε τις βασικές ετικέτες της HTML και θα περιγράψουμε συνοπτικά την λειτουργία που η κάθε μια από αυτές επιτελεί (LaGrone, 2013), (Robson, 2010).

Οι βασικές ετικέτες μιας που όλες οι ιστοσελίδες θα πρέπει να έχουν είναι τέσσερις και επεξηγούνται στον πίνακα (w3schools, n.d.):

ΒΑΣΙΚΑ ΕΤΙΚΕΤΕΣ		
Τύπος εγγράφου	<HTML></HTML>	Ορίζει την αρχή και το τέλος.
Τίτλος	<TITLE></TITLE>	Εμφανίζεται στη μπλε μπάρα του φυλλομετρητή. Πρέπει να βρίσκεται στην επιφυλλίδα.
Επιφυλλίδα	<HEAD></HEAD>	Εδώ μπαίνει ο τίτλος, περιγραφή της σελίδας, τα styles και η javascript.
Βασικός Κορμός	<BODY></BODY>	το κύριο υλικό της σελίδας.

Η δομή μιας ιστοσελίδας, δηλαδή η διάταξη του περιεχομένου της καθορίζεται από τις παρακάτω ετικέτες:

ΕΤΙΚΕΤΕΣ ΔΟΜΗΣ		
Επικεφαλίδα	<H?></H?>	Δέχεται 6 ορισμούς: H1ως καιH6.
Στοιχισήεπικεφαλίδας	<H?ALIGN=LEFT CENTER RIGHT></H?	
Υποδιαίρεση κειμένου	<DIV></DIV>	
Στοιχισήυποδιαίρεσης	<DIVALIGN=LEFT RIGHT CENTER JUSTIFY></DIV>	
Καθορισμένο περιεχόμενο		
Παράθεση ημείωσης	<BLOCKQUOTE></BLOCKQUOTE>	Συνήθως μεεσοχή παραγράφου.
Σημείωση	<Q></Q>	Για σύντομεςσημειώσεις.
Έμφαση		Συνήθως εμφανίζει πλάγια (italics) γράμματα.
Έντονηέμφαση		Συνήθως εμφανίζει έντονα (bold) γράμματα.
Κώδικας	<CODE></CODE>	Για παράθεση γραμμών κώδικα.
Δείγμα	<SAMP></SAMP>	
Είσοδος από πληκτρολόγιο	<KBD></KBD>	
Μεταβλητή	<VAR></VAR>	
Ορισμός	<DFN></DFN>	Δε χρησιμοποιείται ευρέως.
Διεύθυνση συγγραφέα	<ADDRESS></ADDRESS>	
Μεγάλαγράμματα	<BIG></BIG>	
Μικράγράμματα	<SMALL></SMALL>	
Παρεμβολή	<INS></INS>	Σημειώνει τις προσθήκες σε νέα έκδοση.
Ώρααλλαγής	<INSDATETIME=":::"></INS>	
Σχόλια	<INSCITE="URL"></INS>	
Διαγραφή		Σημειώνει τις διαγραφές σε νέα έκδοση.
Ώραδιαγραφής	<DELDATETIME=":::">	
Σχόλιαδιαγραφής	<DELCITE="URL">	

Αρκτικόλεξο	<ACRONYM></ACRONYM>	
Σύντμηση	<ABBR></ABBR>	

Για τη μορφοποίηση του κειμένου μιας ιστοσελίδας χρησιμοποιούνται οι παρακάτω ετικέτες:

ΕΤΙΚΕΤΕΣ ΜΟΡΦΟΠΟΙΗΣΗ ΚΕΙΜΕΝΟΥ		
Έντονη γραφή		Δοκιμή
Πλάγια γραφή	<I></I>	Δοκιμή
Υπογράμμιση	<U></U>	Δοκιμή
Διαγραφή	<STRIKE></STRIKE>	Δοκιμή
Διαγραφή	<S></S>	Δοκιμή
Εκθέτης		Βάση Δοκιμή
Δείκτης		Βάση Δοκιμή
Γραφομηχανή	<TT></TT>	Δοκιμή
Προμορφοποιημένο	<PRE></PRE>	Εμφανίζει τα κενά διαστήματα όπως έχουν γραφεί.
Πλάτος προμορφοποιημένου	<PREWIDTH=?></PRE>	Σε χαρακτήρες.
Στοιχισή στο κέντρο	<CENTER></CENTER>	Στη θέση του χρησιμοποιείται πλέον το <DIV ALIGN="CENTER"></DIV>
Αναβοσβήσιμο	<BLINK></BLINK>	Δοκιμή
Μέγεθος γραμμάτων		Με τιμές από 1 ως και 7.
Χρώμα γραμμάτων		Το χρώμα ορίζεται με τρεις (διψήφιους) δεκαεξαδικούς αριθμούς.
Επιλογή γραμματοσειράς		
Βάρος γραμμάτων		
Βασικό μέγεθος γραμμάτων	<BASEFONT SIZE=?>	Με τιμές από 1 ως και 7 και προεπιλογή το 3. Ισχύει για όλη τη σελίδα.

Ετικέτες που χρησιμοποιούνται για την τοποθέτηση στοιχείων στην ιστοσελίδα:

ΕΤΙΚΕΤΕΣ ΤΟΠΟΘΕΤΗΣΗΣ ΣΤΟΙΧΕΙΩΝ		
Πολλέςστήλες	<MULTICOLCOLS=?></MULTICOL>	
Απόσταση (διάκενο) στηλών	<MULTICOLGUTTER=?></MULTICOL>	
Πλάτοςστήλης	<MULTICOLWIDTH=?></MULTICOL>	
Διαχωριστής	<SPACER>	
Είδοςδιαχωριστή	<SPACERTYPE=HORIZONTAL VERTICAL BLOCK>	
Μέγεθοςδιαχωριστή	<SPACERSIZE=?>	
Διαστάσεις διαχωριστή	<SPACER WIDTH=?HEIGHT=?>	
Στοιχισηδιαχωριστή	<SPACERALIGN=LEFT RIGHT CENTER>	
Επίπεδο(layer)	<LAYER></LAYER>	
Όνομαεπιπέδου	<LAYERID="*"></LAYER>	
Θέσηεπιπέδου	<LAYER LEFT=?TOP=?></LAYER>	
Σειράεπικάλυψης	<LAYERZ-INDEX=?></LAYER>	
Θέσηεπικάλυψης	<LAYERABOVE="*"BELOW="*"></LAYER>	
Διαχωριστής	<SPACER>	
Είδοςδιαχωριστή	<SPACERTYPE=HORIZONTAL	
Μέγεθοςδιαχωριστή	<SPACERSIZE=?>	
Διαστάσεις διαχωριστή	<SPACER WIDTH=?HEIGHT=?>	
Στοιχισηδιαχωριστή	<SPACERALIGN=LEFT RIGHT CENTER>	

Ένα από τα πιο δυνατά χαρακτηριστικά της HTML είναι η δυνατότητα υπέρ-σύνδεσης των ιστοσελίδων. Η χρήση της ετικέτας <a> ευθύνεται για την δημιουργία του παγκοσμίου <<Ιστού>> όπως αυτός αναφέρετε σήμερα. Στον παρακάτω πίνακα παρουσιάζονται οι διάφορες ετικέτες συνδέσμου και πολυμέσων:

ΕΤΙΚΕΤΕΣ ΣΥΝΔΕΣΜΟΥ, ΓΡΑΦΙΚΑ ΚΑΙ ΗΧΟΥ		
Συνδεσμος- παραπομπήσε διεύθυνση	<AHREF="URL">	
Παραπομπήσε σημείοεγγράφου	<AHREF="URL#****">	Αν πρόκειται για άλλο έγγραφο.
	<AHREF="#****">	Αν πρόκειται για το τρέχον έγγραφο.
Παράθυρο όπουθ' ανοίξει οσύνδεσμος		
Ενέργεια με τοκλικ		JavaScript
Σύνδεσμοςγια αποστολήemail	<AHREF="mailto:@">	
Ορισμός θέσηςγια σύνδεσμο	<ANAME="****">	Για χρήση με την Παραπομπή.
Εμφάνισηεικόνας	<IMGSRC="URL">	
Στοιχισεικόνας		
Στοιχιση εικόναςσε σχέση με τοκείμενο		
Εναλλακτικόκείμενο αντί τηςεικόνας		Αν η εικόνα δεν εμφανιστεί.
		Σεpixels.
		Ως ποσοστό των διαστάσεων της
Πλαίσιοεικόνας		Σε pixels.
Ανανέωση περιεχομένουή μεταπήδηση σεάλλη σελίδα	<METAHTTP-EQUIV="Refresh" CONTENT="χρόνοςσεδευτερόλεπτα; URL=ονομασίασελίδας"> Μπαίνει μεταξύ <head> και </head>. Π.χ. <metahttp-equiv="refresh"	
Ενσωματωμένο αντικείμενο	<EMBEDSRC="URL">	Π.χ: κάποιο flash animation.
Μέγεθοςαντι- κειμένου	<EMBED SRC="URL" WIDTH=?HEIGHT=?>	
Αντικείμενο	<OBJECT></OBJECT>	
Παράμετροι	<PARAM>	

Οι ετικέτες διαχωρισμού δίνουν μια βασική στοίχιση του κειμένου:

ΕΤΙΚΕΤΕΣ ΔΙΑΧΩΡΙΣΤΕΣ		
Παράγραφος	<P></P>	Δεν είναι απαραίτητο κλείσιμο του tag.
Στοίχιση παραγράφου	<PALIGN=LEFT CENTER RIGHT></P>	
Πλήρηςστοίχιση παραγράφου	<PALIGN=JUSTIFY></P>	
Αλλαγήσειράς	 	
Χωρίς αναδιπλωση	<BRCLEAR=LEFT RIGHT ALL>	
Οριζόντια γραμμή	<HR>	
Στοίχιση γραμμής	<HRALIGN=LEFT RIGHT CENTER>	
Πάχος γραμμής	<HRSIZE=?>	Σε pixels.
Πλάτος γραμμής	<HRWIDTH=?>	Σε pixels.
Πλάτος γραμμής	<HRWIDTH="%">	Ως ποσοστό του πλάτους σελίδας.
Συμπαγής γραμμή	<HRNOSHADE>	Δίχως την τριδιάστατη όψη.
Χωρίς αλλαγές σειράς	<NOBR></NOBR>	
Αλλαγή γραμμής	<WBR>	

Ακολουθούν οι ετικέτες δημιουργίας λιστών:

ΕΤΙΚΕΤΕΣ ΛΙΣΤΩΝ		
Απλήλιστα		Το μπαίνει μπροστά από κάθε σειρά.
Συμπαγήςλιστα	<ULCOMPACT>	
Τύποςκουκίδας	<ULTYPE=DISC CIRCLE SQUARE>	Για όλη τηλιστα.
Τύποςκουκίδας	<LITYPE=DISC CIRCLE SQUARE>	Για τη σειρά και τιςεπόμενες.
Αριθμημένηλιστα		Το μπαίνει μπροστά από κάθε σειρά.
Συμπαγήςαριθμη μένηλιστα	<OLCOMPACT>	
Μορφήαριθμησης	<OLTYPE=A a I i 1>	Για όλη τηλιστα.

Μορφήαρίθμησης	<LITYPE=A a i 1>	Για τη σειρά και τιςεπόμενες.
Αριθμόςέναρξης αρίθμησης	<OLSTART=?>	Για όλη τηλίστα.
Αριθμόςέναρξης αρίθμησης	<LIVALUE=?>	Για τη σειρά και τιςεπόμενες.
Λίσταεπεξηγήσεων	<DL><DT><DD></DL>	(<DT>=όρος, <DD>=επεξήγηση).
Συμπαγήςλίστα επεξηγήσεων	<DLCOMPACT></DL>	
Λίσταεπιλογών	<MENU></MENU>	Πριν από κάθεσειρά.
Συμπαγήςλίστα επιλογών	<MENUCOMPACT></MENU>	
Λίστα περιεχομένων	<DIR></DIR>	Πριν από κάθεσειρά.
Συμπαγήςλίστα περιεχομένων	<DIRCOMPACT></DIR>	

Η HTML μας δίνει την δυνατότητα να ορίσουμε χρωματισμούς για το κείμενο καθώς και το φόντο του:

ΕΤΙΚΕΤΕΣ ΦΟΝΤΟΥ ΚΑΙ ΧΡΩΜΑΤΑ		
Επαναλαμβανόμενη εικόναφόντου	<BODYBACKGROUND="URL">	
Υδατογράφημα	<BODYBGPROPERTIES="FIXED">	
Χρώμαφόντου	<BODYBGCOLOR="#\$\$\$\$\$">	(order isred/green/blue).
Βασικόχρώμα κειμένου	<BODYTEXT="#\$\$\$\$\$">	
Χρώμασυνδέσμων	<BODYLINK="#\$\$\$\$\$">	
Χρώμασυνδέσμων πουέχουμε επισκεφτεί	<BODYVLINK="#\$\$\$\$\$">	
Ενεργόςσύνδεσμος	<BODYALINK="#\$\$\$\$\$">	

Για να μπορέσουμε να εισάγουμε στο κείμενο μιας ιστοσελίδας σύμβολα θα πρέπει να χρησιμοποιήσουμε μια από τις παρακάτω ετικέτες:

ΕΙΔΙΚΟΙ ΧΑΡΑΚΤΗΡΕΣ		
Ειδικοί χαρακτήρες	&#?;	όπου ? είναι ο κωδικός ISO 859-1.
<	<	
>	>	

&	&	
"	"	
RegisteredTM	®	®
RegisteredTM	®	®
Copyright	©	©
Copyright	©	©
Κενόδιάστημα (δεν επιτρέπει αναδίπλωση)	 	

Ένα καθοριστικό στοιχείο το οποίο δίνει στον χρήστη την δυνατότητα να αλληλεπιδράσει με την ιστοσελίδα και να ανταλλάξει δεδομένα είναι οι φόρμες :

ΕΤΙΚΕΤΕΣ ΦΟΡΜΩΝ		
Ορισμός φόρμας	<FORM ACTION="URL"METHOD=GET POST></FORM>	
'Ανέβασμα' αρχείου	<FORMENCTYPE="multipart/form-data"></FORM>	
Πεδίο εισαγωγής δεδομένων	<INPUTTYPE="TEXT PASSWORD CHECKBOX RADIO FILE BUTTON IMAGE HIDDEN SUBMIT RESET">	
Όνομα πεδίου	<INPUTNAME="****">	
Τιμή πεδίου	<INPUTVALUE="****">	
Τσεκάρισμα	<INPUTCHECKED>	Για τετράγωνα και κυκλικά πεδία τσεκαρίσματος.
Μέγεθος πεδίου	<INPUTSIZE=?>	Σε χαρακτήρες.
Μέγιστο μέγεθος	<INPUTMAXLENGTH=?>	Σε χαρακτήρες.
Κουμπι	<BUTTON></BUTTON>	
Όνομα κουμπιού	<BUTTONNAME="****"></BUTTON>	
Τύπος κουμπιού	<BUTTONTYPE="SUBMIT RESET BUTTON"></BUTTON>	
Προεπιλεγμένη τιμή	<BUTTONVALUE="****"></BUTTON>	
Ετικέτα	<LABEL></LABEL>	
Ετικέτα επιλογής	<LABELFOR="****"></LABEL>	
Λίστα επιλογής	<SELECT></SELECT>	

