

Δημιουργία ιστοσελίδας σε HTML5 για την παραγωγή οπτικοακουστικού ψυχαγωγικού περιεχομένου

Πτυχιακή εργασία

Πέτρος Δαλακούρας, Α.Μ. 2629

Εισηγητές:

Αθανάσιος Μαλάμος

Παρασκευή Σύμπα

Επιβλέπων καθηγητής :

Καπετανάκης Κωνσταντίνος

Abstract

HTML5 is the upcoming version of the HTML language, which aims to address the limitations of the current version of the standard and facilitate the development of impressive web pages as well as powerful applications. In this thesis we will study its differences from its predecessor as well as the new features that are being developed.

Σύνοψη

Η HTML5 είναι η επόμενη έκδοση του προτύπου της γλώσσας HTML, που επιδιώκει να καλύψει τους περιορισμούς που έχει τωρινή προδιαγραφή και να κάνει την δημιουργία εντυπωσιακών σελίδων και εφαρμογών εύκολη σε οποιαδήποτε πλατφόρμα. Σε αυτή την εργασία θα αναλύσουμε τις διαφορές που έχει από την προηγούμενη έκδοση και τις νέες δυνατότητες που προσφέρει.

Περιεχόμενα

Abstract	2
Σύνοψη	3
1 Εισαγωγή	6
1.1 Περίληψη.....	6
1.2 Κίνητρο για την διεξαγωγή της εργασίας.....	6
1.3 Στόχος εργασίας	6
1.4 Δομή εργασίας	6
2 Παρουσίαση της HTML.....	8
2.1 Εισαγωγή	8
2.2 Ιστορία της HTML	8
2.2.1 Εκδόσεις της XHTML.....	10
2.3 Σχετικά με τα πρότυπα.....	11
2.3.1 Διαδικασία υιοθέτησης προτύπων	11
2.3.2 Διαδικασία προτυποποίησης της HTML5	11
2.3.3 Εύρος του προτύπου HTML5	12
2.3.4 Η απόκλιση της HTML5 από τη διαδικασία υιοθέτησης προτύπων.....	13
2.4 Περιορισμοί της HTML	14
2.4.1 Πληθώρα τρίτων τεχνολογιών	14
2.4.2 Αποκλίσεις των υλοποιήσεων από το πρότυπο	15
3 Παρουσίαση της HTML5.....	17
3.1 Εισαγωγή	17
3.2 Βασικά στοιχεία.....	17
3.2.1 Το στοιχείο DOCTYPE	17
3.2.2 Το στοιχείο html.....	18
3.2.3 Το στοιχείο head	18
3.2.4 Το στοιχείο body	19
3.2.5 Αντιμετώπιση των νέων στοιχείων από παλιότερους browsers	22
3.3 Φόρμες	22
3.3.1 Κείμενο κράτησης θέσης.....	23
3.3.2 Αυτόματη εστίαση.....	23
3.3.3 Πεδία διευθύνσεων email.....	24
3.3.4 Διευθύνσεις internet.....	25
3.3.5 Εισαγωγή αριθμών.....	26

3.3.6	Εισαγωγή αριθμού με slider	28
3.3.7	Πεδίο επιλογής ημερομηνίας	29
3.3.8	Πεδία αναζήτησης.....	30
3.3.9	Πεδίο επιλογής χρώματος.....	30
3.3.10	Έλεγχος στοιχείων φόρμας	31
3.4	Στοιχεία CSS.....	32
3.4.1	Σκιά γραμμμάτων	32
3.4.2	Σκιά γύρω από στοιχεία block.....	33
3.4.3	Στρογγυλεμένες γωνίες.....	33
3.4.4	Στήλες	33
3.4.5	Transitions	34
3.4.6	Transformations	34
3.5	Η επιφάνεια ζωγραφικής	35
3.5.1	Σχεδιασμός μονοπατιών (Paths)	37
3.5.2	Σχεδιασμός κειμένου	38
3.5.3	Σχεδιασμός gradient	39
3.5.4	Σχεδιασμός εικόνων	41
3.6	Υποστήριξη εικόνων SVG.....	43
3.7	WebGL	44
3.8	Αναπαραγωγή βίντεο	44
3.9	Εύρεση θέσης.....	46
3.10	Τοπική αποθήκευση.....	47
3.11	Web SQL Database	49
3.12	Σελίδες χωρίς σύνδεση.....	50
4	Παρουσίαση εφαρμογής.....	51
4.1	Γενικά.....	51
4.2	Βιβλιοθήκες και εγκατάσταση	51
4.3	Δημιουργία διαφανειών	53
5	Συμπεράσματα	68
	Βιβλιογραφία	69

