



**ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΕΥΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΔΥΤΙΚΗΣ ΕΛΛΑΔΟΣ
ΣΧΟΛΗ ΔΟΙΚΗΣΗΣ ΚΑΙ ΟΙΚΟΝΟΜΙΑΣ
ΤΜΗΜΑ ΕΠΙΧΕΡΗΜΑΤΙΚΟΥ ΣΧΕΔΙΑΣΜΟΥ ΚΑΙ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Υποστήριξη της HTML5 στους Web Browsers

**Αρλετή Ευαγγελία Μαρία
Φούντα Αικατερίνη**

ΠΑΤΡΑ, 2014

Περιεχόμενα

1 ^ο Κεφάλαιο - Εισαγωγή	9
1.1 - Αντικείμενο της εργασίας.....	9
1.2 - Δομή της εργασίας.....	10
2 ^ο Κεφάλαιο - Ο Παγκόσμιος Ιστός	11
2.1 - Εισαγωγή	11
2.2 - Internet	11
2.2.1 - Τι είναι το internet.....	11
2.2.2 - Δημιουργία του internet	12
2.2.3 - Δομή και λειτουργία του Internet	14
2.3 - Ο Παγκόσμιος Ιστός (WWW)	21
2.3.1 - Τι είναι ο Παγκόσμιος Ιστός (World Wide Web).....	21
2.3.2 - Η ιστορία του Παγκόσμιου Ιστού	21
2.3.3 - Πως λειτουργεί ο Ιστός.....	22
2.3.4 - Web Standards.....	24
2.3.5 - Web 1.0.....	24
2.3.6 - Web2.0.....	25
2.3.7 - Web 3.0.....	28
2.3.8 - Web 4.0.....	31
2.4 - Τεχνολογίες Διαδικτύου	33
2.4.1 - Τι είναι οι Τεχνολογίες Διαδικτύου και πως χρησιμοποιούνται.....	33
2.4.2 - HTML.....	33
2.4.3 - HTML5.....	34
2.4.4 - CSS.....	37
2.4.5 - JavaScript	39
2.4.6 - PHP	42
3 ^ο Κεφάλαιο - Web Browsers.....	44
3.1 - Εισαγωγή	44
3.2 - Λειτουργία των Browsers	44
3.3 - Στοιχεία ενός Browser	45

3.4 - Ιστορία των Web Browsers.....	46
3.5 - Firefox	52
3.6 - Internet Explorer.....	55
3.7 - Opera	58
3.8 - Chrome	60
3.9 - Safari	62
3.10 - Maxthon.....	63
3.11 - Avant.....	64
3.12 - Seamonkey.....	65
3.13 - Σύγκριση των Browsers	66
3.13.1 - SunSpider JavaScript 1.0.1	66
3.13.2 - Peacekeeper Benchmark	67
3.13.3 - Kraken Benchmark.....	68
3.13.4 - RoboHornet	69
3.13.5 - Octane Benchmark	70
3.13.6 - Avant Browser	71
3.13.7 - Αποτελέσματα	71
4 ^ο Κεφάλαιο - HTML5 New Features	72
4.1 – Εισαγωγή.....	72
4.2 – Χαρακτηριστικά	72
4.3 – Semantic Elements.....	73
4.3.1 – Semantic Elements, Υποστήριξη στους Browsers	73
4.3.2 – Semantic Elements, Κώδικας	75
4.3 – Video & Audio	77
4.3.1 - HTML5 Video vs Flash Video.....	77
4.3.2 – Video & Audio, Υποστήριξη στους Browsers	79
4.3.3 – Video & Audio, Κώδικας	80
4.4 – Input Types.....	81
4.4.1 – Input Types, Υποστήριξη στους Browsers	81

4.4 – Form Elements	86
4.4.1 – Form Elements, Υποστήριξη στους Browsers	86
4.4.2 – Form Elements, Κώδικας	90
4.5 – Form Attributes	92
4.4.1 - Form Attributes, Υποστήριξη στους Browsers	93
4.4.2 - Form Attributes, Κώδικας	104
4.6 - Canvas	107
4.6.1 - Canvas, Υποστήριξη στους Browsers	107
4.6.2 - Canvas, Κώδικας	108
4.7 - Drag and Drop	109
4.7.1 - Drag n Drop, Υποστήριξη στους Browsers	110
4.7.2 - Drag n Drop, Κώδικας	110
4.8 - Local Storage	112
4.8.1 - Local Storage, Υποστήριξη στους Browsers	112
4.8.2 - Local Storage, Κώδικας	115
4.9 - HTML5 Βαθμολογίες	117
5° Κεφάλαιο - Συμπεράσματα & Μελλοντικές Μελέτες	118
Βιβλιογραφία	119

Περιεχόμενα Εικόνων

Εικόνα 2. 1: The Internet	11
Εικόνα 2. 2: Network Topology.....	12
Εικόνα 2. 3: Arpanet	13
Εικόνα 2. 4: Χρήστες του Internet	13
Εικόνα 2. 5: Tcp / Ip Layers.....	14
Εικόνα 2. 6: TCP / IP Layers.....	14
Εικόνα 2. 7: TCP vs UDP	16
Εικόνα 2. 8: IP Header.....	17
Εικόνα 2. 9: IPV4 / IPV6 Headers	18
Εικόνα 2. 10: IPV4 vs IPV6.....	18
Εικόνα 2. 11: Internet Networks.....	19
Εικόνα 2. 12: DNS Πίνακας	19
Εικόνα 2. 13: Είδη μεταδόσεων.....	20
Εικόνα 2. 14: Επιλογή μονοπατιού πακέτου	20
Εικόνα 2. 15: DNS.....	22
Εικόνα 2. 16: Λειτουργία του WWW	23
Εικόνα 2. 17: Λειτουργία του WWW	23
Εικόνα 2. 18: Web 1.0.....	24
Εικόνα 2. 19: Web 2.0.....	25
Εικόνα 2. 20: Web 2.0.....	26
Εικόνα 2. 21: Χαρακτηριστικά του Web 2.0	27
Εικόνα 2. 22: Web 1.0 vs Web 2.0 vs Semantic Web	29
Εικόνα 2. 23: Semantic Web	29
Εικόνα 2. 24: Web 1.0 vs Web 2.0 vs Web 3.0	30
Εικόνα 2. 25: Σύνοψη Web 1.0 / Web 2.0 / Web 3.0	30
Εικόνα 2. 26: Web 4.0.....	31
Εικόνα 2. 27: Χρονικό διάγραμμα εκδόσεων του ιστού	32
Εικόνα 2. 28: HTML παράδειγμα κώδικα	33
Εικόνα 2. 29: HTML5 δομικά στοιχεία.....	35
Εικόνα 2. 30: HTML5 κατάσταση νέων στοιχείων.....	36
Εικόνα 2. 31: CSS παράδειγμα.....	37
Εικόνα 2. 32: Λειτουργία του CSS.....	38
Εικόνα 2. 33: JavaScript παράδειγμα	39
Εικόνα 2. 34: AJAX.....	41
Εικόνα 2. 35: PHP λειτουργία	43

Εικόνα 3. 1: Λειτουργία του browser	44
Εικόνα 3. 2: : Λειτουργία του browser	45
Εικόνα 3. 3: Μερίδιο χρήσης Netscape	46
Εικόνα 3. 4: Μερίδια χρήσης 07/2008 - 07/2013.....	47
Εικόνα 3. 5: : Μερίδια χρήσης 2008	48
Εικόνα 3. 6: Μερίδια χρήσης 2009	48
Εικόνα 3. 7: Μερίδια χρήσης 2010.....	49
Εικόνα 3. 8: Μερίδια χρήσης 2011	49
Εικόνα 3. 9: Μερίδια χρήσης 2012	50
Εικόνα 3. 10: Μερίδια χρήσης 2013	50
Εικόνα 3. 11: Μερίδια χρήσης τελευταίου έτους.....	51
Εικόνα 3. 12: Εικονίδια του Phoenix / Firefox	52
Εικόνα 3. 13: Η πρώτη έκδοση του IE.....	55
Εικόνα 3. 14: IE 10.....	57
Εικόνα 3. 15: Spatial Navigation	59
Εικόνα 3. 16: SunSpider Benchmark.....	66
Εικόνα 3. 17: Peacekeeper Benchmark	67
Εικόνα 3. 18: Kraken Benchmark	68
Εικόνα 3. 19: RoboHornet Benchmark	69
Εικόνα 3. 20: Octane Benchmark.....	70
Εικόνα 4. 1: Semantic elements σε Firefox.....	74
Εικόνα 4. 2: Semantic elements σε IE.....	74
Εικόνα 4. 3: HTML5 new Doctype and Charset	75
Εικόνα 4. 4: Semantic elements code	75
Εικόνα 4. 5: Semantic elementsc code	76
Εικόνα 4. 6: Semantic elements code.....	76
Εικόνα 4. 7: HTML5 Video vs Flash Video	77
Εικόνα 4. 8: HTML5 vs Flash.....	78
Εικόνα 4. 9: HTML5 vs Flash.....	79
Εικόνα 4. 10: Avant browser video / audio support.....	79
Εικόνα 4. 11: Opera mp4 support.....	80
Εικόνα 4. 12: HTML5 video / audio code	80
Εικόνα 4. 13: Firefox input type support	82
Εικόνα 4. 14: Seamonkey input type support.....	82
Εικόνα 4. 15: Chrome input type support	83
Εικόνα 4. 16: Safari input type support	83
Εικόνα 4. 17: IE input type support	84

Εικόνα 4. 18: Maxthon input type support.....	84
Εικόνα 4. 19: Opera input type support	85
Εικόνα 4. 20: Avant input type support.....	85
Εικόνα 4. 21: Firefox form elements support.....	86
Εικόνα 4. 22: Seamonkey form elements support	87
Εικόνα 4. 23: Chrome form elements support	87
Εικόνα 4. 24: Opera form elements support	88
Εικόνα 4. 25: Avant form elements support.....	88
Εικόνα 4. 26: Maxthon form element support.....	89
Εικόνα 4. 27: Safari form element support.....	89
Εικόνα 4. 28: IE form element support.....	90
Εικόνα 4. 29: Datalist element.....	91
Εικόνα 4. 30: Keygen form element	91
Εικόνα 4. 31: Output form element.....	91
Εικόνα 4. 32: Chrome min / max attribute	93
Εικόνα 4. 33: Chrome step attribute	94
Εικόνα 4. 34: Chrome pattern attribute	94
Εικόνα 4. 35: Chrome multiple attribute	95
Εικόνα 4. 36: Opera min / max attribute	96
Εικόνα 4. 37: Opera step attribute	96
Εικόνα 4. 38: Opera pattern attribute	97
Εικόνα 4. 39: Safari form attributes support	98
Εικόνα 4. 40: Avant pattern support	98
Εικόνα 4. 41: Avant required support.....	99
Εικόνα 4. 42: Avant min / max support	99
Εικόνα 4. 43: Avant Step support	100
Εικόνα 4. 44: Avant Multiple file support.....	100
Εικόνα 4. 45: : Internet Explorer form attributes support.....	101
Εικόνα 4. 46: Maxthon form attributes support.....	101
Εικόνα 4. 47: Firefox pattern support.....	102
Εικόνα 4. 48: Firefox required support.....	102
Εικόνα 4. 49: Firefox multiple support	103
Εικόνα 4. 50: Seamonkey form attributes support	103
Εικόνα 4. 51: Form novalidate code	104
Εικόνα 4. 52: Autofocus / form attributes code	104
Εικόνα 4. 53: Min / Max attributes code	104
Εικόνα 4. 54: Multiple files code.....	105
Εικόνα 4. 55: Pattern code.....	105

Εικόνα 4. 56: Placeholders code	105
Εικόνα 4. 57: Step code.....	106
Εικόνα 4. 58: Canvas	107
Εικόνα 4. 59: Create canvas item.....	108
Εικόνα 4. 60: Σχεδίαση ορθογωνίου	108
Εικόνα 4. 61: Σχεδίαση κύκλου και εισαγωγή κειμένου	108
Εικόνα 4. 62: Drag n Drop Function	110
Εικόνα 4. 63: Ορισμός draggable αντικειμένου	110
Εικόνα 4. 64: Ορισμός ondragstart.....	111
Εικόνα 4. 65: Ορισμός περιοχής για μετακίνηση	111
Εικόνα 4. 66: Ορισμός ondragover	111
Εικόνα 4. 67: Συνάρτηση ondrop.....	112
Εικόνα 4. 68: Local Storage αρχική	113
Εικόνα 4. 69: Local Storage εισαγωγή τιμής.....	114
Εικόνα 4. 70: Local Storage αποτέλεσμα.....	114
Εικόνα 4. 71: Κάλεισμα συνάρτησης myfun()	115
Εικόνα 4. 72: Δημιουργία Local Storage μεταβλητής.....	115
Εικόνα 4. 73: Κάλεισμα συνάρτησης show()	116
Εικόνα 4. 74: Local Storage Αποτέλεσμα	116

1^ο Κεφάλαιο - Εισαγωγή

1.1 - Αντικείμενο της εργασίας

Τα τελευταία χρόνια οι υπολογιστές έχουν γίνει αναπόσπαστο κομμάτι της καθημερινής μας ζωής. Μία από τις δημοφιλέστερες χρήσεις είναι αυτή του Παγκόσμιου Ιστού (Web). Όλοι μας χρησιμοποιούμε το διαδίκτυο καθημερινά, είτε για τις επαγγελματικές μας υποχρεώσεις, για τη διασκέδασή μας ή για την ενημέρωσή μας.

Από την αρχή της δημιουργίας του ιστού, αυτός έχει εξελιχθεί προσφέροντας όλο και περισσότερες λειτουργίες και περισσότερα χαρακτηριστικά κάνοντας την πλοήγηση πιο εύκολη και πιο ευχάριστη. Μία από τις τελευταίες προσθήκες είναι η νέα έκδοση της HTML, η HTML5.

Έτσι, αντικείμενο της παρούσας πτυχιακής εργασίας καθορίστηκε να είναι η μελέτη της HTML5 και των νέων χαρακτηριστικών που προσφέρει, καθώς επίσης και η έρευνα του κατά πόσο υποστηρίζεται από τους browsers.

Για να επιτευχθεί αυτή η συγκριτική μελέτη, υλοποιήθηκε μία ιστοσελίδα δομημένη αποκλειστικά και μόνο με χαρακτηριστικά της HTML5. Έτσι, και για κάθε ένα από τα χαρακτηριστικά ελέγχουμε το κατά πόσο υποστηρίζονται από τους browsers.

1.2 - Δομή της εργασίας

Η εργασία είναι δομημένη ως εξής:

Στο 2^ο κεφάλαιο γίνεται μία εισαγωγή αναλύοντας καταρχάς τι είναι το Internet, πως δημιουργήθηκε και πως λειτουργεί. Επίσης αναλύεται το τι ακριβώς είναι ο Παγκόσμιος Ιστός, ποια η διαφορά του από το Internet, πως δημιουργήθηκε, πως λειτουργεί και ποια τα χαρακτηριστικά που έχει λάβει από τη στιγμή της δημιουργίας του. Τέλος παραθέτουμε μερικές τεχνολογίες διαδικτύου που είναι απαραίτητες για τη δημιουργία και την προβολή των ιστοσελίδων, όπως είναι για παράδειγμα η HTML, το CSS και η JavaScript.

Στο 3^ο κεφάλαιο παραθέτουμε μερικά ιστορικά στοιχεία για διάφορους browsers με τους οποίους γίνεται η συγκριτική μελέτη. Παραθέτονται επίσης τα αποτελέσματα από benchmarks στα οποία τους υποβάλλαμε με σκοπό την σύγκρισή τους.

Τέλος, στο 4^ο κεφάλαιο παραθέτουμε τα νέα χαρακτηριστικά που προσφέρονται από την HTML5. Στη συνέχεια και έχοντας το κάθε ένα από αυτά τα χαρακτηριστικά ως βάση σύγκρισης, βλέπουμε το κατά πόσο υποστηρίζεται από τους browsers.

2^ο Κεφάλαιο - Ο Παγκόσμιος Ιστός

2.1 - Εισαγωγή

Ο Παγκόσμιος Ιστός (World Wide Web) είναι ένα σύστημα συνδεδεμένων αρχείων μέσω υπέρ-συνδέσμων (hyperlinks) τα οποία είναι προσβάσιμα μέσω του διαδικτύου (Internet). Για την εμφάνιση αυτών των αρχείων χρησιμοποιούνται οι browsers και μπορούν να περιέχουν κείμενο, εικόνες, αρχεία βίντεο και ήχου, σχεδιαγράμματα και άλλα. Η πλοήγηση ανάμεσα στα διάφορα αρχεία γίνεται με τους υπέρ-συνδέσμους (hyperlinks).

2.2 – Internet

2.2.1 - Τι είναι το internet

Το Internet είναι ένα παγκόσμιο σύστημα διασυνδεδεμένων δικτύων τα οποία χρησιμοποιούν ένα συγκεκριμένο πρωτόκολλο επικοινωνίας (TCP/IP). Αποτελείται από διάφορα επιμέρους δίκτυα (ιδιωτικά, δημόσια, ακαδημαϊκά). Το Internet απαρτίζεται από διάφορες υπηρεσίες όπως τα peer-to-peer δίκτυα, οι email παροχές, τα ftp δίκτυα καθώς και τα αρχεία του παγκόσμιου ιστού (WWW). [30]



Εικόνα 2. 1: The Internet [30]

Το internet είναι συντομογραφία του “Interconnection of computer Networks”. Είναι ένας χαώδης συνδυασμός εκατομμυρίων προσωπικών και εταιρικών υπολογιστών. Μέσω του internet μπορεί να συνδέονται υπολογιστές, smartphones, GPS, mp3 players, συναγερμοί και άλλα.

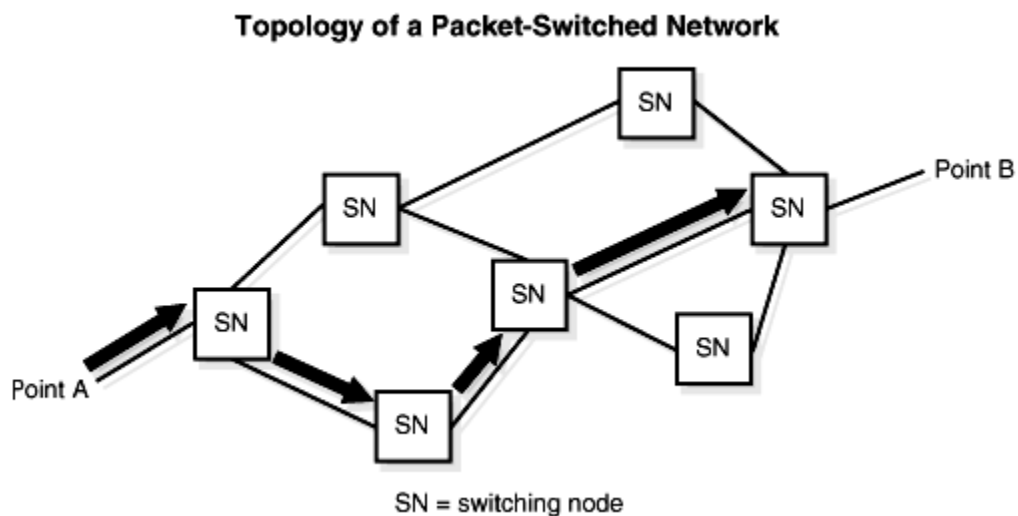
2.2.2 - Δημιουργία του internet

Στις αρχές του 1960 έγιναν πολλές μελέτες για την ανταλλαγή πακέτων μεταξύ των υπολογιστών και αναπτύχθηκαν δίκτυα όπως τα Mark I, ARPANET, CYCLASSES, Merit Network, Tymnet και Telenet, τα οποία χρησιμοποιούσαν διάφορα πρωτόκολλα.

Το ARPANET (Advanced Research Projects Agency's Wide Area Network) καθιερώθηκε από το υπουργείο άμυνας των Ηνωμένων Πολιτειών της Αμερικής με στόχο την στρατιωτική έρευνα.

Αρχικός στόχος ήταν να βρεθεί ένας τρόπος με τον οποίο θα μπορούσε η αμερικανική πολεμική αεροπορία να κρατήσει τον έλεγχο των δυνάμεών της μετά από επίθεση. Έτσι έπρεπε να δημιουργηθεί ένα αποκεντρωμένο και άρα λιγότερο τρωτό δίκτυο μεταγωγής πακέτων. Το κάθε μήνυμα θα χωρίζεται σε πακέτα και αυτά δρομολογούνται στο δίκτυο έχοντας ένα κοινό προορισμό.

Το δίκτυο που αναπτύχθηκε δεν είχε κάποιο κεντρικό εξυπηρετητή που να καθορίζει τη διαδρομή των πακέτων. Το κάθε πακέτο μπορούσε να ακολουθήσει οποιαδήποτε τυχαία διαδρομή. Όλοι οι κόμβοι ήταν ίσοι μεταξύ τους. Ο τελικός κόμβος στη συνέχεια θα έβαζε όλα τα πακέτα στη σειρά και θα συναρμολογούσε το μήνυμα που στάλθηκε. Εάν κάποιος κόμβος παρουσίαζε βλάβη τα πακέτα θα παρέμεναν στον κόμβο μέχρι να επιδιορθωθεί η βλάβη ενώ εάν αυτός καταστρέφονταν, τότε τα πακέτα θα ακολουθούσαν άλλη διαδρομή. Όλα τα παραπάνω εξασφάλιζαν ότι το δίκτυο θα ήταν ανεκτικό σε οποιαδήποτε επίθεση ακριβώς επειδή ο κάθε κόμβος ήταν ανεξάρτητος και δεν υπήρχε κεντρική διοίκηση από κάποια μονάδα.



Εικόνα 2. 2: Network Topology [31]

Το ARPANET οδήγησε στην ανάπτυξη πρωτοκόλλων με τα οποία διάφορα ξεχωριστά δίκτυα μπορούσαν να ενωθούν σε ένα δίκτυο δικτύων. Αργότερα στο δίκτυο προστέθηκαν πανεπιστήμια και ινστιτούτα έχοντας ως αποτέλεσμα το δίκτυο να μεγαλώσει τόσο, ξεπερνώντας κάθε προσδοκία και εντέλει να πάρει την ονομασία Internet.



Εικόνα 2. 3: Arpanet [53]

Τα TCP/IP δίκτυα άρχισαν να επεκτείνονται το 1986 όταν ο National Science Foundation (NSFNET) προμήθευσε τις ΗΠΑ με υπέρ-υπολογιστές. Εμπορικοί πάροχοι (ISPs) άρχισαν να εμφανίζονται στα τέλη του 1980 και στις αρχές του 1990. Το internet εμποροποιήθηκε το 1995 όταν ο NSFNET καταργήθηκε και αφαιρέθηκαν οι τελευταίοι περιορισμοί της χρήσης του για εμπορικούς σκοπούς.

Από τα μέσα του 1990 το internet εξελίσσεται συνεχώς παρέχοντας όλο και περισσότερες υπηρεσίες (instant messaging, video κλήσεις, εύρεση πληροφοριών, κοινωνικά δίκτυα) έχοντας αντίκτυπο σε κάθε σημείο της ζωής μας. Οι χρήστες του αυξάνονται όλο και περισσότερο τα τελευταία χρόνια.[31]

Worldwide Internet users

	2005	2010	2013 ^a
World population^[24]	6.5 billion	6.9 billion	7.1 billion
Not using the Internet	84%	70%	61%
Using the Internet	16%	30%	39%
Users in the developing world	8%	21%	31%
Users in the developed world	51%	67%	77%

^a Estimate.

Source: International Telecommunications Union.^[25]

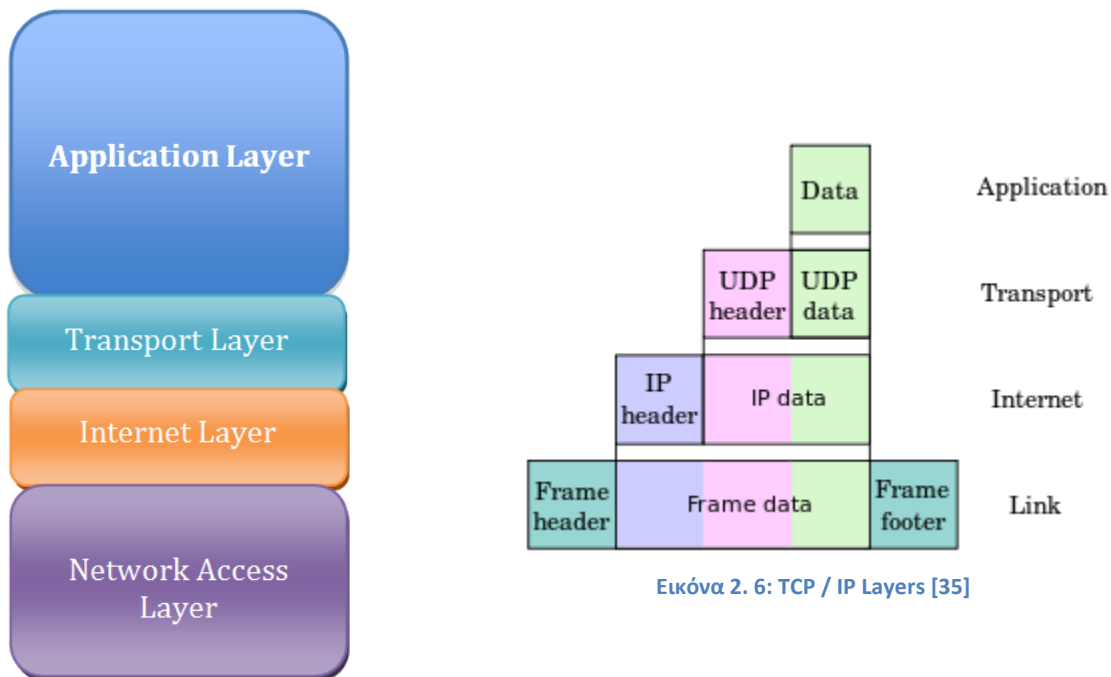
Εικόνα 2. 4: Χρήστες του Internet [58]

2.2.3 - Δομή και λειτουργία του Internet

Η δομή του internet βασίζεται τόσο στο hardware κομμάτι όσο και στο software, το οποίο διαχειρίζεται διάφορες πλευρές της ευρύτερης αρχιτεκτονικής. Η επιτυχία του internet μπορούμε να πούμε ότι βασίζεται στο μεγαλύτερο κομμάτι από το software.

Το internet περιγράφεται από ένα framework, το **Internet Protocol Suite**. Αποτελεί μία αρχιτεκτονική η οποία διαιρεί τις μεθόδους σε διάφορα στρώματα πρωτοκόλλων. Στην κορυφή βρίσκεται το **Application Layer**. Παρέχει τον τρόπο στις διάφορες εφαρμογές να έχουν πρόσβαση στις υπηρεσίες του δικτύου. Σε αυτό το επίπεδο οι εφαρμογές δημιουργούν τα δεδομένα των χρηστών και ανταλλάσσουν μεταξύ τους αυτά τα δεδομένα. Εδώ υποστηρίζονται και τα high - level πρωτόκολλα όπως τα SMTP, FTP, SSH και HTTP. Σημαντικό στοιχείο είναι ότι υποστηρίζει τόσο το TCP (SMTP) όσο και το UDP (FTP) πρωτόκολλο.

TCP/IP



Εικόνα 2. 5: Tcp / Ip Layerrs [54]

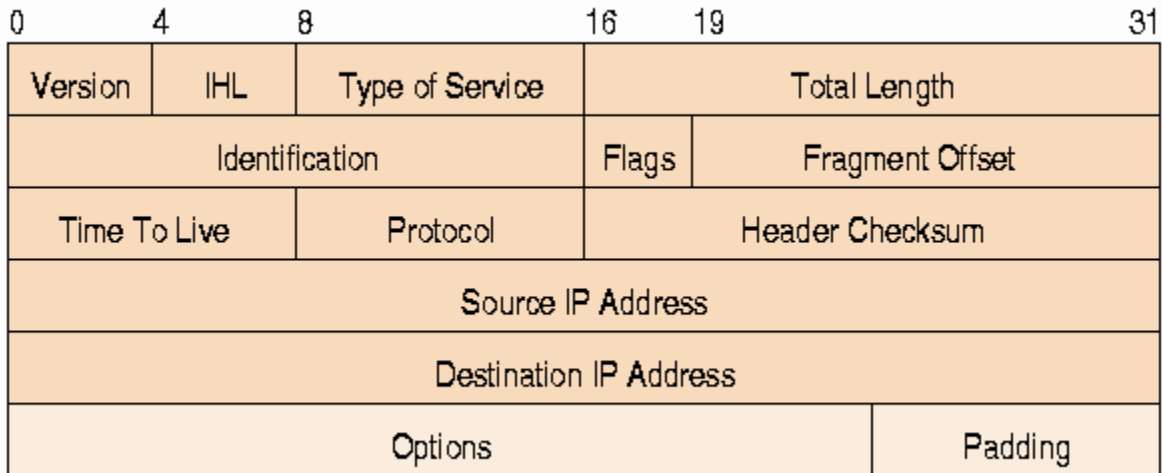
Εικόνα 2. 6: TCP / IP Layers [35]

Κάτω από αυτό το επίπεδο βρίσκεται το **Transport Layer**, το οποίο συνδέει διαφορετικές εφαρμογές μέσω του δικτύου. Με λίγα λόγια είναι το υπεύθυνο κομμάτι για την μεταφορά και παράδοση των μηνυμάτων. Τα πρωτόκολλα που χρησιμοποιούνται για την μεταφορά των πακέτων είναι το **TCP** (Transmission Control Protocol) και το **UDP** (User Datagram Protocol). Ανάλογα με τις προδιαγραφές αξιοπιστίας επιλέγεται το κατάλληλο πρωτόκολλο. Εδώ ανακαλύπτονται τα τυχόν λάθη και ακολουθούνται οι διαδικασίες ανάκαμψης.

Summary Comparison of UDP and TCP		
Characteristic / Description	UDP	TCP
General Description	Simple, <u>high-speed</u> , low-functionality "wrapper" that interfaces applications to the network layer and does little else.	Full-featured protocol that allows applications to send data reliably without worrying about network layer issues.
Protocol Connection Setup	Connectionless; data is sent without setup.	Connection-oriented; connection must be established prior to transmission.
Data Interface To Application	Message-based; data is sent in discrete packages by the application.	Stream-based; data is sent by the application with no particular structure.
Reliability and Acknowledgments	Unreliable, best-effort delivery without acknowledgments.	Reliable delivery of messages; all data is acknowledged.
Retransmissions	Not performed. Application must detect <u>lost data</u> and retransmit if needed.	Delivery of all data is managed, and lost data is retransmitted automatically.
Features Provided to Manage Flow of Data	None	Flow control using sliding windows; window size adjustment heuristics; congestion avoidance algorithms.
Overhead	Very low	Low, but higher than UDP
Transmission Speed	Very high	High, but not as high as UDP
Data Quantity Suitability	Small to moderate amounts of data (up to a few hundred bytes)	Small to very large amounts of data (up to gigabytes)
Types of Applications That Use The Protocol	Applications where data delivery speed matters more than completeness, where	Most protocols and applications sending data that must be

Εικόνα 2. 7: TCP vs UDP [55]

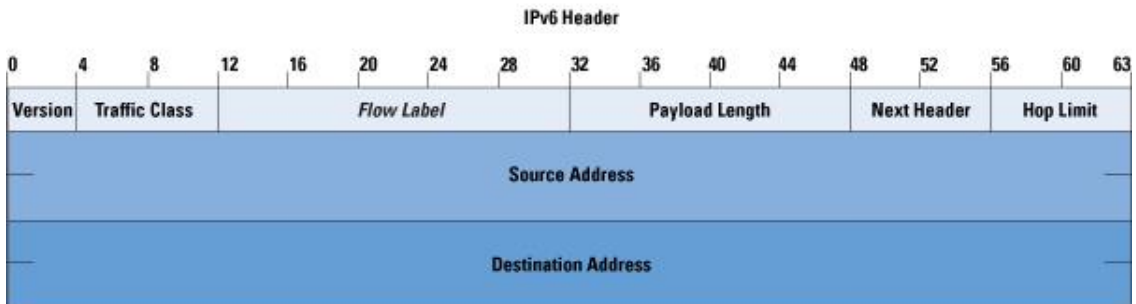
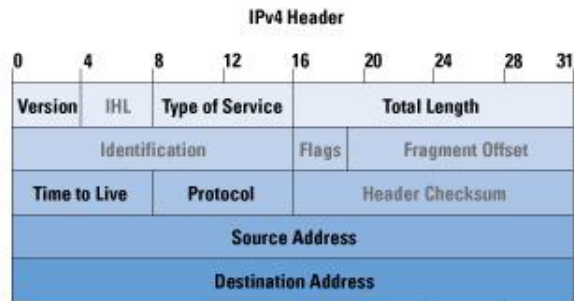
Στη συνέχεια, το **Internet Layer** επιτρέπει στους υπολογιστές να ανακαλύπτουν, να αναγνωρίζουν ο ένας τον άλλο μέσω των **IP** διευθύνσεων και εν τέλει να συνδέονται μεταξύ τους. Είναι το κυρίαρχο συστατικό της αρχιτεκτονικής. Διοχετεύει τα πακέτα μέσα στα δίκτυα με σκοπό να φθάσουν ανεξάρτητα το ένα από το άλλο στον τελικό προορισμό. Τα πακέτα μπορεί να φθάσουν με διαφορετική σειρά από αυτήν που απεστάλησαν.



Εικόνα 2. 8: IP Header [56]

Αυτή τη στιγμή χρησιμοποιούνται δύο εκδόσεις του IP πρωτοκόλλου. Το IPv4 και το IPv6. Το IPv4 χρησιμοποιεί 32bits για τον καθορισμό των διευθύνσεων. Αυτό σημαίνει ότι μπορεί να υποστηρίξει 2^{32} IP διευθύνσεις (περίπου 4.29 δισεκατομμύρια). Ενώ μπορεί να φαίνεται μεγάλος αυτός ο αριθμός, η αλήθεια είναι ότι όλες αυτές οι διευθύνσεις έχουν ήδη αποδοθεί σε συσκευές.

Με το IPv6 προσφέρεται λύση σε αυτό το πρόβλημα καθώς χρησιμοποιεί 128bits για τον καθορισμό των διευθύνσεων, δηλαδή μπορεί να υποστηρίξει 2^{128} διευθύνσεις.



Εικόνα 2. 9: IPv4 / IPv6 Headers [57]

Differences between IPv4 and IPv6

IPv4	IPv6
Source and destination address are 32bit or 4Bytes	Source and Destination addresses are 128bit or 16Byte
Header includes checksum	Header does not include checksum
Header includes options	All optional data in moved to IPv6 extension headers
Broadcast addresses are used to send packets to all nodes on a subnet	No Broadcast addresses. Instead link-local scope all-nodes multicast address is used.
Manual or DHCP based IP configuration	No need of manual or DHCP configuration. Nodes are capable of auto-configuration

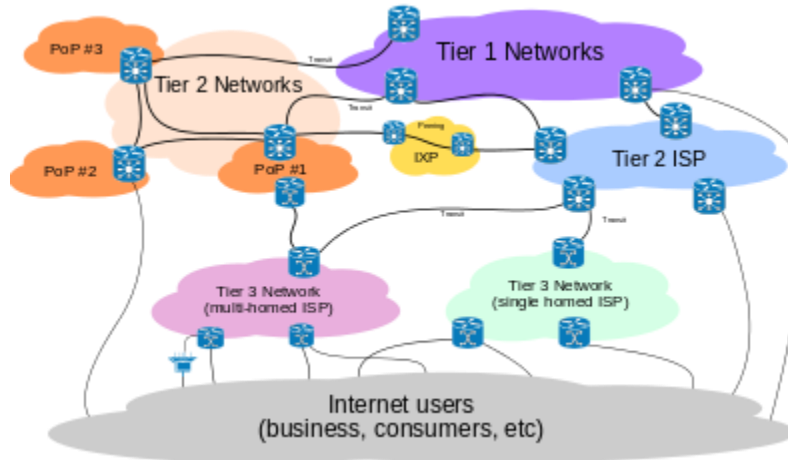
WWW.SecurityTube.NET

2

Εικόνα 2. 10: IPv4 vs IPv6

Το τελευταίο επίπεδο της αρχιτεκτονικής είναι το **Link Layer**. Περιλαμβάνει τις μεθόδους με τις οποίες τα τοπικά δίκτυα συνδέονται με τα routers. Περιγράφει δηλαδή πως θα αποσταλούν τα πακέτα στα γειτονικά hosts.

Άλλο σημαντικό στοιχείο του Internet, είναι το Routing, δηλαδή η επιλογή των διαδρομών για να αποσταλούν τα πακέτα. Η διαδικασία επιτυγχάνεται μέσω διάφορων δικτύων βαθμίδων (Tier Networks).



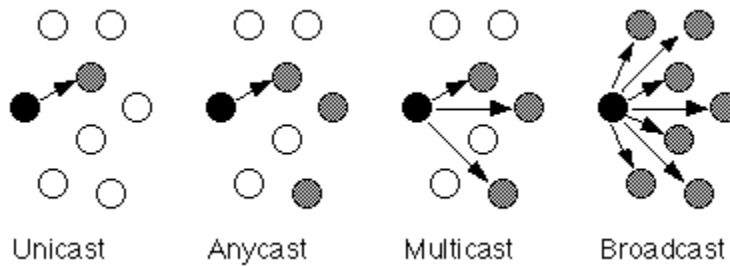
Εικόνα 2. 11: Internet Networks [58]

Οι ενδιάμεσοι σταθμοί των πακέτων αποτελούνται συνήθως από routers, bridges, gateways, firewalls ή switches. Για την ανάθεση των διαδρομών στα πακέτα χρησιμοποιούνται routing tables, τα οποία περιέχουν διαδρομές για διάφορες τοποθεσίες. Αυτοί οι πίνακες είναι αποθηκευμένοι στη μνήμη των routers.