Όνομαλίστας	<SELECTNAME="****"></SELECT>	
Αριθμόςεπιλογών	<SELECTSIZE=?></SELECT>	
Πολλαπλήεπιλογή	<SELECTMULTIPLE>	Για περισσότερες από μία επιλογές.
Επιλογή	<OPTION>	Διαθέσιμα στοιχεία προς επιλογή.
Προεπιλογή	<OPTIONSELECTED>	
Τιμήεπιλογής	<OPTIONVALUE="****">	
Ομαδοποίηση επιλογών	<OPTGROUPLABEL="****"></OPTGROUP>	
Μέγεθοςπεδίου εισαγωγήςκειμένου	<TEXTAREA ROWS=?COLS=?></TEXTAREA>	
Ονομασίαπεδίου εισαγωγήςκειμένου	<TEXTAREANAME="****"></TEXTAREA>	
Αναδίπλωση κειμένου	<TEXTAREAWRAP=OFF HARD SOFT></TEXTAREA>	
Μέληομάδας	<FIELDSET></FIELDSET>	
Υπόμνημα	<LEGEND></LEGEND>	

Επιπλέον δυνατότητες οργάνωσης του περιεχομένου μιας ιστοσελίδας δίνουν οι πίνακες:

ΕΤΙΚΕΤΕΣ ΠΙΝΑΚΩΝ		
Ορισμόςπίνακα	<TABLE></TABLE>	
Στοιχισήπίνακα	<TABLEALIGN=LEFT RIGHT CENTER>	
Περιγραφή πίνακα	<TABLEBORDER></TABLE>	Υπαρκτό ήμη.
Περιγραφή πίνακα	<TABLEBORDER=?></TABLE>	Ορισμός πάχους.
Απόσταση (διάκενο)	<TABLECELLSPACING=?>	
Εσωτερικό περιθώριο	<TABLECELLPADDING=?>	
Επιθυμητό πλάτος	<TABLEWIDTH=?>	Σε pixels.
Ποσοστό	<TABLEWIDTH=%>	Ποσοστό επί του πλάτους της σελίδας.
Χρώμα πίνακα	<TABLEBGCOLOR="\$\$\$\$\$"></TABLE>	

Πλαίσιοπίνακα	<TABLEFRAME=VOID ABOVE BELOW HSIDES LHS RHS VSIDES BOX BORDER></TABLE>	
Χάρακαςπίνακα	<TABLERULES=NONE GROUPS ROWS COLS ALL></TABLE>	
Χρώμα περιγράμματος	<TABLEBORDERCOLOR="\$\$\$\$\$\$">	
Σκούροπερίγραμμα	<TABLEBORDERCOLORDARK="\$\$\$\$\$\$">	
Ανοιχτόχρωμο περίγραμμα	<TABLEBORDERCOLORLIGHT="\$\$\$\$\$\$">	
Σειράπίνακα	<TR></TR>	
Στοιχισησειράς πίνακα	<TRALIGN=LEFT RIGHT CENTER MIDDLE BOTTOM>	
Κελίπίνακα	<TD></TD>	Πρέπει να εμπεριέχεται σε ορισμό σειράς.
Στοιχισηκελιού	<TD ALIGN=LEFT RIGHT CENTERVALIGN=TOP MIDDLE BOTTOM>	
Κείμενοκελιού χωρίςαναδίπλωση	<TDNOWRAP>	
Επέκτασησε στήλες	<TDCOLSPAN=?>	
Επέκτασησε σειρές	<TDROWSPAN=?>	
Επιθυμητόπλάτος κελιού	<TDWIDTH=?>	Σε pixels.
Πλάτος κελιούσε ποσοστό	<TDWIDTH="%">	Επί του πλάτους του πίνακα.
Χρώμακελιού	<TDBGCOLOR="#\$\$\$\$\$\$">	
Επικεφαλίδα στήλης	<TH></TH>	Έντονα και με στοίχιση στο κέντρο.
Στοιχιση επικεφαλίδας	<THALIGN=LEFT RIGHT CENTER MIDDLE BOTTOM>	
Επικεφαλίδαδίχως αναδίπλωση	<THNOWRAP>	
Επέκταση επικεφαλίδαςσε στήλες	<THCOLSPAN=?>	
Επέκταση επικεφαλίδαςσε σειρές	<THROWSPAN=?>	
Επιθυμητόπλάτος επικεφαλίδας	<THWIDTH=?>	Σε pixels.
Πλάτοςσε ποσοστό	<THWIDTH="%">	Επί του πλάτους του πίνακα.
Χρώμακελιού επικεφαλίδας	<THBGCOLOR="#\$\$\$\$\$\$">	
Σώμαπίνακα	<TBODY>	

Υποφυλίδιδαπίνακα	<TFOOT></TFOOT>	Προηγείται του THEAD>.
Επιφυλίδιδαπίνακα	<THEAD></THEAD>	
Επεξήγησηπίνακα	<CAPTION></CAPTION>	
Στοιχισή επεξήγησης	<CAPTIONALIGN=TOP BOTTOM LEFT RIGHT>	
Στήλη	<COL></COL>	
Επέκτασηστηλών	<COLSPAN=?></COL>	
Πλάτοςστήλης	<COLWIDTH=?></COL>	
Πλάτος στήληςσε ποσοστό	<COLWIDTH="%"></COL>	
Ομάδαστηλών	<COLGROUP></COLGROUP>	
Επέκτασηομάδας στηλών	<COLGROUPSPAN=?></COLGROUP>	
Πλάτοςομάδας	<COLGROUPWIDTH=?></COLGROUP>	
Πλάτος ομάδαςσε ποσοστό	<COLGROUPWIDTH="%"></COLGROUP>	

Παρακάτω ακολουθούν πίνακες με τα επιπλέον στοιχεία: Frames(πλαίσια), Εισαγωγής script και applet καθώς και διάφορες άλλες ετικέτες. Στο παρόν εδάφιο δεν θα δώσουμε ιδιαίτερη έμφαση σε αυτές τις ετικέτες.

ΕΤΙΚΕΤΕΣ ΠΛΑΙΣΙΩΝ		
Ξελίδα μεπλαίσια	<FRAMESET></FRAMESET>	Αντί του<BODY>.
Υψοσειρών	<FRAMESETROWS=,,,></FRAMESET>	(Σε pixels ή%).
Υψοσειρών	<FRAMESETROWS=*></FRAMESET>	* = Σχετικό μέγεθος. Συνήθως όσο περισσεύει.
Πλάτοςστηλών	<FRAMESETCOLS=,,,></FRAMESET>	(pixels or%).
Πλάτοςστηλών	<FRAMESETCOLS=*></FRAMESET>	* = Σχετικό μέγεθος. Συνήθως όσο περισσεύει.
Περιγράμμα	<FRAMESETFRAMEBORDER="yes no"></FRAMESET>	
Πλάτος περιγράμματος	<FRAMESETBORDER=?></FRAMESET>	
Χρώμα περιγράμματος	<FRAMESETBORDERCOLOR="#\$\$\$\$\$"></FRAMESET>	

Απόσταση (διάκενο) πλαισίων	<FRAMESETFRAMESPACING=?></FRAMESET>	
Ορισμός πλαισίου	<FRAME>	Περιεχόμενα μεμονωμένου πλαισίου.
Έγγραφο που θα εμφανιστεί	<FRAMESRC="URL">	
Όνομα πλαισίου	<FRAMENAME="****" _blank _self _parent _top>	
Πλάτος περιθωρίου	<FRAME MARGINWIDTH=?>	Αριστερά και δεξιά.
Ύψος περιθωρίου	<FRAME MARGINHEIGHT=?>	Πάνω και κάτω.
Εμφάνιση μπάρας κύλισης	<FRAME SCROLLING="YES NO AUTO">	
Σταθερό μέγεθος	<FRAME NORESIZE>	
Περιγράμμο πλαισίου	<FRAME FRAMEBORDER="yes no">	
Χρώμα περιγράμματος	<FRAME BORDERCOLOR="#\$\$\$\$\$">	
Περιεχόμενο εκτός πλαισίων	<NOFRAMES></NOFRAMES>	Για φυλλομετρητές που δεν υποστηρίζουν πλαίσια.
Εσωτερικό πλαίσιο	<IFRAME></IFRAME>	Όμοιες παράμετροι με το FRAME.
Διαστάσεις	<IFRAME WIDTH=?HEIGHT=?></IFRAME>	
Διαστάσεις	<IFRAME WIDTH=%" HEIGHT=%"></IFRAME>	

ΕΤΙΚΕΤΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΚΑΙ JAVA		
Πρόγραμμα	<SCRIPT></SCRIPT>	
	<SCRIPT SRC="URL"></SCRIPT>	
Είδος προγράμματος	<SCRIPT TYPE="****"></SCRIPT>	
Γλώσσα προγράμματος	<SCRIPT LANGUAGE="****"></SCRIPT>	
Άλλο περιεχόμενο	<NOSCRIPT></NOSCRIPT>	Αν δεν υποστηρίζονται scripts.
Applet	<APPLET></APPLET>	
Όνομα Applet	<APPLET CODE="****">	
Παράμετροι Applet	<APPLET PARAM NAME="****">	
Θέση Applet	<APPLET CODEBASE="URL">	

Ταυτότητα Applet	<APPLETNAME="****">	Για αναφορά-παραπομπές.
Εναλλακτικό κείμενο	<APPLETALT="****">	Αν δεν υποστηρίζεται java.
Στοιχισή Applet	<APPLETALIGN="LEFT RIGHT CENTER">	
Μέγεθος Applet	<APPLET WIDTH=?HEIGHT=?>	Σε pixels.
Απόσταση Applet από άλλα στοιχεία	<APPLET HSPACE=?VSPACE=?>	Σε pixels.
Server Script	<SERVER></SERVER>	

ΔΙΑΦΟΡΕΣ ΕΤΙΚΕΤΕΣ		
Σχόλιο	<!-- ***-->	Δεν εμφανίζεται.
Πρόλογος	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML4.0//EN">	
Κατάλογος αναζήτησης	<ISINDEX>	
Αποστολή αναζήτησης	<AHREF="URL?****">	Με κανονικό ?
URL του αρχείου	<BASEHREF="URL">	Πρέπει να βρίσκεται στην επικεφαλίδα.
Παράθυρο όπου θα κατευθύνονται οι σύνδεσμοι	<BASETARGET="****">	Πρέπει να βρίσκεται στην επικεφαλίδα.
Σχετική σελίδα	<LINK REV="****" REL="****" HREF="URL">	Πρέπει να βρίσκεται στην επικεφαλίδα.
Συνδεδεμένο αρχείο	<LINK TYPE="****" SRC="****"></LINK>	
Meta Πληροφορίες	<META>	Πρέπει να βρίσκεται στην επικεφαλίδα.
Ορισμός στυλ	<STYLE></STYLE>	

2.4 Γλώσσα μορφοποίησης CSS

Η CSS (Cascading Style Sheets - Διαδοχικά Φύλλα Στυλ) είναι μια γλώσσα υπολογιστή που ανήκει στην κατηγορία των γλωσσών φύλλων στυλ που χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που έχει γραφτεί με μια γλώσσα σήμανσης. Χρησιμοποιείται δηλαδή για τον έλεγχο της εμφάνισης μιας σελίδας που γράφτηκε σε γλώσσα HTML, Η CSS είναι μια γλώσσα υπολογιστή προορισμένη να αναπτύσσει στιλιστικά μια ιστοσελίδα δηλαδή να διαμορφώνει περισσότερα χαρακτηριστικά, χρώματα, στοιχισή και δίνει περισσότερες δυνατότητες σε σχέση με την HTML. Για μια καλοσχεδιασμένη ιστοσελίδα η χρήση της CSS κρίνεται ως απαραίτητη. Για ένα έγγραφο HTML θα υπάρχουν παραπάνω από ένα φύλλα

στυλ τα οποία περιέχουν δηλώσεις για την εμφάνιση ενός συγκεκριμένου στοιχείου(LaGrone, 2013). Το Φύλλο στυλ που εφαρμόζεται σε ένα έγγραφο μπορεί να προέρχεται από(Robson, 2010):

- το συγγραφέα μιας ιστοσελίδας
- το χρήστη του πλοηγού
- τον ίδιο τον πλοηγό, αν έχει το δικό του προκαθορισμένο φύλλο στυλ .

Συνεπώς για ένα HTML στοιχείο θα υπάρχουν παραπάνω από μια δηλώσεις που πιθανόν να είναι συγκρουόμενες(w3schools, n.d.). Η CSS για να επιλύσει παρόμοιες συγκρούσεις έχει καθορίσει μια αλληλουχία-σειρά στην οποία θα μπουν αυτές οι δηλώσεις και με βάση την οποία θα επιλεγεί π.χ. η δήλωση που είναι πρώτη στη σειρά.

Ο αλγόριθμος δημιουργίας αυτής της σειράς-αλληλουχίας είναι ο ακόλουθος:

1. Βρες όλες τις δηλώσεις που εφαρμόζονται στο στοιχείο που μας ενδιαφέρει. Οι δηλώσεις εφαρμόζονται στο στοιχείο αν ο επιλογέας του το επιλέξει.

2. Ταξινόμησε με βάση τη σημασία (κανονική η σημαντική) και προσέλευση (συγγραφέας, χρήστη ή πλοηγός χρήστη). Με αύξουσα σειρά προτεραιότητας:

- α. Δηλώσεις πλοηγού χρήστη
- β. Κανονικές δηλώσεις χρήστη
- γ. Κανονικές δηλώσεις συγγραφέα
- δ. Σημαντικές δηλώσεις συγγραφέα
- ε. Σημαντικές δηλώσεις χρήστη

3. Ταξινόμησε τις δηλώσεις ίδιας σημασίας και προέλευσης με κριτήριο την εξειδίκευση του επιλογέα: οι πιο εξειδικευμένοι επιλογείς υπερισχύουν των πιο γενικών. Τα ψεύδο-στοιχεία και οι ψεύδο-κλάσεις λογαριάζονται σαν κανονικά στοιχεία και κλάσεις αντίστοιχα.