1 Εισαγωγή

1.1 Περίληψη

Η γλώσσα HTML δημιουργήθηκε αρχικά για την προβολή απλών σελίδων κειμένου και την σύνδεση τους με άλλους. Με την εξάπλωση της του διαδικτύου άρχισε να υπάρχει η ανάγκη κατασκευής πιο εντυπωσιακών σελίδων, οι οποίες θα αντιδρούν στις ενέργειες του χρήστη, ενώ τα τελευταία χρόνια έχουν αρχίσει να εμφανίζονται σελίδες που είναι ολοκληρωμένες εφαρμογές.

Το πρότυπο της HTML αναθεωρήθηκε αρκετά με την τελευταία έκδοσή του 4.01 να έχει οριστικοποιηθεί το 1999. Καθώς οι απαιτήσεις από την γλώσσα αυξανόντουσαν αναπτύχθηκαν πολλές τεχνολογίες με σκοπό να καλύψουν τις ελλείψεις της γλώσσας οι οποίες με τη σειρά τους δημιούργησαν νέα προβλήματα, καθώς χρειάζεται να υποστηρίζονται από τα αντίστοιχα προγράμματα που χρησιμοποιούν οι χρήστες, κάτι που δεν ήταν δεδομένο.

Η επόμενη έκδοση της HTML είναι η έκδοση 5, η οποία άρχισε να αναπτύσσεται το 2004 και αναμένεται να οριστικοποιηθεί μετά το 2015. Η νέα γλώσσα προσφέρει μία πλειάδα δυνατοτήτων με σκοπό να κάνει την δημιουργία ιστοσελίδων πολύ πιο εύκολη και να μην είναι απαραίτητη η χρήση ξένων εργαλείων, ενώ πολλές από αυτές τις δυνατότητες έχουν ήδη ενσωματωθεί στους υπάρχοντες φυλλομετρητές.

1.2 Κίνητρο για την διεξαγωγή της εργασίας

Η HTML5 είναι ένα πρότυπο που αναπτύσσεται εδώ και πολλά χρόνια με σκοπό να αποτελέσει την κύρια γλώσσα κατασκευής εφαρμογών στο διαδίκτυο. Τον τελευταίο καιρό έχουν παγιωθεί αρκετά από τα στοιχεία της, κάνοντας εφικτή την δοκιμή τους σε πρακτικό επίπεδο.

1.3 Στόχος εργασίας

Ο στόχος αυτής της εργασίας είναι η παρουσίαση των δυνατοτήτων της HTML5 σε αντιδιαστολή με τις πρακτικές που ακολουθούνται στην τρέχουσα έκδοση. Δίνεται ιδιαίτερο βάρος στο πρακτικό κομμάτι, παραθέτοντας τόσο παραδείγματα χρήσης όσο και συζητώντας τον βαθμό στον οποίο έχουν υλοποιηθεί αυτά τα στοιχεία από τους υπάρχοντες browsers και τους τρόπους με τους οποίους αντιμετωπίζονται σε περίπτωση που δεν υποστηρίζονται.

Έπειτα κατασκευάστηκε μία εφαρμογή σε HTML5 η οποία παρουσιάζει κάποια από αυτά τα στοιχεία και κάποια παραδείγματα χρήσης τους από άλλες εφαρμογές.

1.4 Δομή εργασίας

Η εργασία περιλαμβάνει τα παρακάτω κεφάλαια:

1. Εισαγωγή: Αυτό το κεφάλαιο.
2. Παρουσίαση της HTML: Παρουσιάζεται η ιστορία της γλώσσας HTML και τα προβλήματα που υπάρχουν στην ανάπτυξη σελίδων με αυτή. Έπειτα αναλύεται η διαδικασία που ακολουθείται για την υιοθέτηση του προτύπου HTML5/

3. Παρουσίαση της HTML5: Παρουσιάζονται οι δυνατότητες της HTML5. Για όσες από αυτές είναι ήδη υλοποιημένες δίνονται παραδείγματα κώδικα.
4. Ανάλυση της εφαρμογής: Αναλύεται η εφαρμογή που δημιουργήθηκε.
5. Συμπεράσματα: Ένα κεφάλαιο με μία σύντομη ανασκόπηση των αποτελεσμάτων της έρευνας και συζήτηση του μέλλοντος της γλώσσας.