RRAS-ROUTER1 - IP Routing Table					
Destination	Network mask	Gateway	Interface	Metric	Protocol
10.57.76.0	255.255.255.0	10.57.76.1	Local Area C...	1	Local
10.57.76.1	255.255.255.255	127.0.0.1	Loopback	1	Local
10.255.255.255	255.255.255.255	10.57.76.1	Local Area C...	1	Local
127.0.0.0	255.0.0.0	127.0.0.1	Loopback	1	Local
127.0.0.1	255.255.255.255	127.0.0.1	Loopback	1	Local
192.168.45.0	255.255.255.0	192.168.45.1	Local Area C...	1	Local
192.168.45.1	255.255.255.255	127.0.0.1	Loopback	1	Local
224.0.0.0	224.0.0.0	192.168.45.1	Local Area C...	1	Local
224.0.0.0	224.0.0.0	10.57.76.1	Local Area C...	1	Local
255.255.255.255	255.255.255.255	192.168.45.1	Local Area C...	1	Local
255.255.255.255	255.255.255.255	10.57.76.1	Local Area C...	1	Local

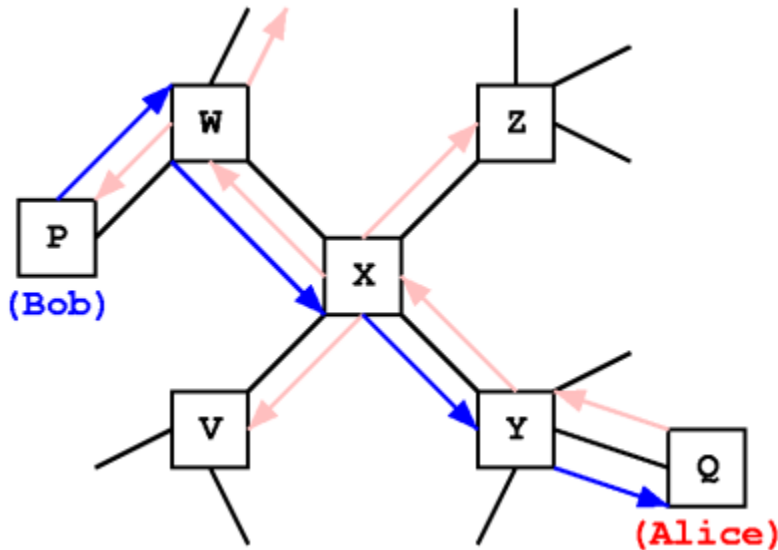
Εικόνα 2. 12: DNS Πίνακας [59]

Υπάρχουν διάφορα είδη routing: το **unicast** (μετάδοση μηνύματος σε συγκεκριμένο κόμβο), **broadcast** (μετάδοση μηνύματος σε όλους τους κόμβους του δικτύου), **multicast** (μετάδοση μηνύματος σε όλους το γκρουπ των κόμβων που θέλουν να λάβουν το μήνυμα), **anycast** (μετάδοση μηνύματος σε οποιονδήποτε έξω από το γκρουπ κόμβων) και **geocast** (μετάδοση μηνύματος σε γεωγραφική περιοχή).



Εικόνα 2. 13: Είδη μεταδόσεων [60]

Η επιλογή του μονοπατιού που θα ακολουθηθεί υπολογίζεται από διάφορους αλγόριθμους οι οποίοι βρίσκουν την καλύτερη δυνατή λύση. Μερικά από τα δεδομένα που λαμβάνονται υπόψη για τον υπολογισμό της διαδρομής είναι το bandwidth, η καθυστέρηση του δικτύου, το τηλεπικοινωνιακό κόστος, η αξιοπιστία, ο φόρτος του δικτύου. [35] [36] [37]



Εικόνα 2. 14: Επιλογή μονοπατιού πακέτου

2.3 - Ο Παγκόσμιος Ιστός (WWW)

2.3.1 - Τι είναι ο Παγκόσμιος Ιστός (World Wide Web)

Ο **Παγκόσμιος Ιστός (World Wide Web)** είναι ένα σύστημα υπέρ-συνδεδεμένων αρχείων τα οποία είναι προσβάσιμα μέσω του internet. Για να δει κανείς αυτά τα αρχεία πρέπει να χρησιμοποιήσει έναν **web browser**. Τα αρχεία μπορεί να περιέχουν κείμενο, εικόνα, βίντεο ή άλλα αρχεία multimedia. [32]

2.3.2 - Η ιστορία του Παγκόσμιου Ιστού

Ο ιστός αναπτύχθηκε στο διάστημα μεταξύ του Μαρτίου 1989 και του Δεκεμβρίου 1990 από τον Tim Berners-Lee (τωρινός διευθυντής του World Wide Consortium - W3C), ο οποίος συνέχισε να τον βελτιώνει μέχρι το 1993. Η αρχική πρόταση του Lee είχε σκοπό την δημιουργία ενός καλύτερου μηχανισμού επικοινωνίας στο CERN, όπου εργαζόταν εκείνο το διάστημα.

Οι αρχές οι οποίες στήριξε την πρότασή του έχουν μείνει αναλλοίωτες μέχρι και σήμερα. Αυτές είναι η **HTML** (Hyper Text Markup Language), το **URI** (Uniform Resource Identifier) και το **HTTP** (Hypertext Transfer Protocol). Στην πρότασή του υποστήριξε πως μπορεί να φτιάξει ένα "**Hypertext project**" με την ονομασία "**WorldWideWeb**", ως έναν ιστό από "hypertext" αρχεία. Θεωρούσε πως μπορεί να κατασκευαστεί ένας read - only ιστός σε μόλις τρεις μήνες, ενώ θα χρειαζόνταν περίπου έξι μήνες για να γίνει δυνατή η δημιουργία νέων link και αρχείων από τους αναγνώστες.

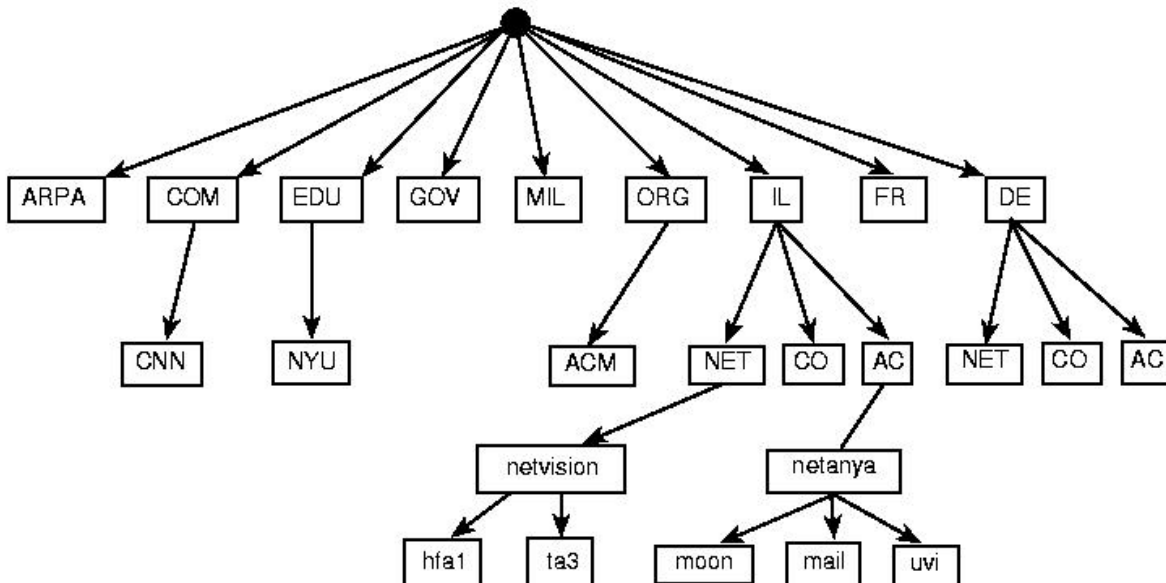
Ο Lee δημιούργησε επίσης τον πρώτο browser καθώς και τον πρώτο web server. Μέχρι το τέλος του 1990 η πρώτη ιστοσελίδα ήταν προσβάσιμη. Μέχρι το 1991 οι άνθρωποι έξω από το CERN ενώθηκαν με την πρώτη Web κοινότητα. Τον Απρίλιο του 1993 το CERN ανακοίνωσε ότι η τεχνολογία του Παγκόσμιου Ιστού θα ήταν ελεύθερη για τον καθένα να την χρησιμοποιήσει.

Η πρόταση που κατέθεσε ο Lee είχε κάποιες διαφορές από άλλα hypertext συστήματα. Ο ιστός απαιτούσε μόνο μη κατευθυνόμενους συνδέσμους παρά κατευθυνόμενους. Με αυτόν τον τρόπο ήταν δυνατόν για κάποιον να δείξει σε άλλο σύνδεσμο - προορισμό, χωρίς να απαιτείται ενέργεια από τον προορισμό. Αυτό το γεγονός έκανε ευκολότερη τη δημιουργία των web servers και των browsers. Στον αντίποδα εμφανίστηκε ένα άλλο πρόβλημα, κατά το οποίο κάποιοι από τους συνδέσμους μπορεί να «δείχνουν» σε άλλους συνδέσμους, οι οποίοι δεν είναι προσβάσιμοι εκείνη τη στιγμή.

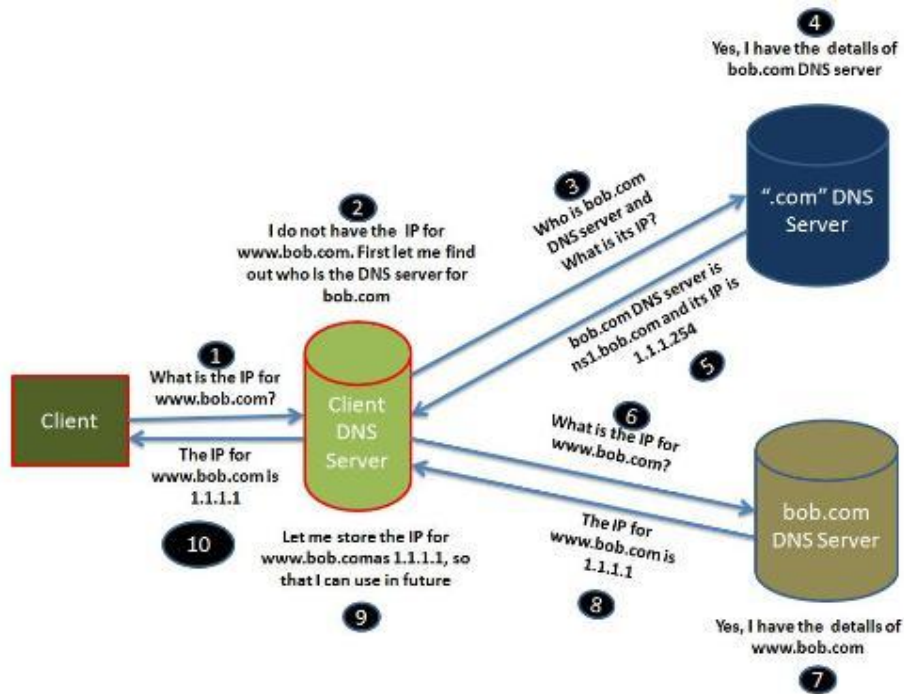
Ο Lee και η ομάδα του γρήγορα συνειδητοποίησαν ότι για να φθάσει η ιδέα του ιστού στις πραγματικές της δυνατότητες θα πρέπει οι ήδη υπάρχουσες για εκείνη τη στιγμή τεχνολογίες να γίνουν παγκόσμια standards. Έτσι, το 1994 ιδρύθηκε το **World Wide Consortium (W3C)**, ένας οργανισμός του οποίου στόχος ήταν να εξασφαλίσει ότι το Web θα ήταν λειτουργικό για όλους. Από τότε ο Lee κατέχει σημαντικό και πρωταγωνιστικό ρόλο στην βελτίωση του ιστού, όπως για παράδειγμα το όραμά του για τον Σημασιολογικό Ιστό (Semantic Web). [32] [33] [34]

2.3.3 - Πως λειτουργεί ο Ιστός

Για να επισκεφθεί κανείς μία ιστοσελίδα πρέπει είτε να γράψει τη διεύθυνσή της (**URL**), είτε να ακολουθήσει κάποιον υπέρ-σύνδεσμο (**hyperlink**) που οδηγεί εκεί. Μόλις δοθεί η διεύθυνση που θέλουμε να επισκεφθούμε ο browser αντιστοιχίζει το URL με την σωστή IP διεύθυνση, χρησιμοποιώντας την **DNS (Domain Name System)** βάση δεδομένων. Στη συνέχεια ο browser στέλνει μία HTTP αίτηση (**request**) στον υπολογιστή που αντιστοιχεί με την IP διεύθυνση που βρέθηκε νωρίτερα. Το αίτημα αποστέλλεται σε μία συγκεκριμένη θύρα που αντιστοιχεί στο IP Suite, για να διαχωριστεί από αιτήματα διαφορετικών υπηρεσιών.

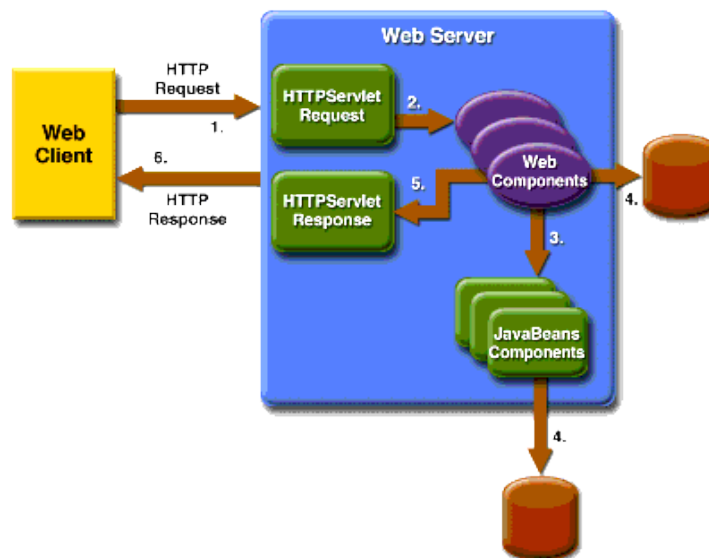


Εικόνα 2. 15: DNS



Εικόνα 2. 16: Λειτουργία του WWW [61]

Στη συνέχεια, ο υπολογιστής που δέχθηκε το αίτημα το μεταφέρει στον web server. Εάν μπορεί να εκτελεστεί επιτυχώς το αίτημα αποστέλλεται πίσω στον αρχικό υπολογιστή ένα HTTP response. Τότε ο browser μεταφράζει το HTML κείμενο (**response**) σε μορφή κατανοητή στον άνθρωπο και εμφανίζεται το αποτέλεσμα στο χρήστη.



Εικόνα 2. 17: Λειτουργία του WWW [62]

2.3.4 - Web Standards

Για την ομαλή λειτουργία του ιστού έχουν αναπτυχθεί διάφορα standards και τεχνικές προδιαγραφές. Πολλές από αυτές τις προδιαγραφές έχουν συνταχθεί από τον Tim Berners-Lee, ως μέρος του project του, ενώ άλλα δημιουργήθηκαν από την Internet Engineering Task Force (IETF) και άλλους οργανισμούς. Τα βασικά αυτά standards είναι:

- Προδιαγραφές για τις markup γλώσσες, ειδικά **HTML**, **XHTML**.
- Stylesheets, ειδικά **CSS**.
- Προδιαγραφές για scripting, ειδικά **JavaScript**.
- Προτάσεις για το **Document Object Model**.
- Προτάσεις για τα URI (**Uniform Resource Identifier**).
- Προτάσεις για το HTTP (**Hyper Text Transfer Protocol**).

2.3.5 - Web 1.0

Ο Web 1.0 ήταν η πρώτη μορφή του Παγκόσμιου Ιστού. Οι χρήστες μπορούσαν μόνο να δουν τις ιστοσελίδες, αλλά δεν μπορούσαν να διαμορφώσουν το περιεχόμενό τους. Τα κύρια χαρακτηριστικά του Web 1.0 ήταν οι στατικές σελίδες, η χρήση των framesets, η χρήση των πινάκων για να τοποθετηθούν και να στοιχηθούν τα στοιχεία στη σελίδα, gif κουμπιά και το ότι οι HTML φόρμες αποστέλλονταν με email.



Εικόνα 2. 18: Web 1.0 [63]

2.3.6 - Web2.0

Με τον όρο Web 2.0 εννοούμε τις ιστοσελίδες οι οποίες δεν μένουν στα πρότυπα των στατικών ιστοσελίδων που ίσχυε στην αρχή της δημιουργία του ιστού. Παρόλο που με τον όρο εννοείται μία νέα έκδοση του Παγκόσμιου Ιστού, δεν αναφέρεται σε ενημέρωση τεχνικών προδιαγραφών παρά σε αλλαγές στον τρόπο που οι ιστοσελίδες δημιουργούνται και χρησιμοποιούνται.



Εικόνα 2. 19: Web 2.0 [39]

Σε αντίθεση με την πρώτη έκδοση του ιστού, οι ιστοσελίδες του Web 2.0, επιτρέπουν την ανάδραση με τον χρήστη και την αλληλεπίδραση με διάφορα social media. Παραδείγματα ιστοσελίδων που συγκαταλέγονται στον Web 2.0, είναι τα μέσα κοινωνικής δικτύωσης, τα ιστολόγια, σελίδες διαμοιρασμού βίντεο και web εφαρμογές.

Οι ιστοσελίδες του Web 2.0 επιτρέπουν στους χρήστες τους να κάνουν πολύ περισσότερα πράγματα από το να ανακτούν απλώς πληροφορίες. Οι χρήστες έχουν το ελεύθερο να συνεισφέρουν. Αυτό από την άλλη έχει τα αρνητικά του στοιχεία, καθώς έχουν δημιουργηθεί τα φαινόμενα του spamming και του trolling.



Εικόνα 2. 20: Web 2.0 [64]

Τα κύρια χαρακτηριστικά του Web 2.0 είναι η συμμετοχή του χρήστη, στοιχειώδης πίστη για την σωστή διαγωγή των επισκεπτών, μαζική συμμετοχή και πλούσιο user interface. Άνετα μπορούμε να πούμε ότι με το Web 2.0, ο ιστός έχει γίνει μία πλατφόρμα η οποία περιέχει εφαρμογές και υπηρεσίες, στις οποίες μπορούμε να έχουμε πρόσβαση ανά πάσα στιγμή επιθυμούμε. [40]

Η εύρεση και η μετάδοση πληροφοριών έχει γίνει εύκολη υπόθεση, καθώς ο ίδιος ο ιστός έχει γίνει πολύ πιο απλός. Πλέον δεν είναι κάτι το στατικό, αλλά συνεχώς ενημερώνεται και αυξάνεται.



Εικόνα 2. 21: Χαρακτηριστικά του Web 2.0 [65]

Ο Web 2.0 μπορεί να περιγραφεί σε τρεις κύριους άξονες:

- **Rich Internet Application (RIA)**: περιγράφει την εμπειρία που έχει μεταφερθεί από το desktop στον browser, είτε ως γραφικά περιβάλλοντα είτε ως ευρηστία.
- **Web Oriented Architecture (WOA)**: περιγράφει πως ο Web 2.0 αποκαλύπτει την λειτουργικότητά τους έτσι ώστε άλλες εφαρμογές να τη χρησιμοποιήσουν προσφέροντας καλύτερες υπηρεσίες.
- **Social Web**: περιγράφει πως η αλληλεπίδραση με το χρήστη είναι ακόμη πιο σημαντική και τον προάγει σε κυρίαρχο συστατικό.

Web 1.0	Web 2.0
Στατικότητα	Δικτύωση και αυτοματισμός
Προϊόντα	Υπηρεσίες
Παραπομπή σε άλλες σελίδες μόνο με links	Real Time ενσωμάτωση στοιχείων (widgets, iframes, RSS feeds)
Read Only Web	Read - Write Web
Περιορισμένες δυνατότητες αμφίδρομης επικοινωνίας	Διαδραστικότητα και συμμετοχή του χρήστη
Owning	Sharing and remix
HTML	HTML, RSS, Ajax, XML
Προσωπική / Εταιρική Ιστοσελίδα	Ιστολόγιο ή wiki
Στατική διαφήμιση	Κόστος ανά επίσκεψη
Τοπική αποθήκευση	Online storage

Παραπάνω φαίνονται οι βασικές διαφορές ανάμεσα στον Web 1.0 και στον Web 2.0. [39][40][52]

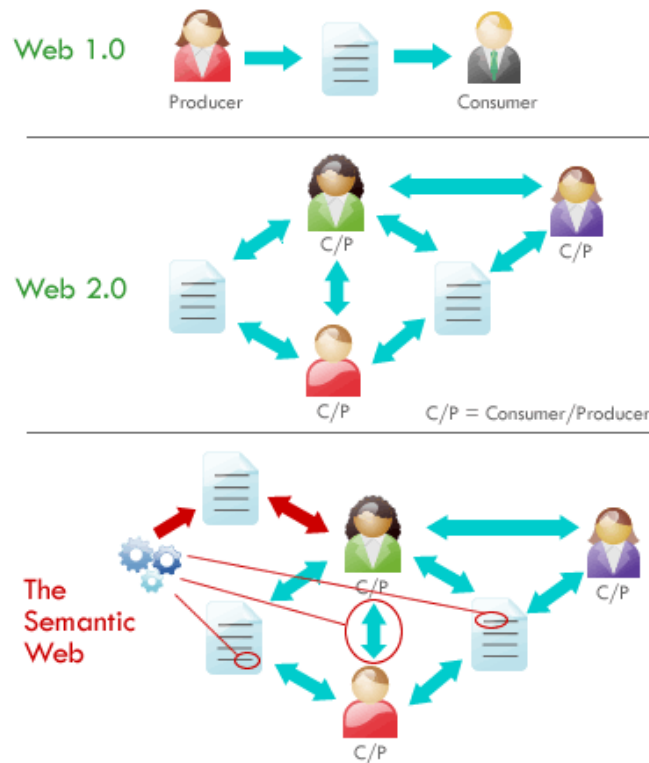
2.3.7 - Web 3.0

Ο **Σημαιολογικός Ιστός** (Web 3.0) προτάθηκε από τον World Wide Consortium. Σκοπός αυτής της πρότασης είναι να χρησιμοποιούνται κοινές μορφές δεδομένων, που να είναι κατανοητές από όλους. Με αυτόν τον τρόπο ο ιστός από μη δομημένα ή ημιδομημένα στοιχεία μετατρέπεται σε έναν ιστό δομημένων και κατανοητών πληροφοριών.

Σύμφωνα με τον W3C, ο Σημαιολογικός Ιστός παρέχει ένα κοινό πλαίσιο κατά το οποίο τα δεδομένα μπορούν να διαμοιράζονται και να επαναχρησιμοποιούνται από τις διάφορες εφαρμογές

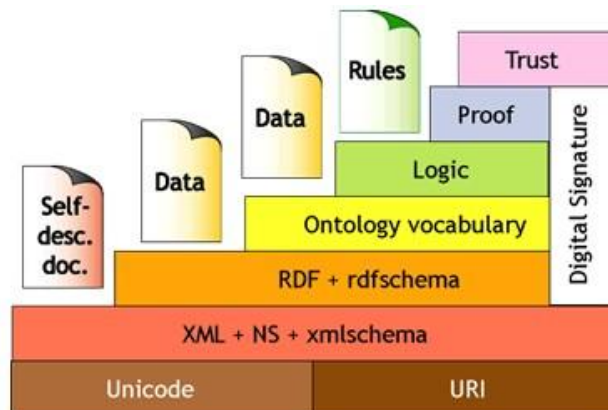
Σκοπός λοιπόν είναι τα δεδομένα να υπάρχουν σε τέτοια μορφή στην οποία οι μηχανές θα είναι ικανές να ερμηνεύσουν την σημασία τους, το περιεχόμενό και να ανταποκριθούν σε περίπλοκα αιτήματα των χρηστών.

Μέχρις στιγμής για την αναπαράσταση των δεδομένων σε μορφή κατανοητή από τους ανθρώπους χρησιμοποιείται η HTML. Δεν είναι δυνατόν όμως η γλώσσα ή το πρόγραμμα που την μεταγλωττίζει (browser), να καταλάβει τι νόημα έχουν τα δεδομένα που διαβάζει και προβάλλει.



Εικόνα 2. 22: Web 1.0 vs Web 2.0 vs Semantic Web [66]

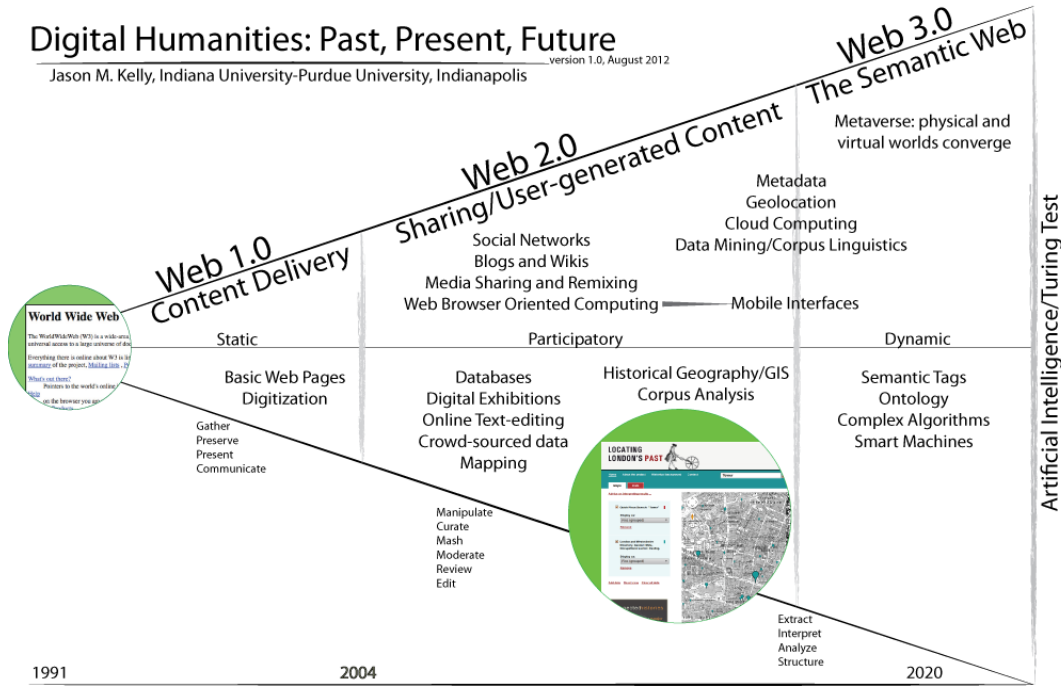
Σε αυτό το σημείο έρχεται ο Σηματολογικός Ιστός και δίνει λύση. Προτείνει γλώσσες ειδικά διαμορφωμένες για αποθήκευση δεδομένων και την κατανόηση των σχέσεων που έχουν αναμεταξύ τους, όπως οι **Resource Description Framework (RDF)**, **Web Ontology Language (OWL)** και η **Extensible Markup Language (XML)**. Η HTML συνεχίζει και χρησιμοποιείται στην αναπαράσταση αυτών των δεδομένων στην οθόνη του χρήστη. [41] [43]



Εικόνα 2. 23: Semantic Web [67]

Digital Humanities: Past, Present, Future

Jason M. Kelly, Indiana University-Purdue University, Indianapolis version 1.0, August 2012



Εικόνα 2. 24: Web 1.0 vs Web 2.0 vs Web 3.0

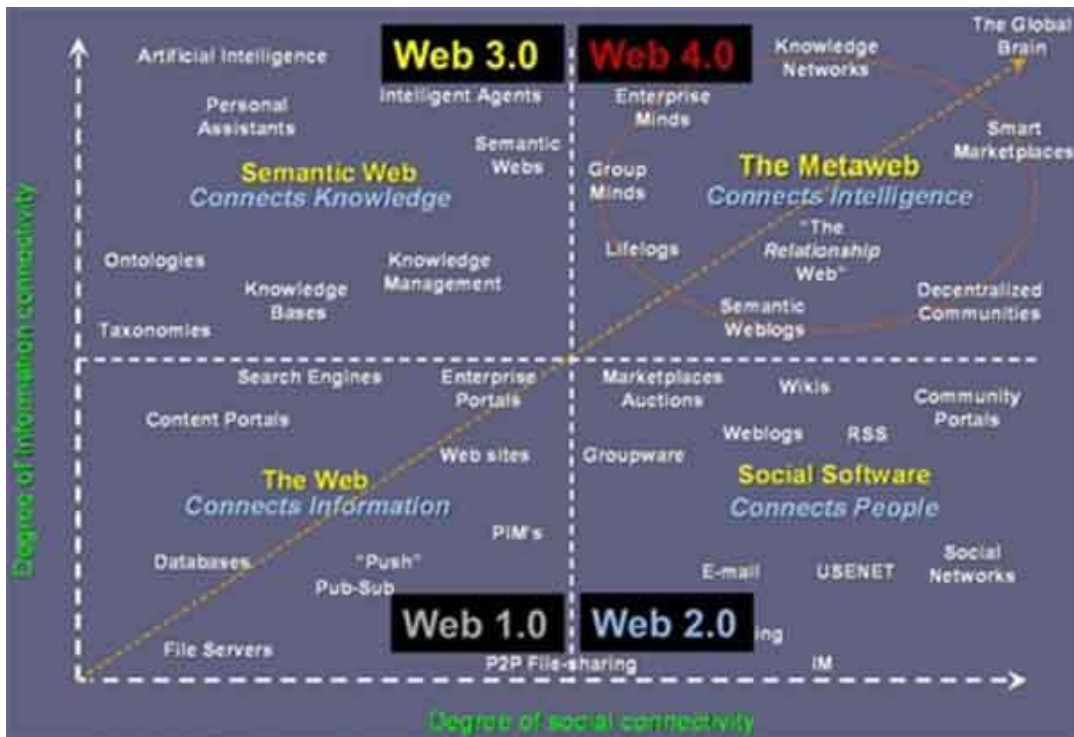
Web 1.0 / 2.0 / 3.0 Summary

Crawl	Walk	Run
Web 1.0	Web 2.0	Web 3.0
Mostly Read-Only	Wildly Read-Write	Portable & Personal
Company Focus	Community Focus	Individual Focus
Home Pages	Blogs / Wikis	Lifestreams / Waves
Owning Content	Sharing Content	Consolidating Content
Web Forms	Web Applications	Smart Applications
Directories	Tagging	User Behavior
Page Views	Cost Per Click	User Engagement
Banner Advertising	Interactive Advertising	Behavioral Advertising
Britannica Online	Wikipedia	The Semantic Web
HTML/ Portals	XML/ RSS	RDF/ RDFS/ OWL

Εικόνα 2. 25: Σύνοψη Web 1.0 / Web 2.0 / Web 3.0 [68]

2.3.8 - Web 4.0

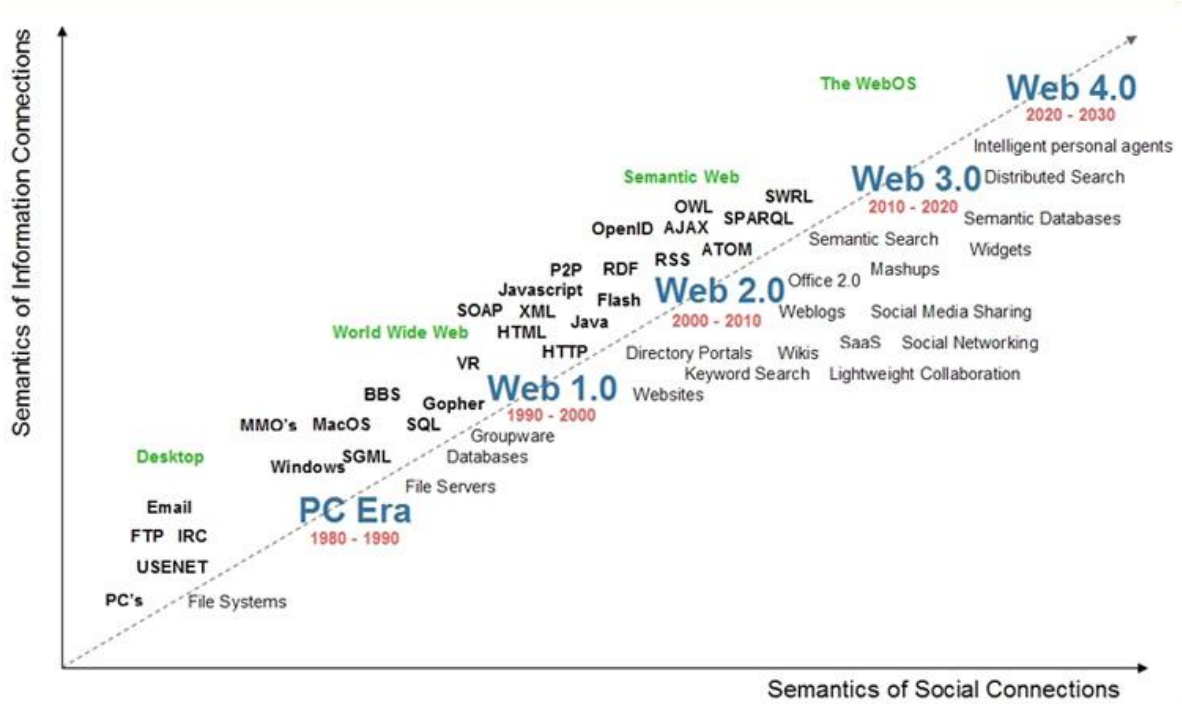
Η επόμενη έκδοση του Παγκόσμιου Ιστού, ο Web 4.0 είναι ακόμη στα πρώτα στάδια ανάπτυξής του. Ο Web 4.0 θα αποτελεί ένα δίκτυο το οποίο θα επικοινωνεί με τους ανθρώπους όπως επικοινωνούμε μεταξύ μας. Θα αποκαλείται «**Συμβιωτικός Ιστός**».



Εικόνα 2. 26: Web 4.0

Όλες οι εφαρμογές θα αναγνωρίζουν τους χρήστες όταν αυτοί στέκονται μπροστά τους και θα επικοινωνούν μαζί τους ως ένας συγκατοικός τους ή φίλος τους. Όλες οι συσκευές θα μπορούν να επικοινωνούν μεταξύ τους και να προβαίνουν σε ανάλογες ενέργειες, όπως για παράδειγμα το ψυγείο να ενημερώνει ότι το γάλα έχει λήξει.

Αν και τα παραπάνω φαίνονται επιστημονική φαντασία που βλέπουμε μόνο στις ταινίες, η αλήθεια είναι ότι είμαστε πιο κοντά στην υλοποίησή τους από όσο νομίζουμε. [42] [44] [43]



Εικόνα 2. 27: Χρονικό διάγραμμα εκδόσεων του ιστού [69]

2.4 - Τεχνολογίες Διαδικτύου

2.4.1 - Τι είναι οι Τεχνολογίες Διαδικτύου και πως χρησιμοποιούνται

Στα προηγούμενα κεφάλαια είδαμε τι είναι το internet, τι είναι ο ιστός, πως δημιουργήθηκε και πως λειτουργεί. Για να λειτουργήσει ο ιστός όπως τον αντιλαμβανόμαστε χρειάζονται διάφορες τεχνολογίες οι οποίες θα ανακτήσουν το επιθυμητό περιεχόμενο, θα το εμφανίσουν με την επιθυμητή μορφή και θα επιτελέσουν ορισμένες λειτουργίες / ενέργειες.

Όλα τα παραπάνω γίνονται δυνατά με τις τεχνολογίες διαδικτύου. Η κάθε μία τεχνολογία χρησιμοποιείται για συγκεκριμένο σκοπό και ρόλο. Αυτές οι τεχνολογίες είναι οι HTML, CSS, JavaScript, XML, PHP, Ajax.

2.4.2 - HTML



Η HTML είναι η κύρια **markup** γλώσσα για την δημιουργία ιστοσελίδων. Μία markup γλώσσα είναι ένα σύστημα για την εισαγωγή μεταδεδομένων σε ένα αρχείο και τα οποία θα διαχωρίζονται από το απλό κείμενο που μπορεί να περιέχει αυτό το αρχείο.

Η HTML είναι γραμμένη χρησιμοποιώντας HTML στοιχεία, δηλαδή συγκεκριμένα tags, τα οποία περικλείονται σε αγκύλες (<p></p>). Συνήθως τα στοιχεία εμφανίζονται σε ζευγάρια, όπου το πρώτο μέρος ονομάζεται **start tag**, ενώ το δεύτερο ονομάζεται **end tag**. Ανάμεσα στα στοιχεία μπορεί να τοποθετηθεί κείμενο, σχόλια ή άλλα στοιχεία.