4. Τέλος ταξινόμησε ανάλογα με τη σειρά καθορισμού: αν δύο δηλώσεις έχουν το ίδιο βάρος , προσέλευση και εξειδίκευση , αυτή που προσδιορίστηκε τελευταία επικρατεί. Οι δηλώσεις σε εισαγόμενα φύλλα στυλ θεωρούνται ότι δηλώνονται πριν από τις δηλώσεις στο ίδιο το φύλλο στυλ .

Αφού λοιπόν προκύψει μια σειρά-αλληλουχία κανόνων εμφάνισης που αφορούν το ίδιο στοιχείο θα επιλεγεί προς εφαρμογή (για την αποφυγή συγκρούσεων) η δήλωση που θα είναι τελευταία στην σειρά που αναλύθηκε πιο πάνω.

Στη συνέχεια θα παραθέσουμε τους βασικούς κανόνες σύνταξης της CSS(Robson, 2010):

Σχόλια

Αν και η ανάγκη για σχόλια στο CSS δεν είναι τόσο επιτακτική όσο σε γλώσσες προγραμματισμού, μπορούμε να εισάγουμε σχόλια μέσα σε /*...*/

Παράδειγμα: /* Αυτό είναι ένα σχόλιο */

CSS κανόνες

Ένας κανόνας CSS αποτελείται από δύο μέρη: Τον επιλογέα (CSS selector) ο οποίος αφορά το τι θα μορφοποιηθεί και τις ιδιότητες (CSS Properties) οι οποίες αφορούν το πώς θα μορφοποιηθεί. Η σύνταξη είναι η εξής:

```
επιλογέας
{
    ιδιότητα1: τιμή1;
    ιδιότητα2: τιμή2;
    ιδιότητα3: τιμή3;
    ...
}
```

Οι ιδιότητες καθώς και οι τιμές, που μπορεί να περιέχει ένας επιλογέας είναι ιδιαίτερα απλές αλλά πάρα πολλές σε πλήθος. Η παρουσίαση τους είναι έξω από τα πλαίσια αυτού του εδαφίου ωστόσο θα μπορούσε κανείς να ανατρέξει στην σελίδα <http://www.w3schools.com/cssref/> όπου και παρουσιάζονται αναλυτικά.

Μπορούμε να εισάγουμε CSS στη σελίδα μας με τρεις διαφορετικούς τρόπους, αναλόγως την περίπτωση. Οι τρόποι αυτοί, κατά σειρά φθίνουσα προτεραιότητας, είναι:

Inline CSS: Αν επιθυμούμε να μορφοποιήσουμε ένα στοιχείο μόνο, και δεν πρόκειται να χρειαστούμε αυτό το είδος μορφοποίησης για τίποτε άλλο στο site, μπορούμε να γράψουμε «χύμα» (δηλ. χωρίς το κομμάτι του CSS selector και χωρίς αγκύλες) CSS properties μέσα στο attribute style, το οποίο το διαθέτει σχεδόν κάθε στοιχείο html. Παραδείγματος χάριν, για να κάνουμε μια συγκεκριμένη παράγραφο κόκκινη, μπορούμε να προσθέσουμε style="color:red;" στο <p> tag (ολοκληρωμένα: <p style="color:red;">). Αξίζει να σημειωθεί, ότι αν κάποιες από τις ιδιότητες που θα γράψουμε εντός της attribute style ενός στοιχείου «συγκρούονται» με κανόνες CSS που έχουν οριστεί για αυτό αλλού και το αφορούν, τότε ο browser θα επιλέξει να εφαρμόσει αυτά που γράψαμε εντός του attribute style, ως πιο συγκεκριμένα για το στοιχείο αυτό.

CSS για μια συγκεκριμένη σελίδα: Πολλές φορές, μπορεί να θέλουμε να εφαρμόσουμε κάποιους κανόνες CSS μόνο για τη συγκεκριμένη σελίδα και όχι για όλο το site (ή να διαφοροποιήσουμε σε κάποιες ιδιότητες τους υπάρχοντες). Ένας τρόπος να το κάνουμε αυτό, για να μην δημιουργήσουμε ξεχωριστό αρχείο CSS είναι να εισάγουμε εντός των tags <head>...</head> της σελίδας τους κανόνες CSS μας μέσα σε <style> tags (με την html attribute type τους σε text/css). Παραδείγματος χάριν, για να κάνουμε το χρώμα γραμμάτων μιας συγκεκριμένης σελίδας γκρι, θα γράφαμε στο head της:

```
<style type="text/css"> body
{
color:gray;
}
</style>
```

Συνήθως όταν ξεκινάμε την ανάπτυξη ενός site, είναι πιο βολικό να χρησιμοποιήσουμε αυτό τον τρόπο αρχικά, μιας και πρέπει να επεξεργαζόμαστε μόνο ένα αρχείο αντί για δύο. Όταν τελειώνουμε την πρώτη σελίδα, συνήθως μεταφέρουμε το CSS που έχουμε γράψει σε εξωτερικό αρχείο (βλ. παρακάτω) ώστε να μπορούμε να χρησιμοποιήσουμε τους ίδιους κανόνες CSS και στις άλλες σελίδες του site μας, χωρίς να πρέπει φυσικά να τους κάνουμε copy-paste σε κάθε σελίδα.

Εξωτερικό αρχείο CSS: Η πιο «σωστή» χρήση του CSS και αυτή που θα έπρεπε να είναι η πρώτη λύση στην οποία θα καταφύγουμε, είναι η χρήση εξωτερικού αρχείου CSS. Για να τη χρησιμοποιήσουμε, γράφουμε τους κανόνες CSS μας σε ένα αρχείο με επέκταση css (πχ main.css) και στο head της κάθε σελίδας του site μας γράφουμε `<link href="main.css" type="text/css" />` (αν το όνομα του αρχείου CSS είναι main.css, αλλιώς προφανώς γράφουμε το σωστό όνομα αρχείου). Μπορούμε να έχουμε πολλά αρχεία css στην ίδια σελίδα, και μεγαλύτερη προτεραιότητα έχει πάντα αυτό που έχει γραφτεί τελευταίο.

2.5 Η γλώσσα Javascript

Η γλώσσα Javascript αποτελεί τη βασική γλώσσα προγραμματισμού της HTML αλλά και γενικότερα του διαδικτύου. Δημιουργήθηκε από την εταιρία Netscape και το αρχικό της όνομα ήταν LiveScript. Η ευκολία μάθησης της αποτελεί σημαντικό κίνητρο για κάποιον νεοεισερχόμενο στο χώρο του web. Επιπλέον, αν αναλογιστούμε την επέκταση του κυβερνοχώρου τα τελευταία χρόνια, το πλήθος των θέσεων εργασίας που εξαρτώνται από το διαδίκτυο και την αυξανόμενη ζήτηση γνώσης, τουλάχιστον των βασικών γνώσεων πάνω σε θέματα Web, από τις διάφορες εταιρίες συμπεραίνουμε την αξία της Javascript(w3schools, n.d.).

Συνεπώς, η Javascript (JS) είναι διερμηνευμένη γλώσσα προγραμματισμού και αποτέλεσε ένα εκ των πρωταρχικών εργαλείων για την εξυπηρέτηση της επικοινωνίας μεταξύ πελάτη και χρήστη στο διαδικτυακό χώρο. Πιο συγκεκριμένα, βοήθησε ουσιαστικά στο:

1. να επιλυθούν τα διάφορα ζητήματα (tasks) από την πλευρά του πελάτη (client-side scripts) γεφυρώνοντας τα αντίστοιχα αιτήματα με τον χρήστη,
2. να ανταλλάσσουν μεταξύ τους δεδομένα ασύγχρονα και
3. να επιτρέπεται η αλλαγή του περιεχομένου ενός εγγράφου με δυναμικό τρόπο από τους συμβαλλόμενους χρήστες.

Επιπλέον, η JS ως μία γλώσσα συγγραφής σεναρίων που χρησιμοποιείται για να προσθέσει εφέ και διαλογικότητα, αλληλεπίδραση και διαδραστικότητα στις ιστοσελίδες μας, πληροί τις εξής ιδιότητες:

1. βασίζεται στα πρωτότυπα (prototype-based),
2. είναι δυναμική,
3. χρησιμοποιεί ασθενείς τύπους και
4. έχει συναρτήσεις ως αντικείμενα πρώτης τάξης.

Ο κώδικας της JS γράφεται σε καθαρό κείμενο (ASCII μορφή) και ενσωματώνεται μέσα στον κώδικα της HTML, μπορεί δε να εκτελεστεί αμέσως ή όταν λαμβάνει χώρα ένα συμβάν (event). Δεν γίνεται μεταγλώττιση (compilation) του κώδικα της JS, αρκεί μόνο ο φυλλομετρητής (browser) να την υποστηρίξει. Προτού προχωρήσουμε στην αναφορά συγκεκριμένων στοιχείων της JS και τη χρήση αντιπροσωπευτικών παραδειγμάτων, θα λύσουμε μία πολύ γενική απορία και θα παραθέσουμε δύο εισαγωγικά χωρία κώδικα για μία πρώτη επαφή. Η συχνή απορία αφορά την ομοιότητα της JAVA με τη Javascript. Αν και ακούγονται ίδιες, η Java και η JavaScript δεν έχουν καμία απολύτως σχέση μεταξύ τους, ούτε στη σύνταξή τους σαν γλώσσες προγραμματισμού ούτε και στις εφαρμογές που χρησιμοποιούνται.

Επομένως, ακολουθούν τα δύο εισαγωγικά παράδειγμα που προαναφέραμε:

- Ένα πρώτο απλό παράδειγμα σε JavaScript είναι το εξής :

```
<html>
<body>
  <script type="text/javascript">
    document.write("Hello from JavaScript")
  </script>
</body>
</html>
```

Ο κώδικας της JavaScript περιέχεται ανάμεσα στα tags `<script>` και `</script>` και είναι χαρακτηριστικό (attribute) μπορούμε να χρησιμοποιήσουμε `type="text/javascript"` ή `language="JavaScript"`. Μέσα σ' ένα αρχείο HTML μπορούμε να έχουμε όσα σύνολα tags `<script>` και `</script>` χρειαστούμε, είτε στο τμήμα head ή στο τμήμα body του εγγράφου.

Η εντολή **document.write()** χρησιμοποιείται για να μπορούμε να εμφανίσουμε κείμενο στην οθόνη του φυλλομετρητή μέσα από εντολές της JavaScript. Το κείμενο *Hello from JavaScript* θα εμφανισθεί μαζί με το υπόλοιπο κείμενο του HTML κώδικα. Αν και εδώ δεν φαίνεται καθαρά η αξία της JavaScript, θα φανεί όταν χρησιμοποιήσουμε εντολές ελέγχου, όπως είναι η *if*, για να μπορούμε να εμφανίσουμε διαφορετικό κείμενο ανάλογα με την τιμή κάποιων μεταβλητών ή ανάλογα με τις επιλογές που έχει κάνει ο χρήστης.

- Για να εμφανίσουμε την ημερομηνία τελευταίας τροποποίησης μιας ιστοσελίδας, γράφουμε τον εξής κώδικα της JavaScript μέσα στον HTML κώδικα :

```
<html>

<body>

<script type="text/javascript">

    document.write("Ημερομηνία τελευταίας τροποποίησης της σελίδας : ")

    document.write(document.lastModified)

</script>

</body>

</html>
```

Στη συνέχεια θα αναφέρουμε κατά σειρά τα βασικά αντικείμενα της JS και τα χειριστήρια συμβάντος, τα οποία παίζουν το σημαντικότερο ρόλο για την κατανόηση και το στήσιμο μιας διαδραστικής εφαρμογής:

Το Αντικείμενο Array

Με το αντικείμενο αυτό μπορούμε να δημιουργήσουμε πίνακες (arrays), δηλ. μεταβλητές με δείκτη. Για να δημιουργήσουμε έναν πίνακα χρησιμοποιούμε τη λέξη κλειδί *new* και δηλώνουμε το μέγεθος (πλήθος των στοιχείων) του πίνακα, ως εξής :

```
Mathites = new Array(20);
```

Η αρίθμηση των δεικτών ενός πίνακα ξεκινάει από το 0 και έτσι για να αναφερθούμε στα περιεχόμενα ενός πίνακα και να πάρουμε (αποσπάσουμε) τις τιμές τους, μπορούμε να γράψουμε τα εξής :

```
FirstMathitis = Mathites[0];
```


SecondMathitis = Mathites[1]

Με την ιδιότητα *length* μπορούμε να βρούμε το πλήθος των στοιχείων ενός πίνακα, όπως :

PlithosMathiton = Mathites.length;

Οι πίνακες διαθέτουν τις εξής μεθόδους :

- *join()*, ενώνει όλα τα στοιχεία ενός πίνακα σε μια συμβολοσειρά και τα διαχωρίζει με κόμματα ή μ' έναν διαχωριστή (delimiter) της επιλογής μας.
- *reverse()*, τοποθετεί τα στοιχεία ενός πίνακα με ανάποδη σειρά.
- *sort()*, ταξινομεί τα στοιχεία ενός πίνακα.

Το Αντικείμενο String

Η JavaScript αντιμετωπίζει τις συμβολοσειρές σαν αντικείμενα string και μπορούμε να δημιουργήσουμε συμβολοσειρές με δύο τρόπους :

text = "Florina per sempre";

text = new String("Florina per sempre");

Μπορούμε να χρησιμοποιήσουμε την ιδιότητα *length* για να βρούμε το μήκος (πλήθος των χαρακτήρων) μιας συμβολοσειράς.

Οι συμβολοσειρές διαθέτουν τις εξής μεθόδους :

- *substring()*, επιστρέφει ένα κομμάτι μιας συμβολοσειράς.
- *substr()*, επιστρέφει ένα κομμάτι μιας συμβολοσειράς.
- *toUpperCase()*, μετατρέπει όλους τους χαρακτήρες μιας συμβολοσειράς σε κεφαλαία γράμματα.
- *toLowerCase()*, μετατρέπει όλους τους χαρακτήρες μιας συμβολοσειράς σε πεζά γράμματα.
- *indexOf()*, ελέγχει αν ένα κομμάτι κειμένου βρίσκεται ή όχι μέσα σε μια συμβολοσειρά, ξεκινώντας από την αρχή μιας συμβολοσειράς.
- *lastIndexOf()*, ελέγχει αν ένα κομμάτι κειμένου βρίσκεται ή όχι μέσα σε μια συμβολοσειρά, ξεκινώντας από το τέλος μιας συμβολοσειράς.
- *link()*, δημιουργεί μια σύνδεση (link) με το κείμενο μιας συμβολοσειράς.
- *anchor()*, δημιουργεί μια σύνδεση (link) τύπου άγκυρας (anchor) με το κείμενο μιας συμβολοσειράς.