2 Παρουσίαση της HTML

2.1 Εισαγωγή

Σε αυτή την ενότητα θα παρουσιάσουμε την γλώσσα HTML, συγκεκριμένα πως δημιουργήθηκε, πως χρησιμοποιήθηκε και κάποια από τα προβλήματα που παρουσιάστηκαν από την χρήση της, όπως και τις προσπάθειες αντιμετώπισής τους.

2.2 Ιστορία της HTML

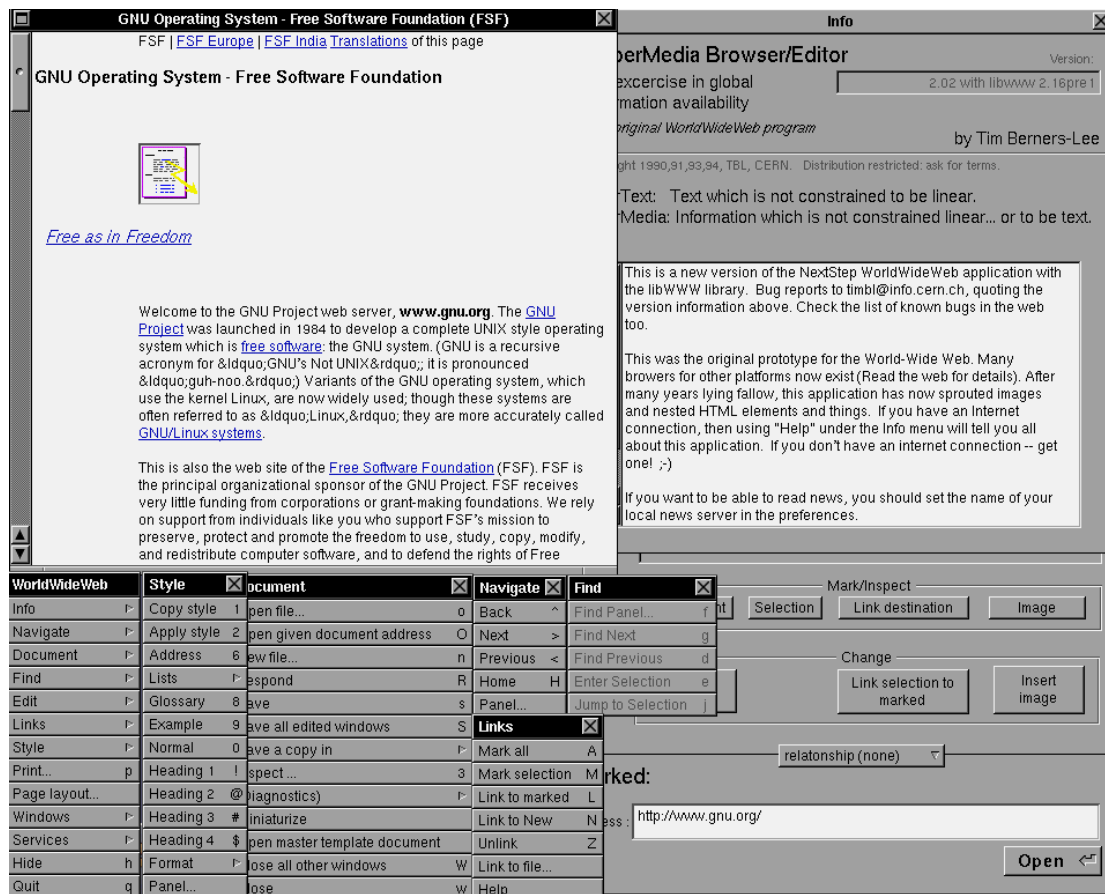
Ο μακρινός πρόγονος της HTML ήταν το ENQUIRE, ένα σύστημα χρήσης και διαμοιρασμού εγγράφων για τους ερευνητές του CERN, το οποίο επινόησε ο φυσικός Τιμ Μπέρνερς Λι το 1980. Αργότερα, το 1989, πρότεινε ένα σύστημα βασισμένο στο διαδίκτυο, το οποίο θα χρησιμοποιούσε υπερκείμενο. Έτσι, έφτιαξε την προδιαγραφή της HTML και έγραψε τον browser και το λογισμικό εξυπηρετητή στα τέλη του 1990. Τον ίδιο χρόνο, ο Μπέρνερς Λι και ο μηχανικός συστημάτων πληροφορικής του CERN Robert Cailliau συνεργάστηκαν σε μια κοινή προσπάθεια εύρεσης χρηματοδότησης, αλλά το έργο δεν υιοθετήθηκε ποτέ επίσημα από το CERN. Στις προσωπικές του σημειώσεις από το 1990, ο Μπέρνερς Λι αριθμεί «μερικές από τις πολλές χρήσεις του υπερκειμένου», και αναφέρει πρώτα από όλες μια εγκυκλοπαίδεια.