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML
2 <html>
3   <head>
4     <title>Example</title>
5     <link href="screen.css" rel="sty
6   </head>
7   <body>
8     <h1>
9       <a href="/">Header</a>
10    </h1>
11    <ul id="nav">
12      <li>
13        <a href="one/">One</a>
14      </li>
15      <li>
16        <a href="two/">Two</a>
17      </li>
```

Εικόνα 2. 28: HTML παράδειγμα κώδικα

Ο browser διαβάζει τα html αρχεία και τα μετατρέπει σε μορφή κατανοητή από τους χρήστες. Τα στοιχεία –tags- αυτά της html, δεν εμφανίζονται στην οθόνη αλλά χρησιμοποιούνται από τον **html parser** για να διαμορφώσει ανάλογα το αρχείο και την δομή του.

Τα στοιχεία της html αποτελούν τους θεμέλιους λίθους κάθε ιστοσελίδας. Μπορούν να ενσωματωθούν εικόνες, αρχεία και να κατασκευασθούν φόρμες. Με τα html elements παράγεται ένα δομημένο κείμενο, όπου τα στοιχεία αποκτούν κάποιο νόημα.

Η HTML δημιουργήθηκε από τον Tim Berners Lee, στα τέλη του 1990. Η πρώτη επίσημη έκδοση της HTML ονομάστηκε “**HTML Tags**”. Περιέγραφε 18 στοιχεία που αποτέλεσαν την αρχική - απλοϊκή έκδοση της HTML. [45]

HTML Έκδοση	Ημερομηνία
HTML 2.0	Νοέμβριος, 1995
HTML 3.2	Ιανουάριος, 1997
HTML 4.0	Δεκέμβριος, 1997
HTML 4.0 (μικρές αλλαγές)	Απρίλιος, 1998
HTML 4.01 (recommendation)	Δεκέμβριος, 1999
HTML 4.01 (standard)	Μάιος, 2000

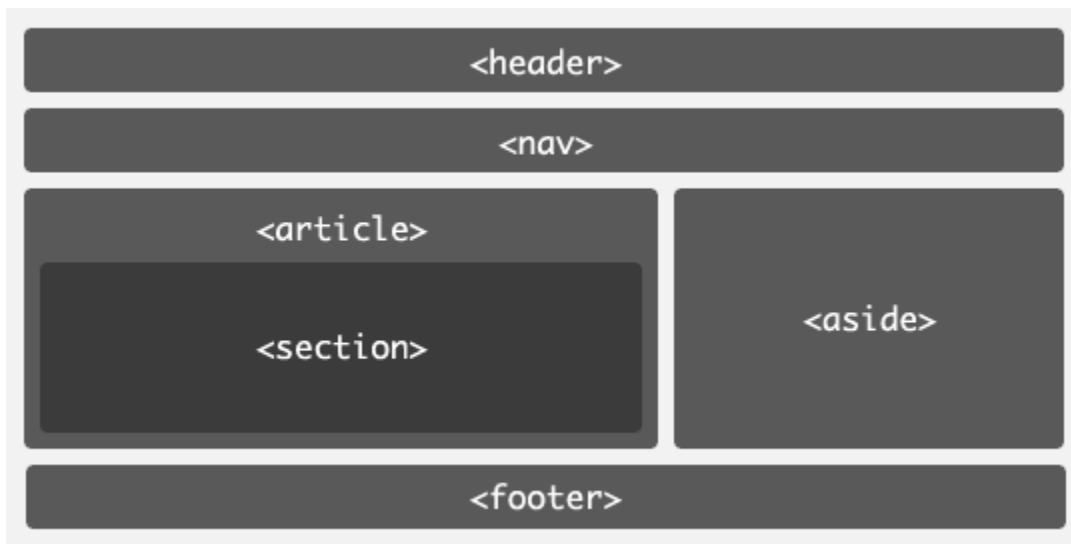
2.4.3 - HTML5



Η τελευταία έκδοση της HTML είναι η HTML5, η οποία από τον Δεκέμβριο του 2012 έχει προταθεί από τον W3C για να καθιερωθεί ως standard. Στόχος των δημιουργών της ήταν η καλύτερη υποστήριξη multimedia περιεχομένου, ενώ θα είναι παράλληλα ακόμη πιο ευανάγνωστη τόσο από τους ανθρώπους, όσο και τις μηχανές (browsers, μηχανές αναζήτησης). Η HTML5 θα αντικαταστήσει όχι μόνο την HTML4, αλλά την XHTML1 καθώς και την Dom Level 2 HTML.

Με την HTML5 γίνεται προσπάθεια να οριστεί μία κοινή markup γλώσσα, όπου θα υποστηρίζει τόσο την HTML όσο και την XHTML. Βελτιώνει και επεκτείνει τα elements, και εισάγει APIs (**Application Programming Interfaces**) για σύνθετες εφαρμογές. Πολλά από τα χαρακτηριστικά της HTML5 κατασκευάστηκαν με την προϋπόθεση να μπορούν να τρέχουν σε συσκευές όπως τα smartphones και τα tablets.

Έχουν εισαχθεί αρκετά νέα χαρακτηριστικά, όπως τα <video>, <audio> και <canvas> με τα οποία είναι ευκολότερη η εισαγωγή και η δημιουργία multimedia περιεχομένων. Τα <section>, <article>, <header>, <nav>, <aside>, <footer> προσθέτουν σημασιολογική υπόσταση στα στοιχεία.



Εικόνα 2. 29: HTML5 δομικά στοιχεία [70]

Για την νέα αυτή έκδοση η ομάδα **Web Hypertext Application Technology Working Group** (WHATWG) ξεκίνησε τις εργασίες το 2004, παρόλο που η προηγούμενη έκδοση δεν είχε ενημερωθεί από το 2000. Αν και η HTML5 ήταν γνωστή για αρκετά χρόνια στους προγραμματιστές έγινε επίκεντρο του ενδιαφέροντος μόνο όταν ο **Steve Jobs** εξέδωσε μία ανακοίνωση τον Απρίλιο του 2010 με τίτλο “**Thoughts on Flash**”, στην οποία ανέφερε πως το Flash δεν θα ήταν πλέον απαραίτητο καθώς ένα νέο web standard δημιουργούνταν για τις κινητές συσκευές, το οποίο θα κυριαρχούσε. Τον Νοέμβριο του 2011 η Adobe ανακοίνωσε πως δεν θα συνέχιζε την ανάπτυξη του Flash για τις κινητές συσκευές και θα επικεντρώνονταν σε εργαλεία τα οποία θα χρησιμοποιούσαν και αξιοποιούσαν την HTML5.

Η πρώτη έκδοση της HTML5 (**1st Draft**) κυκλοφόρησε στις 22 Ιανουαρίου του 2008. Ήδη κάποιοι browsers άρχισαν να ενσωματώνουν κάποια από τα χαρακτηριστικά της. Τον Μάιο του 2011 το project αναβαθμίστηκε σε **Last Call**, δηλαδή καλούσαν όλη την κοινότητα να επιβεβαιώσουν τα τεχνικά χαρακτηριστικά της HTML5. Στον W3C θέλουν να καταλήξουν στις τελικές προδιαγραφές της μέχρι το 2014, όπου και έχει οριστεί το χρονικό όριο της **Recommendation** φάσης του project. [46]

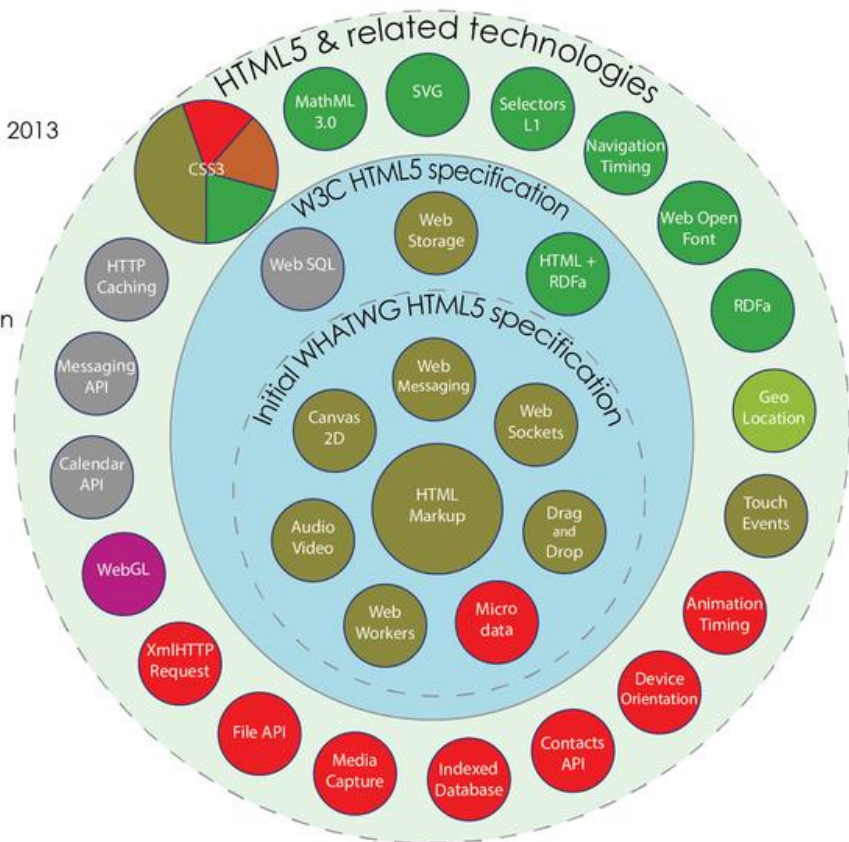
Τα χρονικά όρια των HTML5.0, HTML5.1 και HTML5.2 φαίνονται παρακάτω.

	2012	2013	2014	2015	2016
HTML 5.0	Candidate	Call For Review	Recommendation		
HTML 5.1	1 st Draft		Last Call	Candidate	Recommendation
HTML 5.2				1 st Draft	

HTML5

Taxonomy & Status on January 20, 2013

- W3C Recommendation
- Proposed Recommendation
- Candidate Recommendation
- Last Call
- Working Draft
- Non-W3C Specifications
- Deprecated



by Sergey Mavrody BY · SA

Εικόνα 2. 30: HTML5 κατάσταση νέων στοιχείων [46]

2.4.4 - CSS



Το CSS (**Cascading Style Sheet**) είναι μία style sheet language. Χρησιμοποιείται για να περιγράψει την εμφάνιση και τη μορφή των αντικειμένων που ορίζονται από μία markup γλώσσα όπως είναι η HTML. Το CSS δημιουργήθηκε για να υπάρχει διαχωρισμός της δομής του κειμένου από την εμφάνισή του.

Με τον τρόπο αυτό βελτιώνεται η επεκτασιμότητα του κώδικα, η επαναχρησιμοποίηση του και η προσιτότά του. Χρησιμοποιώντας το CSS μπορούμε να κάνουμε το ίδιο αντικείμενο να εμφανίζεται με διαφορετικό τρόπο σε διαφορετικές πηγές, όπως για παράδειγμα σε desktops, σε smartphones και σε tablets. Χρησιμοποιώντας κάποιους κανόνες είναι δυνατόν να επιλεγεί το σωστό style sheet (**cascade**).

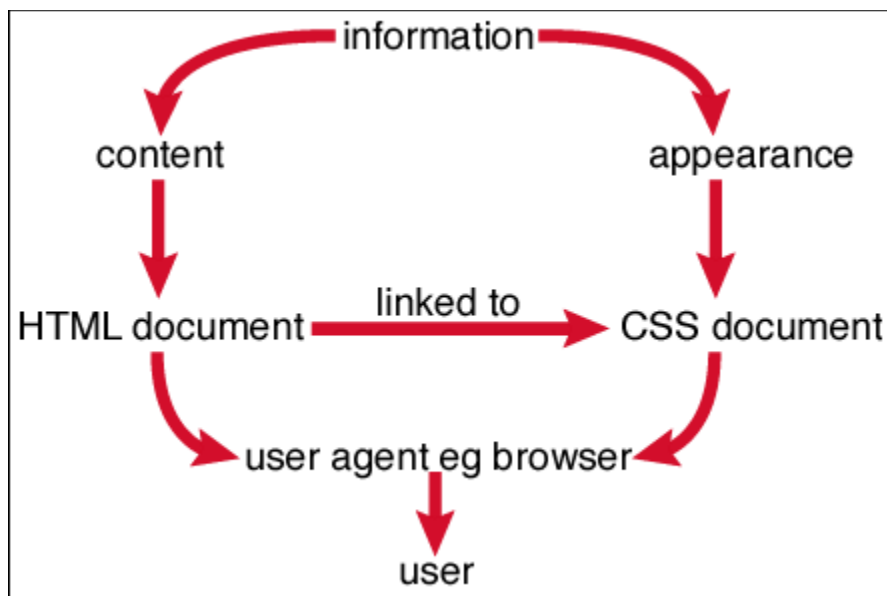
Πριν την εισαγωγή του CSS όλα τα χαρακτηριστικά της μορφολογίας των αντικειμένων των HTML αρχείων περιλαμβάνονταν μαζί με τη δήλωση αυτών των αντικειμένων. Πλέον αυτά τα χαρακτηριστικά μεταφέρονται σε άλλο αρχείο, κάνοντας τη δουλειά των προγραμματιστών πιο εύκολη.



Εικόνα 2. 31: CSS παράδειγμα

Τα style sheets υπάρχουν από την αρχή του Standard Generalized Markup Language το 1980. Το CSS δημιουργήθηκε μετέπειτα για να δώσει μορφή στα HTML κείμενα. Καθώς υπήρχαν διαφορές στις λειτουργίες των browsers, η δουλειά του προγραμματιστή είχε γίνει πιο δύσκολη και περίπλοκη. Έτσι αποφασίστηκε να γίνει διαχωρισμός της δομής του κειμένου από την εμφάνισή του. Προτάθηκαν λοιπόν στον W3C εννιά πρότυπα. Ένα από αυτά ήταν το CSS, το οποίο προτάθηκε από τον Hakon Wium Lie τον Οκτώβριο του 1994.

Η πρόταση του Lie παρουσιάστηκε το 1994 και το 1995 στο συνέδριο “Mosaic and the Web”. Ήδη από εκείνον τον καιρό το W3C είχε εκδηλώσει ενδιαφέρον στην ανάπτυξη του CSS. Μέχρι το τέλος του 1996 το CSS είχε γίνει επίσημο και το CSS Level 1 Recommendation εκδόθηκε τον Δεκέμβριο εκείνου του χρόνου.



Εικόνα 2. 32: Λειτουργία του CSS

Η χρήση του CSS παρουσιάζει πολλά πλεονεκτήματα. Το κυριότερο από αυτά είναι ο **διαχωρισμός του περιεχομένου από την παρουσίαση**. Ως απόρροια αυτού, εάν θελήσουμε να κάνουμε αλλαγές στην εμφάνιση ενός αρχείου τότε θα γίνουν αλλαγές μόνο στο CSS αρχείο. Το HTML αρχείο μένει ανέπαφο, οπότε βλέπουμε ότι η συντήρηση και η επέκταση των αρχείων έχει γίνει ευκολότερη. Ένα css αρχείο (**external stylesheet**) παραμένει στην **cache** του browser οπότε μπορεί να χρησιμοποιηθεί για διάφορα αρχεία χωρίς να επιβαρύνεται το **bandwidth**. [47]

2.4.5 - JavaScript



Η JavaScript είναι μία γλώσσα προγραμματισμού. Επιτρέπει διάφορες client - server εφαρμογές να αλληλεπιδρούν με το χρήστη, χειρίζεται τον browser, επιτρέπει την ασύγχρονη επικοινωνία και μπορεί να μεταβάλλει το περιεχόμενο και την μορφή του. Έχει γίνει πολύ δημοφιλής στον σχεδιασμό παιχνιδιών και desktop εφαρμογών.

```
<!-- BEGIN google analytics -->
<script type="text/javascript">
  var gaJsHost = (("https:" == document.location.protocol) ?
  "https://ssl." : "http://www.");
  document.write(unescape("%3Cscript src='" + gaJsHost +
  "google-analytics.com/ga.js' type='text/javascript'%3E%3C
  /script%3E"));
</script>
<script type="text/javascript">
  var pageTracker = _gat._getTracker("UA-30775-6");
  pageTracker._setDomainName("twitter.com");
  url = '/statuses/rebeccakelley/1207938518';
  pageTracker._setVar('Logged In');
  pageTracker._setVar('lang: en');
  pageTracker._trackPageview(url);
</script>