Οι επόμενες μέθοδοι μπορούν να χρησιμοποιηθούν για να αλλάξουμε τη μορφοποίηση ενός κειμένου μέσα σε μια ιστοσελίδα, όπου το κείμενο αντιμετωπίζεται σαν αντικείμενο συμβολοσειράς :

- *text.big()*, αυξάνει το μέγεθος του κειμένου.
- *text.blink()*, κάνει το κείμενο να αναβοσβήνει (μόνο στον Netscape).

- *text.bold()*, κάνει το κείμενο έντονο.
- *text italics()*, κάνει το κείμενο να έχει πλάγια γραφή.
- *text.small()*, μειώνει το μέγεθος του κειμένου.
- *text.strike()*, κάνει το κείμενο να έχει διακριτή διαγραφή.
- *text.sub()*, μετατρέπει το κείμενο σε μορφή δείκτη (subscript).
- *text.sup()*, μετατρέπει το κείμενο σε μορφή εκθέτη (superscript).

Το Αντικείμενο Math

Το αντικείμενο Math είναι πολύ χρήσιμο για μαθηματικούς υπολογισμούς, αλλά δεν έχουμε τη δυνατότητα να δημιουργήσουμε ένα δικό μας αντικείμενο με τη βοήθειά του. Το αντικείμενο Math διαθέτει τις εξής ιδιότητες (σταθερές) :

- *Math.E*, επιστρέφει τον αριθμό e (περίπου 2.718281), που χρησιμοποιείται ως βάση στους φυσικούς (νεπέριους) λογαρίθμους.
- *Math.LN2*, επιστρέφει τον φυσικό λογάριθμο του 2 (περίπου 0.693147).
- *Math.LN10*, επιστρέφει τον φυσικό λογάριθμο του 10 (περίπου 2.302585).
- *Math.LOG2E*, επιστρέφει τον λογάριθμο του e με βάση το 2 (περίπου 1.442695).
- *Math.LOG10E*, επιστρέφει τον λογάριθμο του e με βάση το 10 (περίπου 0.434294).
- *Math.PI*, επιστρέφει τον αριθμό π (λόγος περιφέρειας ενός κύκλου προς τη διάμετρό του), με τιμή 3.1415926.
- *Math.SQRT1_2*, επιστρέφει την τετραγωνική ρίζα του ½ (περίπου 0.707106).
- *Math.SQRT2*, επιστρέφει την τετραγωνική ρίζα του 2 (περίπου 1.414213).

Το αντικείμενο Math διαθέτει τις εξής μεθόδους (συναρτήσεις) :

- *Math.acos()*, υπολογίζει το τόξο σε ακτίνια με βάσει του συνημιτόνου.
- *Math.asin()*, υπολογίζει το τόξο σε ακτίνια με βάσει του ημιτόνου.
- *Math.atan()*, υπολογίζει το τόξο σε ακτίνια με βάσει της εφαπτομένης.
- *Math.cos()*, υπολογίζει το συνημίτονο ενός αριθμού σε ακτίνια.
- *Math.sin()*, υπολογίζει το ημίτονο ενός αριθμού σε ακτίνια.
- *Math.tan()*, υπολογίζει την εφαπτομένη ενός αριθμού σε ακτίνια.
- *Math.exp()*, επιστρέφει τον αριθμό e υψωμένο σε μια δύναμη.
- *Math.log()*, επιστρέφει τον φυσικό (νεπέριο) ενός αριθμού, δηλ. με βάση το e.
- *Math.max()*, επιστρέφει τον μεγαλύτερο από δύο αριθμούς.
- *Math.min()*, επιστρέφει τον μικρότερο από δύο αριθμούς.
- *Math.abs()*, επιστρέφει την απόλυτη τιμή ενός αριθμού.
- *Math.cell()*, στρογγυλοποιεί έναν αριθμό στον αμέσως επόμενο του ακέραιο αριθμό.
- *Math.floor()*, στρογγυλοποιεί έναν αριθμό στον αμέσως προηγούμενό του ακέραιο αριθμό.
- *Math.pow()*, υπολογίζει τη δύναμη ενός αριθμού υψωμένου σ' έναν άλλον αριθμό.
- *Math.round()*, στρογγυλοποιεί έναν αριθμό στον πλησιέστερό του ακέραιο αριθμό.
- *Math.sqrt()*, υπολογίζει την τετραγωνική ρίζα ενός αριθμού.
- *Math.random()*, επιστρέφει έναν τυχαίο αριθμό ανάμεσα στο 0 και το 1.

Το Αντικείμενο Date

Με το αντικείμενο Date μπορούμε να χειρισθούμε ημερομηνίες και ώρες και μπορούμε είτε να ορίσουμε μια δική μας ημερομηνία είτε να πάρουμε την ημερομηνία του συστήματος και να βρούμε (αποσπάσουμε) την τρέχουσα ημέρα, τον τρέχοντα μήνα, το τρέχοντα έτος και πολλά άλλα.

Πρέπει να έχουμε υπόψη μας ότι η JavaScript αντιμετωπίζει τις ημερομηνίες σαν αριθμούς σε χιλιοστά του δευτερολέπτου, με αρχή από την 1/1/1970. Το αντικείμενο Date διαθέτει πολλές μεθόδους :

- *setDate()*, ορίζουμε την ημέρα του μήνα με μια δική μας τιμή.
- *setMonth()*, ορίζουμε τον μήνα με μια δική μας τιμή, όπου ο Ιανουάριος αντιστοιχεί στο 0 κοκ.
- *setYear()*, ορίζουμε το έτος με μια δική μας τιμή.
- *setFullYear()*, ορίζουμε το έτος με μια δική μας 4ψήφια τιμή.
- *setTime()*, ορίζουμε ολόκληρη την ώρα με μια δική μας τιμή.
- *setHours()*, ορίζουμε την ώρα με μια δική μας τιμή.
- *setMinutes()*, ορίζουμε τα λεπτά της ώρας με μια δική μας τιμή.
- *setSeconds()*, ορίζουμε τα δευτερόλεπτα της ώρας με μια δική μας τιμή.
- *getDate()*, διαβάζουμε την τρέχουσα ημέρα.
- *getMonth()*, διαβάζουμε τον τρέχοντα μήνα.
- *getFullYear()*, διαβάζουμε το τρέχον έτος.
- *getTime()*, διαβάζουμε ολόκληρη την τρέχουσα ώρα.
- *getHours()*, διαβάζουμε την ώρα.
- *getMinutes()*, διαβάζουμε τα λεπτά της ώρας.
- *getSeconds()*, διαβάζουμε τα δευτερόλεπτα της ώρας.
- *getTimeZoneOffset()*, επιστρέφει τη διαφορά της τοπικής ζώνης ώρας από την ώρα GMT.
- *toGMTString()*, επιστρέφει την ώρα σε κείμενο με βάση τη μορφή GMT.
- *toLocalString()*, επιστρέφει την ώρα σε κείμενο με βάση τη μορφή της τοπικής ώρας.
- *parse()*, μετατρέπουμε ένα κείμενο σ' ένα αντικείμενο ημερομηνίας.
- *UTC()*, μετατρέπουμε μια ημερομηνία σε μια μορφή ώρας UTC.

Το Αντικείμενο Window

Με το αντικείμενο window έχουμε πρόσβαση στο παράθυρο του τρέχοντα φυλλομετρητή. Αν έχουμε ανοικτά πολλά παράθυρα φυλλομετρητή ή αν κάποια ιστοσελίδα χρησιμοποιεί πλαίσια (frames), τότε θα υπάρχουν περισσότερα από ένα αντικείμενα window, οπότε θα πρέπει να τους δώσουμε ονόματα για να μπορούμε να τα ξεχωρίσουμε. Όταν αναφερόμαστε σε παράθυρα, χρήσιμα είναι και τα εξής :

- *self*, είναι το τρέχον παράθυρο, αυτό δηλαδή που περιέχει την τρέχουσα ιστοσελίδα.
- *top*, είναι το παράθυρο που βρίσκεται στην κορυφή και περιέχει (είναι γονέας) όλα τα άλλα παράθυρα.
- *parent*, είναι ένα παράθυρο που περιέχει άλλα παράθυρα, όπως συμβαίνει σε μια ιστοσελίδα που περιέχει πλαίσια, η οποία είναι γονέας (parent) όλων των παραθύρων πλαισίων που περιέχει.

- *frames[]*, είναι ένας πίνακας που περιέχει όλα τα πλαίσια μιας ιστοσελίδας, όπου για παράδειγμα `parent.frames[0]` είναι το πρώτο παράθυρο πλαίσιο κοκ.

Το αντικείμενο `window` διαθέτει και τα εξής τρία αντικείμενα παιδιά :

- *location*, είναι η θέση (URL) του HTML εγγράφου που εμφανίζεται μέσα στο παράθυρο της ιστοσελίδας.
- *history*, περιέχει μια λίστα από τα URLs που έχουμε επισκεφθεί.
- *document*, είναι το τρέχον έγγραφο ιστοσελίδας.

Το Αντικείμενο Location

Το αντικείμενο `location` περιέχει πληροφορίες σχετικά με την τρέχουσα διεύθυνση URL, τα περιεχόμενα της οποίας εμφανίζονται μέσα στο παράθυρο του φυλλομετρητή. Διαθέτει τις εξής ιδιότητες :

- *protocol*, είναι το πρωτόκολλο που υπάρχει στη διεύθυνση URL.
- *hostname*, είναι το όνομα του κεντρικού υπολογιστή (host).
- *port*, είναι η πόρτα (θύρα) επικοινωνίας.
- *host*, είναι ο συνδυασμός του `hostname` και της πόρτας (port).
- *pathname*, είναι η διαδρομή και το όνομα του αρχείου ιστοσελίδας.
- *href*, είναι ολόκληρη η διεύθυνση URL.

Το Αντικείμενο History

Το αντικείμενο `history` περιέχει πληροφορίες σχετικά με τις διευθύνσεις URL που έχουμε επισκεφθεί. Περιέχει τις εξής μεθόδους :

- *back()*, πηγαίνει στην προηγούμενη θέση της λίστας ιστορικού.
- *forward()*, πηγαίνει στην επόμενη θέση της λίστας ιστορικού.
- *go()*, πηγαίνει σε μια συγκεκριμένη θέση της λίστας ιστορικού.

Το Αντικείμενο Document

Το αντικείμενο `document` παριστάνει το τρέχον έγγραφο του παραθύρου του φυλλομετρητή. Περιέχει τα εξής αντικείμενα παιδιά :

- *forms[]*, είναι ένας πίνακας που περιέχει όλες τις φόρμες που υπάρχουν μέσα στο έγγραφο.
- *links[]*, είναι ένας πίνακας που περιέχει όλες τις συνδέσεις που υπάρχουν μέσα στο έγγραφο.
- *anchors[]*, είναι ένας πίνακας που περιέχει όλες τις συνδέσεις τύπου άγκυρας που υπάρχουν μέσα στο έγγραφο.

- *images[]*, είναι ένας πίνακας που περιέχει όλες τις εικόνες που υπάρχουν μέσα στο έγγραφο.
- *applets[]*, είναι ένας πίνακας που περιέχει όλα τα Java applets που υπάρχουν μέσα στο έγγραφο.

Το Αντικείμενο Navigator

Το αντικείμενο navigator δεν είναι παιδί ούτε γονέας κανενός άλλου αντικειμένου και περιέχει πληροφορίες σχετικά με την έκδοση του φυλλομετρητή, όπως :

- *appCodeName*, είναι το όνομα κωδικού του φυλλομετρητή, που είναι συνήθως το Mozilla.
- *appName*, είναι το πλήρες όνομα του φυλλομετρητή.
- *appVersion*, είναι ο αριθμός της έκδοσης του φυλλομετρητή.

Δημιουργία Αντικειμένου

Για να μπορέσουμε να δημιουργήσουμε ένα δικό μας αντικείμενο, υπάρχουν οι εξής λέξεις κλειδιά :

- *new*, δημιουργούμε ένα καινούργιο αντικείμενο.
- *this*, αναφερόμαστε στο τρέχον αντικείμενο.
- *with*, κάνουμε προεπιλεγμένο ένα αντικείμενο σε μια ομάδα εντολών για να μην είμαστε αναγκασμένοι να το γράφουμε συνέχεια.

Για να μπορέσουμε να δημιουργήσουμε ένα δικό μας αντικείμενο, γράφουμε πρώτα μια συνάρτηση δημιουργίας αντικειμένων (constructor), με την οποία δίνουμε τιμές στις ιδιότητες του αντικειμένου με τη λέξη κλειδί this, ως εξής :

```
function FullName (firstname, lastname) {
    this.firstname = firstname;
    this.lastname = lastname;
}
```

Μετά, για να μπορέσουμε να δημιουργήσουμε ένα στιγμιότυπο (instance) αυτού του αντικειμένου, χρησιμοποιούμε λέξη κλειδί new, ως εξής :

```
John01 = new FullName("John", "Georgiadis");
```

Τροποποίηση Αντικειμένου

Με τη λέξη κλειδί `prototype` μπορούμε να προσθέσουμε κι άλλες ιδιότητες στο πρωτότυπο ενός αντικειμένου, δηλ. στον ορισμό του, και έτσι όλα τα αντικείμενα αυτού του τύπου που θα δημιουργήσουμε θα περιλαμβάνουν τις ιδιότητες αυτές.