Η πρώτη δημόσια διαθέσιμη περιγραφή της HTML ήταν ένα έγγραφο με το όνομα Ετικέτες HTML (HTML tags), το οποίο πρωτοαναφέρθηκε στο Διαδίκτυο από τον Μπέρνερς Λι στα τέλη του 1991. Περιέγραφε τα 20 στοιχεία τα οποία αποτελούσαν τον αρχικό και σχετικά απλό σχεδιασμό της HTML. Εκτός από την ετικέτα υπερσυνδέσμου, οι υπόλοιπες ήταν έντονα επηρεασμένες από την SGMLguid, μια μορφή δημιουργίας τεκμηρίωσης, φτιαγμένη στο CERN και βασισμένη στην SGML. Δεκατρία από εκείνα τα αρχικά στοιχεία υπάρχουν ακόμα σήμερα στην HTML 4.

Το ίδιο το πρότυπο SGML αναπαράγει μερικές από τις τεχνικές των τυπογράφων, αλλά εκτός από απλή μίμηση της τυπογραφίας προσθέτει γενικευμένη σήμανση βασισμένη σε στοιχεία, τα οποία μπορούν να εμφωλεύονται το ένα μέσα στο άλλο και να φέρουν ιδιότητες. Ακόμα, το SGML διαχωρίζει τη δομή από το περιεχόμενο, κατεύθυνση προς την οποία αργότερα κινήθηκε και η HTML, με τα CSS. Πολλά από τα στοιχεία κειμένου προέρχονται από την τεχνική αναφορά ISO TR 9537, Techniques for using SGML (τεχνικές χρήσης της SGML), η οποία με τη σειρά της καλύπτει τα χαρακτηριστικά των πρώιμων γλωσσών μορφοποίησης κειμένου που χρησιμοποιούνταν από τα TYPSET και RUNOFF, και είχαν αναπτυχθεί στις αρχές της δεκαετίας του 1960 για το λειτουργικό σύστημα CTSS.

Ο Μπέρνερς Λι θεώρησε την HTML ως μια υλοποίηση του SGML. Αυτό ορίστηκε και επίσημα από το Internet Engineering Task Force (IETF) με τη δημοσίευση της πρώτης πρότασης για μια προδιαγραφή της HTML, στα μέσα του 1993, η οποία περιλάμβανε και έναν Ορισμό τύπου εγγράφου (DTD, Document Type Definition) της SGML, ο οποίος όριζε την γραμματική. Αυτό το πρόχειρο (draft) έληξε μετά την πάροδο έξι μηνών, αλλά περιέχει κάτι αξιοσημείωτο: την αναγνώριση της ετικέτας του NCSA Mosaic για την ενσωμάτωση εικόνων μέσα στο κείμενο, η οποία αντικατοπτρίζει την φιλοσοφία του IETF για ενσωμάτωση επιτυχημένων πρωτότυπων μέσα στα πρότυπα. Κάτι παρόμοιο περιείχε και το

ανταγωνιστικό πρόχειρο του Dave Raggett, «HTML+ (Hypertext Markup Format)», από τα τέλη του, το οποίο πρότεινε την προτυποποίηση μερικών ήδη υλοποιημένων δυνατοτήτων, όπως οι πίνακες και οι φόρμες.



Εικόνα 2-1 Ο φυλλομετρητής WorldWideWeb, το πρώτο γραφικό πρόγραμμα περιήγησης για το world wide web που φτιάχτηκε το 1993.

Μετά που τα πρόχειρα HTML και HTML+ έληξαν, στις αρχές του 1994, το IETF δημιούργησε την Ομάδα Εργασίας για την HTML, η οποία το 1995 ολοκλήρωσε την «HTML 2.0», με την πρόθεση να αποτελέσει την πρώτη προδιαγραφή πάνω στην οποία θα βασιζόνταν οι μελλοντικές υλοποιήσεις. Η HTML 2.0 δημοσιεύτηκε ως RFC 1866, και περιείχε ιδέες από τα πρόχειρα HTML και HTML+. Η αρίθμηση 2.0 σκόπευε απλά να ξεχωρίσει την νέα έκδοση από τα πρόχειρα που προηγήθηκαν.