<!-- END google analytics -->
```

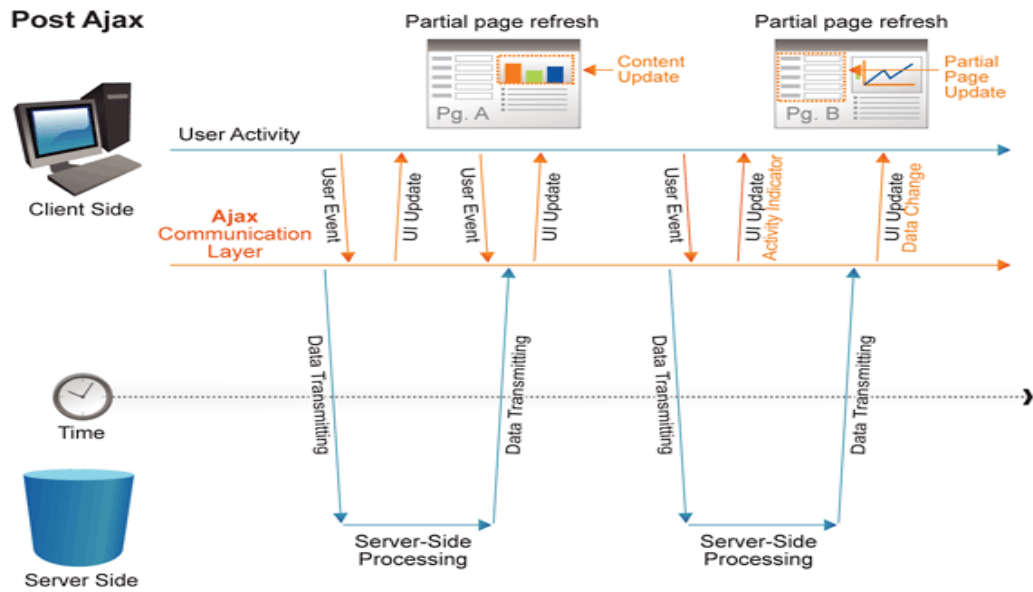
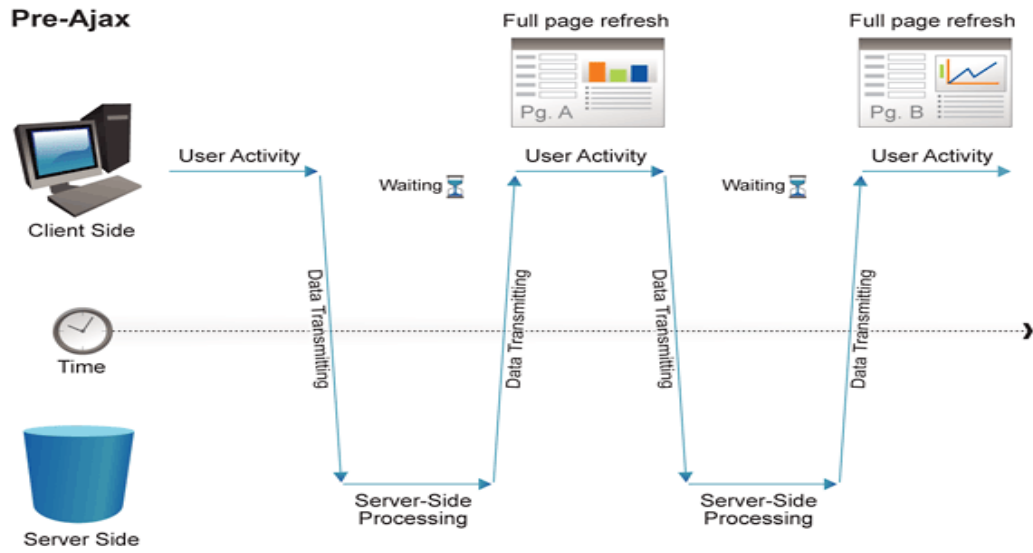
Εικόνα 2. 33: JavaScript παράδειγμα

Η JavaScript δημιουργήθηκε από τον Brendan Eich. Η ομάδα του **Netscape** ήθελε να ενσωματώσει μία ελαφριά γλώσσα που να μην βρίσκει απήχηση μόνο στο προγραμματιστικό κοινό, όπως η Microsoft με την Visual Basic. Έτσι, πήραν το project του Brendan Eich (τότε ήταν υπό το όνομα **LiveScript**) και το ενσωμάτωσαν στην έκδοση Netscape Navigator 2.0 τον Σεπτέμβριο του 1995 όπου και μετονομάστηκε σε **JavaScript**.

Καθώς η JavaScript βρήκε γρήγορα μεγάλη απήχηση, η Microsoft αποφάσισε να εισάγει υποστήριξη της γλώσσας στον Internet Explorer 3, τον Αύγουστο του 1996. Τον Νοέμβριο του 1996, η ομάδα του Netscape αποφάσισε ότι κατέθεσε την JavaScript στο **Ecma International** έτσι ώστε να θεωρηθεί web standard, κάτι που έγινε τον Ιούνιο του 1998.

Η πιο κοινή χρήση της JavaScript είναι η συγγραφή συναρτήσεων οι οποίες ενσωματώνονται ή εισάγονται στα HTML αρχεία και αλληλεπιδρούν με το DOM (Document Object Model) της σελίδας. Οι λειτουργίες που επιτελεί είναι να **εισάγει νέα αντικείμενα** στη σελίδα, ή να υποβάλλει στοιχεία ασύγχρονα με την **AJAX** τεχνική, **animation** σε διάφορα στοιχεία της σελίδας, **interactive** περιεχόμενο όπως παιχνίδια, η **επικύρωση εισόδου** όπως για παράδειγμα το password σε ένα σύστημα και η αποθήκευση πληροφοριών για τις προτιμήσεις και τις κινήσεις ενός χρήστη σε μια σελίδα. [48]

Ο AJAX (**Asynchronous JavaScript and XML**) είναι μία client - side τεχνική κατά την οποία οι εφαρμογές μπορούν να επικοινωνούν ασύγχρονα με έναν server, δηλαδή να στέλνουν και να αποστέλλουν δεδομένα στο παρασκήνιο δίχως να επηρεάζεται η συμπεριφορά της υπάρχουσας σελίδας. [49]



Εικόνα 2. 34: AJAX [71]

2.4.6 - PHP



Η PHP (**Hypertext Preprocessor**) είναι μία γενικού σκοπού server - side γλώσσα προγραμματισμού, σχεδιασμένη για web εφαρμογές. Δημιουργήθηκε από τον **Rasmus Lerdorf** το 1995. Ο PHP κώδικας εκτελείται στον web server, ο οποίος παράγει τα αποτελέσματα και επιστρέφονται στην HTML σελίδα. Οι εντολές μπορούν είτε να κληθούν από ένα εξωτερικό αρχείο, είτε να ενσωματωθούν κατευθείαν μέσα στο HTML αρχείο.

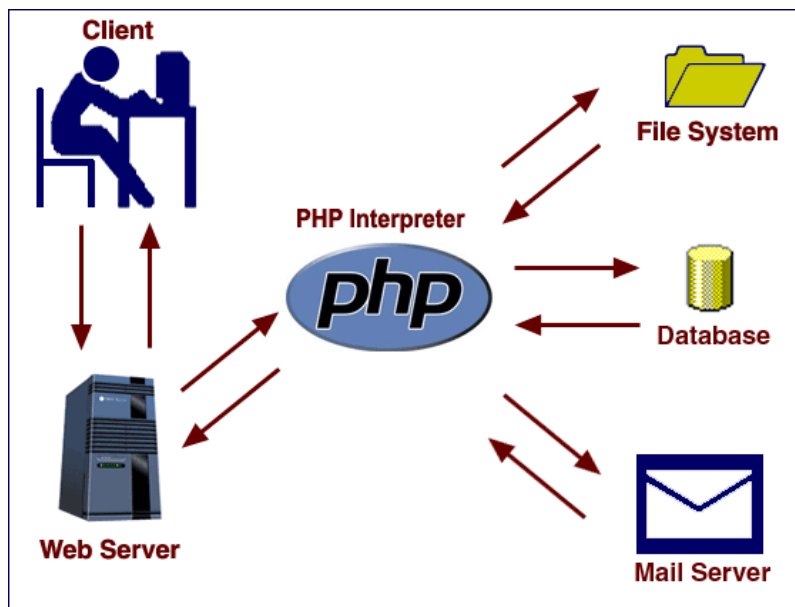
Η δημιουργία της PHP ξεκίνησε το 1994 όταν ο **Lerdorf** έγραψε ένα σύνολο **Perl scripts** για να συντηρήσει την προσωπική του σελίδα. Τα scripts αυτά εκτελούσαν ενέργειες όπως να εμφανίζουν το βιογραφικό του και να καταγράφουν την κίνηση στην ιστοσελίδα του. Στη συνέχεια ξαναέγραψε αυτά τα scripts με την C για λόγους απόδοσης και τα επέκτεινε έτσι ώστε να μπορεί να εργάζεται με φόρμες και να επικοινωνεί με βάσεις δεδομένων. Ονόμασε την υλοποίησή του "**Personal Home Page/Forms Interpreter**" ή PHP/FI. Αυτή η υλοποίηση μπορούσε να χρησιμοποιηθεί για την κατασκευή απλών δυναμικών web εφαρμογών.

Οι **Zeev Suraski** και **Andi Gutmans** ξαναέγραψαν τον parser το 1997, όπου το μετονόμασαν σε Hypertext Preprocessor και δημιουργήθηκε η βάση για την PHP3. Έπειτα, τον Ιούνιο του 1998 ξεκίνησε η δοκιμή της τρίτης αυτής έκδοσης. Η PHP4 κυκλοφόρησε τον Μάιο του 2000, υποστηριζόμενη από την Zend Engine I. Τον Ιούλιο του 2004 κυκλοφόρησε η PHP5, υποστηριζόμενη από την Zend Engine II.

Η PHP δέχεται την είσοδο από αρχεία ή από stream χαρακτήρων μαζί με τις PHP εντολές και παράγει την έξοδο που συνήθως είναι HTML αντικείμενα, JSON ή XML στοιχεία. Πλέον η PHP εστιάζει σε server - side scripting όπως το ASP.NET ή το JavaServer Pages της Sun Microsystems. Επιπλέον έχουν δημιουργηθεί διάφορα MVC (**Model View Control**) frameworks όπως το CakePHP, Symfony, Yii, Zend και CodeIgniter τα οποία διαχωρίζουν την εμφάνιση της πληροφορίας από την αλληλεπίδραση του χρήστη με την εφαρμογή. Στο MVC, το **model** αντιστοιχεί στις συναρτήσεις και τα δεδομένα, το **view** στην εμφάνιση των δεδομένων και το **control** στην είσοδο των δεδομένων.

Αντίθετα από μια συνηθισμένη HTML σελίδα, η σελίδα PHP δεν στέλνεται άμεσα σε έναν πελάτη (client), αντ' αυτού πρώτα αναλύεται και μετά αποστέλλεται το παραγόμενο αποτέλεσμα. Τα στοιχεία HTML στον πηγαίο κώδικα μένουν ως έχουν, αλλά ο PHP κώδικας ερμηνεύεται και εκτελείται. Ο κώδικας PHP μπορεί να θέσει ερωτήματα σε βάσεις δεδομένων, να δημιουργήσει εικόνες, να διαβάσει και να γράψει αρχεία ή να συνδεθεί με απομακρυσμένους υπολογιστές.

Όταν ένας χρήστης ανοίγει μια σελίδα, ο server επεξεργάζεται τις εντολές PHP και στην συνέχεια στέλνει τα αποτελέσματα στον browser του χρήστη. Εκτός από την διαχείριση των περιεχομένων της σελίδας, η PHP μπορεί επίσης να στείλει HTTP κεφαλίδες (headers). Μπορεί να στείλει cookies ή να διαχειριστεί την ταυτοποίηση χρηστών. [50]



Εικόνα 2. 35: PHP λειτουργία [72]

3ο Κεφάλαιο - Web Browsers

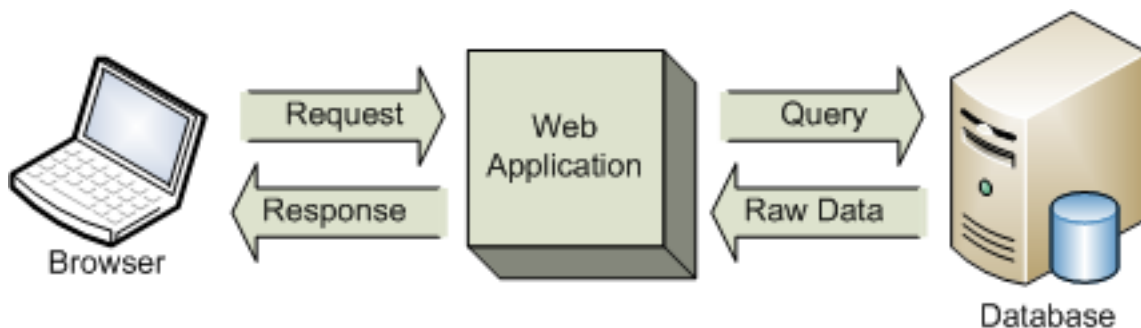
3.1 - Εισαγωγή

Ο web browser είναι μία εφαρμογή η οποία χρησιμοποιείται για την εύρεση, την ανάκτηση και την παρουσίαση πληροφοριών από τον ιστό. Οι πηγές αυτών των πληροφοριών χαρακτηρίζονται από το URL (Uniform Resource Identifier) και μπορεί να είναι μία ιστοσελίδα, μία εικόνα, ένα βίντεο, κείμενο ή άλλο είδος περιεχομένου.

Οι σημερινοί browsers είναι συστήματα τα οποία μπορούν να παρουσιάσουν HTML σελίδες, εφαρμογές, JavaScript, AJAX και άλλα διάφορα περιεχόμενα. Πολλοί από αυτούς προσφέρουν διάφορα plug ins τα οποία αυξάνουν τις δυνατότητές τους. [9][11]

3.2 - Λειτουργία των Browsers

Ο browser είναι ένας client ο οποίος επικοινωνεί με τον server στον ιστό και ζητάει πληροφορίες. Στη συνέχεια ο server στέλνει τα δεδομένα πίσω στον client / browser και αυτός εμφανίζει τα αποτελέσματα.

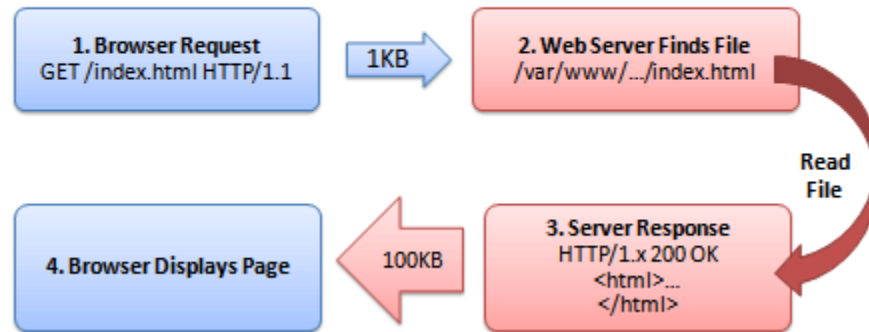


Εικόνα 3. 1: Λειτουργία του browser

Για τη λειτουργία του browser πρέπει να του δοθεί ως όρισμα μία τοποθεσία έτσι ώστε να την επισκεφθεί (URL). Το πρόθεμα στα URLs (URI), δηλώνει τον τρόπο που θα ερμηνευτεί το URL. URIs που ξεκινούν με **http**: δηλώνουν ότι τα περιεχόμενα θα μεταφερθούν χρησιμοποιώντας το Hypertext Transfer Protocol (HTTP). Υπάρχουν και άλλου είδους προθέματα όπως **https** (Hypertext Transfer Protocol Secure), **ftp** (File Transfer Protocol) και **file** (τοπικά αρχεία).

Όταν βρεθεί η πηγή, ο browser παίρνει το περιεχόμενό της και το εμφανίζει στο χρήστη. Για την εμφάνιση των πληροφοριών σε μορφή κατανοητή από τους ανθρώπους χρησιμοποιούνται η HTML και το CSS.

HTTP Request and Response



Εικόνα 3. 2: : Λειτουργία του browser [73]

3.3 - Στοιχεία ενός Browser

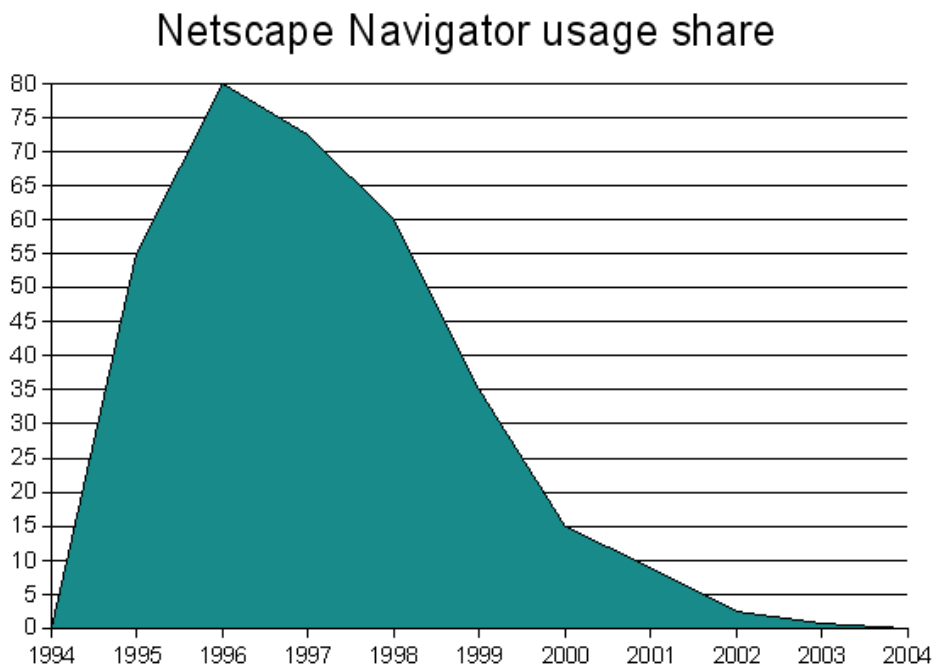
Όλοι οι browsers αποτελούνται από κάποια κοινά στοιχεία. [10]

Αυτά είναι τα εξής:

- **Address Bar:** το πεδίο όπου συμπληρώνουμε τις διευθύνσεις (URLs) που θέλουμε να επισκεφθούμε.
- **Title Bar:** ο τίτλος της ιστοσελίδας που επισκεπτόμαστε.
- **Navigation Toolbar:** εργαλειοθήκη η οποία βοηθάει στην πλοήγησή μας, όπως τα κουμπιά home, back, refresh.
- **Workspace:** το πεδίο που εμφανίζονται οι πληροφορίες από τα urls.
- **Scroll bars:** γραμμές κύλισης με τις οποίες μπορούμε να βλέπουμε ολόκληρο το περιεχόμενο των ιστοσελίδων.

3.4 - Ιστορία των Web Browsers

Ο πρώτος web browser εφευρέθηκε το 1990 από τον Tim Berners-Lee. Ονομαζόταν **WorldWideWeb** και στη συνέχεια μετονομάστηκε σε **Nexus**. Το 1993 το λογισμικό των browsers αναπτύχθηκε επιπλέον από τον Marc Andreessen με την ονομασία **Mosaic** (αργότερα ονομάστηκε σε **Netscape**). Ο Netscape έγινε ο πρώτος δημοφιλής browser που έκανε τον ιστό πιο προσβάσιμο στο μέσο χρήστη. Μέσα σε ένα χρόνο κατείχε το 90% της συνολικής χρήσης.



Εικόνα 3. 3: Μεριδίο χρήσης Netscape [74]

Η Microsoft απάντησε στον Netscape το 1995 με τον **Internet Explorer**, ο οποίος είχε στοιχεία από το λογισμικό του Netscape. Σε συνδυασμό με τα Windows, ο IE μέχρι το 2002 κατείχε το 95% της συνολικής χρήσης. Το 1996 ο Opera έκανε το ντεμπούτο του. Παρόλα αυτά δεν κατάφερε ποτέ να πάρει μεγάλο μερίδιο αγοράς.

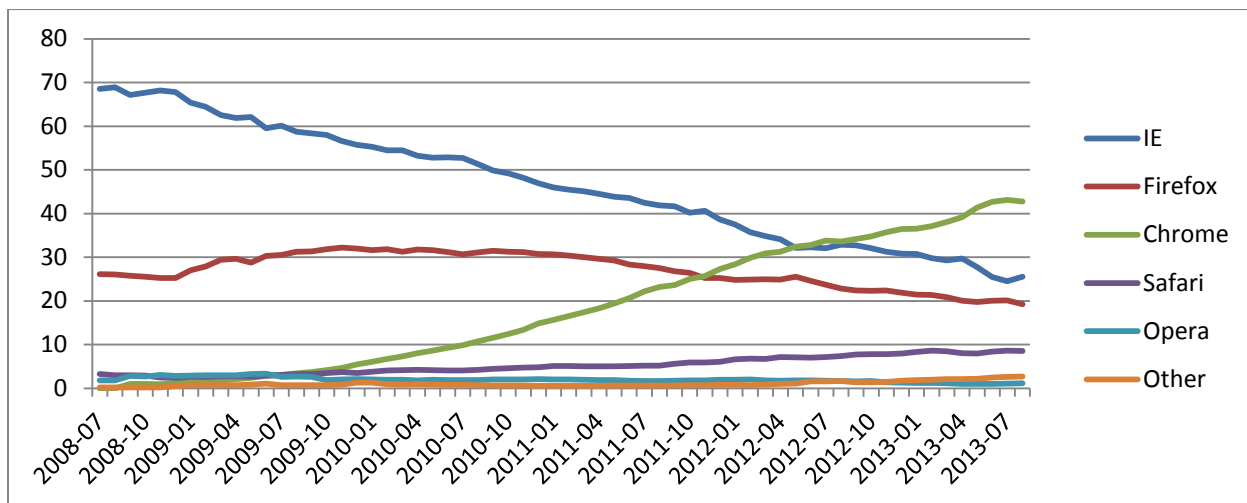
Το 1998 ο Netscape δημιούργησε την Mozilla Foundation σε μία προσπάθεια να δημιουργηθεί ένας ανταγωνιστικός browser βασισμένος στο open source λογισμικό. Ο browser που δημιουργήθηκε εξελίχθηκε στον **Firefox**, ο οποίος τα τελευταία χρόνια είναι ένας από τους κυρίαρχους browsers. Από τότε έχει ξεκινήσει ένας αγώνας ανάμεσα στους διάφορους browsers, για να κατακτήσουν την αγορά.

Άλλοι browsers που έκαναν την εμφάνισή τους είναι ο **Safari** της Apple τον Ιανουάριο του 2003. Κατέχει πλέον το κυρίαρχο μερίδιο στα συστήματα της Apple και συνολικά περίπου

το 7% της παγκόσμιας χρήσης. Ένα από τα νεότερα μέλη της αγοράς είναι ο **Chrome**, ο οποίος έκανε την εμφάνισή του τον Σεπτέμβριο του 2008. Το μερίδιο της χρήσης που κατέχει αυξάνεται σημαντικά τα τελευταία χρόνια.

Οι παραπάνω web browser είναι οι κυρίαρχοι στον τομέα και συνολικά κατέχουν περισσότερο από το 90% της συνολικής χρήσης τόσο σε σταθερούς υπολογιστές όσο και σε φορητούς. Από τα διάφορα στοιχεία που προκύπτουν ανά τα χρόνια ο IE έχει χάσει σημαντικό έδαφος. Ο Firefox αποτελεί σταθερή αξία, ενώ έκπληξη αποτελεί ο Chrome τα μερίδια αγοράς του οποίου εκτοξεύονται τα τελευταία χρόνια. [29]

Στην παρακάτω εικόνα φαίνονται τα ποσοστά χρήσης στους παραπάνω browsers, από τον Ιούλιο του 2008, μέχρι τον Ιούλιο του 2013.



Εικόνα 3. 4: Μερίδια χρήσης 07/2008 - 07/2013 [27]

Στο παραπάνω συμπέρασμα μπορούμε να καταλήξουμε κοιτώντας και τις παρακάτω εικόνες, οι οποίες δείχνουν το μερίδιο των browsers ανά χώρα από το 2008 μέχρι το 2013. Φαίνεται ξεκάθαρα πως η κυριαρχία του Internet Explorer το 2008, έχει καταρρεύσει και την θέση του πλέον κατέχει ο Chrome. [27][12]

StatCounter Global Stats
Top Browsers Per Country from Aug to Sept 2008



Εικόνα 3. 5: : Μεριδία χρήσης 2008 [27]

StatCounter Global Stats
Top Browsers Per Country from Aug to Sept 2009



Εικόνα 3. 6: Μεριδία χρήσης 2009 [27]

StatCounter Global Stats
Top Browsers Per Country from Aug to Sept 2010



Εικόνα 3. 7: Μέρδια χρήσης 2010 [27]

StatCounter Global Stats
Top Browsers Per Country from Aug to Sept 2011



Εικόνα 3. 8: Μέρδια χρήσης 2011 [27]

StatCounter Global Stats
Top Browsers Per Country from Aug to Sept 2012



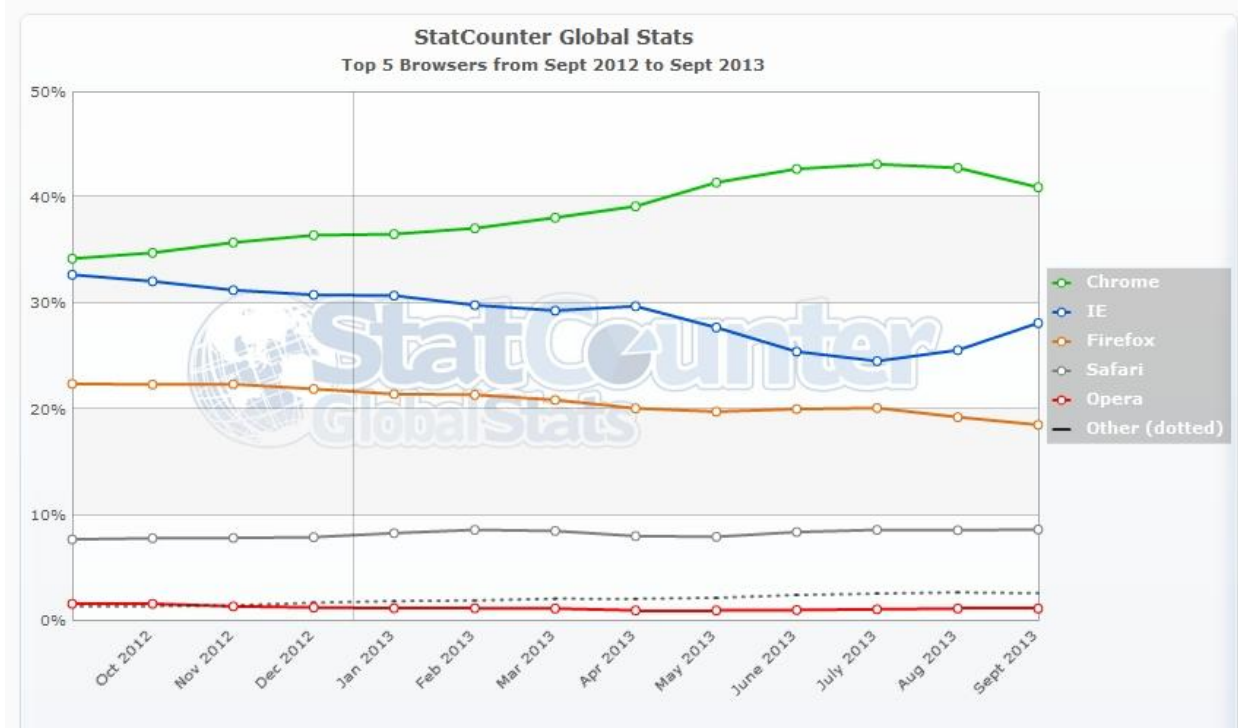
Εικόνα 3. 9: Μερίδια χρήσης 2012 [27]

StatCounter Global Stats
Top Browsers Per Country, Aug 2013



Εικόνα 3. 10: Μερίδια χρήσης 2013 [27]

Στο τελευταίο διάγραμμα φαίνονται τα ποσοστά χρήσης των browsers μέσα στο τελευταίο έτος, Σεπτέμβριος 2012 - Σεπτέμβριος 2013.



Εικόνα 3. 11: Μερίδια χρήσης τελευταίου έτους [27]

3.5 - Firefox



Ο Mozilla δημιουργήθηκε το 1998, μετά την δημοσίευση του κώδικα του Netscape. Σκοπός αυτής της κίνησης ήταν να χρησιμοποιηθούν οι γνώσεις και οι ιδέες χιλιάδων προγραμματιστών ανά τον κόσμο με σκοπό να δημιουργηθεί ένας καινοτόμος browser που θα κυριαρχήσει. Μέσα στον πρώτο χρόνο τα νέα μέλη είχαν ήδη συμβάλλει στο έργο προσθέτοντας νέες δυνατότητες.

Δημιουργώντας μία ανοιχτή κοινότητα ο Mozilla έγινε μεγαλύτερος από κάθε άλλη εταιρία. Αντί τα μέλη της οργάνωσης να εργάζονται για την δημιουργία της επόμενης έκδοσης του Netscape, άρχισαν να δημιουργούν διάφορους browsers, εργαλεία και διάφορα άλλα projects. Το κοινό σημείο που τους παρακινούσε ήταν η δυνατότητα που θα είχε ο κάθε ένας χρήστης του ιστού να επιλέξει τα εργαλεία που ήθελε για την περιήγησή του στο διαδίκτυο.

Μετά από αρκετά χρόνια εκδόθηκε ο **Mozilla 1.0**, το 2002. Η έκδοση περιείχε πολλά νέα χαρακτηριστικά όπως email client, αλλά παρόλα αυτά δεν βρήκε μεγάλη υποστήριξη από το ευρύ κοινό. Μέχρι το 2002, περισσότεροι από 90% των χρηστών πλοηγούνταν στο διαδίκτυο με τον Internet Explorer. Την ίδια χρονιά εκδόθηκε η πρώτη έκδοση του **Phoenix**, που στη συνέχεια θα μετονομάζονταν σε **Firefox**. Στόχος αυτής της έκδοσης ήταν να παρέχει στους χρήστες την καλύτερη δυνατή εμπειρία πλοήγησης.



Εικόνα 3. 12: Εικονίδια του Phoenix / Firefox

Το 2003 το project Mozilla δημιούργησε το ίδρυμα Mozilla (Mozilla Foundation), μία ανεξάρτητη, μη κερδοσκοπική οργάνωση που υποστηρίζονταν από δωρεές και επιχειρήσεις. Το νέο αυτό ίδρυμα ανέλαβε την συνέχεια του προγράμματος και δημιουργήθηκαν νέες εκδόσεις τόσο του Firefox, όσο και του Thunderbird (email client).

Η έκδοση Firefox 1.0 εκδόθηκε το 2003. Από τότε μεσολάβησαν πολλές νέες εκδόσεις και ο Firefox κατέληξε να είναι ένας από τους συχνότερα χρησιμοποιούμενους browsers. [13][14][15][16]

Παρακάτω απαριθμούνται μερικές ημερομηνίες - σταθμοί στην ιστορία του Firefox.

Ημερομηνία	Γεγονός
31/05/1998	Ο κώδικας του Netscape μοιράζεται ελεύθερα στο διαδίκτυο.
28/10/1998	Ανακοινώνεται η νέα κατεύθυνση που θα ακολουθηθεί για τον κώδικα του browser.
17/03/1999	Η America Online Inc. εξαγοράζει την Netscape Communications.
14/11/2000	Ο Netscape 6 βγαίνει στην αγορά, στηριζόμενος σε λογισμικό ανοιχτού κώδικα.
23/09/2002	Ανακοινώνεται ο Phoenix 0.1.
15/07/2003	Δημιουργείται ο οργανισμός Mozilla Foundation σαν μία start-up στηριζόμενη από την America Online και τον Mitch Kapor.
09/06/2004	Ο αριθμός των χρηστών που κατέβασαν τον Firefox εκτοξεύεται μετά την ανακοίνωση σημαντικών λαθών στην ασφάλεια του IE.
14/09/2004	Ο Mozilla προχωράει στις πρώτες επί πληρωμή έρευνες ασφαλείας.
15/12/2004	Εμφανίζεται διαφήμιση του Firefox στους New York Times.
03/08/2005	Η Mozilla Corporation δημιουργείται ως θυγατρική της Mozilla Foundation.
13/02/2007	Το πρώτο κείμενο του μανιφέστου του Firefox ανακοινώνεται στο διαδίκτυο.
28/12/2007	Η America Online ανακοινώνει το τέλος των εκδόσεων του Netscape και προτείνει στους χρήστες τον Firefox.
02/07/2008	Ο Mozilla δημιουργεί ρεκόρ Γκίνες ως προς τα downloads του Firefox 3.
28/01/2010	Ανακοινώνεται η πρώτη έκδοση του Firefox για κινητές συσκευές.
22/03/2011	Ανακοινώνεται ο Firefox 4 και ακολουθούν ραγδαίες αναβαθμίσεις κάθε εβδομάδα.
25/07/2001	Ανακοινώνεται η έναρξη ενός project για δημιουργία λειτουργικού Firefox OS.

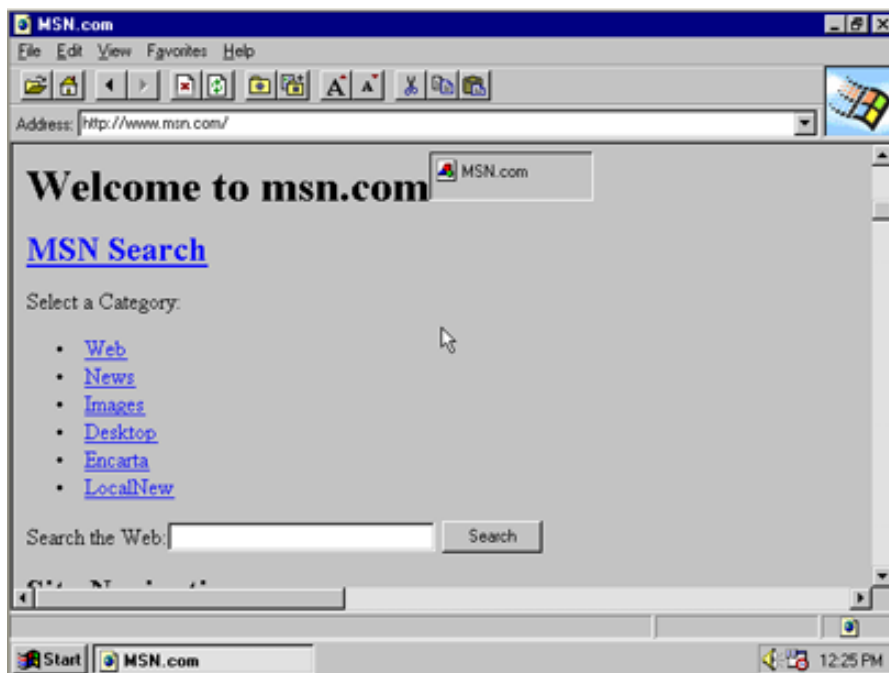
Στον παρακάτω πίνακα φαίνονται ενδεικτικά μερικές εκδόσεις του browser και ορισμένα σημαντικά χαρακτηριστικά που ενσωματώθηκαν.

Ημερομηνία	Έκδοση	Χαρακτηριστικά
01/10/2002	0.2	Δημιουργία search bar
19/10/2002	0.4	Pop up blocking
07/12/2002	0.5	Μείωση μεγέθους και μνήμης Βελτίωση απόδοσης
15/10/2003	1.5	Password Manager
29/11/2005	1.8	Εκκαθάριση ιστορικού και στοιχείων πλοήγησης Βελτιωμένα web standards
21/10/2006	2.0	Ενσωματωμένη ant phishing προστασία JavaScript 1.7
17/06/2008	3.0	Malware protection Antivirus protection with download manager JSON leak protection
30/06/2009	3.5	HTML5 video / audio support Private browsing mode HTML5 local storage / offline application storage Canvas SVG transforms
21/06/2011	5.0	Καλύτερη HTML5 υποστήριξη
08/01/2013	18.0	Ταχύτερη JavaScript Καλύτερη αναπαράσταση των εικόνων
09/02/2013	19.0	Ενσωματωμένος PDF Reader
25/06/2013	22.0	Υποστηρίζονται τα στοιχεία data / time της HTML5
06/08/2013	23.0	Υποστηρίζεται το <input type="range"> της HTML5

3.6 - Internet Explorer



Η πρώτη έκδοση του IE ολοκληρώθηκε στις 16 Αυγούστου του 1995. Στην ουσία όμως υπήρξε απλώς μία επανέκδοση του Mosaic. Ο IE ερχόταν και ως προ-εγκατεστημένος στα Windows 95. Αρχικός σκοπός του εγχειρήματος ήταν να βοηθηθούν οι άνθρωποι να χρησιμοποιούν το διαδίκτυο. Τότε μόνο 6 άνθρωποι ασχολούταν με την παραγωγή του IE. Η δεύτερη έκδοση του IE ήρθε τον Νοέμβριο του 1995 και παρείχε υποστήριξη για SSL (Secure Sockets Layer: πρωτόκολλο που σχεδιάστηκε από τη Netscape για να παρέχει ασφάλεια κατά την μετάδοση ευαίσθητων δεδομένων στο διαδίκτυο) και Internet Newsgroups. Ήταν η πρώτη έκδοση που υποστηρίζονταν τόσο από τα Windows όσο και από τα Macintosh.



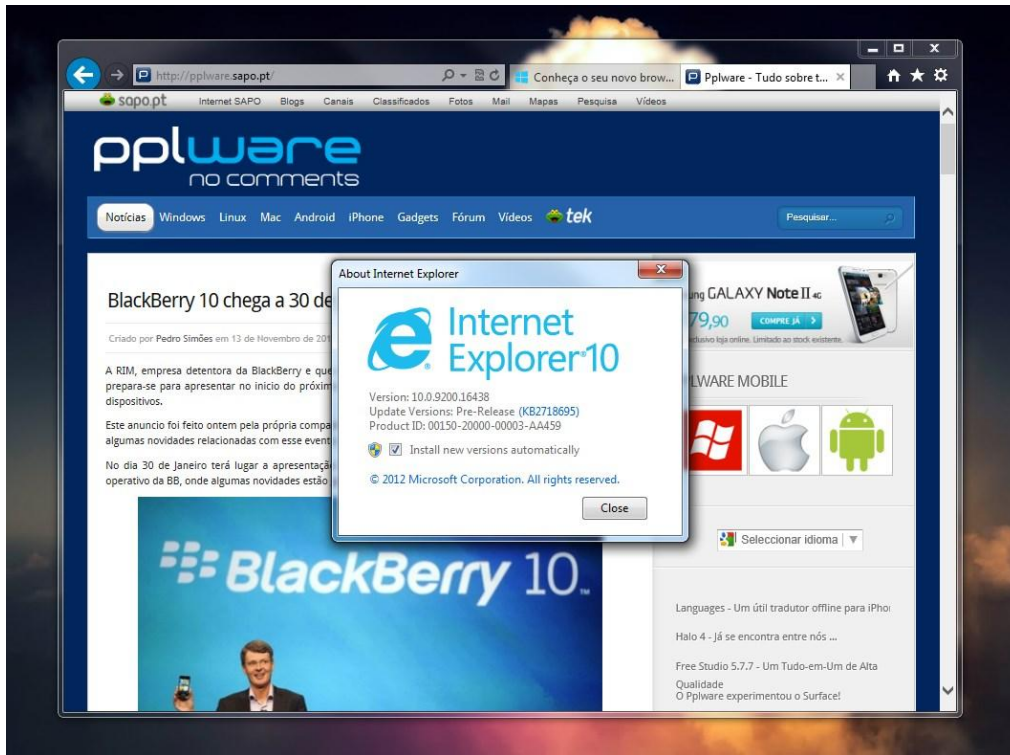
Εικόνα 3. 13: Η πρώτη έκδοση του IE [17]

Η 3^η έκδοση κυκλοφόρησε τον Αύγουστο του 1996. Παρείχε πλήθος νέων χαρακτηριστικών όπως **CSS**, **ActiveX** και **Java** υποστήριξη, **mail**, **windows media player**, windows address book, News 1.0, Microsoft NetMeeting και μπορούσε να εμφανίσει εικόνες μορφής JPEG και GIF. Σημαντικό χαρακτηριστικό ήταν ότι μπορούσε να υποστηρίξει **streaming** αρχεία ήχου χωρίς την υποστήριξη τρίτων εφαρμογών. Μέσα στην πρώτη εβδομάδα διαθεσιμότητας του προγράμματος, το «κατέβασαν» περισσότεροι από ένα εκατομμύριο χρήστες.

Η 4^η έκδοση, τον Οκτώβριο του 1997 υποστήριζε την **DHTML** (Dynamic HTML). Αυτό σαν γεγονός έφερε επανάσταση στον κόσμο του διαδικτύου, καθώς πλέον οι ιστοσελίδες αφαίρεσαν από πάνω τους αυτή τη στατικότητα και έγιναν δυναμικές. Υπήρχε πλέον η δυνατότητα αλληλεπίδρασης με το χρήστη. Σε αυτό το σημείο η Microsoft κατηγορήθηκε για αθέμιτο ανταγωνισμό καθώς ο IE είχε δεθεί πλέον με το λειτουργικό των Windows. Από την άλλη πλευρά, εάν κάποιος ήθελε να χρησιμοποιήσει κάποιον άλλο browser, τότε θα έπρεπε να τον κατεβάσει κάτι το οποίο ήταν δύσκολο με τις αργές συνδέσεις της εποχής.

Η 5^η έκδοση εμφανίστηκε το 1999. Σε αυτή την έκδοση παρέχονταν υποστήριξη AJAX (ο όρος ανακαλύφθηκε αργότερα). Μέχρι το τέλος του 1999 για τον IE εργάζονταν περίπου 100 άτομα με ένα συνολικό budget 100 εκατομμυρίων δολαρίων. Ο IE απέκτησε άσχημη φήμη με την 6^η έκδοση του προγράμματος, καθώς όταν χρησιμοποιούνταν από μηχανήματα παλαιού τύπου με κρίσιμες πληροφορίες αποθηκευμένες, παρουσιάζονταν σοβαρά ζητήματα ασφαλείας. Η 7^η έκδοση ήρθε αρκετά αργότερα, το 2006. Η νέα αυτή έκδοση υποστήριζε τις λεγόμενες καρτέλες και προστασία απέναντι σε phishing attacks.

Η 9^η έκδοση του IE κυκλοφόρησε με αρκετά νέα χαρακτηριστικά, τα σημαντικότερα από αυτά να είναι αυτόματη ανάκαμψη από βλάβες (**automated recovery enhanced**), περαιτέρω υποστήριξη της HTML5 και καλύτερο σκορ σε ένα benchmark τεστ, ονόματι Acid 3.0. Η τελευταία έκδοση (10) κυκλοφόρησε το 2012 και παρέχει CSS3 υποστήριξη, hardware acceleration και ακόμη περισσότερη HTML5 υποστήριξη. [17]



Εικόνα 3. 14: IE 10 [17]

3.7 – Opera



Ο Opera web browser ξεκίνησε το 1994 σαν ερευνητικό project της Telenor, της μεγαλύτερης εταιρίας τηλεπικοινωνιών στη Νορβηγία. Το 1995 έγινε ξεχωριστή εταιρία με το όνομα Opera Software ASA. Η πρώτη δημόσια έκδοση κυκλοφόρησε το 1996. Τότε ήταν συμβατή μόνο με τα Windows. Το 1998 αποφασίστηκε να ξεκινήσει έρευνα για να προωθηθεί ο Opera σε κινητές συσκευές. Το 2000 κυκλοφόρησε η έκδοση 3.0 στην οποία υποστηρίζονταν πολλά λειτουργικά συστήματα και πλατφόρμες.

Στην έκδοση 9.1 το 2006 περιλαμβάνονταν προστασία από απάτες χρησιμοποιώντας τεχνολογία από τον GeoTrust καθώς και προστασία από phishing. Στην έκδοση 9.5 προστέθηκε επιπλέον προστασία από malware.

Ο Opera περιλαμβάνει πολλά και διάφορα features όπως **ad blocking**, λογισμικό προστασίας από απάτες, download manager, Bit Torrent client, **RSS Reader** και ενσωματωμένους έναν **email client** και IRC Chat.

Ανά τις διάφορες εκδόσεις του Opera browser, υπήρξαν διάφορες καινοτομίες οι οποίες με τον καιρό υιοθετήθηκαν και από τους υπόλοιπους browsers. Ο Opera ήταν ο πρώτος browser που υποστήριζε την λειτουργία **tabbed browsing** από τις πρώτες κιόλας εκδόσεις του, το 1993. Επίσης το **caching** των σελίδων γινόταν εν μέρει στη μνήμη RAM, παρά μόνο στον σκληρό δίσκο.

Στον Opera 2 το 1996, εμφανίστηκαν τα **sessions**. Οι χρήστες μπορούσαν να επανακτούν τις σελίδες τις οποίες είχαν ανοίξει, τη θέση του κάθε παραθύρου και το σημείο στο οποίο είχαν κάνει scroll σε κάθε σελίδα. Στην 3^η έκδοση του προγράμματος εισήχθη το χαρακτηριστικό **UserCSS**, με το οποίο οι χρήστες μπορούσαν να διαμορφώσουν το στυλ των σελίδων όπως επιθυμούσαν αυτοί.

Στην 4^η έκδοση του προγράμματος το 2000, εμφανίστηκε το χαρακτηριστικό της **διαγραφής των ψηφιακών αποτυπωμάτων** που άφηνε ο χρήστης όπως history, cookies, cache. Την ίδια χρονιά στην 5^η έκδοση, εισήχθη η λειτουργία ad blocking.

Το 2003 στην 7^η έκδοση εμφανίστηκε το χαρακτηριστικό της χωρικής μετακίνησης (**spatial navigation**), με το οποίο οι χρήστες μπορούσαν να μετακινηθούν σε διάφορα στοιχεία μιας ιστοσελίδας, όπως links, σύμφωνα με την τοποθέτησή τους στο χώρο και όχι σύμφωνα με την HTML δομή.



Εικόνα 3. 15: Spatial Navigation [75]

Στο Opera 8 το 2005 οι developers εισήγαγαν το χαρακτηριστικό της φωνητικής πλοήγησης, με το οποίο ήταν δυνατό οι χρήστες να μετακινηθούν στις σελίδες με φωνητικές εντολές. Το πρόγραμμα μπορούσε να τους επιστρέφει φωνητικά τα περιεχόμενα των σελίδων. Στην 9^η έκδοση του προγράμματος ο browser υποστήριζε έναν ενσωματωμένο BitTorrent client.

Παρόλο που ο browser έχει εισάγει ανά τα χρόνια πολλά και σημαντικά στοιχεία τα οποία σήμερα αποτελούν default στοιχεία όλων των browsers, ο Opera δεν έχει καταφέρει να πάρει μεγάλο μερίδιο αγοράς. Σημαντικός παράγοντας για αυτήν την κατάσταση είναι ότι στις αρχές της δημιουργίας του μετά από σύντομο χρονικό διάστημα έπρεπε οι χρήστες να τον αγοράσουν ενώ στη συνέχεια επέτρεπε να εμφανίζονται διαφημίσεις μέσα στο πρόγραμμα, με το κατάλληλο αντίτιμο. Πλέον όμως, ο Opera είναι ένας δωρεάν browser όπως όλοι οι υπόλοιποι. [18] [19] [20] [21]

3.8 - Chrome



Ο Chrome είναι ένας freeware browser που αναπτύχθηκε από την Google. Η πρώτη beta έκδοση κυκλοφόρησε για τα Windows στις 2 Σεπτεμβρίου του 2008, ενώ η πρώτη δημόσια stable έκδοση στις 11 Δεκεμβρίου του ίδιου χρόνου. Η Google δημοσιοποίησε το μεγαλύτερο μέρος του κώδικα του Chrome στα πλαίσια ενός open source project του Chromium. Όλες οι εκδόσεις του Chrome βασίζονται πάνω σε αυτό το project.

Ο Eric Schmidt (executive chairman της Google) αρχικά δεν συμφωνούσε με την ιδέα της δημιουργίας ενός browser καθώς θεωρούσε την Google ακόμη μικρή εταιρία και δεν ήθελε να συμμετάσχει στον πόλεμο των browsers. Μετέπειτα και αφού η Google είχε προσλάβει αρκετούς developers της Mozilla Firefox, και αφού είδε ένα προσχέδιο του Chrome ενθουσιάστηκε με την ιδέα και συμφώνησε να προχωρήσει το project.

Ο Chrome στοχεύει στην ασφάλεια, την ταχύτητα, την απλότητα και την σταθερότητα. Στα μεγαλύτερα πλεονεκτήματα του Chrome συγκαταλέγονται η απόδοσή του, και η ταχύτητα επεξεργασίας της JavaScript. Πολλά από τα χαρακτηριστικά του Chrome υλοποιήθηκαν πρώτα από αυτόν παρόλο που είχαν ανακοινωθεί και από άλλους browser developers. Για παράδειγμα η δυνατότητα της αναζήτησης μέσα από το address bar (**Omnibox**) είχε ανακοινωθεί πρώτα από την Mozilla αλλά εμφανίστηκε πρώτα στον Chrome. Μία άλλη πρωτοπορία του Chrome είναι η δυνατότητα του συγχρονισμού των bookmarks και των κωδικών πρόσβασης μέσω ενός Google λογαριασμού. Με αυτόν τον τρόπο σε κάθε νέα εγκατάσταση του Chrome, απλώς με τη σύνδεση με τον Google λογαριασμό ενσωματώνονται όλα τα προσωπικά στοιχεία του χρήστη.

Ο Chrome παρέχει και αυξημένη ασφάλεια για τους χρήστες του. Περιοδικά η Google δημοσιεύει δύο **blacklists** με επικίνδυνα sites και προειδοποιεί τους χρήστες να μην τα επισκέπτονται. Επίσης έχουν αναπτύξει ένα μοντέλο με το οποίο κάθε καρτέλα (tab) στον browser θεωρείται ανεξάρτητη από την άλλη και δεν μπορούν να αλληλεπιδράσουν. Εκτός αυτών έχουν διαμορφώσει έτσι το σύστημά τους ώστε οι διεργασίες που χρησιμοποιούνται δεν μπορούν να αλληλεπιδράσουν με ευαίσθητες περιοχές του υπολογιστή όπως η μνήμη ή τα αρχεία του χρήστη. Ενδιαφέρον παρουσιάζει το θέμα του **Master Password** (κοινός κωδικός πρόσβασης για όλα τα sites). Πολλές ομάδες έχουν συμπεριλάβει το χαρακτηριστικό αυτό στους browser τους, ενώ η Google θεωρεί πως δεν παρέχει ουσιαστική ασφάλεια από τους hackers και όλα τα bugs που κατοχυρώνονται με αυτό το θέμα χαρακτηρίζονται ως "WontFix".[22]

Στον παρακάτω πίνακα φαίνονται ενδεικτικά μερικές ημερομηνίες - σταθμοί για τον browser, καθώς και σημαντικά στοιχεία που ενσωματώθηκαν στην αντίστοιχη έκδοση.

Έκδοση	Ημερομηνία	Σημαντικά Στοιχεία
0.2.149	02/09/2008	Πρώτη έκδοση
1.0.154	11/12/2008	Πρώτη σταθερή έκδοση
2.0.172	24/05/2009	35% αύξηση ταχύτητας της JavaScript στο SunSpider benchmark
3.0.159	12/10/2009	25% ταχύτερη JavaScript HTML5 video / audio υποστήριξη
3.0.249	25/01/2010	Extensions Bookmark συγχρονισμός Περαιτέρω HTML5 υποστήριξη Βελτίωση στην ταχύτητα 100% Acid3 Pass Test
5.0.375	21/05/2010	Περαιτέρω HTML5 υποστήριξη (App Cache, drag n drop) Ενσωμάτωση του Adobe Flash Player
8.0.552	02/12/2010	Chrome web store για plug-ins Ενσωματωμένος PDF viewer για αυξημένη ασφάλεια
12.0.742	07/06/2011	Hardware accelerated 3D CSS Ασφάλεια κατά το downloading
17.0.963	08/02/2012	Download scanning protection
29.0.1547	20/08/2013	«Τελευταία έκδοση»

3.9 - Safari



Ο Safari browser αναπτύχθηκε από την Apple και συμπεριλαμβάνεται στα λειτουργικά Mac και iOS. Η πρώτη έκδοση κυκλοφόρησε στις 7 Ιανουαρίου του 2003. Για τα Windows η πρώτη έκδοση κυκλοφόρησε στις 11 Ιουνίου του 2007.

Αρχικά στις εκδόσεις του Mac συμπεριλαμβάνονταν οι Internet Explorer και Cyberdog. Στις 7 Ιανουαρίου 2003 ο Steve Jobs ανακοίνωσε ότι η Apple είχε αναπτύξει τον δικό της browser χρησιμοποιώντας μία δική τους μηχανή αναπαράστασης γραφικών. Η πρώτη σταθερή έκδοση ανακοινώθηκε στις 23 Ιουνίου του 2003. Στις 29 Απριλίου του 2005 κυκλοφόρησε η δεύτερη έκδοση του προγράμματος. Αυτός ήταν ο μόνος browser που συμπεριλαμβανόταν πλέον στα λειτουργικά Mac. Η έκδοση 2.0.2 που κυκλοφόρησε στις 31 Οκτωβρίου του 2005 συμπεριλάμβανε τις διορθώσεις για να περάσει ο browser το Acid2 test.

Στις 2 Ιουνίου του 2008, οι υπεύθυνοι της ομάδας για την ανάπτυξη του Safari, ανακοίνωσαν στην 4^η έκδοση του προγράμματος μία νέα JavaScript μηχανή, την **SquirrelFish**, η οποία βελτίωνε κατά πολύ την ταχύτητα επεξεργασίας των scripts. Η εν λόγω μηχανή εξελίχθηκε στην Nitro καταφέροντας στο τέλος να ξεπεράσει τις ταχύτητες του IE8 κατά οκτώ φορές και του Firefox κατά τέσσερις. Με τη συγκεκριμένη έκδοση ο Safari πέρασε 100% το Acid3 test. Επιπλέον παρέχονταν εργαλεία για τους developers όπως Web Inspector, CSS element viewing, JavaScript debugger, resource graphs και database management με SQL υποστήριξη.

Στην 5^η έκδοση του Safari υπήρξε μία βελτίωση της τάξεως του 30% στην επεξεργασία της JavaScript. Επίσης παρείχαν βελτιωμένα εργαλεία για τους developers καθώς και περισσότερη υποστήριξη για την HTML5. Παρέχονταν επίσης ο **Safari Reader** ο οποίος αφαιρούσε το στυλ και τις διάφορες διαφημίσεις από τις ιστοσελίδες επιστρέφοντας δηλαδή μόνο το κείμενο, extensions, επιπρόσθετη υποστήριξη της HTML5 και **DNS prefetching** (ο browser βρίσκει τους συνδέσμους και επιστρέφει τις διευθύνσεις τους πριν ζητηθούν). [26]

3.10 - Maxthon



Ο Maxthon είναι ένας freeware browser ο οποίος κατασκευάζεται στην Κίνα. Σύμφωνα με τους κατασκευαστές του ο browser κατασκευάζεται έτσι ώστε να έχει υψηλή ταχύτητα, περισσότερα χαρακτηριστικά από τους υπόλοιπους browsers και το χαρακτηριστικό “Seamless browsing”, δηλαδή να έχουν οι χρήστες τη δυνατότητα να συνεχίζουν την πλοήγησή τους από εκεί που την άφησαν τελευταία φορά. Αυτό γίνεται έχοντας μέρη του browser

κωδικοποιημένα σε cloud υπηρεσίες.

Ο Maxthon προσφέρει μερικά ιδιαίτερα χαρακτηριστικά όπως τα **Smart Accelerator** (μηχανή για αύξηση της ταχύτητας πλοήγησης), **Undo List** (δυνατότητα επαναφοράς κλεισμένων ιστοσελίδων), **Online Syncing**, **Feed Reader**, **Super Drag and Drop** (οι χρήστες μπορούν να επιλέξουν και να σύρουν ένα κείμενο μέσα σε μια μηχανή αναζήτησης, ή να ανοίξουν ένα σύνδεσμο ή εικόνα σε ξεχωριστά tabs σέρνοντάς τα) και **Ad Blocking**.

Ο Maxthon 4 (**Maxthon Cloud Browser**) κυκλοφόρησε στις 10 Δεκεμβρίου του 2012. Από αυτή την έκδοση υποστηρίζονται **cloud** λειτουργίες όπως ο διαμοιρασμός αρχείων online, η αποθήκευσή τους σε cloud λογαριασμό, ο συγχρονισμός των δεδομένων του χρήστη μέσω του cloud, **Reader Mode** όπου αφαιρούνται οι όποιες διαφημίσεις από το κείμενο και διαμορφώνεται αναλόγως το μέγεθος των γραμμάτων, **Ad Hunter** (αφαίρεση διαφημίσεων), **New Sessions** (ταυτόχρονη είσοδος σε websites με διαφορετικό λογαριασμό) καθώς και **Source Sniffer** (ανάκτηση φωτογραφιών και αρχείων βίντεο / ήχου από μία ιστοσελίδα). Τέλος ο Maxthon καταφέρνει τη μεγαλύτερη βαθμολογία στην συμβατότητα με την HTML5 σε σχέση με τους υπόλοιπους browsers σε σχετικά τεστ. [23]

3.11 - Avant



Ο Avant είναι ένας freeware browser ο οποίος δημιουργήθηκε από τον κινέζο προγραμματιστή Anderson Che. Η έκδοση 2012 (που κυκλοφόρησε τον Οκτώβριο του 2011) περιλάμβανε δύο μηχανές δημιουργίας των γραφικών του browser, την Trident layout engine (Internet Explorer shell) και την Gecko layout engine (Mozilla Firefox shell). Ο χρήστης είχε τη δυνατότητα να επιλέξει ποια από τις δύο μηχανές να χρησιμοποιείται.

Ο Avant δημιουργήθηκε έχοντας ως πρότυπο τον Opera, τον πρώτο browser που υποστήριζε το “multiple document interface” (η δυνατότητα να εμφανίζονται πολλά παράθυρα υπό ένα υπέρ-παράθυρο). Στόχος λοιπόν ήταν να δημιουργηθεί ένας browser χρησιμοποιώντας την μηχανή γραφικών του IE, παρέχοντας την εργονομία του Opera αλλά χωρίς τα προβλήματα του πρώτου (IE).

Κατά την πορεία της ανάπτυξης του λογισμικού προστέθηκαν διάφορα χαρακτηριστικά στον browser όπως popup blocking, ad blocking και ActiveX, Java, JavaScript one-click disabling. Η μηχανή αναζήτησης που χρησιμοποιεί ο Avant είναι μία έκδοση της Yahoo!search μηχανής αναζήτησης ωστόσο παρέχονται και άλλες μηχανές.

Μερικά από τα χαρακτηριστικά που προσφέρονται από τον Avant είναι η ανάκτηση των σελίδων ύστερα από βλάβη του προγράμματος, ενσωματωμένος RSS reader, συμβατότητα με πολλά add-ins και toolbars του Internet Explorer και online bookmark synchronization. [24]

3.12 - Seamonkey



Το SeaMonkey είναι ένα δωρεάν και open source cross - platform Internet Suite. Αποτελεί τη συνέχεια του πρώην Mozilla Application Suite και η ανάπτυξή του βασίζεται στην κοινότητά του.

Αποτελείται από έναν web browser, ο οποίος είναι απόγονος του Netscape, έναν email client και έναν news client, έναν HTML editor καθώς και έναν IRC client. Οι mail λειτουργίες περιλαμβάνουν υποστήριξη για πολλαπλούς λογαριασμούς, εντοπισμό των junk mails, φίλτρα, λεξικό, κατάλογο διευθύνσεων και άλλα.

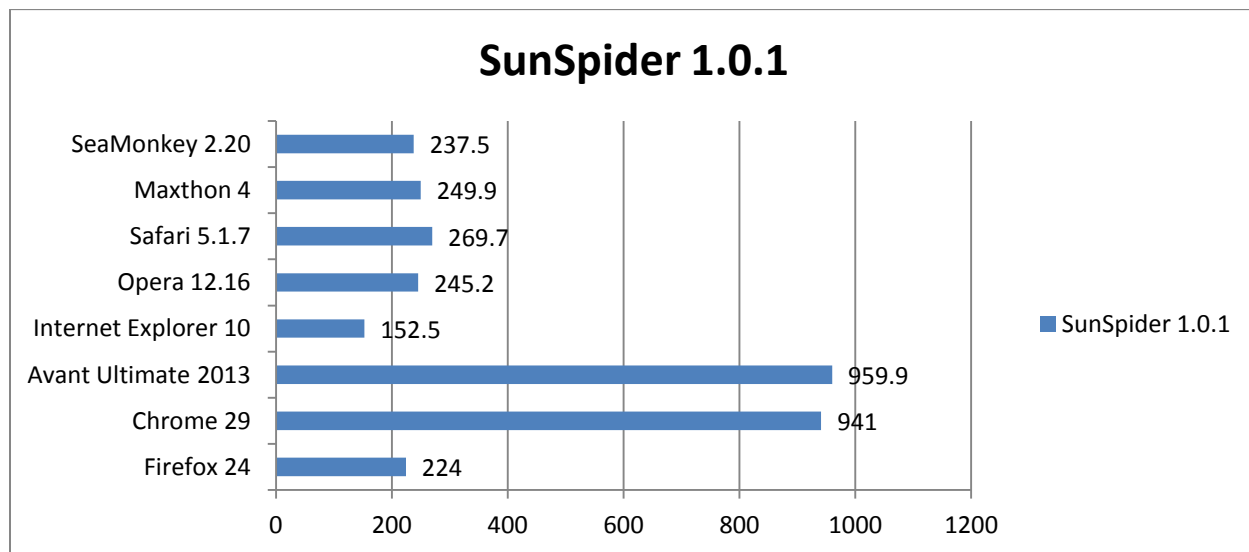
Στα 10 Μαρτίου 2005, ο οργανισμός Mozilla (Mozilla Foundation) ανακοινώνει ότι δεν θα κυκλοφορεί πλέον επίσημες εκδόσεις του Mozilla Application Suite, καθώς επικεντρώνεται στις standalone εφαρμογές Firefox και Thunderbird (email client). Παρόλα αυτά δηλώνει ότι θα παρέχει υποστήριξη σε όποια μέλη της κοινότητας επιθυμούν να συνεχίσουν το project. Έτσι συνεχίζεται η ανάπτυξη του project υπό την επίβλεψη του SeaMonkey Council. Η πρώτη έκδοση (SeaMonkey 1.0) κυκλοφόρησε στις 10 Σεπτεμβρίου του 2005. Η πρώτη σταθερή έκδοση κυκλοφόρησε στις 30 Ιανουαρίου 2006. Η τελευταία έκδοση του προγράμματος είναι η SeaMonkey 2.17.1. [25]

3.13 - Σύγκριση των Browsers

Για την σύγκριση των προηγούμενων browsers, χρησιμοποιήσαμε διάφορα benchmarks και παραθέτουμε παρακάτω τα αποτελέσματά τους. Όλες οι δοκιμές έγιναν σε μηχανήμα με Windows 7 64bit, 4GB Ram, GeForce 8600 και CPU Intel Dual Core E6750. [28]

3.13.1 - SunSpider JavaScript 1.0.1

Το εν λόγω benchmark αναπτύχθηκε από την Apple το 2007 και επικεντρώνεται σε καθημερινές χρήσεις της JavaScript όπως κωδικοποίηση και χειρισμό κειμένου (encryption / text manipulation). Θεωρείται ως ένα από τα ανταγωνιστικότερα benchmarks. Όσο μικρότερος ο χρόνος που χρειάστηκαν για να ολοκληρώσουν το benchmark, τόσο το καλύτερο.

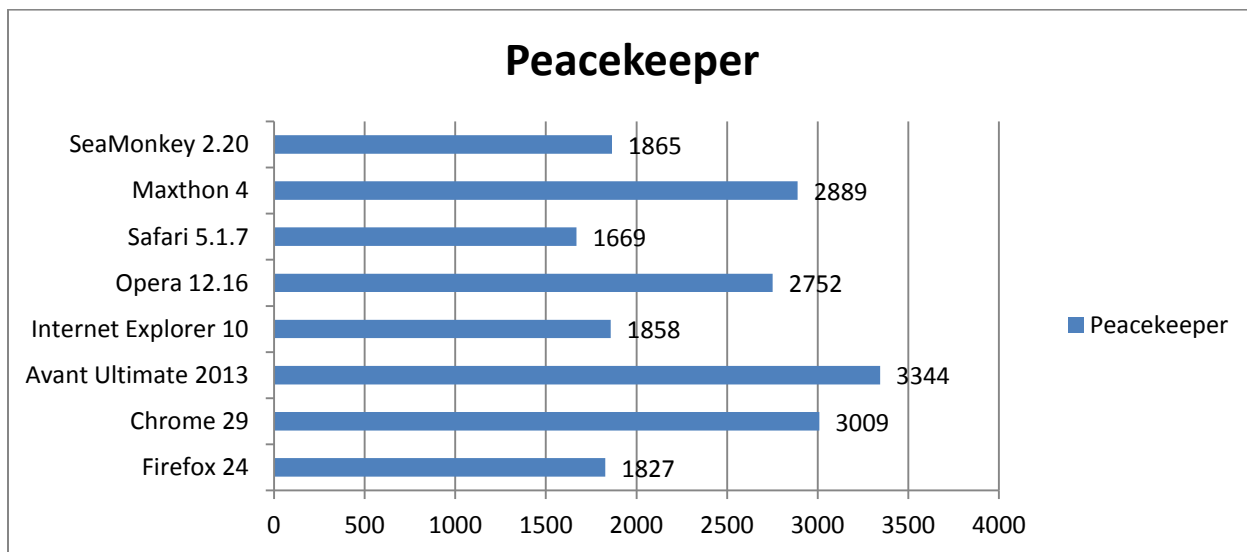


Εικόνα 3. 16: SunSpider Benchmark

Όπως φαίνεται και από το παραπάνω γράφημα νικητής σε αυτό το benchmark είναι ο Internet Explorer 10 με τον Firefox 24 να ακολουθεί. Εντύπωση προκαλεί ο χρόνος που χρειάστηκε ο Chrome 29 για να ολοκληρώσει το τεστ, καθώς είναι τρεις και τέσσερις φορές μεγαλύτερος των υπολοίπων.

3.13.2 - Peacekeeper Benchmark

Κατασκευάστηκε από την Futuremark. Καλύπτει μεγαλύτερο φάσμα από την JavaScript μηχανή και είναι ένας καλός δείκτης για την συνολική εικόνα του browser σε front - face επίπεδο. Όσο μεγαλύτερο σκορ πετυχαίνει ένας browser τόσο το καλύτερο.

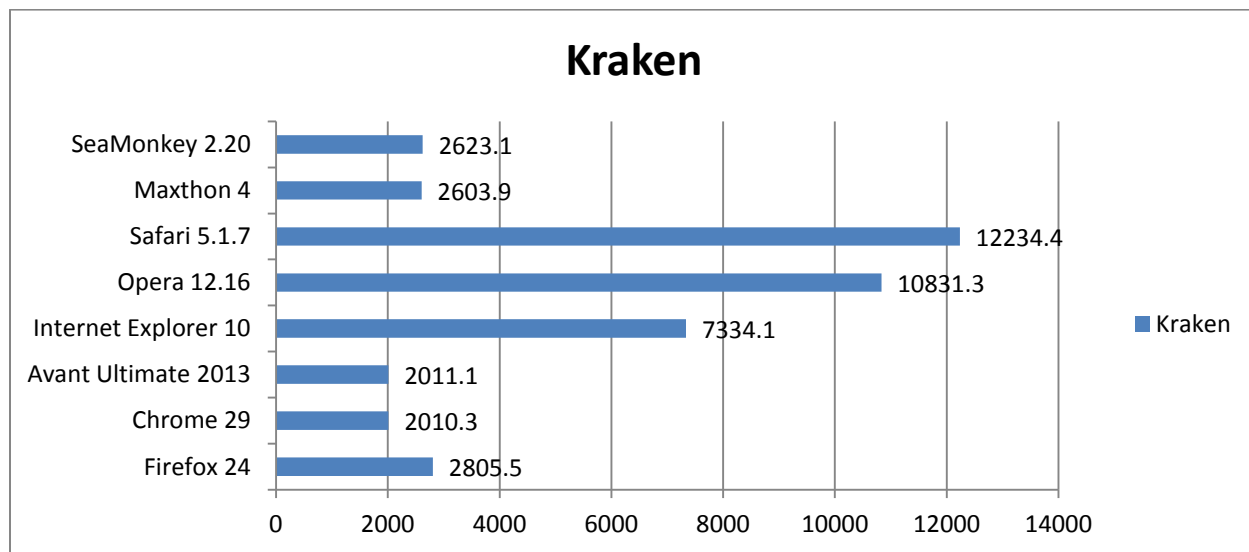


Εικόνα 3. 17: Peacekeeper Benchmark

Προς έκπληξη, στα αποτελέσματα βλέπουμε ότι την πρώτη θέση την κατέχει ο Avant Ultimate 2013 browser με σκορ 3344 και ενώ χρησιμοποιούσε την Chrome layout engine. Στη δεύτερη θέση έρχεται ο Chrome με σκορ 3009 ενώ λίγο πίσω του ακολουθεί ο Maxthon.

3.13.3 - Kraken Benchmark

Το Kraken είναι το JavaScript benchmark εργαλείο της Firefox. Ένα απαιτητικό τεστ το οποίο επικεντρώνεται όπως το SunSpider σε ρεαλιστικές / καθημερινές χρήσης της JavaScript. Παρόμοια όσο μικρότερος ο χρόνος εκτέλεσης τόσο το καλύτερο.

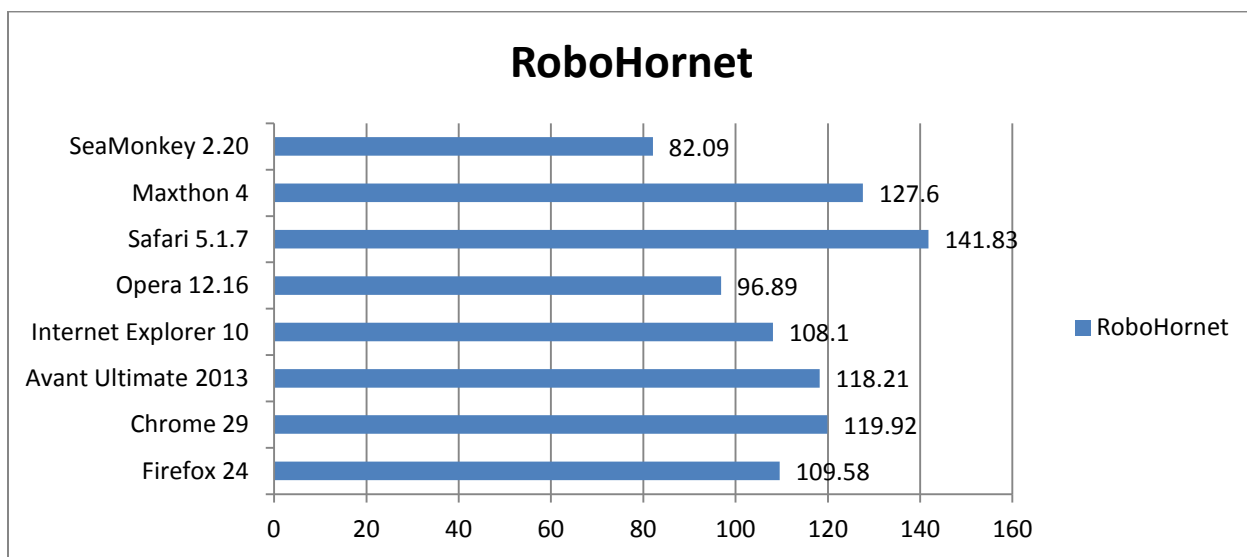


Εικόνα 3. 18: Kraken Benchmark

Παρατηρούμε ότι τους καλύτερους χρόνους εκτέλεσης τους έχει καταφέρει ο Avant browser με τον Chrome. Αντίθετα τους τρεις χειρότερους τους έχουν καταφέρει οι Safari, Opera και Internet Explorer. Ενδιαφέρον παρουσιάζει ότι τα αποτελέσματα του Avant φαίνονται να είναι μέχρι τώρα κοντά με αυτά του Chrome, πιθανότατα επειδή στις δοκιμές μας αυτός χρησιμοποιούσε τη μηχανή του Chrome.

3.13.4 - RoboHornet

Το RoboHornet αποτελεί μία από τις τελευταίες προσθήκες στα benchmarks. Έχει κατασκευαστεί από την Google και μέχρι στιγμής βρίσκεται στο στάδιο του alpha testing, οπότε τα αποτελέσματά του θα πρέπει να λαμβάνονται υπόψη με λίγη προσοχή. Όσο μικρότερο το σκορ, τόσο το καλύτερο.

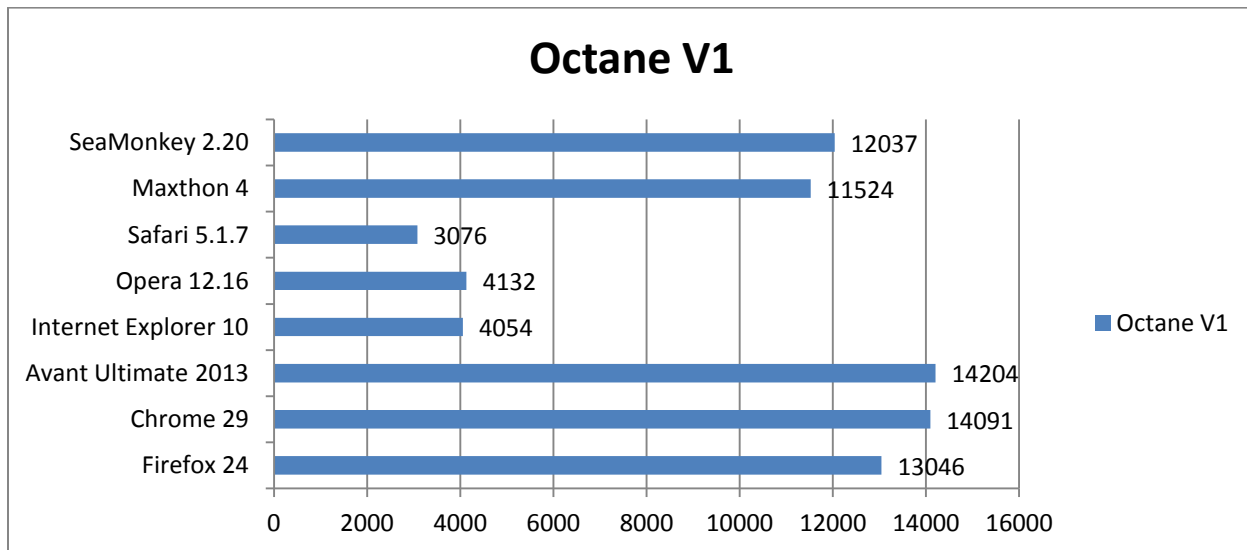


Εικόνα 3. 19: RoboHornet Benchmark

Παρατηρούμε ότι στο εν λόγω benchmark τον καλύτερο χρόνο τον επιτυγχάνει ο SeaMonkey, ενώ τον ακολουθεί ο Opera. Αντίθετα τους χειρότερους χρόνους τους επιτυγχάνουν οι Maxthon και Safari.

3.13.5 - Octane Benchmark

Αποτελεί το νέο benchmark εργαλείο της Google με το οποίο δοκιμάζεται ο Chrome. Όσο μεγαλύτερο σκορ καταφέρνουν οι browsers τόσο το καλύτερο.



Εικόνα 3. 20: Octane Benchmark

Παρατηρούμε ότι τα καλύτερα αποτελέσματα τα έφεραν οι Chrome, Avant, Firefox, Seamonkey και Maxthon. Οι υπόλοιποι τρεις έχουν μείνει πίσω με διαφορά.

3.13.6 - Avant Browser

Στη συνέχεια δοκιμάσαμε πάλι το SunSpider JavaScript benchmark στον Avant, αυτή τη φορά όμως χρησιμοποιώντας την μηχανή του Internet Explorer 10. Ο χρόνος που χρειάστηκε να ολοκληρώσει το τεστ έπεσε στα 172,4ms από τα 959,9ms. Έτσι, συμπεραίνουμε ότι ανάλογα με την μηχανή που χρησιμοποιεί ο Avant, θα καταφέρει και τα αντίστοιχα αποτελέσματα. Επίσης φαίνεται ξεκάθαρα ότι υπάρχει διαφορά ανάμεσα σε αυτές τις rendering engines.

3.13.7 - Αποτελέσματα

Τα παραπάνω αποτελέσματα δεν προσφέρουν κάποιο ξεκάθαρο νικητή όπως θα περιμέναμε στην αρχή. Ο **Chrome 29** έχει καταφέρει γενικώς πολύ καλά αποτελέσματα και θα μπορούσαμε να πούμε πως παίρνει την πρωτιά με τον **Firefox**, τον **Maxthon** και τον **Avant** να ακολουθούν.

Σε ρεαλιστικές συνθήκες ενός μέσου χρήστη, δύσκολα θα φανούν οι διαφορές τους, οπότε η επιλογή του browser επαφίεται στον κάθε χρήστη ξεχωριστά.

4^ο Κεφάλαιο - HTML5 New Features

4.1 – Εισαγωγή

Σύμφωνα με τα [4][5], η HTML5 είναι η νέα έκδοση της HTML (Hyper Text Markup Language) από το W3C (World Wide Web Consortium). Η νέα αυτή έκδοση βγήκε στη δημοσιότητα το 2011 και από τότε άρχισε να χρησιμοποιείται κατά κόρον. Εκείνη τη στιγμή η υποστήριξή της από τους διάφορους browsers ήταν ελάχιστη. Όσο όμως περνάει ο καιρός όλοι οι browsers φροντίζουν έτσι ώστε να υποστηρίζουν τα νέα χαρακτηριστικά της.

Το σημαντικό με την HTML5 είναι ότι δεν πρόκειται για μία έκδοση η οποία αντικαθιστά την προηγούμενη (HTML4.01). Απλώς προσθέτει διάφορα νέα στοιχεία και χαρακτηριστικά. Οτιδήποτε προβλεπόταν από την HTML4 συνεχίζει να λειτουργεί και να υποστηρίζεται.

Από το 2011 η HTML5 είναι σε συνεχή εξέλιξη, με τους ανθρώπους του W3C να προσπαθούν να εισάγουν όλο και περισσότερα χαρακτηριστικά και να την κάνουν ακόμη καλύτερη. Η νέα αυτή έκδοση της HTML, δημιουργήθηκε έχοντας ως γνώμονα τα παρακάτω:

- Ελάττωση των εξωτερικών plug-ins όπως για παράδειγμα το Flash Player. Γίνεται προσπάθεια οι περισσότερες λειτουργίες να εκτελούνται τοπικά στον browser του χρήστη (native).
- Η HTML5 θα πρέπει να είναι ανεξάρτητη της συσκευής που εμφανίζεται (device – independent).
- Τα νέα χαρακτηριστικά θα πρέπει να βασίζονται στην HTML, το CSS, το DOM και την JavaScript.

4.2 – Χαρακτηριστικά

Σύμφωνα με τα [1][2][3][4][6] συνοπτικά τα νέα στοιχεία της HTML5 είναι τα εξής:

- Semantic Elements
- Video and Audio Support
- Input Types
- Form Attributes / Form Elements
- Canvas
- Drag and Drop Function
- Local Storage

4.3 – Semantic Elements

Η HTML5 έχει εισάγει τα semantic elements (σημασιολογικά στοιχεία). Αυτά αποτελούν στοιχεία τα οποία έχουν κάποια σημασία τόσο για τον browser όσο και για τον προγραμματιστή.

Τα semantic elements αποτελούν μέρος της ευρύτερης οικογένειας του σημασιολογικού ιστού (semantic web), μία προσπάθεια δηλαδή που γίνεται να αναδιαμορφωθεί όλος ο ιστός. Σκοπός δηλαδή είναι το κάθε τι στον ιστό να αποκτήσει νόημα το οποίο θα είναι κατανοητό τόσο από το χρήστη όσο και από τους browsers και τις μηχανές αναζήτησης. Απώτερος στόχος είναι ένας ιστός πιο φιλικός στο χρήστη και πιο κοντά στον ανθρώπινο τρόπο σκέψης και λειτουργίας. Για παράδειγμα εάν μέχρι τώρα για να βρούμε τα δρομολόγια του ΚΤΕΛ από Αθήνα για Πάτρα κάναμε αναζήτηση με τους όρους «ΚΤΕΛ Αθήνα Πάτρα» και στη συνέχεια αναζητούσαμε την πληροφορία που μας ενδιαφέρει, με το σημασιολογικό ιστό θα μπορούμε να στέλνουμε την εξής επερώτηση στη μηχανή αναζήτησης, «τι ώρα έχει λεωφορείο από Αθήνα για Πάτρα το απόγευμα», και να μας επιστρέφονται τα αποτελέσματα που επιθυμούμε.

Όσον αφορά τα semantic elements της HTML5 τώρα, ενώ μέχρι τώρα όλα τα στοιχεία περικλείονταν σε divs χωρίς να έχουμε γνώση του περιεχομένου τους, πλέον έχουν προστεθεί τα **nav** (δηλώνει το navigation menu της σελίδας), **header** (η κεφαλίδα της σελίδας), **footer** (η κάτω κεφαλίδα της σελίδας), **section** (δηλώνει ένα κομμάτι πληροφορίας), **article** (κομμάτι πληροφορίας), **aside** (επιπρόσθετη πληροφορία – sidebar), **figure** και **figcaption** (διαγράμματα, φωτογραφίες κ.λ.π), **time** (δηλώνει την ημερομηνία) και **mark** (χρωματισμένο κείμενο).


Τα semantic elements βοηθούν στην καλύτερη δόμηση της πληροφορίας, κάνουν τις ιστοσελίδες πιο εύχρηστες στην διαχείρισή τους και στην ανάκτηση των πληροφοριών που επιθυμούμε. [1][2][3]

4.3.1 – Semantic Elements, Υποστήριξη στους Browsers

Όπως θα δούμε στη συνέχεια τα semantic elements, υποστηρίζονται από όλους τους σύγχρονους browsers. Οι όποιες διαφορές ανάμεσά τους έγκεινται στην εμφάνιση των στοιχείων και συγκεκριμένα στο μέγεθός τους και στις αποστάσεις τους. Ανάλογα με τον περιηγητή χρειάζονται διαφορετικά μεγέθη των στοιχείων και διαφορετικές αποστάσεις μεταξύ τους. Ενδεικτικά φαίνονται οι διαφορές στους Firefox και IE.

HTML5

[Αρχική Σελίδα](#) | [Video / Audio](#) | [Input Types](#) | [Form Elements](#) | [Form Attributes](#) | [Canvas](#) | [Drag And Drop](#) | [Local Storage](#)



Η HTML5 είναι η τελευταία έκδοση της HTML (Hyper Text Markup Language) από το W3C. Μέχρι αυτή τη στιγμή βρίσκεται στο στάδιο της ανάπτυξης.

Όλοι οι μεγάλοι browsers υποστηρίζουν την HTML5, σε διαφορετικό βαθμό ο καθένας. Η HTML5 έρχεται με πολλά νέα στοιχεία και προσθήκες.

Αυτή είναι μία πειραματική σελίδα για να δούμε αυτά τα στοιχεία και την συμβατότητά τους με τους διάφορους browsers.

Νέα χαρακτηριστικά της HTML5

- Νέο Doctype
- Νέα εισαγωγή Scripts
- Video / Audio
- Canvas
- Form Elements
- Form Attributes
- Input Types
- Semantic / Structural Elements
- Drag n Drop
- Local Storage

Firefox

Επεξήγηση

Η συγκεκριμένη σελίδα είναι υλοποιημένη με χαρακτηριστικά της HTML5. Συγκεκριμένα χρησιμοποιήθηκαν τα νέα Semantic Elements (section, nav, header, figure, figcaption, article, aside, footer).

Semantic element είναι ένα δομικό στοιχείο, το οποίο σημαίνει κάτι, δηλαδή η λειτουργία του είναι ξεκάθαρη τόσο στον προγραμματιστή, όσο και στον browser.

Για παράδειγμα το element nav, δηλώνει ξεκάθαρα ότι περιέχει το navigation menu της σελίδας.

Αντίθετα ένα div δεν καθορίζει κάποιο νόημα στο περιεχόμενό του. Μπορεί να περιέχει το οτιδήποτε.

Αρλετή Ευαγγελία, Φούντα Κατερίνα - 2013

Εικόνα 4. 1: Semantic elements σε Firefox

HTML5

[Αρχική Σελίδα](#) | [Video / Audio](#) | [Input Types](#) | [Form Elements](#) | [Form Attributes](#) | [Canvas](#) | [Drag And Drop](#) | [Local Storage](#)



Η HTML5 είναι η τελευταία έκδοση της HTML (Hyper Text Markup Language) από το W3C. Μέχρι αυτή τη στιγμή βρίσκεται στο στάδιο της ανάπτυξης.

Όλοι οι μεγάλοι browsers υποστηρίζουν την HTML5, σε διαφορετικό βαθμό ο καθένας. Η HTML5 έρχεται με πολλά νέα στοιχεία και προσθήκες.

Αυτή είναι μία πειραματική σελίδα για να δούμε αυτά τα στοιχεία και την συμβατότητά τους με τους διάφορους browsers.

Νέα χαρακτηριστικά της HTML5

- Νέο Doctype
- Νέα εισαγωγή Scripts
- Video / Audio
- Canvas
- Form Elements
- Form Attributes
- Input Types
- Semantic / Structural Elements
- Drag n Drop
- Local Storage

IE

Επεξήγηση

Η συγκεκριμένη σελίδα είναι υλοποιημένη με χαρακτηριστικά της HTML5. Συγκεκριμένα χρησιμοποιήθηκαν τα νέα Semantic Elements (section, nav, header, figure, figcaption, article, aside, footer).

Semantic element είναι ένα δομικό στοιχείο, το οποίο σημαίνει κάτι, δηλαδή η λειτουργία του είναι ξεκάθαρη τόσο στον προγραμματιστή, όσο και στον browser.

Για παράδειγμα το element nav, δηλώνει ξεκάθαρα ότι περιέχει το navigation menu της σελίδας.

Αντίθετα ένα div δεν καθορίζει κάποιο νόημα στο περιεχόμενό του. Μπορεί να περιέχει το οτιδήποτε.

Αρλετή Ευαγγελία, Φούντα Κατερίνα - 2013

Εικόνα 4. 2: Semantic elements σε IE

4.3.2 – Semantic Elements, Κώδικας

Μερικά νέα χαρακτηριστικά της HTML5 όπως φαίνεται και στην εικόνα παρακάτω είναι η διαφορετική μορφή δήλωσης του DOCTYPE, του CHARSET καθώς και ο τρόπος εισαγωγής διάφορων scripts ή αρχείων όπως τα CSS.