Για παράδειγμα, για να προσθέσουμε την ιδιότητα `fathersname` στο αντικείμενο `FullName` που δημιουργήσαμε προηγουμένως, γράφουμε την εξής εντολή :

```
FullName.prototype.fathersname = "George";
```

Εν συνεχεία έπονται τα χειριστήρια συμβάντος (event handlers), από τα οποία αντλεί τη δύναμή της η JavaScript. Παρακολουθώντας τις ενέργειες του χρήστη, η JavaScript μπορεί να κάνει επιβεβαίωση ή κάποια ενέργεια ανάλογα με τις καταχωρήσεις ή τις κινήσεις του χρήστη και αυτό χωρίς να χρειασθεί να μεσολαβήσει ο server, όπως συμβαίνει με τις δυναμικά παραγόμενες ιστοσελίδες. Οι ενέργειες του χρήστη σε μια ιστοσελίδα (HTMLpage) περιορίζονται στην κίνηση του ποντικιού και στις καταχωρήσεις (input) στα στοιχεία (πεδία) των φορμών (formelements). Επομένως, καταγράφουμε τα εξής event handlers:

onBlur - Το συμβάν αυτό λαμβάνει χώρα όταν χάνεται η εστίαση (lost focus), δηλ. όταν απομακρυνόμαστε είτε με κλικ με το ποντίκι είτε πατώντας το πλήκτρο tab σ' ένα πεδίο κειμένου (text field) ή σε μια περιοχή κειμένου (text area) ή και σ' ένα πλαίσιο λίστας (select) μιας φόρμας.

onChange - Το συμβάν αυτό λαμβάνει χώρα όταν τροποποιείται το περιεχόμενο ενός πεδίου κειμένου (text field) ή μιας περιοχής κειμένου (text area) ή και ενός πλαισίου λίστας (select) μιας φόρμας, πριν ακόμα απομακρυνθούμε από το στοιχείο αυτό.

onClick - Το συμβάν αυτό λαμβάνει χώρα όταν κάνουμε κλικ με το ποντίκι σ' ένα αντικείμενο, όπως είναι για παράδειγμα ένα πλήκτρο εντολής (button) ή ένα πλαίσιο ελέγχου (checkbox).

onFocus - Το συμβάν αυτό λαμβάνει χώρα όταν κάνουμε εστίαση (focus) μέσα σ' ένα πεδίο μιας φόρμας. Μπορούμε να εστιάσουμε (receivefocus) σ' ένα στοιχείο (πεδίο) μιας φόρμας είτε πατώντας επανειλημμένα το πλήκτρο tab μέχρι να φθάσουμε στο επιθυμητό πεδίο της φόρμας ή κάνοντας κλικ με το ποντίκι μέσα στο συγκεκριμένο πεδίο. Η επιλογή κάποιου κειμένου μέσα σ' ένα πεδίο δημιουργεί το event select.

onLoad - Το συμβάν αυτό λαμβάνει χώρα όταν τελειώσει το φόρτωμα μιας ιστοσελίδας σ' ένα παράθυρο ή όταν τελειώσει το φόρτωμα όλων των πλαισίων (frames) που βρίσκονται μέσα σ' ένα tag <frameset>.

onMouseOver - Το συμβάν αυτό λαμβάνει χώρα όταν ο δείκτης του ποντικιού τοποθετείται πάνω από ένα αντικείμενο ή και από έναν σύνδεσμο (link).

onSelect - Το συμβάν αυτό λαμβάνει χώρα όταν επιλέγουμε ένα μέρος ή όλο το κείμενο ενός πεδίου κειμένου (text field) ή μιας περιοχής κειμένου (textarea).

onSubmit - Το συμβάν αυτό λαμβάνει χώρα όταν κάνουμε κλικ με το ποντίκι σ' ένα πλήκτρο υποβολής (submit button) για να υποβάλλουμε μια φόρμα. Το χειριστήριο

συμβάντος πρέπει να επιστρέφει την τιμή true για να μπορέσουν να υποβληθούν (σταλούν) τα περιεχόμενα της φόρμας στον server. Αν επιστρέφει την τιμή false, δεν θα μπορέσει να γίνει η υποβολή των στοιχείων της φόρμας.

onUnload - Το συμβάν αυτό λαμβάνει χώρα όταν φεύγουμε από ένα έγγραφο (ιστοσελίδα). Κανονικά θα πρέπει να τοποθετείται μέσα σ' ένα από τα tags <body> ή <frameset>.

2.6 Ανάλυση και ανάπτυξη Ιστοσελίδων

Είναι πολύ βασικό να κατανοήσουμε την διαφορά των δυναμικών ιστοσελίδων με τις στατικές πριν ξεκινήσουμε ανάπτυξη της δικής μας ιστοσελίδας, αφού θα πρέπει να καθορίσουμε σε ποια από τις δύο κατηγορίες αυτή ανήκει και να προχωρήσουμε αναλόγως στην υλοποίησή της. Στην συνέχεια θα πρέπει να αναλύσουμε διεξοδικά τον τρόπο ανάπτυξης της. Επομένως, έχουμε τον εξής διαχωρισμό(wikipedia, n.d.):

1. Στατικές ιστοσελίδες ονομάζονται αυτές που αποτελούνται κυρίως από αρχεία HTML και CSS που μεταδίδονται αυτούσια από τον διακομιστή που τις φιλοξενεί, χωρίς να υπάρχει κάποια επεξεργασία πριν. Μια στατική ιστοσελίδα είναι πολύ εύκολο να αναγνωριστεί αφού κάθε φορά που την επισκεπτόμαστε η εμφάνισή της παραμένει ίδια και αυτή δεν αλληλεπιδρά με τον χρήστη. Πλέον δεν συναντάμε συχνά τέτοιες ιστοσελίδες σε σχέση με παλαιότερα, αφού έχουν πολύ περιορισμένες δυνατότητες και δεν ανταποκρίνονται στις ανάγκες του σύγχρονου χρήστη. Πλέον χρησιμοποιούνται μόνο σε σπάνιες περιπτώσεις που αφορούν την προβολή πολύ απλών πραγμάτων, όπως πχ για να δηλωθεί ότι μια ιστοσελίδα είναι υπό κατασκευή κλπ.
2. Δυναμική είναι μια ιστοσελίδα που χρησιμοποιεί μία ή περισσότερες γλώσσες προγραμματισμού για να επεξεργαστεί το περιεχόμενο των αρχείων που θα μεταφέρει στον χρήστη για προβολή. Πλέον δυναμικές είναι οι περισσότερες ιστοσελίδες που υπάρχουν αυτή τη στιγμή στο διαδίκτυο, αφού μπορούν να καλύψουν ένα πολύ ευρύ φάσμα αναγκών(Duckett, 2014). Σε αυτή την κατηγορία ανήκουν οι ιστοσελίδες που αλληλεπιδρούν με τον χρήστη, αυτές που έχουν την δυνατότητα να διαχειρίζονται δεδομένα και γενικότερα να τροποποιούν το αποτέλεσμα που βλέπει ο τελικός χρήστης ανάλογα με την εφαρμογή. Κάποια χαρακτηριστικά παραδείγματα τέτοιων ιστοσελίδων μπορεί να είναι ένα ενημερωτικό site, ένα ηλεκτρονικό κατάστημα μέχρι και ένα κοινωνικό δίκτυο. Οι δυνατότητες που έχουν αυτού του είδους οι ιστοσελίδες είναι πραγματικά πάρα πολλές και ποικίλουν ανάλογα με την τεχνολογία που θα επιλέξουμε να αναλάβει το δυναμικό κομμάτι του προγραμματισμού.

Με βάση τις παραπάνω κατηγοριοποιήσεις συμπεράνουμε εύκολα ότι εφαρμογή που πραγματεύεται αυτή η πτυχιακή εργασία είναι μια ξεκάθαρα δυναμική ιστοσελίδα. Συνεπώς, αρμόζει σε αυτό το σημείο να πραγματοποιηθεί η αναφορά των βασικών σημείων μίας τέτοιας σελίδας.

Το πρώτο πράγμα που πρέπει να κάνουμε είναι να τη χωρίσουμε σε μικρότερα μέρη και

να τα αναλύσουμε επιμέρους, εξετάζοντας τις διαθέσιμες τεχνολογίες. Όλες οι δυναμικές ιστοσελίδες χωρίζονται σε 2 βασικά μέρη, την εμφάνιση (front-end) και το δυναμικό κομμάτι (back-end). Η εμφάνιση μίας ιστοσελίδας είναι η πρώτη και η σημαντικότερη εικόνα που συναντάει ένας νέος χρήστης και προφανώς παίζει καθοριστικό ρόλο στον αν τελικά θα κινηήσει το ενδιαφέρον του. Ουσιαστικά αποτελείται από αρχεία δομημένα κατά τέτοιο τρόπο ώστε να μπορούν αν διαβαστούν από τους browsers. Τα κύρια εργαλεία για αυτό το σκοπό αποτελούν η HTML και η CSS.

Γράφοντας τις σελίδες μας μόνο με HTML κώδικα, μπορούμε να ορίσουμε το χρώμα και το μέγεθος του κειμένου αλλά και άλλων στοιχείων της σελίδας (όπως πίνακες, links, εικόνες κ.τ.λ.). Για να αλλάξουμε το χρώμα κάποιου κειμένου ή το χρώμα ενός πίνακα, θα πρέπει να βρούμε το χρώμα αυτό μέσα στον κώδικα και να το αλλάξουμε. Η διαδικασία αυτή μπορεί να φαντάζει εύκολη όταν έχουμε να διαχειριστούμε μια μόνο σελίδα, αλλά ένα site αποτελείται από δεκάδες σελίδες τις οποίες χρειάζεται να διαχειριζόμαστε εύκολα και γρήγορα. Αυτό το πρόβλημα λύνεται με τη χρήση CSS σε εξωτερικά αρχεία όπου μπορούμε να ορίζουμε χρώματα και μεγέθη οργανωμένα σε στυλ και έπειτα να εφαρμόζουμε τα στυλ αυτά στα στοιχεία των σελίδων της ιστοσελίδας μας. Με αυτόν τον τρόπο, κάθε φορά που αλλάζουμε το χρώμα ενός στυλ, αλλάζει το χρώμα όλων των στοιχείων που έχουν αναφορά στο στυλ αυτό. Έτσι αν έχουμε ορίσει ένα στυλ για το κύριο μενού του site, τότε θα χρειάζεται να αλλάξουμε το χρώμα του στυλ αυτού και αυτόματα θα εφαρμοστεί σε όλες τις σελίδες.

Το DOM είναι ένα μοντέλο παρουσίασης της W3C (World Wide Web Consortium), ανεξάρτητο από γλώσσες και πλατφόρμες που χρησιμοποιείται για την αναπαράσταση HTML και XML αρχείων. Στην ουσία είναι οι κανόνες που ορίζουν την δομή ενός αρχείου και δημιουργούν την ιεραρχία των στοιχείων που το συνθέτουν, ώστε να επιτρέψουν την πρόσβαση σε εξωτερικά scripts. Αυτή η δομή χρησιμοποιείται κυρίως από την JavaScript για να αναγνωρίσει τα στοιχεία τα οποία θα επηρεάσει δυναμικά και όχι στην εμφάνιση του html αρχείου από τον browser.

Αναλυτικά η JavaScript μπορεί μέσω του DOM να:

- Αλλάξει όλα τα HTML στοιχεία μιας σελίδας
- Αλλάξει όλες τις ιδιότητες των HTML στοιχείων μιας σελίδας
- Αλλάξει όλους τους κανόνες των HTML στοιχείων μιας σελίδας
- Αφαιρέσει υπάρχοντα HTML στοιχεία μιας σελίδας
- Προσθέσει νέα HTML στοιχεία σε μια σελίδα
- Έχει πρόσβαση σε όλα τα γεγονότα (events) που συμβαίνουν στα HTML στοιχεία
- Να δημιουργήσει καινούρια γεγονότα που συμβαίνουν στα HTML στοιχεία

Στην καθημερινή μας ζωή πολλές φορές πρέπει να πάρουμε κάποιες αποφάσεις που αφορούν τα εργαλεία που θα χρησιμοποιήσουμε για να κάνουμε μια συγκεκριμένη δουλειά. Συνήθως επιλέγουμε αυτά που ταιριάζουν περισσότερο στις ανάγκες μας και μας γλιτώνουν πολύτιμο χρόνο. Αυτό μπορεί να γίνει με την επιλογή ενός εργαλείου που έχει φτιάξει κάποιος άλλος, καθώς δεν χρειάζεται κάθε φορά να «ανακαλύπτουμε τον τροχό».

Αν λοιπόν το θέμα μας είναι η σχεδίαση μιας ιστοσελίδας, δεν χρειάζεται να το κάνουμε από την αρχή για κάθε στοιχείο της, αλλά μπορούμε να χρησιμοποιήσουμε ένα front-end framework. Δηλαδή στην ουσία, κάποια έτοιμα κομμάτια κώδικα HTML, CSS και JavaScript που μπορούν να χρησιμοποιηθούν από τους σχεδιαστές του front-end, δηλαδή του οπτικού κομματιού μιας ιστοσελίδας(w3schools, n.d.), (Yee-King, n.d.). Με την επιλογή αυτή εξοικονομούμε πολύτιμο χρόνο καθώς επίσης με τη χρήση ενός framework είμαστε σίγουροι ότι η εκάστοτε ιστοσελίδα που κατασκευάζουμε θα εμφανίζεται σωστά σε όλους τους browsers όλων των συσκευών που κυκλοφορούν στην αγορά, αν φυσικά αυτές υποστηρίζονται στις προδιαγραφές του framework που επιλέξαμε. Αυτή τη στιγμή υπάρχουν πάνω από 40 διαφορετικά front-end frameworks διαθέσιμα για download δωρεάν και έτοιμα για χρήση, πράγμα που καθιστά πιο δύσκολη την απόφαση ποιο από όλα πρέπει να χρησιμοποιήσουμε. Για να πάρουμε την σωστή απόφαση θα πρέπει να γνωρίζουμε τις ανάγκες που πρέπει αυτό να καλύπτει και να συγκρίνουμε τις διαφορές δυνατοτήτες που αυτά έχουν. Οι ιδιότητες μερικών εξ αυτών είναι οι ακόλουθες:

1. Οι δυνατότητες εμφάνισης, δηλαδή αν είναι οι στήλες που φιλοξενούν το περιεχόμενο, προσδιορίζονται σε αριθμό pixels (fixed) ή ως ένας αριθμός % του πλάτους της οθόνης προβολής.
2. Μέγιστος αριθμός στηλών που μπορεί να δημιουργήσει.
3. Πρόσθετες δυνατότητες, προσθετοί κανόνες που μπορεί να φιλοξενεί, εικονίδια που περιέχονται ή γραμμές, κώδικας JavaScript κλπ.
4. Άδεια που μας δίνεται από τον δημιουργό για τις διάφορες χρήσεις που μπορούμε να κάνουμε.

Το δυναμικό κομμάτι (back-end) είναι αυτό που την κάνει να ξεχωρίζει από μια στατική ιστοσελίδα και έχει να κάνει με την γλώσσα προγραμματισμού που τρέχει στην πλευρά του server και αναλαμβάνει να διαμορφώσει τα αρχεία πριν αυτά σταλούν στον τελικό χρήστη. Οι διαθέσιμες τεχνολογίες που υπάρχουν και είναι κατασκευασμένες γι' αυτή τη δουλειά είναι πολλές και η κάθε μια προσφέρει τα δικά της χαρακτηριστικά αλλά και καταναλώνει και τους ανάλογους πόρους. Οι πιο δημοφιλείς server-side γλώσσες προγραμματισμού είναι οι επόμενες: Php, Java, Python, Ruby, Perl, Asp.net.

ΚΕΦΑΛΑΙΟ 3: Καινοτομίες της HTML5

3.1 Από την HTML στην HTML5

Η ανάπτυξη της HTML5 ξεκίνησε το 2004 από μέλη της Apple, της Mozilla Foundation και της Opera software. Το 2008 κυκλοφόρησε η πρώτη έκδοση της με τους ειδικούς να λένε πως πρόκειται για μια εξελισσόμενη γλώσσα που μπορεί να μην ολοκληρωθεί ποτέ. Η μοίρα της ήταν αβέβαιη για κάποιο χρονικό διάστημα, ειδικά κατά την περίοδο που οι μεγάλες παίχτες Apple και Adobe βρίσκονταν σε διαμάχη για το ποια τεχνολογία θα επικρατήσει στις νέες τότε πλατφόρμες των κινητών συσκευών (Robson, 2010).