Η HTML 3.0 προτάθηκε ως πρότυπο από το IETF, αλλά η πρόταση έληξε πέντε μήνες αργότερα χωρίς άλλη ενέργεια. Περιείχε πολλές από τις δυνατότητες που συμπεριλαμβάνονταν στην πρόταση του Raggett για την HTML+, όπως την υποστήριξη για πίνακες, τη ροή κειμένου γύρω από εικόνες και την προβολή πολύπλοκων μαθηματικών τύπων.

Το W3C ξεκίνησε την ανάπτυξη του δικού του Arena browser ως δοκιμαστική πλατφόρμα για την HTML 3 και για τα CSS αλλά η HTML 3.0 δεν πέτυχε, για διάφορους λόγους. Το πρόχειρο θεωρήθηκε υπερβολικά μεγάλο, με 150 σελίδες, και ο ρυθμός ανάπτυξης του browser, καθώς και ο αριθμός των ενδιαφερομένων μερών υπερέβαιναν τις δυνατότητες του IETF. Οι εταιρείες που διέθεταν browser, συμπεριλαμβανομένων των

Microsoft και Netscape εκείνο τον καιρό, επέλεξαν να υλοποιήσουν διαφορετικά υποσύνολα των δυνατοτήτων του πρόχειρου της HTML 3, καθώς και να εισάγουν τις δικές του επεκτάσεις σ' αυτό. Οι επεκτάσεις αυτές έλεγχαν εμφανισιακά στοιχεία των εγγράφων, αντίθετα με την «πεποίθηση [της ακαδημαϊκής κοινότητας μηχανικών] ότι το χρώμα, το παρασκήνιο, το μέγεθος και ο τύπος της γραμματοσειράς ήταν οπωσδήποτε έξω από το στόχος μιας γλώσσας της οποίας η μοναδική πρόθεση ήταν να καθορίσει πώς οργανώνεται ένα έγγραφο.» Ο Dave Raggett, συνεργάτης του W3C για πολλά χρόνια, σχολίασε ότι «Μέχρι ενός σημείου, η Microsoft έκτισε την επιχειρηματική της δραστηριότητα στον Ιστό επεκτείνοντας τις δυνατότητες της HTML.»

Η περαιτέρω ανάπτυξη κάτω από την επίβλεψη του IETF καθυστέρησε λόγω σύγκρουσης ενδιαφερόντων. Από το 1996 και μετά, οι προδιαγραφές της HTML τηρούνται, μαζί με ανάδραση από τους δημιουργούς λογισμικού, από το World Wide Web Consortium (W3C). Ωστόσο, το 2000 η HTML έγινε επίσης παγκόσμιο πρότυπο (ISO/IEC 15445:2000). Η τελευταία προδιαγραφή της HTML, η HTML 4.01 δημοσιεύτηκε από το W3C το 1999, και το 2001 δημοσιεύτηκαν επίσης και τα λάθη και οι παραλείψεις της (errata).

2.2.1 Εκδόσεις της XHTML

Η XHTML είναι ξεχωριστή γλώσσα η οποία ως αναδιαμόρφωση της HTML 4.01 με χρήση της XML 1.0 και η οποία συνεχίζει να αναπτύσσεται.

Η XHTML 1.0, δημοσιεύτηκε στις 26 Ιανουαρίου 2000, ως Σύσταση του W3C, μετά αναθεωρήθηκε και επανεκδόθηκε στις 1 Αυγούστου 2002. Προσφέρει τις ίδιες τρεις εκδοχές όπως η HTML 4.0 και 4.01, αναδιαμορφωμένες ως XML, με μικρούς περιορισμούς.

Η XHTML 1.1, δημοσιεύτηκε στις 31 Μαΐου 2001, ως Σύσταση του W3C. Βασίζεται στην XHTML 1.0 Strict, αλλά περιέχει μικρές αλλαγές, μπορεί να παραμετροποιηθεί, μπορεί να αναμορφωθεί χρησιμοποιώντας αρθρώματα της XHTML, τα οποία δημοσιεύτηκαν στις 10 Απριλίου 2001, ως Σύσταση του W3C.

Για την XHTML 2.0 δεν υπάρχει πρότυπο, και αυτή τη στιγμή είναι ένα πρόχειρο έγγραφο και θεωρείται ακόμα έργο σε εξέλιξη. Η XHTML 2.0 δεν είναι συμβατή με την XHTML 1.x και επομένως μπορεί πιο σωστά να χαρακτηριστεί ως μια νέα γλώσσα που είναι εμπνευσμένη από την XHTML παρά ως αναβάθμιση της υπάρχουσας XHTML 1.x.