```
<!DOCTYPE html>
<html>
  <head>
    <!-- New charset -->
    <meta charset="UTF-8">
    <!-- New script insert - Χωρίς type attribute-->
    <link rel="stylesheet" href="css/common.css" />
    <link rel="stylesheet" href="css/home.css" />
    <title>HTML5</title>
```

Εικόνα 4. 3: HTML5 new Doctype and Charset

Στις παρακάτω εικόνες φαίνεται η δόμηση του αρχείου με τα semantic elements. Συγκεκριμένα βλέπουμε το navigation menu που περικλείεται στο nav στοιχείο, ένα sidebar που περικλείεται στο aside στοιχείο και τέλος ένα article στοιχείο το οποίο περιέχει πληροφορίες (κείμενο) καθώς και μία φωτογραφία (figure / figcaption elements).

```
<!--
Δόμηση του αρχείου με τα νέα semantic στοιχεία της HTML
section, header, nav, article, aside, footer, figure, figcaption
-->
</head>
<body>
  <!-- Όλα τα elements περικλείονται σε ένα κλασικό div -->
  <div id="main">
    <!-- Header Element-->
    <header><h2 align="center">HTML5</h2</header>

    <!-- Navigation Menu -->
    <nav>
      <ul>
        <li><a href="index.html">Αρχική Σελίδα</a></li>
        <li><a href="video_audio.html">Video / Audio</a></li>
        <li><a href="input_types.html">Input Types</a></li>
        <li><a href="form_elements.html">Form Elements</a></li>
        <li><a href="form_attributes.html">Form Attributes</a></li>
        <li><a href="canvas.html">Canvas</a></li>
        <li><a href="dragndrop.html">Drag And Drop</a></li>
        <li><a href="local_storage.html">Local Storage</a></li>
      </ul>
    </nav>
```

Εικόνα 4. 4: Semantic elements code

```

<!-- Aside element -->
<aside id="html5_new">
  <h3 align="center">Νέα χαρακτηριστικά της HTML5</h3>
  <ul id="sidebar">
    <li>Νέο Doctype</li>
    <li>Νέα εισαγωγή Scripts</li>
    <li>Video / Audio</li>
    <li>Canvas</li>
    <li>Form Elements</li>
    <li>Form Attributes</li>
    <li>Input Types</li>
    <li>Semantic / Structural Elements</li>
    <li>Drag n Drop</li>
    <li>Local Storage</li>
  </ul>
</aside>

```

Εικόνα 4. 5: Semantic elements code

```

<!-- Article - HTML5 -->
<article id="ar1">
  <!-- Figure + Figurecaption -->
  <figure>

    <figcaption><i>HTML5 Logo</i></figcaption>
  </figure>
  <p>
    Η HTML5 είναι η τελευταία έκδοση της HTML (Hyper Text Markup Language) από το W3C.
    Μέχρι αυτή τη στιγμή βρίσκεται στο στάδιο της ανάπτυξης.
    </br></br>Όλοι οι μεγάλοι browsers υποστηρίζουν την HTML5,
    σε διαφορετικό βαθμό ο καθένας.
    Η HTML5 έρχεται με πολλά νέα στοιχεία και προσθήκες.
    </br></br>Αυτή είναι μία πειραματική σελίδα για να δούμε
    αυτά τα στοιχεία και την συμβατότητά τους
    με τους διάφορους browsers.
  </p>
</article>

```

Εικόνα 4. 6: Semantic elements code

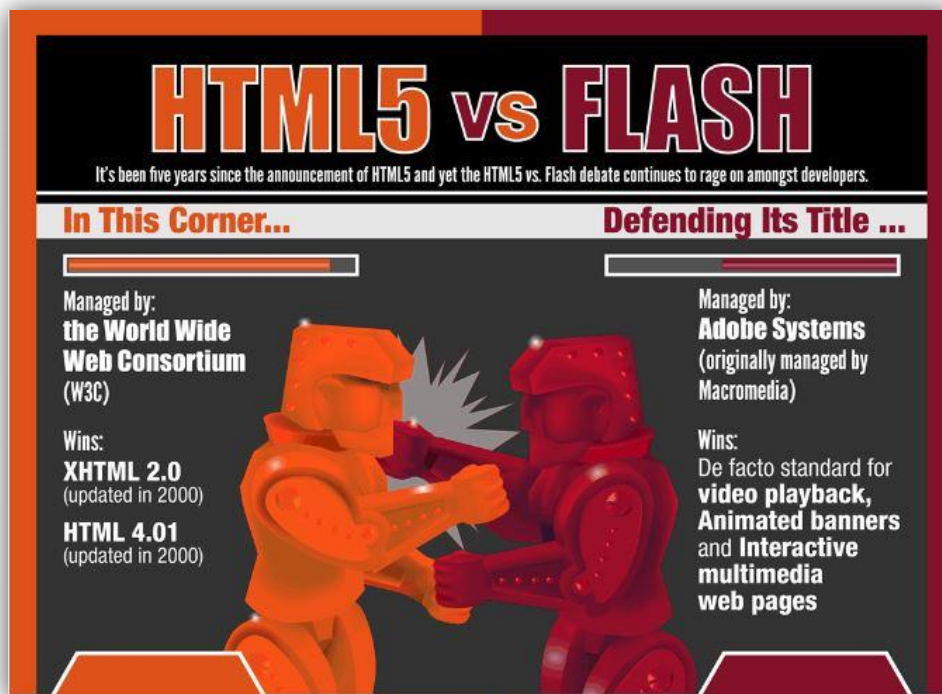
4.3 – Video & Audio

Με αυτά τα νέα στοιχεία (video / audio) καθορίζεται ένας κοινός τρόπος προβολής αρχείων video και ήχου. Έτσι δεν είναι αναγκαία η ύπαρξη άλλων, τρίτων plug-ins όπως είναι για παράδειγμα το Adobe Flash Player.

Μέχρι στιγμής σχεδόν όλα τα video στις ιστοσελίδες ήταν της μορφής του flash player. Με την HTML5 αυτή η κυριαρχία του flash έρχεται να κλονιστεί, αφού προσφέρονται αρκετά πλεονεκτήματα από τη native χρήση του video της HTML5. [1][2][3][7]

4.3.1 - HTML5 Video vs Flash Video

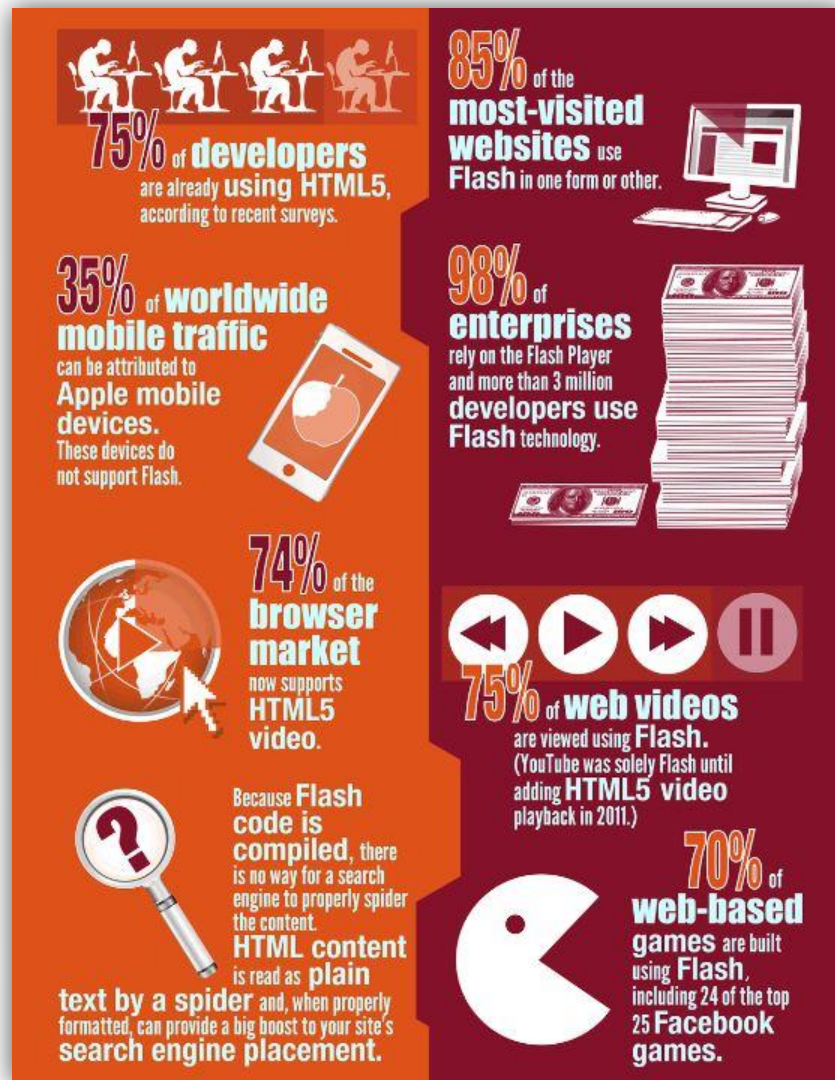
Μέχρι τώρα σχεδόν το αποκλειστικό μερίδιο αγοράς στην παρουσία video στις ιστοσελίδες το είχε το Flash Player της Adobe. Πλέον όμως όλο και περισσότεροι developers αρχίζουν να σχεδιάζουν τις ιστοσελίδες χρησιμοποιώντας το video της HTML5.



Εικόνα 4. 7: HTML5 Video vs Flash Video [8]

Ένα από τα σημαντικότερα πλεονεκτήματα της HTML5 έναντι του Adobe Flash Player είναι ότι μπορεί να ανακαλυφθεί από τις μηχανές αναζήτησης. Αντίθετα το Flash Player γίνεται compile και ενσωματώνεται μέσα στις σελίδες. Αποτέλεσμα αυτού είναι να μην μπορεί να προσφέρει τίποτα στο SEO των σελίδων.

Αντίθετα με την HTML5 οι μηχανές αναζήτησης ανακαλύπτουν τα video που υπάρχουν, αφού τα διαβάζουν ως απλό κείμενο και έτσι μπορούν να αυξήσουν την ποιότητα της σελίδας στον ιστό.



Εικόνα 4. 8: HTML5 vs Flash [8]

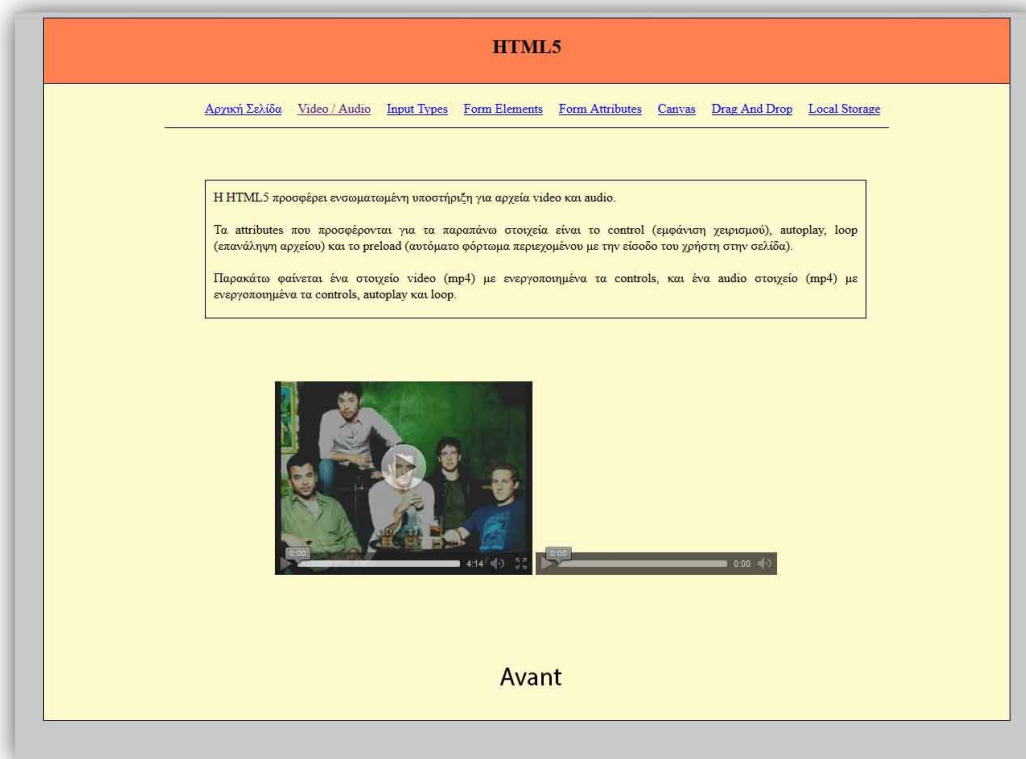
Άλλο πλεονέκτημα της HTML5 είναι ότι το Flash Player ανήκει σε μία μόνο εταιρεία (Adobe), ενώ αυτή αποτελείται από πρότυπα ανοικτά σε πειραματισμό και τροποποιήσεις προς το καλύτερο (open standards). [8]



Εικόνα 4. 9: HTML5 vs Flash [8]

4.3.2 – Video & Audio, Υποστήριξη στους Browsers

Τα video και audio elements δεν υποστηρίζονται ακόμη από όλους τους browsers. Συγκεκριμένα δεν υποστηρίζεται από τον Safari. Επίσης ο Opera δεν υποστηρίζει τη μορφή των αρχείων (mp4). Παρακάτω φαίνεται ενδεικτικά ο Avant browser καθώς και ο Opera.



Εικόνα 4. 10: Avant browser video / audio support



Εικόνα 4. 11: Opera mp4 support

4.3.3 – Video & Audio, Κώδικας

Για την εισαγωγή ενός αρχείου βίντεο ή ήχου, χρησιμοποιούνται τα στοιχεία video και audio, όπως φαίνεται παρακάτω στον κώδικα. Σε περίπτωση που δεν υποστηρίζονται αυτά, τότε εμφανίζεται το ανάλογο μήνυμα.