Το 2011 πλέον το 34% των κορυφαίων ιστοσελίδων χρησιμοποιούν HTML5. Το ίδιο έτος η Adobe σταματάει τη δημιουργία της τεχνολογίας Flash για κινητά και επικεντρώνεται στην ανάπτυξη της HTML5. Η γλώσσα πλέον επικρατεί ξεκάθαρα και ξεκινά μια σταθερή πορεία ανάπτυξης και υιοθέτησης της στις διάφορες πλατφόρμες (wikipedia, n.d.).

Σήμερα, το μόνο αρνητικό που μπορεί κάποιος να ισχυριστεί είναι η συμβατότητα της HTML5 ως καινούργια σχετικά γλώσσα. Όλοι όμως οι mainstream browsers (Chrome, Firefox, Internet Explorer, Safari, Opera) υποστηρίζουν τα νέα στοιχεία της HTML5 και τα APIs, και συνεχίζουν να προσθέτουν τα νέα χαρακτηριστικά της HTML5 στις τελευταίες εκδόσεις τους.

Η ομάδα εργασίας της HTML 5 περιλαμβάνει πλέον εργαζόμενους των εταιρειών AOL, Apple, Google, IBM, Microsoft, Mozilla, Nokia, Opera, και εκατοντάδες άλλους τεχνολογικούς κολοσσούς.

Η τρέχουσα έκδοση της προσφέρει πολλά νέα στοιχεία, από κινούμενα σχέδια με τα γραφικά, μουσική για ταινίες, και μπορεί επίσης να χρησιμοποιηθεί για την κατασκευή πολύπλοκων εφαρμογών web. Επιπλέον η HTML5 είναι cross-platform. Είναι σχεδιασμένη για να λειτουργεί και να χρησιμοποιείται σε υπολογιστές, Tablets, Smartphones, ή Smart TVs.

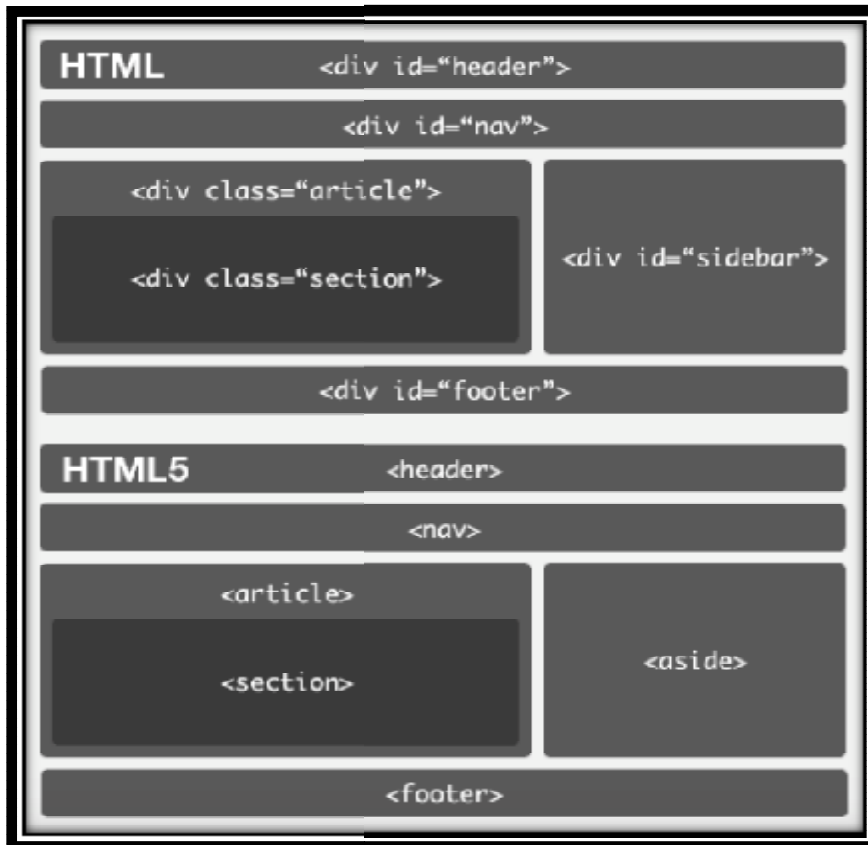
3.2 Καινούργια στοιχεία της HTML5

Η HTML5 εισάγει μια σειρά από καινοτομίες και απλοποιεί πολλές διεργασίες που πριν ήταν αρκετά πολύπλοκες να γίνουν. Ενσωματώνει αρκετά στοιχεία JavaScript μέσα σε ένα μόλις tag και απαλλάσσει τους developers από τη χρήση παλαιότερων τεχνολογιών που αποσύρονται (wikipedia, n.d.), (Duckett, 2014).

Για τη βασική δομή των σελίδων η HTML5 πλέον χρησιμοποιεί νέα tags τα όπως φαίνονται παρακάτω στον πίνακα (Robson, 2010):

ΕΤΙΚΕΤΕΣ ΔΟΜΗΣ	
<code><header></code>	Εισάγει την κεφαλίδα στο έγγραφο
<code><hgroup></code>	Κατηγοριοποιεί τις κεφαλίδες
<code><nav></code>	Εισάγει το μενού περιήγησης
<code><section></code>	Εισάγει μια περιοχή
<code><main></code>	Καθορίζει την κύρια περιοχή ενός έγγραφου
<code><article></code>	Εισάγει ένα άρθρο
<code><aside></code>	Εισάγει δευτερεύον περιεχόμενο
<code><footer></code>	Εισάγει το τελείωμα ενός έγγραφου
<code><details></code>	Εισάγει λεπτομέρειες που μπορούν και να κρυφτούν
<code><summary></code>	Εισάγει μια σύνοψη για τις λεπτομέρειες
<code><figure></code>	Εισάγει μια εικόνα ή ένα γράφημα
<code><figcaption></code>	Εισάγει τις πληροφορίες του παραπάνω tag
<code><mark></code>	Εισάγει επιλεγμένο κείμενο
<code><time></code>	Εισάγει ημερομηνία/ώρα
<code><wbr></code>	Εισάγει πιθανή αλλαγή γραμμής
<code><dialog></code>	Εισάγει πεδίο διαλόγου
<code><command></code>	Εισάγει μια εντολή που μπορεί να δοθεί
<code><meter></code>	Εισάγει την κεφαλίδα στο έγγραφο
<code><progress></code>	Κατηγοριοποιεί τις κεφαλίδες

Η παρακάτω εικόνα συνοψίζει τις διαφορές στη δομή μεταξύ HTML και HTML5:



Εικόνα 11: Δομικές διαφορές HTML και HTML5

Όσον αφορά τις Φόρμες, νέα πεδία καταχώρισης έχουν κάνει την εμφάνισή τους στην HTML5. Τα πλεονεκτήματα από αυτά τα νέα πεδία είναι ότι πλέον δεν χρειάζεται η χρήση της JavaScript για να γίνει ο αυτόματος έλεγχος των δεδομένων πριν ακόμα εκείνα καταχωρηθούν και η καλύτερη διαχείριση τους. Τα καινούρια πεδία είναι:

ΕΤΙΚΕΤΕΣ ΦΟΡΜΩΝ		
search	<code><input type="search" name="search_field" /></code>	Το πεδίο της αναζήτησης.
color	<code><input type="color" name="color_field" /></code>	Το πεδίο για το χρώμα.
email	<code><input type="email" name="email_field" /></code>	Το πεδίο για την διεύθυνση email
time	<code><input type="time" name="time_field" /></code>	Το πεδίο για την Ώρα.
datetime-local	<code><input type="datetime-local" name="gr_time_field" /></code>	Το πεδίο για την τοπική ώρα και την ημερομηνία.
tel	<code><input type="tel" name="telephone_field" /></code>	Το πεδίο για το νούμερο τηλεφώνου.
month	<code><input type="month" name="month_field" /></code>	Το πεδίο κειμένου για το μήνα.
week	<code><input type="week" name="week_field" /></code>	Το πεδίο κειμένου για την εβδομάδα.

date	<code><input type="date" name="date_field" /></code>	Το πεδίο για την ημέρα.
datetime	<code><input type="datetime" name="datetime_field" /></code>	Το πεδίο για την ώρα και την ημερομηνία.
url	<code><input type="url" name="url_field" /></code>	Το πεδίο για την διεύθυνση url.
number	<code><input type="number" name="number_field" /></code>	Το πεδίο για αριθμούς.
range	<code><input type="range" name="range_field" /></code>	Το πεδίο για επιλογή αριθμού από εύρος τιμών.

Το πεδίο που χρησιμοποιείται για να εισαχθεί μια αριθμητική τιμή ονομάζεται Number. Σε αυτό το πεδίο δεν μπορεί να εισαχθεί κείμενο και μπορεί να δεχθεί περιορισμούς. Για να μπορούμε να διαλέξουμε μέσα από ένα εύρος τιμών χρησιμοποιούμε το πεδίο Range. Και αυτό όπως το πεδίο Number, δέχεται περιορισμούς μέσω των παρακάτω ιδιοτήτων:

ΙΔΙΟΤΗΤΕΣ NUMBER & RANGE	
Max	Ο μέγιστος αριθμός που μπορεί να λάβει το πεδίο.
Min	Ο ελάχιστος αριθμός που μπορεί να λάβει το πεδίο.
Step	Το πόσο μπορεί να μεταβληθεί το ποσό που εισάγεται στο πεδίο, δηλαδή το ποίους αριθμούς μπορούμε να εισάγουμε.
Value	Είναι η αρχική τιμή που έχει το πεδίο.

3.3 Canvas & SVG

Μια περιοχή canvas εισάγετε με το tag `<canvas></canvas>` και δίνει την δυνατότητα εμφάνισης διαφόρων γραφικών σε μια σελίδα, όπως απλά διαγράμματα, εντυπωσιακά interfaces, κινούμενα γραφικά, γραφικές παραστάσεις και εσωτερικές εικόνες. Η σχεδίαση γραφικών εντός του canvas γίνεται ορίζοντας σημεία μέσα στην περιοχή, με την τεχνική pixel-based drawing. Η γλώσσα JavaScript χρησιμοποιείται για να σχεδιάσουμε μέσα στο canvas και μας δίνει τη δυνατότητα να σχεδιάσουμε κάθε τύπου γραφικά(Yee-King, n.d.).

Τα στοιχεία svg εισάγονται και αυτά παρόμοια με το tag `<svg></svg>` και μας δίνουν τη δυνατότητα σχεδίασης 2D γεωμετρικών σχημάτων και διανυσμάτων πολύ γρήγορα και εύκολα με τη χρήση των σχετικών παραμέτρων.

Κάθε μέσο έχει τα θετικά του και τα αρνητικά του. Το SVG δεν εξαρτάται από την ανάλυση πράγμα το οποίο το κάνει κατάλληλο για εφαρμογές που χρησιμοποιούν μεγέθυνση ή τροποποίηση ανάλογα την ανάλυση. Επίσης υποστηρίζει πάρα πολύ καλά την προσθήκη κίνησης στις εικόνες είτε με τη μορφή κώδικα είτε με τη βοήθεια της JavaScript και μπορεί κανείς να έχει τον πλήρη έλεγχο κάθε στοιχείου. Ακόμα, επειδή το SVG είναι στην ουσία ένα

XML αρχείο είναι πολύ πιο εύκολο να χρησιμοποιηθεί από online εφαρμογές. Όμως το SVG έχει και δύο μεγάλα μειονεκτήματα. Πρώτον όσο πιο πολύπλοκο είναι το στοιχείο που χρησιμοποιεί το SVG τόσο πιο αργή απόδοση θα έχει και δεύτερον το SVG από μόνο του δεν κάνει για παιχνίδια.

Το canvas είναι ιδανικό για επίπεδη σχεδίαση λόγω της πολύ καλής απόδοσης πλατφόρμας του. Όλα αποδίδονται ως pixel, έτσι δεν μειώνεται η οπτική απόδοση του στοιχείου. Επίσης ότι σχεδιάσουμε μπορούμε να το αποθηκεύσουμε σαν εικόνα της μορφής .jpg η .png. Είναι το καλύτερο για επεξεργασία εικόνας σε επίπεδο pixel. Όμως έχει και τα μειονεκτήματα του, όπως ότι ό,τι σχεδιαστεί είναι εξολοκλήρου pixel. Η απόδοση κειμένου είναι πολύ κακή. Επίσης η προσθήκη κίνησης είναι πολύ δύσκολο να επιτευχθεί και δεν ενδείκνυται για την επεξεργασία εικόνων σε online εφαρμογές γιατί πρέπει να σχεδιάζετε το στοιχείο όλο από την αρχή.

Canvas και SVG έχουν και τα δύο τα πλεονεκτήματα τους και τα μειονεκτήματα. Τα χρησιμοποιούμε ανάλογα του τι θέλουμε να κάνουμε. Εάν θέλουμε στατικές εικόνες που να μην μετατρέπονται η να σχεδιάσουμε διαστάσιμα γραφικά ο canvas είναι το ιδανικό εργαλείο. Αν θέλουμε κίνηση στα γραφικά μας, μεταβαλλόμενες εικόνες και απολυτό έλεγχο τότε χρειαζόμαστε το SVG.

3.4 Βίντεο και Ήχος

Μαζί με την HTML5 ήρθαν και αρκετές ριζικές αλλαγές. Μία από αυτές ήταν και η εισαγωγή ήχου και βίντεο στην σελίδα. Στις προηγούμενες εκδόσεις της HTML τα εργαλεία για να προσθέσουμε ήχο και βίντεο ήταν διαφορά προσθετά όπως το Quick time της Apple η το flash της Adobe. Στην HTML5 εισάγεται ένας καινούριος τρόπος ώστε να ενσωματώνονται στην σελίδα. Αυτός είναι τα στοιχεία <audio> και <video> αντίστοιχα(wikipedia, n.d.).

Προς το παρόν οι τύποι βίντεο και ήχου που υποστηρίζονται από την HTML5 είναι ο τύπος Ogg (ogg αρχεία με το Theora video codec και το Vorbis audio codec) και ο τύπος MPEG4 (MPEG4 αρχεία με το H.264 video codec και το AAC audio codec) για τα βίντεο, Ogg Vorbis, MP3 και WAV για τον ήχο(Yee-King, n.d.).

Τα στοιχεία <video> και <audio> μπορούν να δεχθούν κάποιες ιδιότητες, οι οποίες εμφανίζονται στους παρακάτω πίνακες.