Εικόνα 2-2 Το λογότυπο της HTML του W3C (αριστερά) και το νέο λογότυπο της HTML5.

Τον Οκτώβριο του 2009 το W3C σταμάτησε την εξέλιξη της XHTML με σκοπό να επικεντρωθεί στην ανάπτυξη της HTML5.

2.3 Σχετικά με τα πρότυπα

Όπως φαίνεται από τα παραπάνω δίνεται βάρος στις διαφορετικές εκδόσεις της γλώσσας και στην διαδικασία με την οποία υιοθετούνται διάφορα πρότυπα σαν διαφορετικές εκδόσεις. Παρακάτω θα εξετάσουμε αυτή τη διαδικασία και την χρησιμότητά της.

2.3.1 Διαδικασία υιοθέτησης προτύπων

Ένα πρότυπο αναφέρεται σε μία τεχνολογία που χρησιμοποιείται πρακτικά και καθορίζει τις λεπτομέρειες σχετικά με το πώς λειτουργεί αυτή η τεχνολογία. Η υιοθέτηση ενός προτύπου είναι πολύ σημαντική διαδικασία γιατί εγγυάται ότι προϊόντα από διαφορετικές εταιρείες έχουν την δυνατότητα να συνεργάζονται απρόσκοπτα μεταξύ τους.

Για τις τεχνολογίες που έχουν σχέση με το internet ο φορέας που είναι υπεύθυνος για την δημιουργία είναι το Internet Engineering Task Force (IETF). Όταν ξεκινάει η συζήτηση για μία νέα έκδοση μίας τεχνολογίας τότε ανοίγει ένα νέο πρόχειρο (working draft) που αναφέρεται σε αυτή. Οι εταιρείες, οι ειδικοί που ασχολούνται με τον τομέα καθώς και άλλοι σχετικοί φορείς καταθέτουν έγγραφα RFC (Request for Comments), τα οποία περιέχουν προτάσεις σχετικά με την τεχνολογία υπό συζήτηση. Κάθε έγγραφο RFC έχει έναν μοναδικό αριθμό, για παράδειγμα η έκδοση 2.0 της HTML είναι το έγγραφο RFC 1866. Δεν καταλήγουν όλα τα έγγραφα να γίνονται πρότυπα, καθώς κάποια από αυτά απορρίπτονται ή απορροφούνται από άλλα έγγραφα.

Από τη στιγμή που κατατίθεται ένα έγγραφο RFC γίνεται μία συζήτηση σχετικά με το περιεχόμενό του και τυχών βελτιώσεις. Οι φορείς που συμμετέχουν κάθε φορά στη συζήτηση διαφέρουν ανάλογα με την τεχνολογία που περιέχεται. Συνήθως για την νέα έκδοση της HTML αναλαμβάνει την συζήτηση το World Wide Web Consortium (W3C), ενώ για την έκδοση 5 της γλώσσας αυτό συνεργάζεται με το Web Hypertext Application Technology Working Group (WHATWG), το οποίο είναι μία άλλη επιτροπή που ιδρύθηκε από στελέχη εταιρειών που ασχολούνται με την γλώσσα. Οποιοσδήποτε μπορεί να καταθέσει μία πρόταση ή να σχολιάσει μία ήδη υπάρχουσα, ενώ όταν οι συμμετέχοντες στη συζήτηση συμφωνήσουν η πρόταση συνήθως εγκρίνεται ως πρότυπο.

2.3.2 Διαδικασία προτυποποίησης της HTML5

Η HTML5 είναι μια υπό ανάπτυξη γλώσσα σήμανσης για τον Παγκόσμιο Ιστό που όταν ετοιμαστεί θα είναι η επόμενη μεγάλη έκδοση της HTML (Γλώσσα Υπερκειμένου, HyperText Markup Language). Η ομάδα Web Hypertext Application Technology Working Group (WHATWG) άρχισε δουλειά σε αυτή την έκδοση τον Ιούνιο του 2004 με το όνομα Web Applications 1.0. Το Φεβρουάριο του 2010 το πρότυπο ήταν ακόμη σε κατάσταση "Last Call" στο WHATWG.