```

<!-- Εισαγωγή αρχείων video & audio -->
<section>
  <video width="320" height="240" controls>
    <source src="files/shattered.mp4" type="video/mp4">
    Your browser does not support the video tag.
  </video>

  <audio src="files/shattered.mp4" controls loop preload>
    <p>Your browser does not support the audio element </p>
  </audio>
</section>

```

Εικόνα 4. 12: HTML5 video / audio code

4.4 – Input Types

Πλέον με την HTML5 παρέχονται νέες μορφές εισόδου δεδομένων (input types). Αυτό οδηγεί σε καλύτερο έλεγχο της εισόδου δεδομένων καθώς και επικύρωσής τους. [1][2][3][4]

Αυτοί οι τύποι αναλυτικά είναι οι εξής:

- **Color**: προσφέρει τη δυνατότητα επιλογής χρώματος από μία παλέτα χρωμάτων
- **Date**: δίνεται η δυνατότητα επιλογής ημερομηνίας
- **Datetime**: οι χρήστες μπορούν να επιλέξουν ημερομηνία και ώρα χωρίς να λαμβάνεται υπόψη η ζώνη ώρας τους
- **Datetime-local**: παρόμοια επιλέγεται η ημερομηνία και η ώρα, αυτή τη φορά όμως λαμβάνεται υπόψη η ζώνη ώρας που βρίσκεται ο χρήστης
- **Month**: επιλέγεται ο μήνας και η χρονολογία
- **Number**: εισαγωγή μόνο αριθμού. Δέχεται επιπλέον attributes όπως ελάχιστη (min) / μέγιστη (max) τιμή, αρχική τιμή (value), και βηματισμός (step).
- **Range**: επιλογή αριθμού από ένα εύρος. Δέχεται τα attributes min, max, value, step.
- **Search**: συμπεριφέρεται ως ένα απλό στοιχείο εισόδου κειμένου, αλλά χρησιμοποιείται για να δέχεται όρους αναζήτησης.
- **Tel**: στοιχείο για είσοδο τηλεφωνικού αριθμού
- **Time**: επιτρέπεται η εισαγωγή ώρας
- **Url**: δίνεται η δυνατότητα εισαγωγής ενός url
- **Email**: στοιχείο το οποίο δέχεται είσοδο μόνο στη μορφή ενός email
- **Week**: επιλογή εβδομάδας και χρόνου.

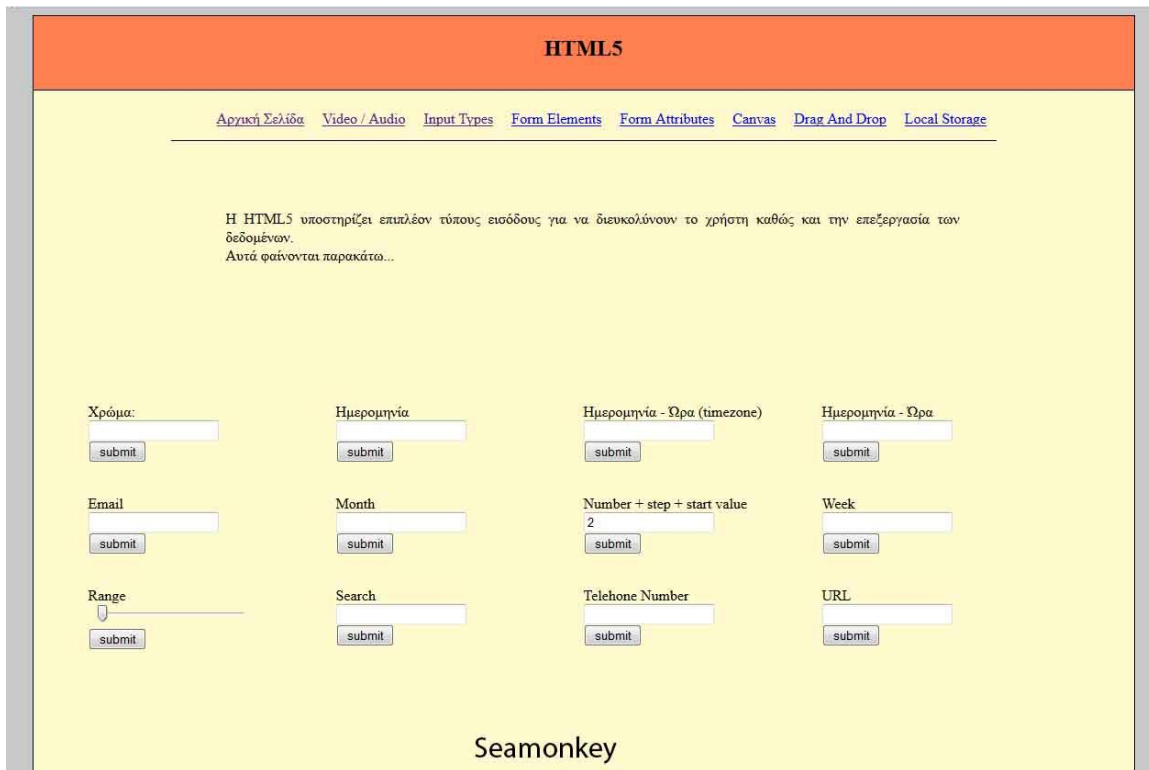
4.4.1 – Input Types, Υποστήριξη στους Browsers

Οι παραπάνω τύποι δεν υποστηρίζονται ακόμη 100% από τους browsers. Ο καθένας υποστηρίζει ένα μέρος (μικρό η μεγάλο) από αυτούς. Αναλυτικότερα αυτό φαίνεται στις παρακάτω εικόνες.

Στον Firefox βλέπουμε ότι τα μόνα στοιχεία που υποστηρίζονται είναι το range, το email και το URL στα οποία εάν δεν δοθεί σωστή είσοδος τότε δεν επικυρώνονται και δεν αποστέλλονται τα δεδομένα. Όμοια συμπεριφορά έχει και ο Seamonkey.

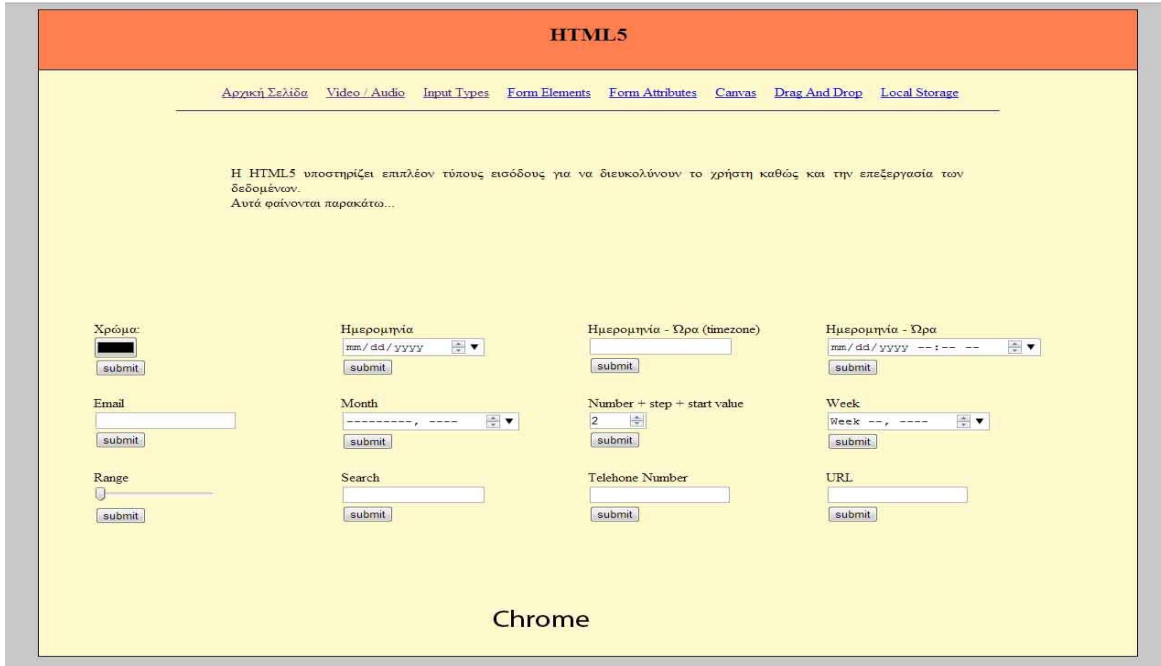


Εικόνα 4. 13: Firefox input type support



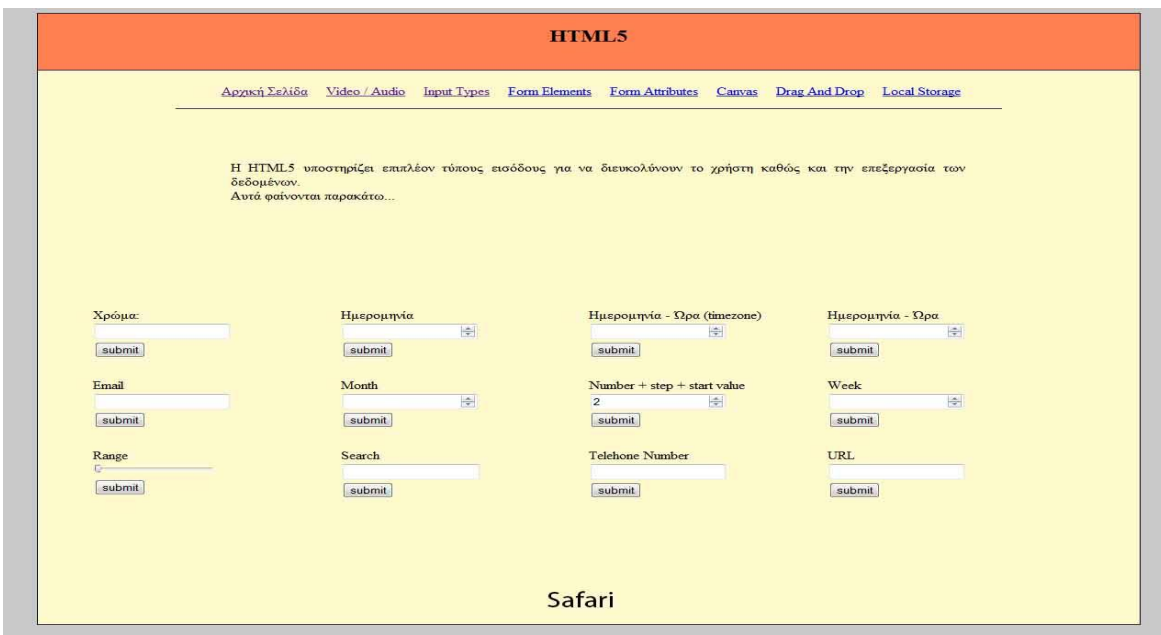
Εικόνα 4. 14: Seamonkey input type support

Ο Chrome αντίθετα έχει πολύ μεγαλύτερο ποσοστό υποστήριξης καθώς οι μόνοι τύποι που δεν υποστηρίζει είναι της ημερομηνίας / ώρας δίχως την ώρα ζώνης του χρήστη και του τηλεφωνικού αριθμού.



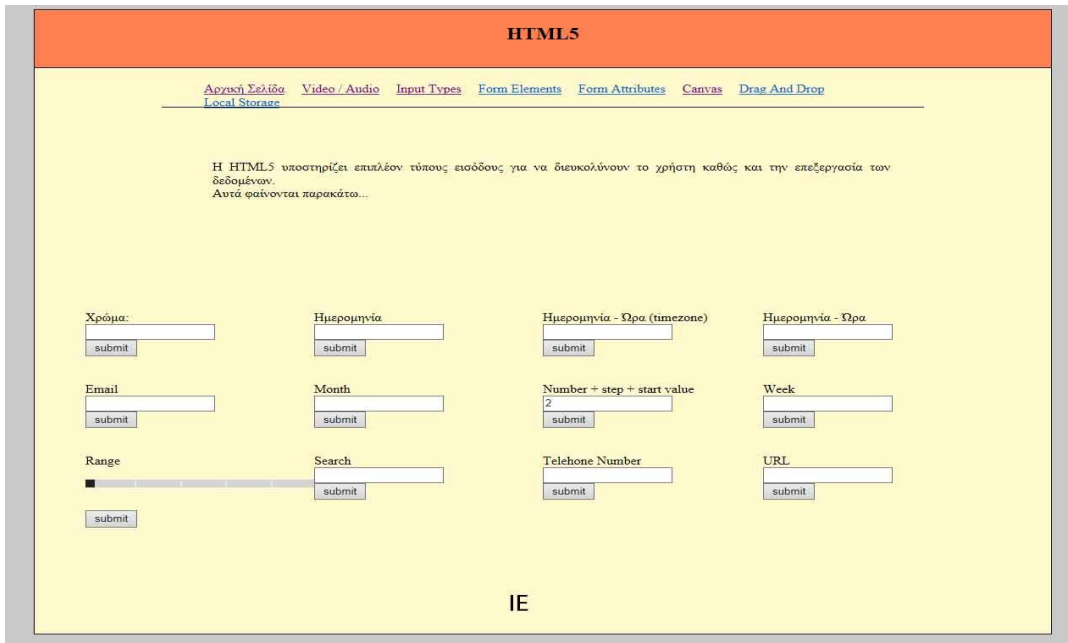
Εικόνα 4. 15: Chrome input type support

Ο Safari δεν έχει καλά ποσοστά υποστήριξης καθώς δεν υποστηρίζει αρκετούς τύπους όπως χρώμα, email, URL, τηλεφωνικός αριθμός.



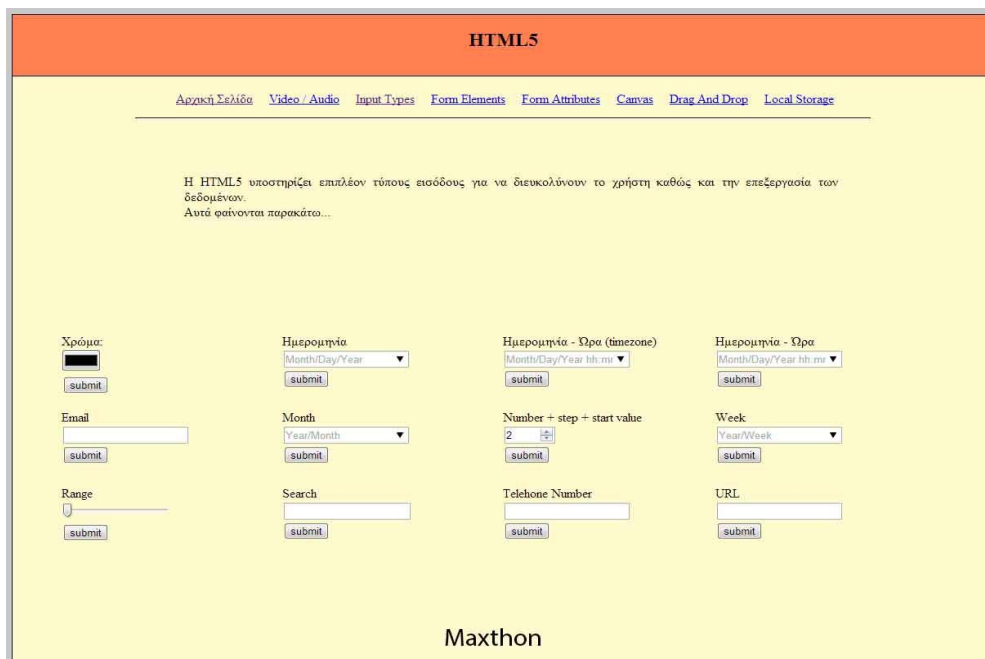
Εικόνα 4. 16: Safari input type support

Ο ΙΕ υποστηρίζει μόνο τις μορφές email, number, range και url.



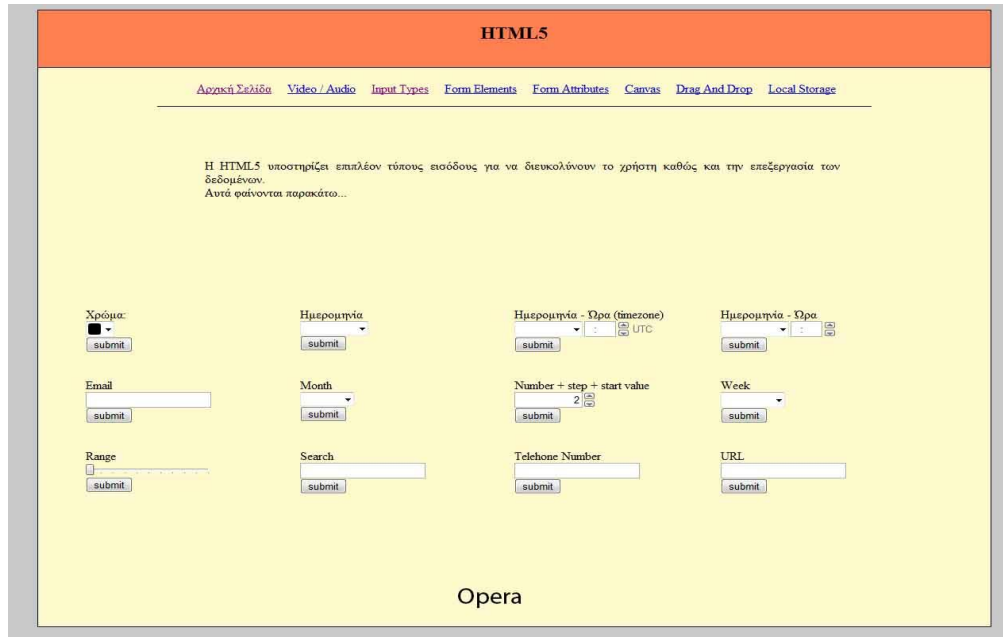
Εικόνα 4. 17: IE input type support

Ο Maxthon επιτυγχάνει από τα μεγαλύτερα ποσοστά επιτυχίας, καθώς υποστηρίζει όλες τις μορφές εισόδου.



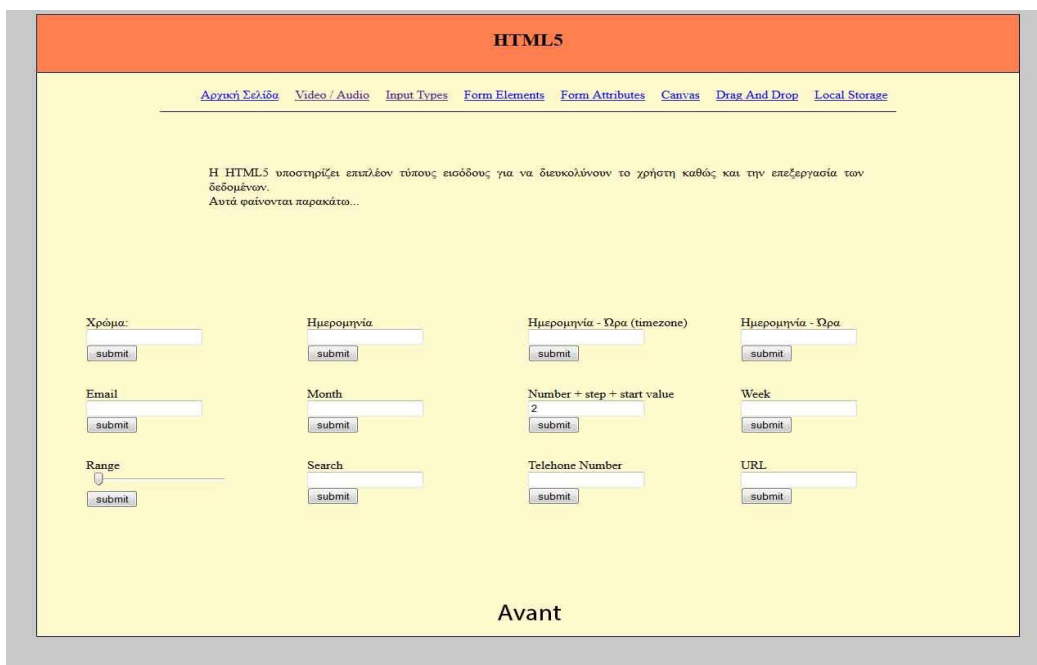
Εικόνα 4. 18: Maxthon input type support

Αρκετά μεγάλα ποσοστά επιτυχίας επιτυγχάνει και ο Opera, καθώς οι μόνοι τύποι που δεν υποστηρίζει είναι της αναζήτησης και του τηλεφωνικού αριθμού.



Εικόνα 4. 19: Opera input type support

Ο Avant το μόνο στοιχείο που δεν υποστηρίζει, είναι αυτό της ημερομηνίας / ώρας χωρίς να λαμβάνεται υπόψη η ζώνη ώρας του χρήστη.



Εικόνα 4. 20: Avant input type support

4.4 – Form Elements

Με την HTML5 παρέχονται νέα στοιχεία φορμών (form elements). Αυτά είναι τα `datalist`, `keygen` και `output`. [1][2][3][4]

Το **datalist** στοιχείο παρέχει ορισμένες προκαθορισμένες τιμές για μία είσοδο. Εμφανίζεται με μία dropdown μορφή.

Το **keygen** στοιχείο δημιουργήθηκε για να συμβάλλει στην ταυτοποίηση των χρηστών στην είσοδο ενός συστήματος. Δημιουργεί έναν κωδικό με βάση μία τιμή που θα το δοθεί. Αυτός ο κωδικός μπορεί να δημιουργηθεί με δύο διαφορετικούς τρόπους (δύο διαφορετικές κωδικοποιήσεις).

Το **output** στοιχείο, εμφανίζει το αποτέλεσμα μίας αριθμητικής πράξης.

4.4.1 – Form Elements, Υποστήριξη στους Browsers

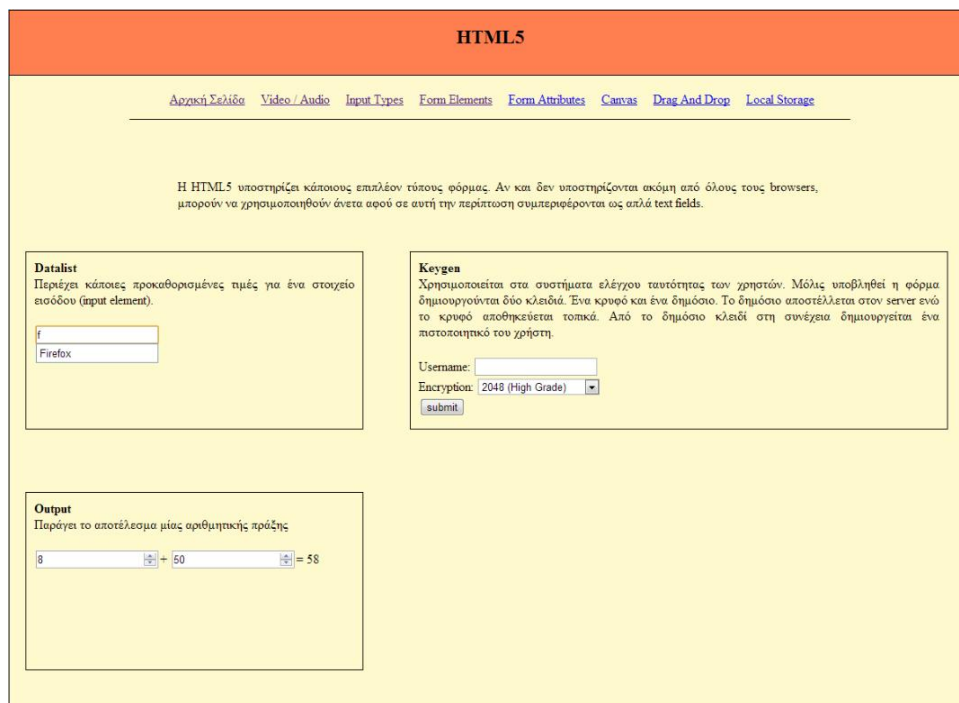
Όπως και με τα `input types`, δεν υπάρχει 100% υποστήριξη των στοιχείων από όλους τους browsers. Αναλυτικότερα αυτό φαίνεται στις παρακάτω εικόνες. Και τους τρεις τύπους τους υποστηρίζουν οι Firefox, Seamonkey, Chrome, Opera και Avant browser.



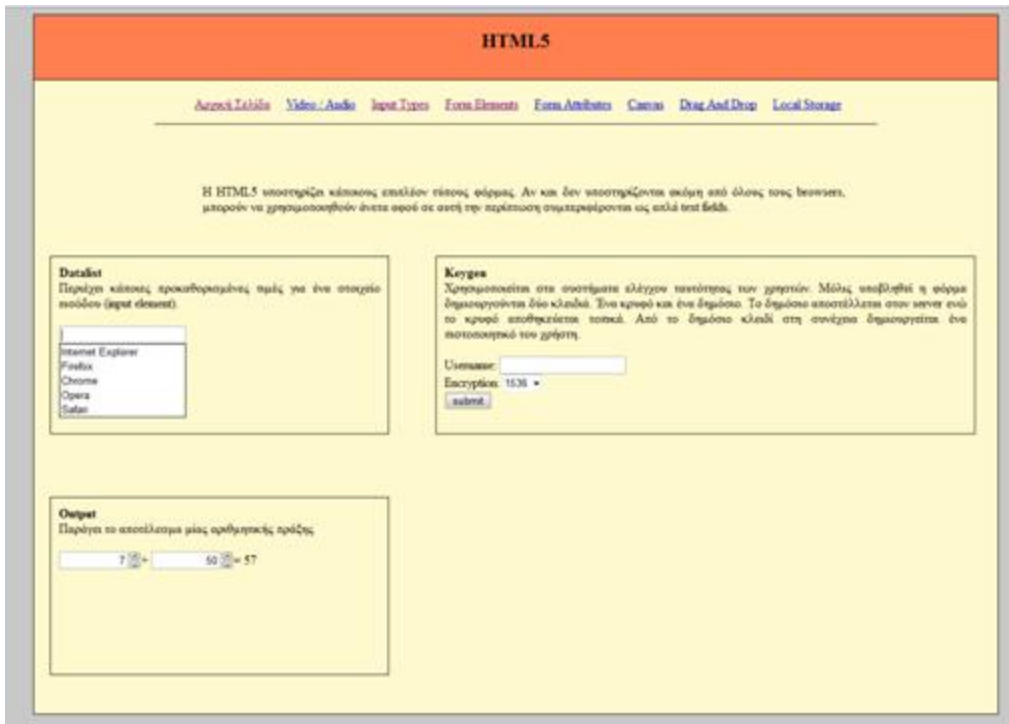
Εικόνα 4. 21: Firefox form elements support



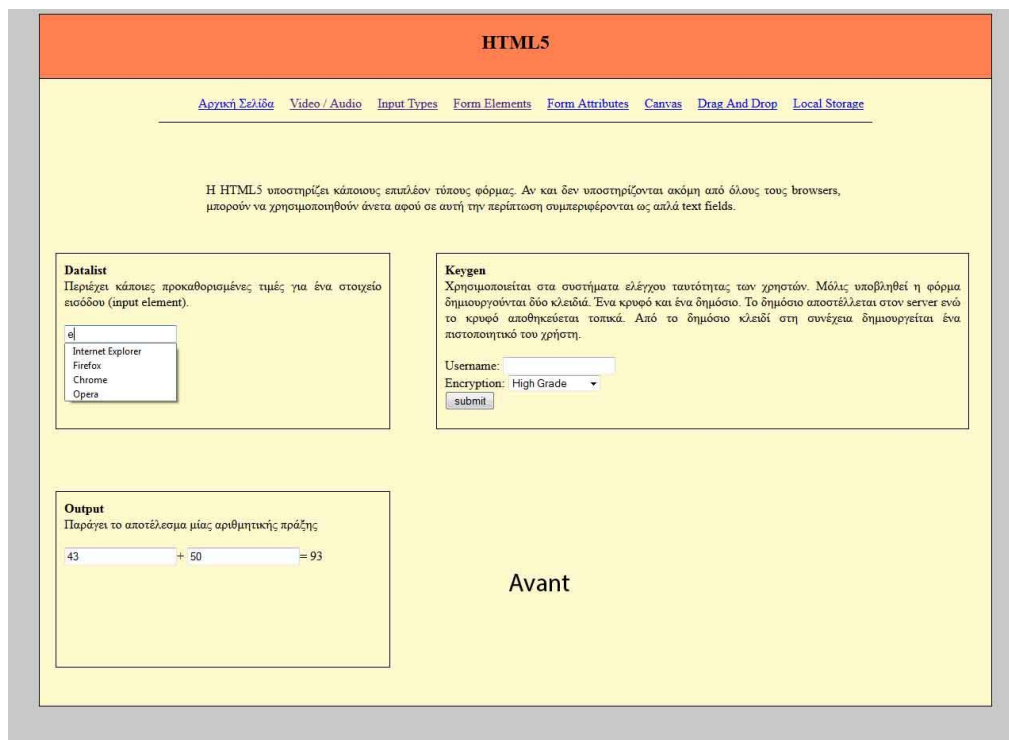
Εικόνα 4. 22: Seamonkey form elements support



Εικόνα 4. 23: Chrome form elements support



Εικόνα 4. 24: Opera form elements support



Εικόνα 4. 25: Avant form elements support

Αντίθετα, ο Safari και ο Maxthon δεν υποστηρίζουν το datalist element,

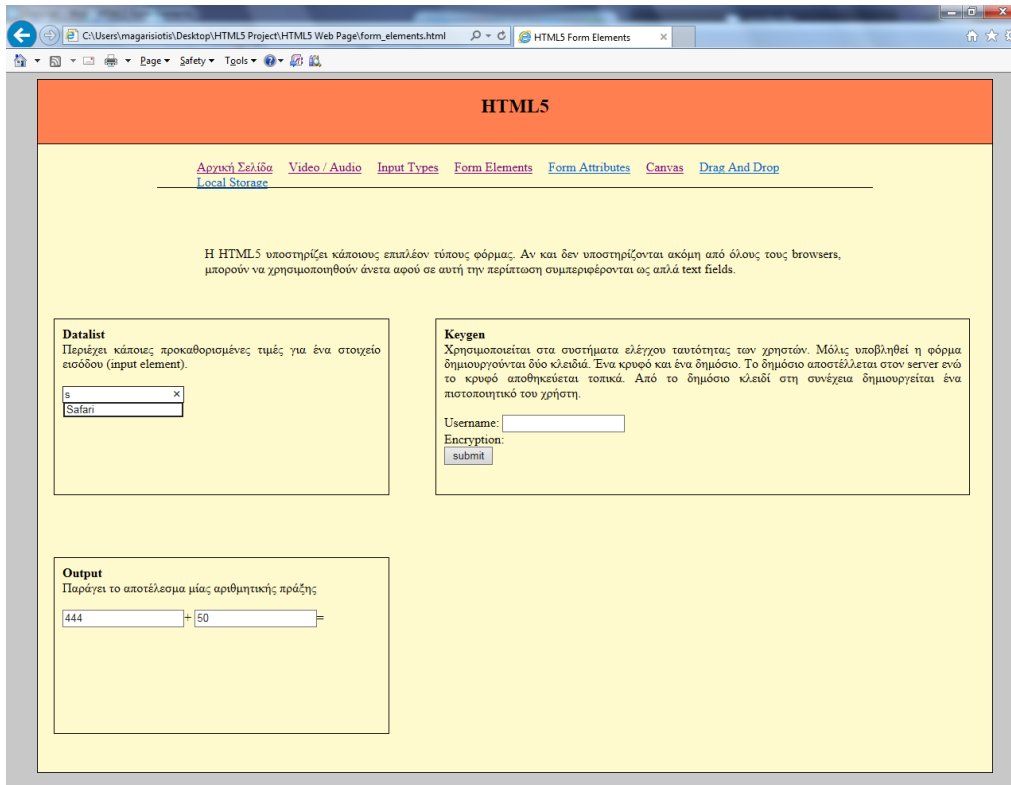


Εικόνα 4. 26: Maxthon form element support



Εικόνα 4. 27: Safari form element support

Ενώ ο IE δεν υποστηρίζει το keygen και το output.



Εικόνα 4. 28: IE form element support

4.4.2 – Form Elements, Κώδικας

Στις παρακάτω εικόνες φαίνονται οι κώδικες για την εισαγωγή των προηγούμενων νέων form elements.

Στην αμέσως επόμενη εικόνα βλέπουμε το datalist στοιχείο. Συγκεκριμένα σε ένα input, δηλώνουμε μία λίστα με ορισμένες προκαθορισμένες τιμές. Αυτές στη συνέχεια (όπως είδαμε και στα screenshots των browsers) εμφανίζονται με μία drop down μορφή.

```

<section class="form_elements" id="datalist">
  <span class="explain"><b>Datalist</b><br/>
</span>
<br/><br/>
<input list="browsers">
<datalist id="browsers">
  <option value="Internet Explorer">
  <option value="Firefox">
  <option value="Chrome">
  <option value="Opera">
  <option value="Safari">
</datalist>
</section>

```

Εικόνα 4. 29: Datalist element

Για το keygen στοιχείο, αρκεί να δηλώσουμε τη μεταβλητή αναγνωριστικό (username) και στη συνέχεια εισάγουμε το <keygen>.

```

<form action="">
  Username: <input type="text"><br/>
  Encryption: <keygen><br/>
  <input type="submit" value="submit">
</form>
</section>

```

Εικόνα 4. 30: Keygen form element

Τέλος, για το output στοιχείο στα attributes της φόρμας (form) εισάγουμε την αριθμητική πράξη που θα εκτελεστεί (javascript μορφή) και το αποτέλεσμα εμφανίζεται μέσα στο tag <output></output>.