ΕΤΙΚΕΤΕΣ VIDEO	
Autoplay	Ξεκινάει την αναπαραγωγή του βίντεο μόλις ολοκληρωθεί η φόρτωση του.
controls	Τα κουμπιά ελέγχου, όπως το play, το stop κτλ που εμφανίζονται στο βίντεο.
src	Ορίζει το url του βίντεο

loop	Κάθε φορά που το βίντεο τελειώνει, ξεκινάει και πάλι αυτόματα η αναπαραγωγή του από την αρχή.
preload	Το βίντεο φορτώνεται όταν η σελίδα φορτωθεί
height	Ορίζει το ύψος του βίντεο.
width	Ορίζει το μήκος του βίντεο.

ΕΤΙΚΕΤΕΣ AUDIO	
Autoplay	Ξεκινάει την αναπαραγωγή του ήχου μόλις ολοκληρωθεί η φόρτωση του.
controls	Τα κουμπιά ελέγχου, όπως το play, το stop κτλ που εμφανίζονται στο ήχο.
src	Ορίζει το url του ήχου
loop	Κάθε φορά που ο ήχος τελειώνει, ξεκινάει και πάλι αυτόματα η αναπαραγωγή του από την αρχή.
preload	Ο ήχος φορτώνεται όταν η σελίδα φορτωθεί.

3.5 Αποθήκευση Πληροφοριών

Όταν επισκεπτόμαστε μία ιστοσελίδα, τα δεδομένα της αποθηκεύονται στον υπολογιστή μας με τη χρησιμοποίηση της Ιενειδοτίρτ, ως τα λεγόμενα cookies. Τα αρχεία αυτά σαρώνονται με κάθε αίτηση του διακομιστή κι αυτό είναι το μεγάλο πλεονέκτημα τους. Αυτό γιατί όταν αποθηκεύονταν πολλά δεδομένα, τα cookies έκαναν την διαδικασία χρονοβόρα. Αυτό είχε ως αποτέλεσμα την καταφυγή σε διαφορετικές μεθόδους αποθήκευσης των δεδομένων της ιστοσελίδας (Robson, 2010).

Έτσι στην HTML5 τα δεδομένα σαρώνονται μόνο όταν είναι απαραίτητο και όχι με κάθε αίτηση του διακομιστή. Κάθε ιστοσελίδα αποθηκεύει σε διαφορετικό μέρος τα δικά της δεδομένα, έχοντας έτσι πρόσβαση μόνο στα δικά της. Με αυτό τον τρόπο δεν επηρεάζεται ο χρόνος απάντησης της σελίδας παρά την αποθήκευση τεράστιου όγκου δεδομένων. Έτσι είναι εφικτό να αποθηκεύονται μεγάλου όγκου δεδομένα χωρίς να επηρεάζεται η απόδοση μιας ιστοσελίδας. Επίσης τα δεδομένα αποθηκεύονται σε διαφορετικές περιοχές του δίσκου για κάθε ιστοσελίδα, έτσι ώστε μια ιστοσελίδα να έχει πρόσβαση μόνο στα δικά της δεδομένα. Και σε αυτή την μέθοδο αποθήκευσης δεδομένων η HTML5 χρησιμοποιεί την JavaScript για να αποθηκεύσει τα δεδομένα της κάθε ιστοσελίδας. Οι δυο νέοι τρόποι αποθήκευσης δεδομένων είναι οι εξής:

- SessionStorage
- LocalStorage

Η μέθοδος SessionStorage:

Η μέθοδος αυτή αποθηκεύει τα δεδομένα για μία μόνο επίσκεψη. Αφού η επίσκεψη τελειώσει τα δεδομένα διαγράφονται. Δηλαδή μπαίνοντας σε μία σελίδα ξεκινά η αποθήκευση των πληροφοριών και μόλις αποσυνδεθούμε η κλείσουμε την ιστοσελίδα τα δεδομένα αυτομάτως διαγράφονται. Αυτή η μέθοδος χρησιμοποιείται για εφαρμογές χωρίς σύνδεση ή για διασταυρούμενη ανταλλαγή μηνυμάτων.

Η μέθοδος LocalStorage:

Η μέθοδος αποθηκεύει τα δεδομένα μόνιμα, δηλαδή είναι διαθέσιμα για πάντα.

3.6 Η CSS3

Το CSS3 προσφέρει κάποια καινούρια χαρακτηριστικά, τα οποία δίνουν μια σειρά από πλεονεκτήματα στον σχεδιασμό μιας ιστοσελίδας αφού έρχεται για να απλοποιήσει κάποιες διαδικασίες που παλιότερα απαιτούσαν περισσότερους πόρους και χρόνο για να φτιαχτούν. Τα κύρια οφέλη του CSS3 είναι ότι επιτρέπει την δημιουργία χαρακτηριστικών που αλλιώς θα χρειαζόντουσαν την JavaScript για να εφαρμοστούν, έτσι απλοποιεί πολύ τον κώδικα και κάνει τις σελίδες πιο ελαφριές. Ένα άλλο πλεονέκτημα στο CSS3 είναι η δυνατότητα δημιουργίας γραφικών, που μέχρι τώρα χρειαζόνταν το Photoshop για να δημιουργηθούν και να εμφανιστούν μέσω εικόνων στην ιστοσελίδα.

Μερικές καινοτομίες που φέρνει το CSS3 είναι οι εξής:

- **Media queries**

Δυνατότητα εφαρμογής κανόνων CSS ανάλογα με μέγεθος οθόνης της συσκευής του χρήστη.

Κώδικας:

```
@media screen and (min-width: 480px) {  
  
body {
```



```
background-color: lightgreen;
}
}
```

· **Gradients**

Μια καινοτομία του CSS3 καθώς στις προηγούμενες εκδόσεις του CSS δεν υπήρχε η δυνατότητα μίξης των χρωμάτων γραμμικά.

Κώδικας:

```
body{
background: linear-gradient(top, #000 0%, #111 50%,
#222 70%, #333 100%);
}
```

· **Animation και Transitions**

Τα Transitions είναι ένα set κανόνων CSS που μας επιτρέπουν να αλλάζουμε τιμές στις ιδιότητες ενός στοιχείου, ομαλά και σταδιακά σε συγκεκριμένο χρόνο. Με άλλα λόγια μας επιτρέπουν να αλλάζουμε τιμές στα στοιχεία με τέτοιο τρόπο ώστε να πετυχαίνουμε ένα εφέ σαν το animation.

Κώδικας:

```
body{
transition: width 2s;
}
```

· **Custom γραμματοσειρές**

Απλώς παίρνουμε το αρχείο της γραμματοσειράς που θέλουμε, το φορτώνουμε στη σελίδα και καλούμε τη γραμματοσειρά με το @font-face. Αυτό μας δίνει την δυνατότητα να

δώσουμε ξεχωριστή εμφάνιση στα διάφορα κείμενα χωρίς την χρήση εικόνων.

Κώδικας:

```
@font-face{  
  font-family: myfamily;  
  src: url(font.woff);  
}  
  
body{  
  font-family:myfamily;  
}
```

· Φόντο με πολλαπλές εικόνες

Αυτό μας επιτρέπει να ορίζουμε πολλές εικόνες ως φόντο για ένα στοιχείο, τη θέση της κάθε εικόνας και την επανάληψή της.

Κώδικας:

```
body{  
  
  background:url(img.jpg), url(img2.jpg);  
  
  background-size:100% 100%;  
  
  background-repeat:no-repeat;  
}
```

· Rounded Corners

Στις προηγούμενες εκδόσεις του CSS, η χρήση στρογγυλεμένων γωνιών ήταν μια χρονοβόρα διαδικασία καθώς κάθε φορά ήμασταν υποχρεωμένοι να ‘κόβουμε’ τέσσερις κυκλικές εικόνες και να τις τοποθετούμε σε κάθε γωνία του div ώστε να παραχθεί το εφέ. Πλέον, αυτό το χαρακτηριστικό προστίθεται με τη χρήση μιας γραμμής CSS. Αυτό που χρειάζεται είναι να ορίσουμε την ακτίνα του κύκλου για κάθε γωνία.

Κώδικας:

```
body{  
  
border-radius: 10px;  
  
}
```

· **Text shadow**

Ο κύριος στόχος αυτής της ιδιότητάς είναι να δώσει στους σχεδιαστές ένα εργαλείο για να δημιουργήσουν εφέ κειμένου, το οποίο θα μπορούσε να γίνει πριν μόνο με την χρήση εικόνων. Αυτό έχει το βασικό πλεονέκτημα ότι το κείμενο σε αντίθεση με πριν, παραμένει ανιχνεύσιμο από τις μηχανές αναζήτησης. Επίσης εξοικονομούνται πόροι, καθώς οι εικόνες συνήθως καταλαμβάνουν περισσότερο χώρο, από μία γραμμή κώδικα.

Κώδικας:

```
h1 {  
  
text-shadow: 1px 1px 1px rgba(255,255,255, 1);  
  
}
```

ΚΕΦΑΛΑΙΟ 4: Εφαρμογή της HTML5

4.1 Εισαγωγή

Στα προηγούμενα κεφάλαια της εργασίας αυτής είδαμε σταδιακά όλη την τεχνολογία γύρω από το αντικείμενο του προγραμματισμού με τη γλώσσα HTML5. Πιο συγκεκριμένα, στο 1^ο κεφάλαιο κάναμε μια εισαγωγή στο χαμηλότερο επίπεδο της επιστήμης των δικτύων υπολογιστών, είδαμε την ιστορική εξέλιξη τους, αναφερθήκαμε στα διάφορα πρωτόκολλα που αναπτύχθηκαν καθώς και στις διάφορες τεχνολογίες επικοινωνίας εξυπηρέτησης αιτημάτων με αποκορύφωμα την server-client αρχιτεκτονική και τα web services.

Στη συνέχεια, στο κεφάλαιο 2 έγινε παρουσίαση των πρώτων εκδόσεων της γλώσσας HTML καθώς και των βασικών στοιχείων που την απαρτίζουν. Είδαμε επίσης την γλώσσα μορφοποίησης CSS καθώς και την γλώσσα JavaScript η οποία προσδίδει απαραίτητη δυναμικότητα στις σελίδες μας. Στο κεφάλαιο 3, προχωρήσαμε στην παρουσίαση των καινούργιων στοιχείων που εισήγαγε η εξελιγμένη πλέον HTML5 καθώς και τις νέες δυνατότητες μορφοποίησης με τη χρήση της CSS3 αλλά και κάποιες επιμέρους δυνατότητες απεικόνισης γραφικών (Yee-King, n.d.).

Μετά από όλα τα παραπάνω είμαστε πλέον σε θέση να αναπτύξουμε τη δική μας εφαρμογή με τη γλώσσα HTML5 προκειμένου να παρουσιάσουμε και πρακτικά τις δυνατότητες και τη χρήση της γλώσσας (LaGrone, 2013). Η εφαρμογή αυτή ουσιαστικά θα είναι μια σελίδα blog που στο αριστερό της μέρος θα δέχεται ως είσοδο άρθρα από τον χρήστη θα τα αποθηκεύει με τη χρήση της τεχνολογίας local storage και θα τα προβάλλει στο δεξί της μέρος ως δημοσιευμένα άρθρα πλέον.

4.2 Η βασική δομή της εφαρμογής

Το πρώτο βήμα που θα πρέπει να κάνουμε προκειμένου να αναπτύξουμε την εφαρμογή blog είναι φυσικά να δημιουργήσουμε το σκελετό της, χρησιμοποιώντας τις γνώσεις μας από τις ενότητες 2.3 και 3.2.

Συνθέτοντας τα γνωστά tags της γλώσσας δημιουργούμε την παρακάτω δομή:

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
```

```
<title>Εφαρμογή Blog</title>
```

```
</head>
```

```
<body>
```

```
<header id="menu">
```

```
<nav>
```

```
<ul>
```

```
<li><a href="#" >Φόρτωση αποθηκευμένων άρθρων</a></li>
```

```
<li><a href="#" >Πληροφορίες</a></li>
```

```
</ul>
```

```
</nav>
```

```
</header>
```

```
<section id="inputForm" style="float:left;">
```

```
<form>
```

```
</form>
```

```
</section>
```

```
<aside id="articles" style="float:right;">
```

```
<h1>Λίστα Άρθρων</h1>
```

```
<article>
```

```
</article>
```

```
</aside>
```

```
<footer>
```

```
<p id="copyright"><em>Copyright 2016 Blog</em></p>
```

```
</footer>
```

```
</body>
```

```
</html>
```

Ο σκελετός που δημιουργήσαμε απαρτίζεται από το menu στο πάνω μέρος της σελίδας, από που μας δίνεται η δυνατότητα να φορτώσουμε τα ήδη αποθηκευμένα άρθρα (locally stored) καθώς επίσης και από την επιλογή <<Πληροφορίες>> η οποία εμφανίζει ένα popup box με τα credits της εφαρμογής. Στο κέντρο της σελίδας, στα αριστερά έχουμε προσθέσει την ετικέτα <section> μέσα στην οποία σκοπεύουμε να προσθέσουμε την φόρμα εισαγωγής δεδομένων για την προσθήκη άρθρων καθώς και ένα <audio> tag για την αναπαραγωγή ήχου. Στα δεξιά της σελίδας με την ετικέτα <aside> έχουμε δημιουργήσει ένα χώρο όπου θα παρουσιάζονται τα άρθρα της εφαρμογής. Στο κάτω μέρος υπάρχει το <footer> όπου αναγράφεται το copyright της εφαρμογής.

4.3 Στοιχεία εισόδου

Στη συνέχεια θα πρέπει προσθέσουμε τον κώδικα για τη φόρμα προσθήκης άρθρων. Τα στοιχεία που πρόκειται να προσθέσουμε είναι:

- Ένα inputbox που θα δέχεται text και θα χρησιμοποιηθεί για την προσθήκη τίτλου στο εκάστοτε άρθρο.
- Ένα textarea που ο χρήστης θα γράφει το κείμενο του άρθρου που πρόκειται να προστεθεί.
- Ένα inputbox για text όπου θα προστίθεται το όνομα του αρθρογράφου.
- Και φυσικά ένα input τύπου submit το οποίο θα καταχωρεί το άρθρο.

Παρακάτω φαίνεται ο κώδικας που υλοποιεί την φόρμα:

```
<form>
<fieldset>
<legend><b>Προσθήκη Άρθρου</b></legend>
<p>
<input type="text" id="articleTitle" placeholder="Τίτλος Άρθρου" required/>
</p>
<p>
<textarea id="articleText" rows="10" cols="50" required
placeholder="Κείμενο Άρθρου"></textarea>
</p>
<p>
<input id="articleAuthor" type="text" placeholder="Όνομα Συγγραφέα" required/>
</p>
<p>
<input onclick="addArticle()" type="submit" value="Δημοσίευση"/>
</p>
```

</fieldset>

</form>

4.4 Γραφικά εφαρμογής

Προκειμένου να επιδείξουμε και την λειτουργία των γραφικών στοιχείων θα προσθέτουμε δυναμικά στο κάτω μέρος του κάθε άρθρου ένα SVG λογότυπο με τα πρώτα γράμματα του κάθε αρθρογράφου. Το αποτέλεσμα που θα προσπαθήσουμε να επιτύχουμε φαίνεται στην παρακάτω εικόνα.



Εικόνα 12: Πρότυπο εικόνας για χρήση λογοτύπου.

Οι ετικέτες που θα χρησιμοποιήσουμε είναι:

- Η ετικέτα <svg> ώστε να δημιουργηθεί το στοιχείο SVG
- Η ετικέτα <ellipse> που σχεδιάζει το σχήμα της έλλειψης χρησιμοποιώντας τις διάφορες παραμέτρους της.
- Η ετικέτα <text> με την οποία θα εμφανίσουμε το κείμενο και συγκεκριμένα τα 4 πρώτα γράμματα του αρθρογράφου.
- Και τις ετικέτες <defs> και <linearGradient> που σχετίζονται με το πως θα εμφανιστούν τα χρώματα του σχήματος.

Παρακάτω ακολουθεί ο κώδικας του SVG σχήματος:

```
<svg height="130" width="500">
```

```
<defs>
```

```
<linearGradient id="grad1" x1="0%" y1="0%" x2="100%" y2="0%">
```



```

<stop offset="0% "
style="stop-color:rgb(255,255,0);stop-opacity:1" />

    <stop offset="100% "
        style="stop-color:rgb(255,0,0);stop-opacity:1" />

</linearGradient>

</defs>

<ellipse cx="100" cy="70" rx="85" ry="55" fill="url(#grad1)" />

<text fill="#ffffff" font-size="45" font-family="Verdana"
        x="50" y="86">Dem</text>

    Sorry, your browser does not support inline SVG.