Η HTML5 προορίζεται για αντικατάσταση της HTML 4.01, της XHTML 1.0, και της DOM Level 2 HTML. Ο σκοπός είναι η μείωση της ανάγκης για ιδιόκτητα plug-in και πλούσιες διαδικτυακές εφαρμογές (RIA) όπως το Adobe Flash, το Microsoft Silverlight, το Apache Pivot, και η Sun JavaFX. Οι ιδέες πίσω από την HTML5 εμφανίστηκαν αρχικά το 2004 από την ομάδα WHATWG. Η HTML5 εμπεριέχει το πρότυπο Web Forms 2.0 που είναι επίσης της WHATWG.

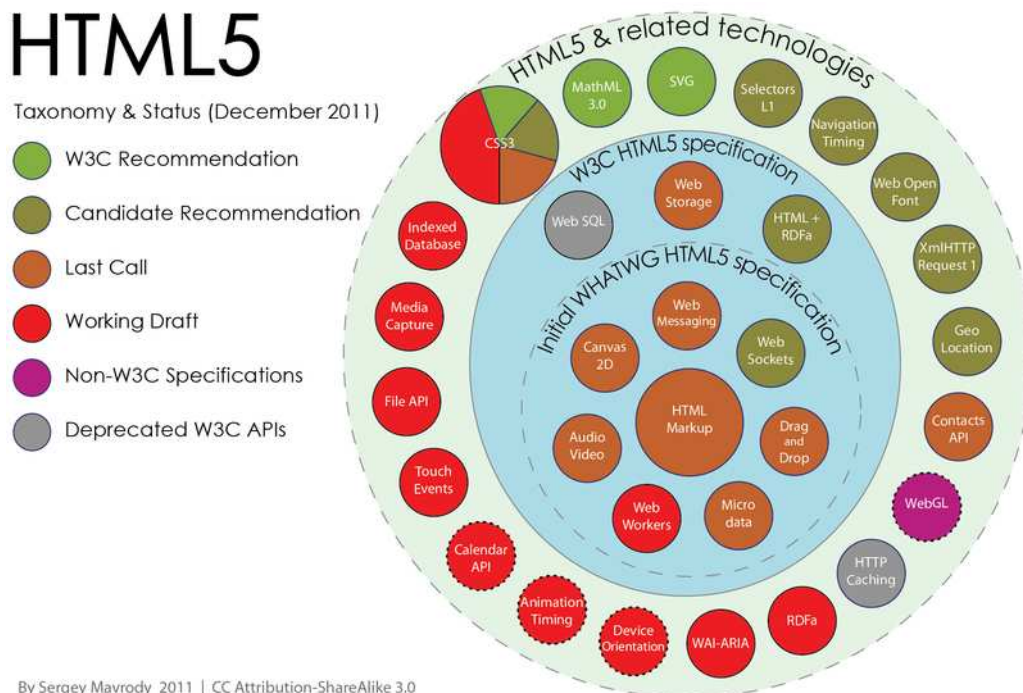
Το πρότυπο HTML5 υιοθετήθηκε ως αρχικό βήμα για τις εργασίες της νέας ομάδας εργασίας HTML του W3C το 2007. Αυτή η ομάδα εργασίας δημοσίευσε το Πρώτο Δημόσιο Working Draft του προτύπου στις 22 Ιανουαρίου 2008. Το πρότυπο είναι ακόμη υπό ανάπτυξη, και αναμένεται να παραμείνει έτσι για πολλά χρόνια, παρόλο που μέρη της HTML5 θα τελειώσουν και θα υποστηριχτούν από περιηγητές πριν το όλο πρότυπο φτάσει στη τελική κατάσταση Recommendation.

2.3.3 Εύρος του προτύπου HTML5

Σε αντίθεση με τις προηγούμενες εκδόσεις της HTML οι οποίες στόχευαν στο να δημιουργήσουν μία γλώσσα κατάλληλη για την περιγραφή της εμφάνισης των σελίδων και την βασική αλληλεπίδρασή τους με τον χρήστη, η HTML5 περιλαμβάνει πολλά τμήματα με τα οποία θα μπορούν να δημιουργηθούν ανεξάρτητες και αρκετά πολύπλοκες εφαρμογές. Βάση το εύρος των θεμάτων που καλύπτουν οι προτάσεις που αναπτύσσονται μπορεί να την χαρακτηριστεί σαν μία πλήρης γλώσσα προγραμματισμού.