```

<form oninput="x.value=parseInt (a.value)+parseInt (b.value) ">
  <input type="number" id="a">+
  <input type="number" id="b" value="50">=
  <output name="x" for="a b"></output>
</form>

```

Εικόνα 4. 31: Output form element

4.5 – Form Attributes

Προηγουμένως είδαμε ότι υπάρχουν νέοι input και form τύποι. Έτσι, έχουν εισαχθεί και νέα attributes (χαρακτηριστικά) για αυτούς τους τύπους. [1][2][3][4]

Τα νέα attributes για τις φόρμες είναι:

- **Autocomplete:** αυτόματη συμπλήρωση των πεδίων.
- **Novalidate:** δεν γίνεται η επικύρωση των πεδίων σύμφωνα με τις προδιαγραφές.

Τα νέα attributes για τα input elements είναι:

- **Autocomplete**
- **Autofocus:** Ο δείκτης του ποντικιού κεντράρει στο συγκεκριμένο πεδίο μόλις φορτώσει η σελίδα.
- **Form:** δηλώνει ότι ένα στοιχείο ανήκει σε μία φόρμα ακόμα και εάν αυτό βρίσκεται έξω από αυτήν.
- **Formaction:** δηλώνει το αρχείο που θα επεξεργαστεί τη φόρμα όταν αυτή υποβληθεί. Το συγκεκριμένο attribute υπερισχύει την ενέργεια που δηλώνεται μέσα στο σώμα της φόρμας.
`<form action="file1.php">...<input type="submit" formaction="file2.php"></form>`
- **Formenctype:** καθορίζει με τι μορφή θα κωδικοποιηθούν τα δεδομένα στην υποβολή της φόρμας. Υπερισχύει της δήλωσης στο σώμα της φόρμας. (`<form enctype="">`).
`<form enctype="multipart/form-data"><input type="submit" formenctype="application/x-www-form-urlencoded"></form>`
- **Formmethod:** καθορίζει την HTTP μέθοδο που θα σταλεί η φόρμα (get / post). Παρόμοια με τα formenctype και formaction, υπερισχύει της δήλωσης στο σώμα της φόρμας.
- **Formnovalidate:** δηλώνει ότι κατά την υποβολή της φόρμας δεν θα γίνει επικύρωση των στοιχείων της. Έτσι για παράδειγμα σε ένα πεδίο email μπορεί να σταλεί μία τιμή η οποία όμως δεν αντιστοιχεί σε αποδεκτή μορφή.
- **Formtarget:** καθορίζει που θα εμφανιστεί η απάντηση του συστήματος αφού υποβληθεί μία φόρμα.
- **Height / Width:** καθορίζονται το ύψος και το πλάτος ενός input - image στοιχείου.
`<input type="image" src=".." width="50" height="50">`
- **List:** Αναφέρεται σε ένα input στοιχείο με δυνατότητα επιλογής προκαθορισμένων τιμών (datalist).
- **Min / Max:** ορίζονται ελάχιστες και μέγιστες δυνατές τιμές ενός στοιχείου.

- **Multiple:** ορίζει τη δυνατότητα να επιλεγούν πολλαπλά στοιχεία σε μία φόρμα.
- **Pattern:** δίνεται η δυνατότητα η είσοδος σε ένα στοιχείο να ακολουθεί κάποια συγκεκριμένη μορφή, π.χ. να αποτελείται από 10 αριθμούς (για τηλεφωνικό αριθμό).
- **Placeholder:** δίνεται σε ένα element, μία τιμή. Λειτουργεί ως βοήθημα για να καταλάβει ο χρήστης τι είσοδο πρέπει να εισάγει.
- **Required:** καθορίζει ότι ένα στοιχείο είναι απαραίτητο να συμπληρωθεί πριν την υποβολή της φόρμας.
- **Step:** ορίζεται το διάστημα των αριθμών που θα πρέπει να έχει η είσοδος ενός στοιχείου, π.χ. αριθμοί ανά 3 (0, 3, 6, 9...).

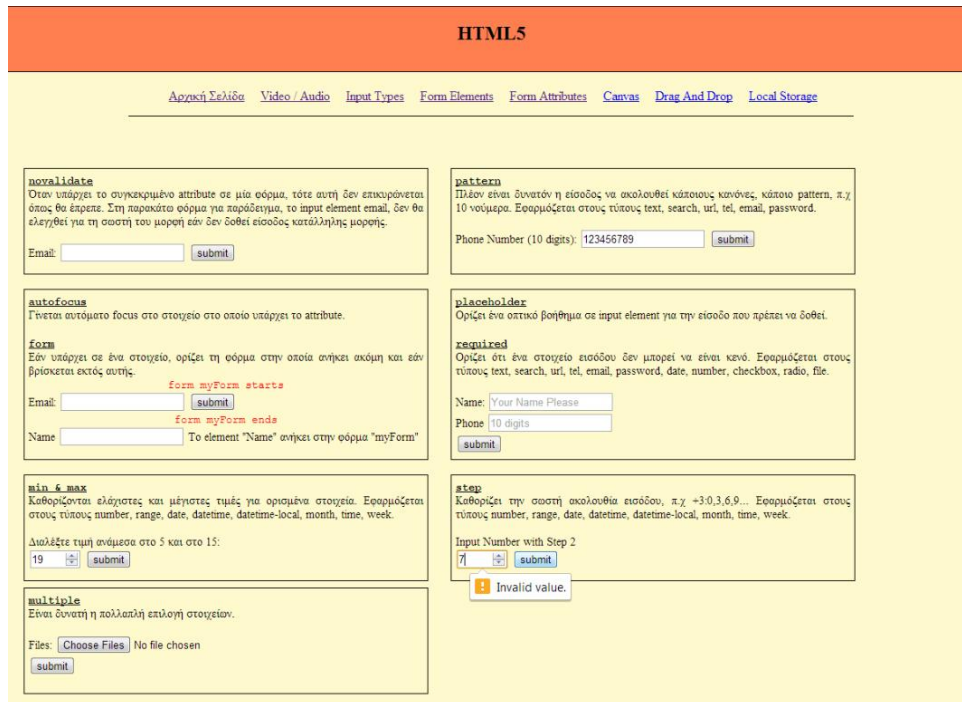
4.4.1 - Form Attributes, Υποστήριξη στους Browsers

Τα Form Attributes είναι ακόμη μία περίπτωση όπου δεν υπάρχει πλήρη υποστήριξη από όλους τους browsers. Αυτό φαίνεται καλύτερα στα παρακάτω παραδείγματα.

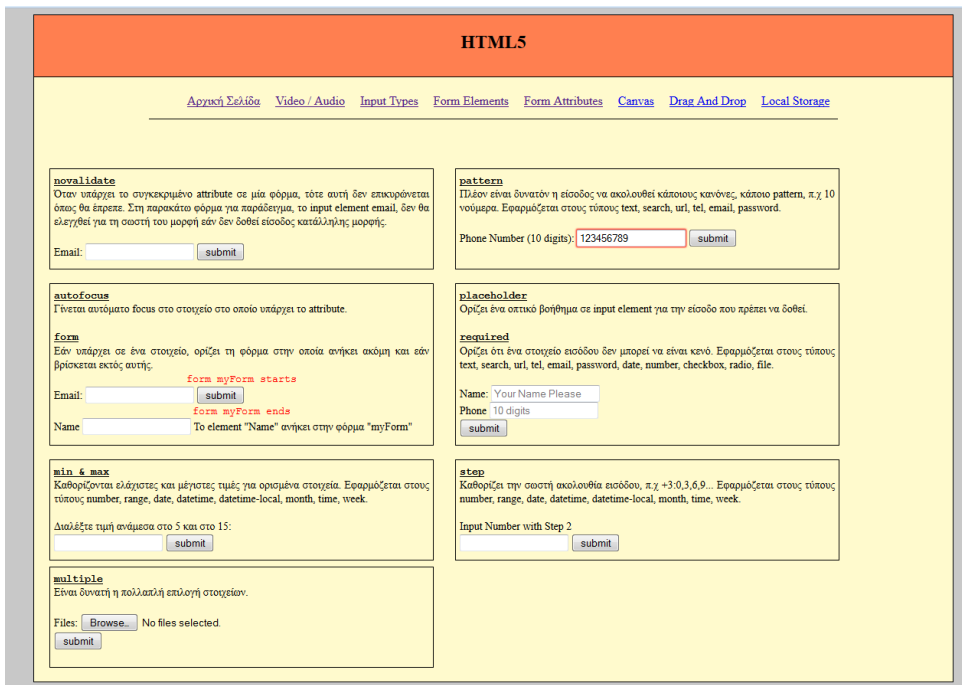
Ο chrome στις δοκιμές που κάναμε είχε καθολική επιτυχία, αφού υποστηρίζει όλα τα attributes. Στην επόμενη εικόνα φαίνεται το error handling για το min / max attribute, ενώ εμφανή είναι τα placeholders. Επίσης το novalidate attribute υποστηρίζεται κανονικά, αφού στο πεδίο του email δέχεται όλες τις τιμές.

Εικόνα 4. 32: Chrome min / max attribute

Παρακάτω φαίνεται το error handling στην περίπτωση του step, εάν δοθεί δηλαδή αριθμός έξω από την αναμενόμενη ακολουθία. Όμοια για το pattern attribute, το οποίο αναμένει 10 νούμερα σαν είσοδο.

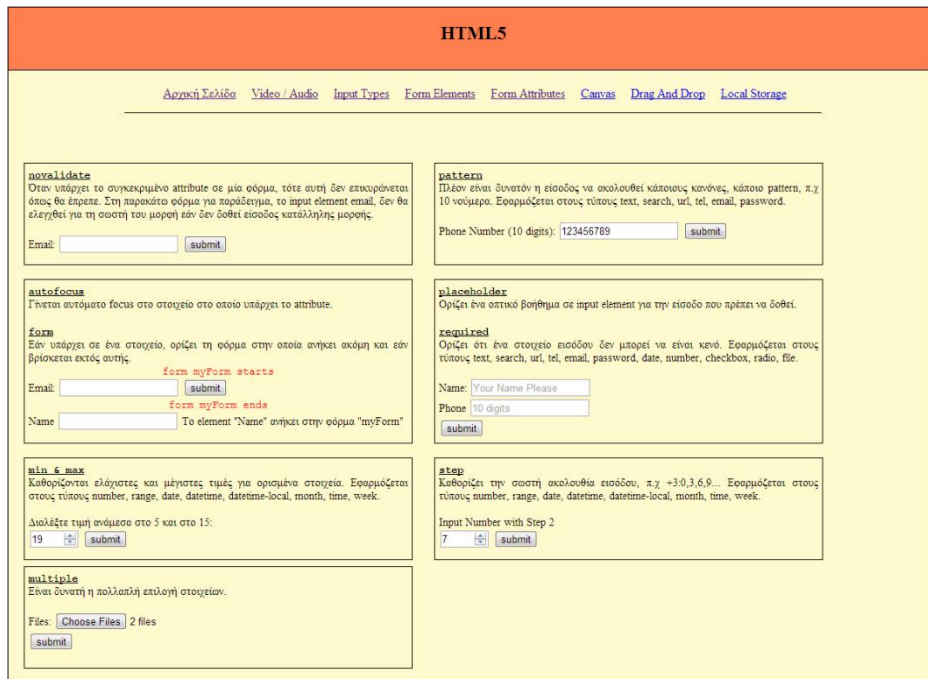


Εικόνα 4. 33: Chrome step attribute



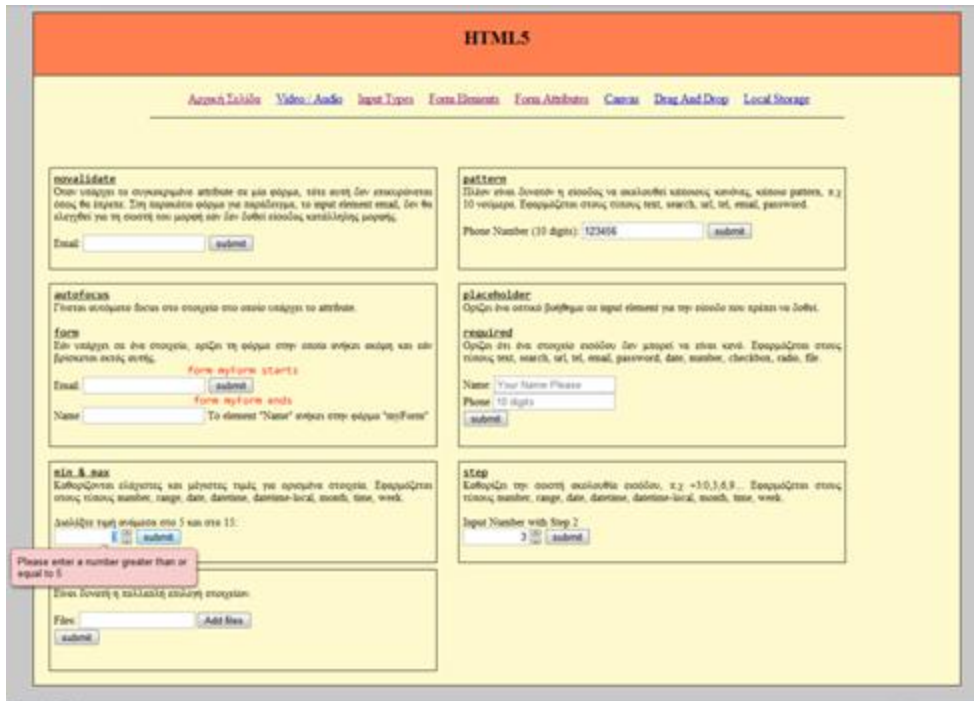
Εικόνα 4. 34: Chrome pattern attribute

Φαίνεται επίσης ότι παρέχεται υποστήριξη στο multiple attribute, αφού βλέπουμε ότι έχουν επιλεγεί 2 στοιχεία.

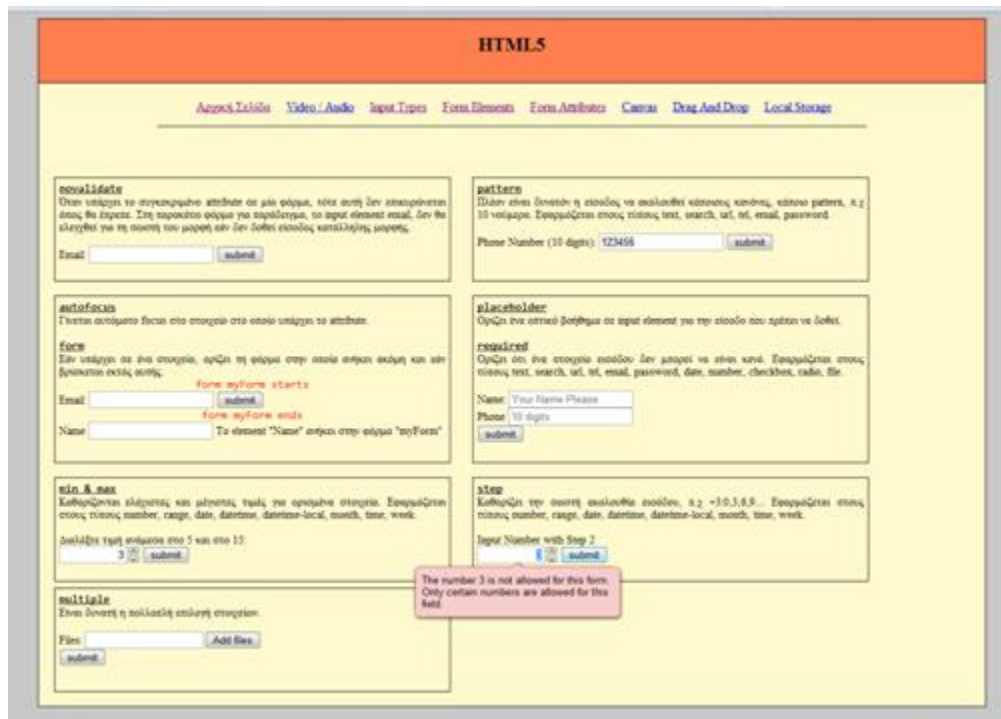


Εικόνα 4. 35: Chrome multiple attribute

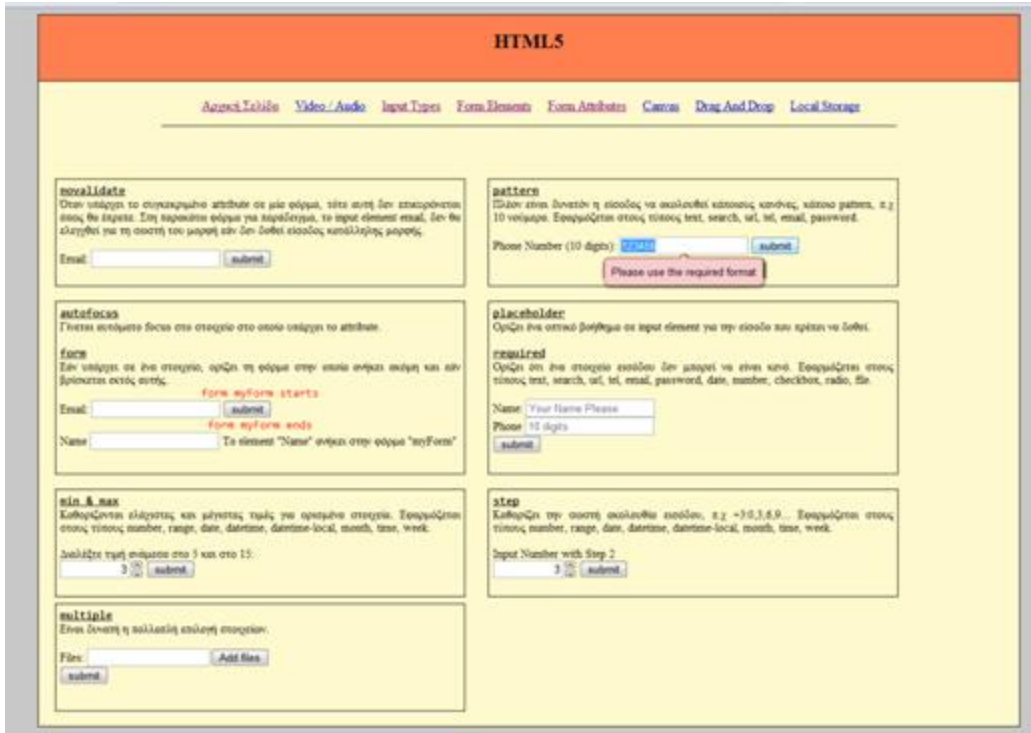
Στη συνέχεια φαίνεται η υποστήριξη που παρέχει ο Opera browser, που όπως βλέπουμε υποστηρίζει και αυτός όλα τα attributes εμφανίζοντας τα ανάλογα μηνύματα λάθους.



Εικόνα 4. 36: Opera min / max attribute

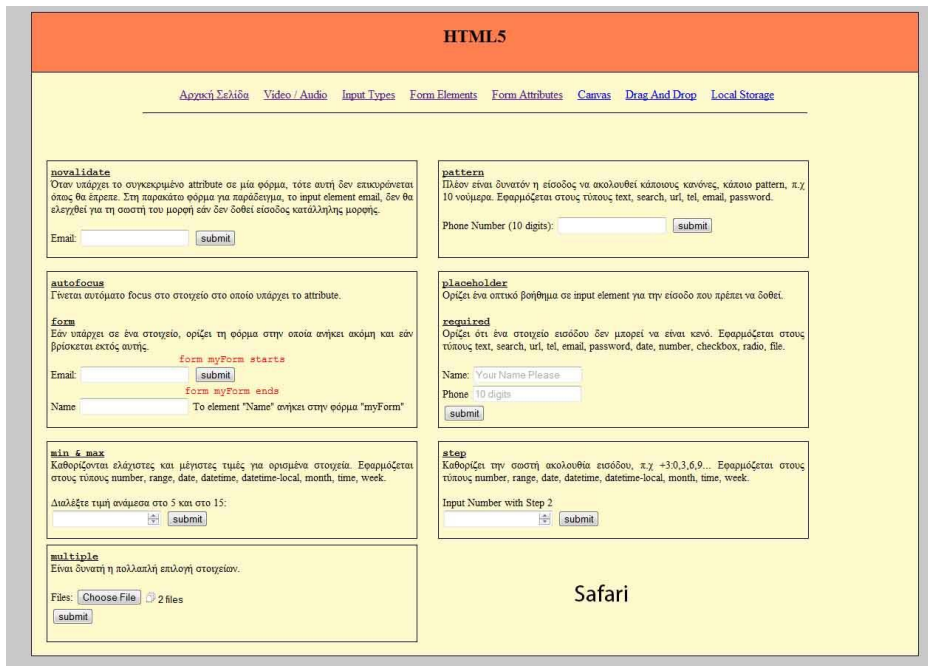


Εικόνα 4. 37: Opera step attribute



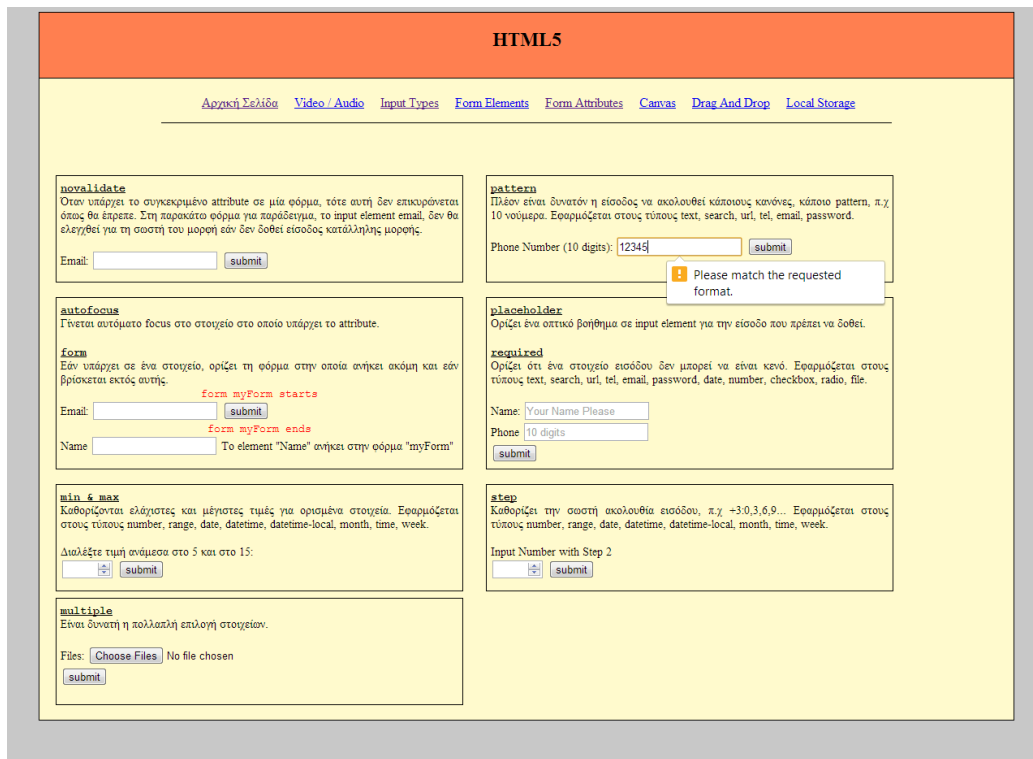
Εικόνα 4. 38: Opera pattern attribute

Ο Safari σύμφωνα με τις προδιαγραφές θα έπρεπε να υποστηρίζει τα attributes step, placeholder, multiple, min / max, form και autofocus. Από αυτά όμως στην πράξη δεν υποστηρίζονται τα min / max και step, αφού δέχονται όλες τις τιμές εισόδου.



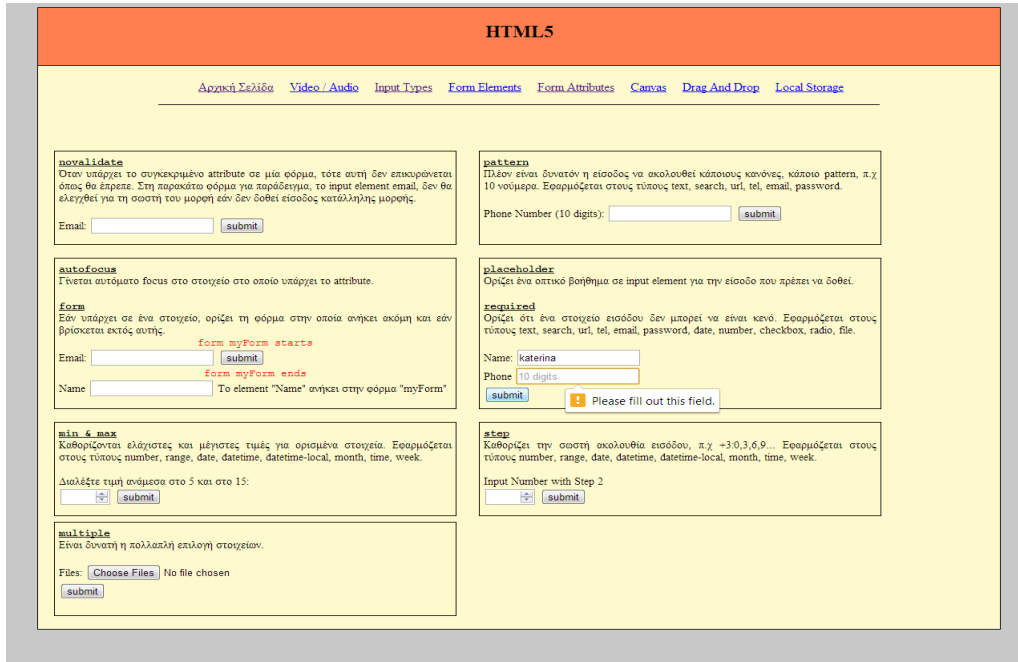
Εικόνα 4. 39: Safari form attributes support

Ο Avant browser υποστηρίζει πλήρως τα attributes τα οποία δοκιμάσαμε. Στην επόμενη εικόνα φαίνεται το error handling για το pattern του τηλεφωνικού αριθμού.



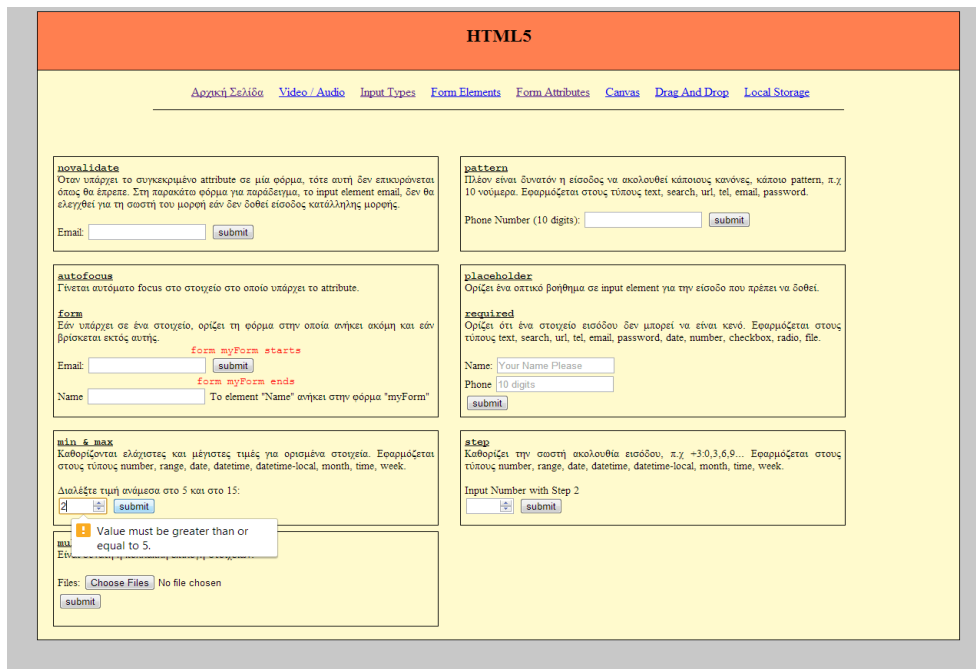
Εικόνα 4. 40: Avant pattern support

Εδώ βλέπουμε το μήνυμα λάθους εάν υποβάλλουμε τη φόρμα και δεν συμπληρώσουμε ένα απαιτούμενο πεδίο (required).

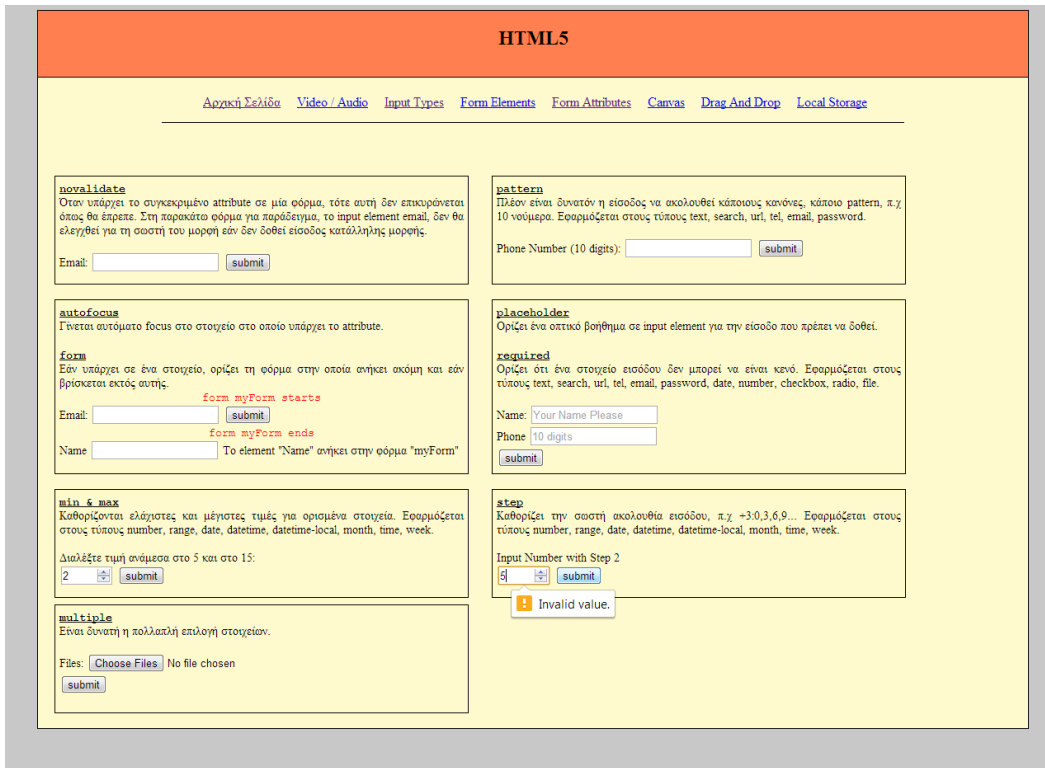


Εικόνα 4. 41: Avant required support

Στις παρακάτω εικόνες φαίνονται τα μηνύματα λάθους για τα min / max και step attributes.

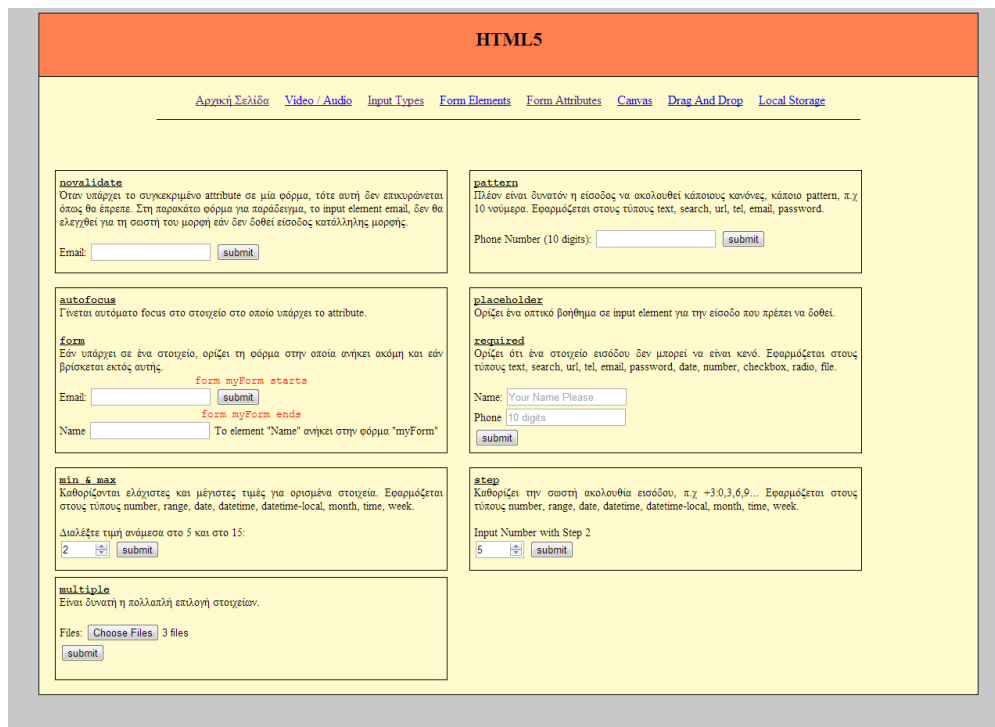


Εικόνα 4. 42: Avant min / max support



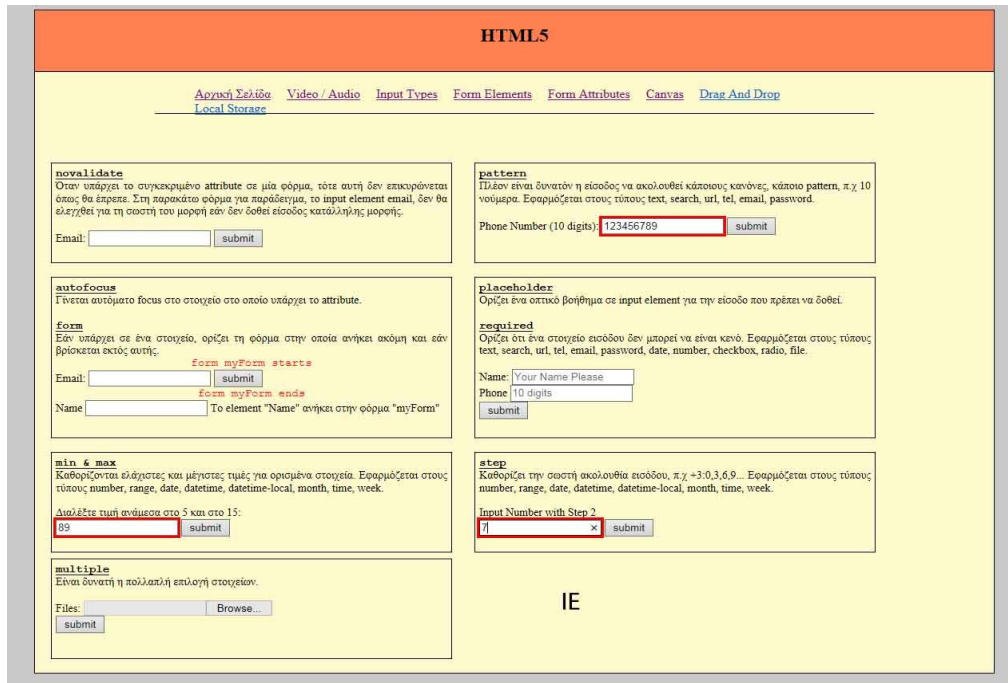
Εικόνα 4. 43: Avant Step support

Τέλος, υποστηρίζεται η πολλαπλή επιλογή αρχείων.



Εικόνα 4. 44: Avant Multiple file support

Ο Internet Explorer υποστηρίζει όλα τα attributes.



Εικόνα 4. 45: : Internet Explorer form attributes support

Ο Maxthon παρέχει και αυτός πλήρη υποστήριξη για τα attributes.



Εικόνα 4. 46: Maxthon form attributes support

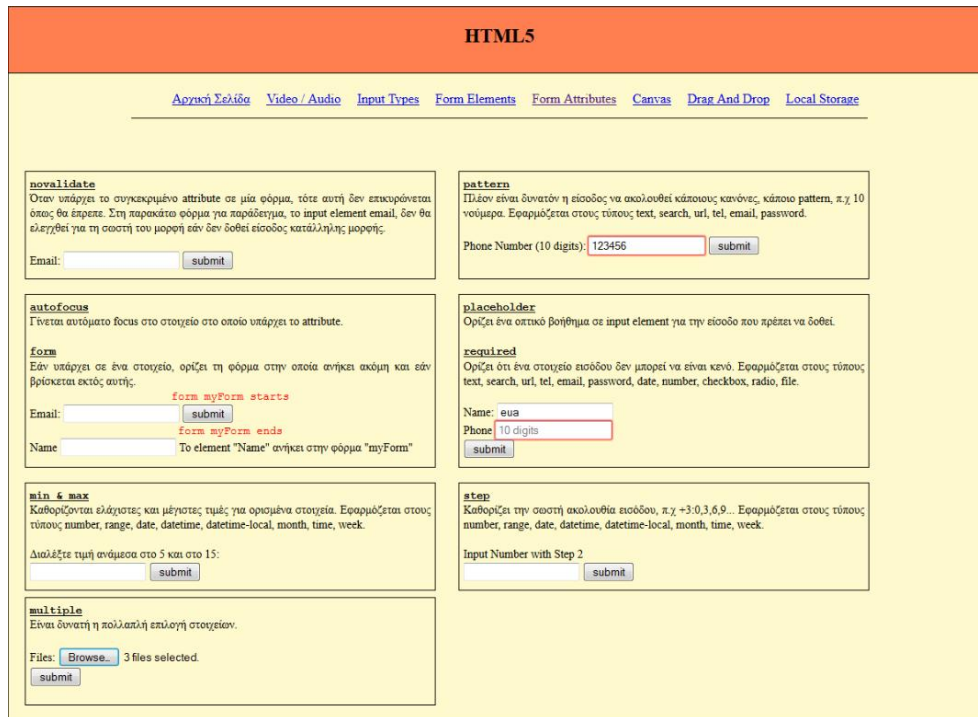
Ο Firefox παρέχει υποστήριξη στα attributes εκτός από τα min / max και step.

The screenshot shows a demo page for HTML5 form attributes. The 'pattern' attribute is highlighted, showing an input field with the value '123456' and a tooltip that says 'Please match the requested format.' The 'required' attribute is also highlighted, showing an input field with the value 'Your Name Please' and a tooltip that says 'Please fill out this field.' The 'step' attribute is highlighted, showing an input field with the value '10' and a tooltip that says 'Please fill out this field.'

Εικόνα 4. 47: Firefox pattern support

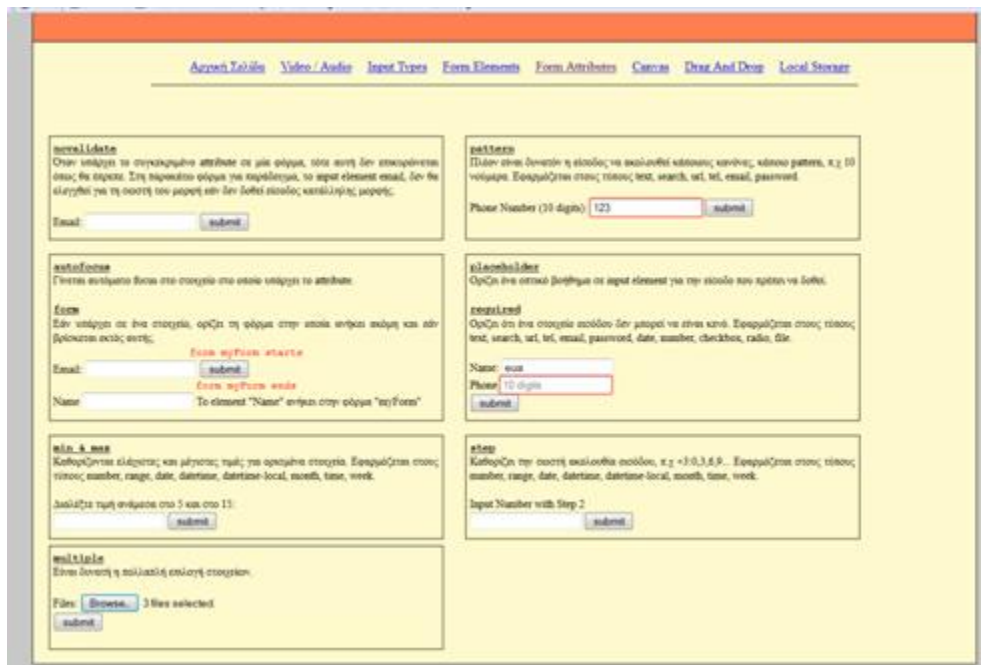
The screenshot shows a demo page for HTML5 form attributes. The 'required' attribute is highlighted, showing an input field with the value 'Your Name Please' and a tooltip that says 'Please fill out this field.' The 'step' attribute is highlighted, showing an input field with the value '10' and a tooltip that says 'Please fill out this field.'

Εικόνα 4. 48: Firefox required support



Εικόνα 4. 49: Firefox multiple support

Όμοια συμπεριφορά με τον Firefox, έχει και ο Seamonkey όπως φαίνεται παρακάτω.



Εικόνα 4. 50: Seamonkey form attributes support

4.4.2 - Form Attributes, Κώδικας

Παρακάτω φαίνεται το novalidate attribute, το οποίο οδηγεί σε μη επικύρωση της φόρμας κατά την υποβολή της. [1][2][3][4]

```
<form action="" novalidate>
  Email: <input type="email">
  <input type="submit" value="submit">
</form>
```

Εικόνα 4. 51: Form novalidate code

Στην επόμενη εικόνα βλέπουμε το autofocus attribute, το οποίο όταν τοποθετηθεί σε κάποιο element, τότε μόλις φορτώσει η σελίδα ο κέρσορας του ποντικιού πηγαίνει αυτόματα σε εκείνο το σημείο. Επίσης φαίνεται το form attribute, με το οποίο μπορούμε να εισάγουμε ένα element σε μία φόρμα, ακόμα και αν βρίσκεται έξω από αυτήν.