</svg>

```

4.5 Ήχος

Ο ήχος σε μια σελίδα πάντα προσθέτει μια ζωντάνια οπότε και εμείς επιλέξαμε να προσθέσουμε έναν audio player οποίος δίνει την δυνατότητα στο χρήστη να ελέγξει το playback (παράμετρος controls) καθώς επίσης έχουμε επιλέξει η μουσική να ξεκινάει κατά την φόρτωση της σελίδας (παράμετρος autoplay). Το κομμάτι που θα αναπαραχθεί είναι ένα mp3 και βρίσκεται τοπικά στον υπολογιστή μας όπως φαίνεται από την ετικέτα source. Ακολουθεί ο σχετικός κώδικας:

```

<audio controls autoplay>

<source src="vivaldi.mp3" type="audio/mpeg">

    Your browser does not support the audio element.

</audio>

```

4.6 Μορφοποίηση εφαρμογής

Σε όλες τις εφαρμογές πολύ σοβαρό ρόλο παίζει η Διεπαφή Χρήστη. Για το λόγο αυτό έχουμε αναπτύξει σε CSS3 την μορφοποίηση των στοιχείων της σελίδας. Φυσικά δεν θα ήταν εφικτό να αναλύσουμε μια προς μια όλες τις παραμέτρους που έχουμε χρησιμοποιήσει ωστόσο έχουμε ήδη κάνει παρουσίαση της CSS στις ενότητες 2.4 και 3.6 όπου και μπορεί κανείς να ανατρέξει για λεπτομέρειες. Ο κώδικας μορφοποίησης που χρησιμοποιήσαμε φαίνεται παρακάτω.

```
<style>

    html{

    }

    body{

        background-color:#BBBCC;

    }

    header#menu{

        margin-bottom:20px;

        background-color:#000066;

        border-radius:10px;

    }

    headernav{

        height:30px;

        width:100%;
```

```
}
```

```
navul{  
    text-align:center;  
}
```

```
navli{  
    display:inline-block;  
    list-style:none;  
    margin:5px;  
}
```

```
navlia{  
    color:#cccccc;  
}
```

```
section{  
    background-color:#cccccc;  
    padding:10px;  
}
```

```
forminput[type=text]{  
    width:200px;  
    padding:5px;
```

```
}
```

```
article#text{  
    max-width:400px;  
}
```

```
audio{  
    margin-top:10px;  
    margin-left:20px;  
}
```

```
footer{  
    clear:both;  
    padding-top:100px;  
}
```

```
footerp{  
    background-color:white;  
    color:black;  
    padding:5px;  
    border-radius:5px;  
    text-align:center;  
}
```

</style>

4.7 JavaScript, JQuery & localStorage

Για να υλοποιήσουμε την λειτουργικότητα της εφαρμογής αναπτύξαμε τις JavaScript συναρτήσεις:

- addArticle()
- getCurrentDate()
- loadArticles()

Επιπλέον των συναρτήσεων που εμείς αναπτύξαμε, χρησιμοποιήσαμε και την ευρέως διαδεδομένη βιβλιοθήκη της JavaScript που ονομάζεται JQuery. Ουσιαστικά την χρησιμοποιήσαμε ώστε να απλοποιήσουμε την των επιλογή στοιχείων της HTML, η οποία γίνεται με μια πολύ απλή σύνταξη: `$("#element_id")`. Επιπλέον, η JQuery χρησιμοποιήθηκε προκειμένου να καλέσουμε τις συναρτήσεις: `.val()` η οποία επιστρέφει τα δεδομένα που μπορεί να περιέχει ένα στοιχείο και την `.append("html_code")` που απλώς προσθέτει των html κώδικα που έχει δεχθεί ως παράμετρο, σε κάποιο άλλο HTML element. Περαιτέρω ανάλυση της JQuery θα ξέφευγε από το πλαίσιο αυτής της εργασίας όποτε και δε θα επεκταθούμε άλλο.

Ένα άλλο πολύ σημαντικό στοιχείο που χρησιμοποιήθηκε είναι το αντικείμενο `localStorage`. Όπως είδαμε και στην ενότητα 3.5 το αντικείμενο αυτό μας δίνει την δυνατότητα να αποθηκεύσουμε δεδομένα τοπικά στο φυλλομετρητή μας. Οι συναρτήσεις που χρησιμοποιήσαμε ήταν οι:

- `setItem()` η οποία δημιουργεί μια εγγραφή δεδομένων
- `getItem()` η οποία μας επιστρέφει μια εγγραφή δεδομένων

Η συνάρτηση `addArticle()` που αναπτύξαμε ουσιαστικά ελέγχει με μια `if` ότι τα απαραίτητα πεδία της φόρμας έχουν συμπληρωθεί και στη συνέχεια συνθέτει τη δομή ενός άρθρου μαζί με τα απαραίτητα δεδομένα του και το αποστέλλει προς αποθήκευση στην `localStorage` αλλά και προς εμφάνιση στην ετικέτα με `id 'articles'`. Αντίστοιχα, η συνάρτηση `loadArticles()` καλείται από την επιλογή 'Φόρτωση αποθηκευμένων άρθρων' που βρίσκεται στο menu της εφαρμογής. Ουσιαστικά η συνάρτηση αυτή όταν καλεστεί προσπελαύνει όλες τις εγγραφές άρθρων που θα βρει στο `localStorage` και τις στέλνει προς εμφάνιση.

Η τελευταία συνάρτηση που έχει υλοποιηθεί είναι η `getCurrentDate()` η οποία όταν κληθεί αναλαμβάνει να δημιουργήσει το αλφαριθμητικό της παρούσας ημερομηνίας. Προκειμένου να το πετύχει αυτό χρησιμοποιεί το αντικείμενο `Date` που είχε παρουσιαστεί στην ενότητα 2.5.

Τον πλήρη κώδικα JavaScript της εφαρμογής μπορεί κανείς να τον βρει στο Παράρτημα στο τέλος της εργασίας. Στη συνέχεια παραθέτουμε ένα Screen Capture της εφαρμογής εν ώρα δράσης.



Εικόνα 13: Διεπαφή εφαρμογής Blog σε HTML5

Βιβλιογραφία

Duckett, J. (2014). *HTML and CSS: Design and Build Websites*.

Jean, W. (1997). *Δίκτυα επικοινωνιών*. Παπασωτηρίου.

LaGrone, B. (2013). *HTML5 and CSS3 Responsive Web Design Cookbook*. PACKT Publishing.

Robson, E. F. (2010). *Head First HTML5 Programming*. O'REILLY.

Tanenbaum, A. S. (2003). *Δίκτυα υπολογιστών*. Κλειδάριθμος.

w3schools. (n.d.). Ανάκτηση από <http://www.w3schools.com/js/>

w3schools. (n.d.). Ανάκτηση από http://www.w3schools.com/html/html5_intro.asp

wikipedia. (n.d.). Ανάκτηση από <https://en.wikipedia.org/wiki/HTML5>

wikipedia. (n.d.). Ανάκτηση από https://el.wikipedia.org/wiki/Επιστήμη_του_Web

Yee-King, M. (n.d.). *coursera*. Ανάκτηση από <https://www.coursera.org/learn/website-coding>

Γεώργιος Αλεξόπουλος, Α. Λ. (2015). *Τηλεπικοινωνίες και δίκτυα υπολογιστών*. Κλειδάριθμος.

Παράρτημα

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
```

```
<title>Εφαρμογή Blog</title>
```

```
<style>
```

```
    html{
```

```
    }
```

```
    body{
```

```
        background-color:#BBBCC;
```

```
    }
```

```
    header#menu{
```

```
        margin-bottom:20px;
```

```
        background-color:#000066;
```

```
        border-radius:10px;
```

```
    }
```



```
headernav{  
    height:30px;  
    width:100%;  
}  
  
navul{  
    text-align:center;  
}  
  
navli{  
    display:inline-block;  
    list-style:none;  
    margin:5px;  
}  
  
navlia{  
    color:#cccccc;  
}  
  
section{  
    background-color:#cccccc;  
    padding:10px;  
}
```

```
forminput[type=text]{  
    width:200px;  
    padding:5px;  
}
```

```
article#text{  
    max-width:400px;  
}
```

```
audio{  
    margin-top:10px;  
    margin-left:20px;  
}
```

```
footer{  
    clear:both;  
    padding-top:100px;  
}
```

```
footerp{  
    background-color:white;  
    color:black;
```

```
padding:5px;

border-radius:5px;

text-align:center;

}

</style>

<script src="jquery-3.0.0.min.js"></script>

<script type="text/javascript">

function getDate(){

var today = new Date();

var dd = today.getDate();

var mm = today.getMonth()+1;

var yyyy = today.getFullYear();

if(dd<10) {

dd='0'+dd

}

if(mm<10) {

mm='0'+mm

}

}
```

```

        today = dd+'/'+mm+'/'+yyyy;

        return today;
    }

```

```

function loadArticles(){

```

```

    keys = Object.keys(localStorage),

```

```

    i = keys.length;

```

```

    while ( i-- ) {

```

```

        $("#articles").append(localStorage.getItem(keys[i]));

```

```

    }

```

```

}

```

```

function addArticle(){

```

```

    if( $("#articleTitle").val().length>0 &&$("#articleText").val().length>0
&&$("#articleAuthor").val().length>0){

```

```

        article

```

```

        =

```

```

'<article><header><h2>'+$("#articleTitle").val()+'</h2><p>Δημοσιεύτηκεστις<time

```

```

datetime="">'+getCurrentDate()+'>'+getCurrentDate()+'</time>απότον/ην<a

```

```

href="#">'+$("#articleAuthor").val()+'</a></p></header><p

```

```

id="text">'+$("#articleText").val()+'</p><p><svg

```

```

height="130"

```

```

width="500"><defs><linearGradient id="grad1" x1="0%" y1="0%" x2="100%"

```

```

y2="0%"><stop offset="0%" style="stop-color:rgb(255,255,0);stop-opacity:1" /><stop

```

```

offset="100%"

```

```

style="stop-color:rgb(255,0,0);stop-opacity:1" /></linearGradient></defs><ellipse cx="100"

```

```

cy="70" rx="85" ry="55"

```

```
fill="url(#grad1)" /><text fill="#ffffff" font-size="45"
```

```
font-family="Verdana"
```

```
x="50"
```

```
y="86">'+$("#articleAuthor").val().substring(0,4)+'</text> Sorry, your browser does not  
support inline SVG. </svg></p></article>';
```

```
localStorage.setItem(Date.now()+"",article);
```

```
$("#articles").append(article);
```

```
}
```

```
}
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<header id="menu">
```

```
<nav>
```

```
<ul>
```

```
<li><a
```

```
href="#"
```

```
onclick="loadArticles()">Φόρτωση αποθηκευμένων άρθρων</a></li>
```

```
<li><a href="#" onclick="alert('Πτυχιακή εργασία, εφαρμογή
```

```
Blog σε HTML5!')">Πληροφορίες</a></li>
```

```
</ul>
```

```

        </nav>

</header>

<section id="inputForm" style="float:left;">

    <form>

        <fieldset>

            <legend><b>Προσθήκη Άρθρου</b></legend>

            <p>

                <input type="text" id="articleTitle" placeholder="Τίτλος Άρθρου"
required/>

            </p>

            <p>

                <textarea id="articleText" rows="10" cols="50" required
placeholder="Κείμενο Άρθρου"></textarea>

            </p>

            <p>

                <input id="articleAuthor" type="text"
placeholder="Όνομα Συγγραφέα" required/>

            </p>

            <p>

                <input onclick="addArticle()" type="submit" value="Δημοσίευση"/>

            </p>

        </fieldset>

    </form>

```

```
<audio controls autoplay>
```

```
<source src="vivaldi.mp3" type="audio/mpeg">
```

Your browser does not support the audio element.

```
</audio>
```

```
</section>
```

```
<aside id="articles" style="float:right;">
```

```
<h1>Λίστα Άρθρων</h1>
```

```
<article>
```

```
<header>
```

```
<h2>Demo Τίτλος</h2>
```

```
<p>
```

```
Δημοσιεύτηκε στις <time datetime="2009-09-04T16:31:24+02:00">15 Ιουνίου 2016</time> από τον/ην <a href="#">Demo User</a>
```

```
</p>
```

```
</header>
```

```
<p id="text">Αυτό είναι ένα Demo κείμενο. Μπορείτε να προσθέσετε και άλλα άρθρα από τη φόρμα στα αριστερά σας!</p>
```

```
<p>
```

```
<svg height="130" width="500">
```

```

        <defs>
        <linearGradient id="grad1" x1="0%" y1="0%" x2="100%"
y2="0%">
        <stop offset="0%"
style="stop-color:rgb(255,255,0);stop-opacity:1" />
        <stop offset="100%"
style="stop-color:rgb(255,0,0);stop-opacity:1" />
        </linearGradient>
        </defs>
        <ellipse cx="100" cy="70" rx="85" ry="55" fill="url(#grad1)"
/>
        <text fill="#ffffff" font-size="45" font-family="Verdana"
x="50" y="86">Dem</text>
        Sorry, your browser does not support inline SVG.
        </svg>
    </p>
</article>
</aside>
<footer>
    <p id="copyright"><em>Copyright 2016 Blog</em></p>
</footer>
</body>

```


</html>