```
<form action="">
  Email: <input type="email" autofocus>
  <input type="submit" value="submit">
</form>
<div style="width:490px; text-align:center; color:red;">
  <code>form myForm ends</code></div>
Name <input type="text" name="lname" form="form1">
<span>Το element "Name" ανήκει στην φόρμα "myForm"</span>
```

Εικόνα 4. 52: Autofocus / form attributes code

Εδώ φαίνονται τα min / max attributes, τα οποία καθορίζουν μία ελάχιστη και μέγιστη τιμή που δέχεται ένα στοιχείο.

```
Διαλέξτε τιμή ανάμεσα στο 5 και στο 15:
<br/><input type="number" name="quantity" min="5" max="15">
<input type="submit" value="submit">
</form>
</article>
```

Εικόνα 4. 53: Min / Max attributes code

Παρακάτω φαίνεται ο κώδικας για την πολλαπλή επιλογή αρχείων (multiple attribute).

```
<form action="">
  Files: <input type="file" multiple>
  <br/>
  <input type="submit" value="submit">
</form>
```

Εικόνα 4. 54: Multiple files code

Στην επόμενη εικόνα βλέπουμε το pattern attribute, όπου στην προκειμένη περίπτωση ζητάμε να δοθούν σαν είσοδος 10 νούμερα.

```
<form action="">
Phone Number (10 digits): <input type="tel" pattern="[0-9]{10}">
  <input type="submit" value="submit">
</form>
```

Εικόνα 4. 55: Pattern code

Τα placeholders λειτουργούν ως hints, για να καταλάβει ο χρήστης την τιμή που πρέπει να εισάγει. Επίσης εάν σε κάποιο στοιχείο δοθεί το attribute required, τότε αυτό πρέπει να συμπληρωθεί οπωσδήποτε προτού υποβληθεί η φόρμα.

```
<form action="">
  Name: <input type="text" placeholder="Your Name Please" ><br/>
  Phone <input type="tel" pattern="[0-9]{10}" required placeholder="10 digits">
  <br/><input type="submit" value="submit">
</form>
```

Εικόνα 4. 56: Placeholders code

Το step καθορίζει την ακολουθία αριθμών που είναι αποδεκτή από το σύστημα.

```
<form action="">  
  Input Number with Step 2<br/>  
  <input type="number" min="0" max="12" step="2">  
  <input type="submit" value="submit">  
</form>
```

Εικόνα 4. 57: Step code

4.6 - Canvas

Μία από τις νέες και σημαντικότερες προσθήκες της HTML5 είναι ο Canvas. Η δυνατότητα δηλαδή του σχεδιασμού γραφικών τοπικά στον εκάστοτε browser. Τα στοιχεία canvas απλώς περιέχουν τα γραφικά, για το σχεδιασμό τους χρησιμοποιείται κάποιο script, συνήθως javascript. [1][2][3][4]

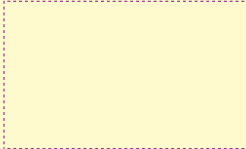
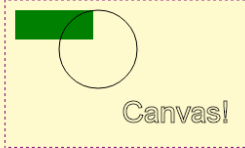
4.6.1 - Canvas, Υποστήριξη στους Browsers

Ο canvas υποστηρίζεται από όλους τους browsers δίχως να υπάρχουν διαφορές μεταξύ τους.

HTML5

[Αρχική Σελίδα](#) [Video / Audio](#) [Input Types](#) [Form Elements](#) [Form Attributes](#) [Canvas](#) [Drag And Drop](#) [Local Storage](#)

Το canvas της HTML5 χρησιμοποιείται για τον σχεδιασμό γραφικών. Στην ουσία απλώς περιέχει τα γραφικά, για τον σχεδιασμό τους χρησιμοποιείται κάποιο script (javascript).

<p>Για να οριστεί ένα στοιχείο canvas, απλώς το δηλώνουμε και δίνουμε τις διαστάσεις του:</p> <pre>< canvas id='first' width='250' height='150' style='margin-left:50px; border:1px dashed purple;' ></pre> 	<p>Ο σχεδιασμός του αντικειμένου που θέλουμε γίνεται με κάποιο script.</p> <pre>var c=document.getElementById("second"); var ctx=c.getContext("2d"); ctx.fillStyle="#FF0000"; ctx.fillRect(10,10,150,75); ctx.beginPath(); ctx.arc(95,50,40,0,2*Math.PI); ctx.stroke(); ctx.font="30px Arial"; ctx.strokeText("Canvas!",120,125);</pre> 	<p>Δηλώνουμε το id του canvas και καλούμε τη μέθοδο για τον σχεδιασμό.</p> <pre>var c=document.getElementById("second"); var ctx=c.getContext("2d");</pre> <hr/> <p>Σχεδιάζουμε ένα ορθογώνιο με διαστάσεις 80x30, στις συντεταγμένες 10,10 του canvas.</p> <pre>ctx.fillStyle="green"; ctx.fillRect(10,10,80,30);</pre> <hr/> <p>Δημιουργούμε έναν κύκλο στις συντεταγμένες 98,50 με διάμετρο 40px.</p> <pre>ctx.beginPath(); ctx.arc(95,50,40,0,2*Math.PI); ctx.stroke();</pre> <hr/> <p>Δημιουργούμε ένα πεδίο κειμένου με φόντο Arial, μέγεθος γραμματοσειράς 30px, στις συντεταγμένες 120,25.</p> <pre>ctx.font="30px Arial"; ctx.strokeText("Canvas!",120,125);</pre>
---	---	---

Εικόνα 4. 58: Canvas

4.6.2 - Canvas, Κώδικας

Στην παρακάτω εικόνα φαίνεται η δημιουργία ενός canvas αντικειμένου. Ορίζουμε τις διαστάσεις του καθώς και την εξωτερική του εμφάνιση. Σχεδιάζουμε λοιπόν ένα ορθογώνιο με μωβ περίμετρο.

```
<canvas id='second' width='250' height='150'  
style='margin-left:10px; border:1px dashed purple;'>
```

Εικόνα 4. 59: Create canvas item

Στη συνέχεια ο όποιος σχεδιασμός γραφικών μέσα στον canvas γίνεται με τη χρήση script. Παρακάτω βλέπουμε τον κώδικα για τον σχεδιασμό ενός πράσινου ορθογωνίου. Δίνονται οι συντεταγμένες των τεσσάρων γωνιών του καθώς και το χρώμα που θα έχει το γέμισμά του.

```
<script>  
var c=document.getElementById("second");  
var ctx=c.getContext("2d");  
ctx.fillStyle="green";  
ctx.fillRect(10,10,80,30);
```

Εικόνα 4. 60: Σχεδίαση ορθογωνίου

Έπειτα σχεδιάζουμε έναν κύκλο δίνοντας ως είσοδο τις συντεταγμένες και την διάμετρο, καθώς και ένα κείμενο.

```
ctx.beginPath();  
ctx.arc(95,50,40,0,2*Math.PI);  
ctx.stroke();  
  
ctx.font="30px Arial";  
ctx.strokeText("Canvas!",120,125);  
</script>
```

Εικόνα 4. 61: Σχεδίαση κύκλου και εισαγωγή κειμένου

4.7 - Drag and Drop

Η HTML5 υποστηρίζει την Drag and Drop λειτουργία, δηλαδή τη δυνατότητα να μετακινεί κανείς αντικείμενα (εικόνες, κείμενο, σχεδιαγράμματα κ.α.) από ορισμένα σημεία της σελίδας σε ορισμένα άλλα σημεία. Μέχρι πρότινος κάτι τέτοιο ήταν δυνατό μόνο μέσω της χρήση script. Πλέον όμως υποστηρίζεται η εκτέλεση τοπικά από τους browsers. [1][2][3][4]

Τα αντικείμενα τα οποία θα είναι δυνατόν να μετακινηθούν παίρνουν το όρισμα `draggable="true"`. Στη συνέχεια, πρέπει να καθοριστεί ποια javascript συνάρτηση θα εκτελεστεί μόλις αρχίσει να μετακινείται το αντικείμενο (`ondragstart=""`).

Τα αντικείμενα δεν μπορούν να μετακινηθούν παντού μέσα στην σελίδα παρά μόνο σε εκείνα τα στοιχεία που θα τα ορίσουμε εμείς. Έτσι, ορίζουμε το τι θα γίνει όταν το αντικείμενο βρεθεί από πάνω από ένα τέτοιο στοιχείο (`ondragover=""`) καθώς και το τι θα γίνει όταν αφήσουμε το αντικείμενο πάνω στο στοιχείο (`ondrop=""`).


4.7.1 - Drag n Drop, Υποστήριξη στους Browsers

Η συγκεκριμένη λειτουργία υποστηρίζεται από όλους τους browsers εκτός από τον Safari.

HTML5

[Αρχική Σελίδα](#) [Video / Audio](#) [Input Types](#) [Form Elements](#) [Form Attributes](#) [Canvas](#) [Drag And Drop](#) [Local Storage](#)

Ένα από τα νέα και πολύ χρήσιμα χαρακτηριστικά της HTML5 είναι η Drag and Drop λειτουργία. Ενώ μέχρι τώρα τέτοιου είδους δυνατότητες ήταν δυνατές μέσω κάποιου script, πλέον αυτό γίνεται τοπικά στον browser του χρήστη.

<p>Σύρτε την εικόνα από το ένα πλαίσιο στο άλλο.</p> 	<p>Αρχικά πρέπει να δηλωθεί ένα αντικείμενο το οποίο θα μπορεί να μετακινηθεί.</p> <p>Εισάγουμε το attribute <code>draggable="true"</code></p>	<p>Δηλώνουμε ποια συνάρτηση θα εκτελεστεί όταν ξεκινήσει το drag.</p> <pre>ondragstart="drag(event) "</pre>	<p>Ορίζουμε που θα μεταφερθούν τα αντικείμενα.</p> <pre>ondrop="drop(event) " ondragover="allowDrop(event) "</pre>
--	--	---	--

```
function drag(event)
Καθορίζεται ο τύπος του αντικείμενου που θα σύρουμε και του ανατίθεται μία τιμή.
ev.dataTransfer.setData("Text", ev.target.id);
Εισάγουμε σε μεταβλητή τύπου "Text" το id του αντικείμενου που σέρνουμε.

function allowdrop(event)
Επειδή τα αντικείμενα δεν μπορούν να μεταφερθούν πάνω σε άλλα αντικείμενα, στην αρχή ακυρώνουμε την default συμπεριφορά.
event.preventDefault();

function drop(event)
Ορίζουμε τι θα συμβεί κατά το drop.

ev.preventDefault();
Αποτρέπουμε την default συμπεριφορά.

var data=ev.dataTransfer.getData("Text");
Παίρνουμε την τιμή που έχει το "Text" και την τοποθετούμε στην μεταβλητή "data".

ev.target.appendChild(document.getElementById(data));
Τοποθετούμε το συρόμενο στοιχείο.
```

Εικόνα 4. 62: Drag n Drop Function

4.7.2 - Drag n Drop, Κώδικας

Στην παρακάτω εικόνα βλέπουμε τη δήλωση ενός αντικειμένου (μίας εικόνας στην προκειμένη περίπτωση) ως δυνατής για μετακίνηση και τη συνάρτηση που εκτελείται καθώς προσπαθούμε να την μετακινήσουμε.

```

```

Εικόνα 4. 63: Ορισμός draggable αντικειμένου

Στην περίπτωση μας η συνάρτηση που εκτελείται παίρνει το id του αντικειμένου και το αποθηκεύει σε μία μεταβλητή.

```
/* Παίρνουμε το id του συρόμενου αντικειμένου */  
function drag(ev)  
{  
    ev.dataTransfer.setData("Text", ev.target.id);  
}
```

Εικόνα 4. 64: Ορισμός ondragstart

Δηλώνουμε το στοιχείο όπου θα μπορεί να μετακινηθεί το αντικείμενο.

```
</div>  
  
<div id="to" ondrop="drop(event)" ondragover="allowDrop(event)">
```

Εικόνα 4. 65: Ορισμός περιοχής για μετακίνηση

Αρχικά, όταν μετακινήσουμε το αντικείμενο πάνω από το στοιχείο που θέλουμε, πρέπει να ακυρώσουμε την default συμπεριφορά (αυτή μπορεί να είναι να ανοίξει μία νέα σελίδα).

```
/* Ακυρώνουμε την default συμπεριφορά */  
function allowDrop(ev)  
{  
    ev.preventDefault();  
}
```

Εικόνα 4. 66: Ορισμός ondragover

Στο τέλος, δηλώνουμε τι θα κάνουμε με το αντικείμενο που μετακινείται. Συγκεκριμένα, βρίσκουμε το id του αντικειμένου από τη μεταβλητή που το αποθηκεύσαμε και το εισάγουμε στο στοιχείο που θέλουμε.

```
/* Συνάρτηση για το drop του αντικειμένου */  
function drop(ev)  
{  
    ev.preventDefault();  
    var data=ev.dataTransfer.getData("Text");  
    ev.target.appendChild(document.getElementById(data));  
}
```

Εικόνα 4. 67: ondrop

4.8 - Local Storage

Η HTML5 έχει εισάγει ένα νέο τρόπο τοπικής αποθήκευσης πληροφοριών στους browsers των χρηστών από τα cookies. Με τη νέα αυτή λειτουργία (Local / Web Storage) η αποθήκευση των δεδομένων είναι πιο ασφαλής και γρηγορότερη. Τα δεδομένα χρησιμοποιούνται μόνο όταν ζητηθούν. Η λειτουργία υποστηρίζεται από όλους τους browsers. [1][2][3][4]

4.8.1 - Local Storage, Υποστήριξη στους Browsers

Για να διαπιστωθεί εάν υποστηρίζεται η λειτουργία, δημιουργήσαμε μία φόρμα και δώσαμε μία τιμή («2013»). Στη συνέχεια κλείσαμε το παράθυρο του browser και όταν το ξανά-ανοίξαμε η τιμή που είχαμε δώσει υπήρχε ως όρισμα στο input text.

Ενδεικτικά το αποτέλεσμα φαίνεται στις παρακάτω εικόνες:

HTML5

[Αρχική Σελίδα](#) [Video/Audio](#) [Input Types](#) [Form Elements](#) [Form Attributes](#) [Canvas](#) [Drag And Drop](#) [Local Storage](#)

Ένα από τα επιπρόσθετα νέα χαρακτηριστικά της HTML5, είναι μία βελτιωμένη μορφή τοπικής αποθήκευσης σε σχέση με τα cookies.

Οι σελίδες μπορούν να αποθηκεύονται τοπικά στον browser του χρήστη, πιο γρήγορα και με μεγαλύτερη ασφάλεια. Τα δεδομένα δεν βρίσκονται συνέχεια αποθηκευμένα αλλά λαμβάνονται μόνο όταν ζητηθούν.

Εισάγετε Όνομα:

Παίρνουμε την τιμή που έχει εισαχθεί στο text field ως εξής:
`document.getElementById('name').value`

Η αλλαγή αυτής της τιμής γίνεται με παρόμοιο τρόπο:
`document.getElementById("name").value=.....`

myfun_function
Αποθηκεύουμε την τιμή του text field σε μία τοπική μεταβλητή με όνομα name.

```
localStorage.setItem('name',document.getElementById('name').value);
```

show_function
Κάθε φορά που ανοίγει η σελίδα, καλείται η συνάρτηση show.

Ελέγχεται εάν υπάρχει η τοπική μεταβλητή. Εάν ναι τότε αναθέτει την τιμή της στο πεδίο του text field.
Έτσι ο browser θυμάται τι έχουμε εισάγει στο πεδίο ακόμα και εάν κλείσουμε το παράθυρο.

```
if(localStorage.getItem('name'))  
document.getElementById("name").value=localStorage.getItem('name');
```

Εικόνα 4. 68: Local Storage αρχική

HTML5

[Αρχική Σελίδα](#)
[Video / Audio](#)
[Input Types](#)
[Form Elements](#)
[Form Attributes](#)
[Canvas](#)
[Drag And Drop](#)
[Local Storage](#)

Ένα από τα επρόσθετα νέα χαρακτηριστικά της HTML5, είναι μία βελτιωμένη μορφή τοπικής αποθήκευσης σε σχέση με τα cookies.

Οι σελίδες μπορούν να αποθηκεύονται τοπικά στον browser του χρήστη, πιο γρήγορα και με μεγαλύτερη ασφάλεια. Τα δεδομένα δεν βρίσκονται συνέχεια αποθηκευμένα αλλά λαμβάνονται μόνο όταν ζητηθούν.

Εισάγετε Όνομα:

Εισάγετε το όνομα 2013

Παίρνουμε την τιμή που έχει εισαχθεί στο text field ως εξής:

```
document.getElementById('name').value
```

Η αλλαγή αυτής της τιμής γίνεται με παρόμοιο τρόπο:

```
document.getElementById("name").value=.....
```

myfun_function

Αποθηκεύουμε την τιμή του text field σε μία τοπική μεταβλητή με όνομα name.

```
localStorage.setItem('name',document.getElementById('name').value);
```

show_function

Κάθε φορά που ανοίγει η σελίδα, καλείται η συνάρτηση show.

Ελέγχεται εάν υπάρχει η τοπική μεταβλητή. Εάν ναι τότε αναθέτει την τιμή της στο πεδίο του text field. Έτσι ο browser θυμάται τι έχουμε εισάγει στο πεδίο ακόμα και εάν κλείσουμε το παράθυρο.

```
if(localStorage.getItem('name'))
document.getElementById("name").value=localStorage.getItem('name');
```

Εικόνα 4. 69: Local Storage εισαγωγή τιμής

HTML5

[Αρχική Σελίδα](#)
[Video / Audio](#)
[Input Types](#)
[Form Elements](#)
[Form Attributes](#)
[Canvas](#)
[Drag And Drop](#)
[Local Storage](#)

Ένα από τα επρόσθετα νέα χαρακτηριστικά της HTML5, είναι μία βελτιωμένη μορφή τοπικής αποθήκευσης σε σχέση με τα cookies.

Οι σελίδες μπορούν να αποθηκεύονται τοπικά στον browser του χρήστη, πιο γρήγορα και με μεγαλύτερη ασφάλεια. Τα δεδομένα δεν βρίσκονται συνέχεια αποθηκευμένα αλλά λαμβάνονται μόνο όταν ζητηθούν.

Εισάγετε Όνομα:

Παίρνουμε την τιμή που έχει εισαχθεί στο text field ως εξής:

```
document.getElementById('name').value
```

Η αλλαγή αυτής της τιμής γίνεται με παρόμοιο τρόπο:

```
document.getElementById("name").value=.....
```

myfun_function

Αποθηκεύουμε την τιμή του text field σε μία τοπική μεταβλητή με όνομα name.

```
localStorage.setItem('name',document.getElementById('name').value);
```

show_function

Κάθε φορά που ανοίγει η σελίδα, καλείται η συνάρτηση show.

Ελέγχεται εάν υπάρχει η τοπική μεταβλητή. Εάν ναι τότε αναθέτει την τιμή της στο πεδίο του text field. Έτσι ο browser θυμάται τι έχουμε εισάγει στο πεδίο ακόμα και εάν κλείσουμε το παράθυρο.

```
if(localStorage.getItem('name'))
document.getElementById("name").value=localStorage.getItem('name');
```

Εικόνα 4. 70: Local Storage αποτέλεσμα

4.8.2 - Local Storage, Κώδικας

Το πρώτο πράγμα που πρέπει να κάνουμε είναι να ρυθμίσουμε το σύστημα έτσι ώστε να θυμάται την τιμή που θέλουμε. Έτσι, με το submit καλούμε μία συνάρτηση η οποία κάνει ακριβώς αυτό.

```
<input type='submit' value='send' onclick="myfun();" >
```

Εικόνα 4. 71: Κάλεισμα συνάρτησης myfun()

Στην συνάρτησή μας λοιπόν myfun(), αρχικά γίνεται έλεγχος για το εάν υποστηρίζεται η λειτουργία Local Storage. Εάν δεν υποστηρίζεται τότε εμφανίζεται το ανάλογο μήνυμα, «Δεν υποστηρίζεται η τοπική αποθήκευση».

Στην περίπτωση που υποστηρίζεται τότε παίρνουμε την τιμή του input text (document.getElementById('name').value) και το αναθέτουμε σε μία μεταβλητή Local Storage ονόματος name.

Στη συνέχεια εμφανίζουμε ανάλογο μήνυμα επιτυχίας ανάθεσης της τιμής.

```
/* Η συνάρτηση που καλείται */
function myfun() {
  /* Εάν υποστηρίζεται η Τοπική Αποθήκευση */
  if(typeof(Storage) !== "undefined")
  {
    /* Κάθε φορά που καλείται η συνάρτηση γίνεται ανατίθεται στην τοπική
    * μεταβλητή η τιμή του text field. */
    localStorage.setItem('name', document.getElementById('name').value);

    document.getElementById("show").innerHTML="Εισάγατε το όνομα "+localStorage.getItem('name');
  }
  /* Εάν δεν υποστηρίζεται η τοπική αποθήκευση */
  else
  {
    document.getElementById("show").innerHTML="Δεν υποστηρίζεται η τοπική αποθήκευση.";
  }
}
```

Εικόνα 4. 72: Δημιουργία Local Storage μεταβλητής

Μόλις φορτώσει η σελίδα, καλείται μία δεύτερη συνάρτηση ονόματος show().

```
<body onload='show();'>
```

Εικόνα 4. 73: Κάλεισμα συνάρτησης show()

Σε αυτή τη συνάρτηση ελέγχουμε εάν έχει ανατεθεί κάποια τιμή στη μεταβλητή name και εάν ναι, τότε εμφανίζουμε αυτή τη τιμή μέσα στο input text. Έτσι, όταν κλείνουμε και έπειτα ανοίγουμε πάλι τη σελίδα η μεταβλητή έχει τη τιμή που της δόθηκε νωρίτερα και εμφανίζεται στο πεδίο.

```
/* Με το φόρτωμα της σελίδας, εισάγουμε την τιμή της τοπικής μεταβλητής  
 * στο text field (εάν έχει δημιουργηθεί) */  
function show() {  
    if (localStorage.getItem('name'))  
        document.getElementById("name").value=localStorage.getItem('name');  
}
```

Εικόνα 4. 74: Local Storage Αποτέλεσμα

4.9 - HTML5 Βαθμολογίες

Η ιστοσελίδα «The HTML5 test - how well does your browser support HTML5» (<http://html5test.com/index.html>) είναι ένας δείκτης του κατά πόσο οι διάφοροι browsers υποστηρίζουν την HTML5 και τα διάφορα χαρακτηριστικά τους.

Τα αποτελέσματα των μετρήσεών τους φαίνονται στον παρακάτω πίνακα:

Browser	Score
Maxthon 4.1.2	476
Chrome 29	463
Avant	463
Firefox 23.0	414
Seamonkey 2.20	414
Opera 12.16	404
IE 10.0	320
Safari 4.1.7	278

Από την πειραματική ιστοσελίδα που δημιουργήσαμε με τα νέα χαρακτηριστικά της HTML5 μπορεί να επιβεβαιωθεί ότι οι browsers με τα καλύτερα ποσοστά υποστήριξης είναι ο Chrome και ο Maxthon. Ο Avant ακολουθεί κατά πόδας με τους Firefox, Seamonkey και Opera να βρίσκονται λίγο πιο πίσω. Αντίθετα οι IE και Safari υστερούν προς το παρόν αρκετά στην υποστήριξη των νέων στοιχείων της HTML4.

Όλα αυτά βέβαια είναι εύκολο να αλλάξουν, καθώς έγκειται στους κατασκευαστές των παραπάνω browsers να τους κάνουν πιο συμβατούς στις επόμενες εκδόσεις τους. [6]

5^ο Κεφάλαιο - Συμπεράσματα & Μελλοντικές Μελέτες

Στη συγκεκριμένη πτυχιακή εργασία έγινε μία συγκριτική μελέτη του κατά πόσο υποστηρίζεται η HTML5 και τα νέα χαρακτηριστικά της από τους διάφορους browsers, δημιουργώντας μία ιστοσελίδα με τα νέα αυτά χαρακτηριστικά.

Όπως φάνηκε και από τα αποτελέσματα, δεν υπάρχει ακόμη κάποιος browser ο οποίος να υποστηρίζει κατά 100% τα χαρακτηριστικά της. Ορισμένοι ωστόσο όπως ο Chrome και ο Maxthon καταφέρνουν αρκετά μεγάλα ποσοστά υποστήριξής τους.

Δεν παρατηρήσαμε το γεγονός να υπάρχει κάποιος browser (τουλάχιστον από τους δημοφιλέστερους με τους οποίους διεξήγαμε την έρευνα) ο οποίος να μην παρέχει καθόλου υποστήριξη της HTML5. Σε γενικές γραμμές όλοι παρέχουν ένα ικανοποιητικό ποσοστό υποστήριξης των χαρακτηριστικών της.

Όσο συνεχίζεται η δημιουργία νέων εκδόσεων των browsers, και ίσως η δημιουργία κάποιων καινούργιων, θα παρατηρούμε όλο και μεγαλύτερη υποστήριξη της HTML5.

Στο μέλλον θα μπορούσαν να γίνουν επιπλέον μελέτες και συγκριτικά τεστ. Μία επιπλέον παράμετρος θα μπορούσε να είναι η συμμετοχή νέων browser, είτε αυτοί είναι καινούργιοι είτε απλώς δεν είναι τόσο γνωστοί στο ευρύ κοινό.

Επίσης σημαντική παράμετρος είναι η εξέλιξη των browsers που μελετήσαμε με την κυκλοφορία νέων εκδόσεων, όπου λογικά θα είναι ακόμα πιο φιλικό προς την HTML5.

Τέλος είναι λογικό να υποθέτουμε ότι στο μέλλον σε επόμενες εκδόσεις της HTML θα δημιουργηθούν και άλλα χαρακτηριστικά τα οποία θα μπορούν να ενσωματωθούν στη μελέτη.

Βιβλιογραφία

1. W3Schools - HTML5 New Elements, http://www.w3schools.com/html/html5_new_elements.asp
2. Nettuts+ - 28 HTML5 Features, Tips and Techniques you Must Know, <http://net.tutsplus.com/tutorials/html-css-techniques/25-html5-features-tips-and-techniques-you-must-know/>
3. TechRepublic - 10 new HTML5 tags you need to know about, <http://www.techrepublic.com/blog/10things/10-new-html5-tags-you-need-to-know-about/3219>
4. 1stWebDesigner: HTML5 Introduction - What is HTML5 Capable of, Features and Resources, <http://www.1stwebdesigner.com/design/html5-introduction/>
5. Dive Into HTML5 - <http://diveintohtml5.info/introduction.html>
6. The HTML5 Test: How well does your browser support HTML5? - <http://html5test.com/>
7. Wikipedia: HTML5 Video - http://en.wikipedia.org/wiki/HTML5_video
8. HTML5 VS FLASH - <http://blog.accusoft.com/posts/2013/january/html5-vs-flash-infographic.html>
9. What is browser - <http://www.webopedia.com/TERM/B/browser.html>
10. The purpose, function and components of web browsers - <http://webdesignpeeps.com/purpose-function-components-of-web-browsers/>
11. Web browser - http://en.wikipedia.org/wiki/Web_browser
12. Usage share of web browsers - http://en.wikipedia.org/wiki/Usage_share_of_web_browsers#Desktop_and_mobile_browser_usage
13. Firefox - <http://en.wikipedia.org/wiki/Firefox>
14. Features of Firefox - http://en.wikipedia.org/wiki/Features_of_Firefox
15. History of the Mozilla Project - <http://www.mozilla.org/en-US/about/history/>
16. Mozilla Wiki Timeline - <https://wiki.mozilla.org/Timeline>
17. Internet Explorer, a dive through history - <http://www.neowin.net/news/internet-explorer-version-1-10-a-dive-through-history>

18. Opera - <http://en.wikipedia.org/wiki/Opera> (web browser)
19. Innovations by Opera - <http://operawiki.info/operainnovations>
20. Spatial navigation - http://en.wikipedia.org/wiki/Spatial_navigation
21. 5 features Opera did first - <http://www.slashgeek.net/2012/06/08/5-features-opera-browser-did-first/>
22. Google Chrome - http://en.wikipedia.org/wiki/Google_Chrome
23. Maxthon - <http://en.wikipedia.org/wiki/Maxthon>
24. Avant Browser - http://en.wikipedia.org/wiki/Avant_Browser
25. Seamonkey - <http://en.wikipedia.org/wiki/SeaMonkey>
26. Safari - [http://en.wikipedia.org/wiki/Safari_\(web_browser\)](http://en.wikipedia.org/wiki/Safari_(web_browser))
27. StatCounter - <http://gs.statcounter.com/#browser-ww-monthly-200807-201308>
28. The BIG browser benchmark - <http://www.zdnet.com/the-big-browser-benchmark-january-2013-edition-7000009776/>
29. Web Browser - http://en.wikipedia.org/wiki/Web_browser
30. What is the Internet - <http://netforbeginners.about.com/od/i/f/What-Is-The-Internet.htm>
31. History of the Internet - http://en.wikipedia.org/wiki/History_of_the_Internet
32. World Wide Web - http://en.wikipedia.org/wiki/World_Wide_Web
33. History of the World Wide Web - http://en.wikipedia.org/wiki/History_of_the_World_Wide_Web
34. History of the Web - <http://www.webfoundation.org/vision/history-of-the-web/>
35. Internet Protocol Suite - http://en.wikipedia.org/wiki/Internet_protocol_suite
36. Layers of the TCP / IP model - http://www.sis.pitt.edu/~icucart/networking_basics/4LayersofTCIPModel.html
37. IPV4 / IPV6 - <http://mashable.com/2011/02/03/ipv4-ipv6-guide/>
38. Routing - <http://en.wikipedia.org/wiki/Routing>
39. Web 2.0 - http://en.wikipedia.org/wiki/Web_2.0
40. Ten characteristics of web 2.0 - <http://www.vitodibari.com/en/ten-characteristics-web-20-internet-changed.html>
41. Semantic Web - http://en.wikipedia.org/wiki/Semantic_Web
42. Web 4.0: The Ultra Intelligent Electronic Agent is coming - <http://bigthink.com/big-think-tv/web-40-the-ultra-intelligent-electronic-agent-is-coming>
43. Web 1.0 vs Web 2.0 vs Web 3.0 vs Web 4.0 - <http://flatworldbusiness.wordpress.com/flat-education/previously/web-1-0-vs-web-2-0-vs-web-3-0-a-bird-eye-on-the-definition/>
44. Web 4.0 era is upon us - <http://www.pcworld.com/article/143110/article.html>
45. HTML - <http://en.wikipedia.org/wiki/HTML>
46. HTML5 - <http://en.wikipedia.org/wiki/HTML5>

47. Cascading Style Sheets - http://en.wikipedia.org/wiki/Cascading_Style_Sheets
48. JavaScript - <http://en.wikipedia.org/wiki/JavaScript>
49. AJAX - [http://en.wikipedia.org/wiki/Ajax_\(programming\)](http://en.wikipedia.org/wiki/Ajax_(programming))
50. PHP - <http://en.wikipedia.org/wiki/PHP>
51. Model - View - Controller -
<http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>
52. Νέος Παγκόσμιος Ιστός (Web 2.0) και Τεχνολογίες Δημιουργίας Περιεχομένου από το Χρήστη. Μελέτη της διείσδυσής τους σε Έλληνες Φοιτητές, Διπλωματική Εργασία της Γιώτη Ελένης, ΕΜΠ, Μηχανολόγοι Μηχανικοί - Τομές Βιομηχανικής Διοίκησης & Επιχειρησιακής Έρευνας.
53. The Internet, where did it come from - <http://www.bubblews.com/news/1091328-the-internet-where-did-it-come-from>
54. The TCP / IP family of internet protocols -
http://www.cellbiol.com/bioinformatics_web_development/doku.php/chapter_1_-_internet_networks_and_tcp-ip/1-2_the_tcp-ip_family_of_internet_protocols
55. TCP / UDP - http://commons.wikimedia.org/wiki/File:Tcp_udp.jpg
56. IP Packet Structure - <http://www.freesoft.org/CIE/Course/Section3/7.htm>
57. IPv4 vs IPv6 - http://commons.wikimedia.org/wiki/File:IPv6_vs_IPv4.jpg
58. Internet - <http://en.wikipedia.org/wiki/Internet>
59. Understanding the IP routing table - [http://technet.microsoft.com/en-us/library/cc787509\(v=ws.10\).aspx](http://technet.microsoft.com/en-us/library/cc787509(v=ws.10).aspx)
60. Multicast routing - <http://www.cse.tkk.fi/fi/opinnot/T-110.5190/1996/mcast.html>
61. Using google public dns - <http://webhost.pro/blog/2009/12/google-dns/>
62. Getting started with web applications - http://www.cs.umsl.edu/~he/GS_WebApp.html
63. Web 1.0 vs Web 2.0 - <http://drdyer.edublogs.org/2008/05/13/web-10-vs-web-20-day-13-write-a-post-using-comments/>
64. Web and Mobile Interest Group Emerges from W3C to Further Mobile Web Initiative - <http://lorenzfz.blog.de/2013/08/27/web-and-mobile-interest-group-emerges-from-w3c-to-further-mobile-web-initiative-16340176/>
65. Web 2.0 και Εκπαίδευση 2.0 - <http://e-mathisi2011.wikispaces.com/web+2.0+%CE%BA%CE%B1%CE%B9+++%CE%95%CE%BA%CF%80%CE%B1%CE%AF%CE%B4%CE%B5%CF%85%CF%83%CE%B7+2.0>
66. What is semantic web - <http://www.rockingteam.com/2009/05/what-is-semantic-web/>
67. Agents and the semantic web - <http://www.cs.rpi.edu/~hendler/AgentWeb.html>
68. Web 2.0 and Web 3.0, Impact on hospitality technology -
<http://rockcheetah.com/blog/technology/web-2-0-3-0-impact-on-hospitality-technology-hftp-hitec/>

69. Web 2.0 vs Web 3.0 - <http://thepaisano.wordpress.com/2008/03/08/web-20-vs-web-30/>
70. A preview of HTML5 - <http://alistapart.com/article/previewofhtml5>
71. AJAX: A new approach to web applications - <http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications/>
72. PHP Tutorial - <http://www.learnphp-tutorial.com/PHPBasics.cfm>
73. Optimizing JavaScript and CSS files in drupal - <http://shvetsgroup.com/blog/optimizing-javascript-and-css-files-drupal>
74. Netscape navigator - http://en.wikipedia.org/wiki/Netscape_Navigator
75. Spatial navigation and opera - <http://www.codinghorror.com/blog/2008/02/spatial-navigation-and-opera.html>