Αυτό οφείλεται στο ότι καλείται να αντικαταστήσει ένα πλήθος τρίτων τεχνολογιών που σήμερα χρησιμοποιούνται στις υπάρχουσες ιστοσελίδες χωρίς να υπολείπεται σε δυνατότητες από αυτές. Επιπλέον πολλές από τις αλλαγές που έχουν προταθεί επιδιώκουν να διορθώσουν άμεσες ελλείψεις και κενά της γλώσσας.

Οι αλλαγές προφανώς δεν περιέχουν αλλαγές μόνο στην γλώσσα HTML αλλά προσθέτουν πολλά καινούργια στοιχεία στη Javascript και το CSS. Επίσης, με σκοπό να μην επαναληφθούν προβλήματα που παρουσιάστηκαν στο παρελθόν, η διατύπωση των προτύπων γίνεται με αρκετά μεγάλη λεπτομέρεια ώστε να είναι ξεκάθαρος ο τρόπος υλοποίησής τους.



By Sergey Mavrody 2011 | CC Attribution-ShareAlike 3.0

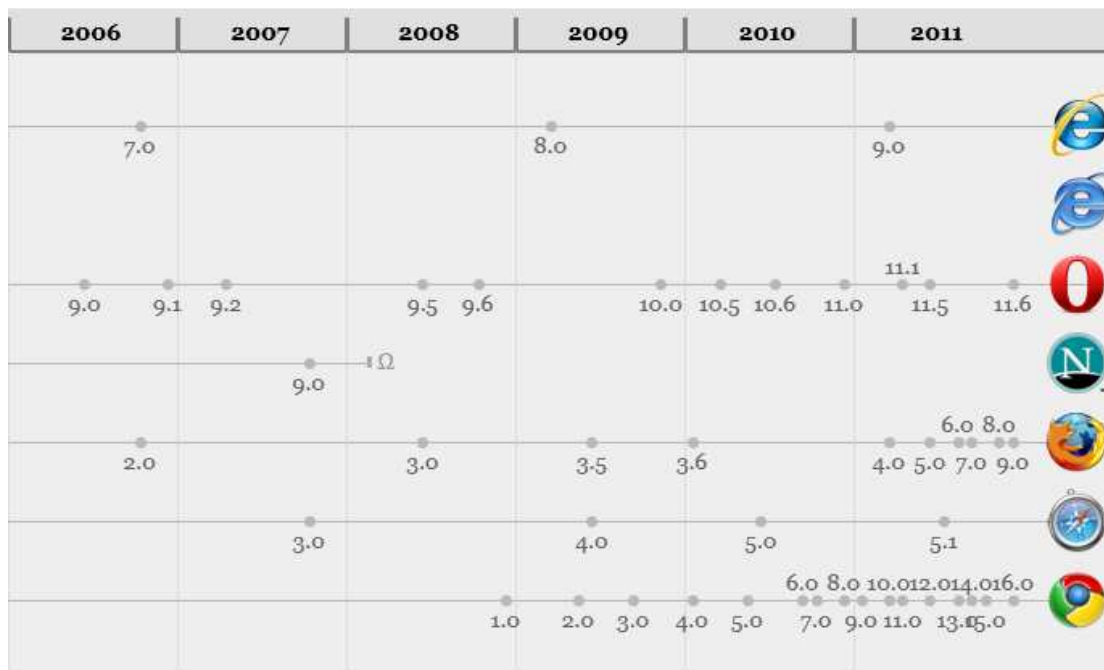
Εικόνα 2-3 Οι τεχνολογίες που περιλαμβάνονται στο πρότυπο HTML5. Φαίνονται οι τεχνολογίες που έχουν προταθεί από το W3C και το WHATWG, και η κατάσταση που βρίσκεται η κάθε μία.

2.3.4 Η απόκλιση της HTML5 από τη διαδικασία υιοθέτησης προτύπων

Ένα πρότυπο τέτοιου μεγέθους τυπικά χρειάζεται είκοσι χρόνια για να οριστικοποιηθεί, κάτι που τοποθετεί την τελική ημερομηνία στο 2022. Όμως η μεγαλύτερη διαφορά που έχει η γλώσσα HTML5 σε σχέση με παρόμοια πρότυπα είναι το γεγονός ότι οι ανάγκες που καλείται να ικανοποιήσει είναι ιδιαίτερα άμεσες και πιεστικές. Επιπλέον οι συμμετέχοντες στην διαδικασία και αυτοί που τελικά θα το υλοποιήσουν είναι οι εταιρείες και οι οργανισμοί που δημιουργούν τους σύγχρονους browsers.

Εξαιτίας αυτών η διαδικασία έχει οριστικοποίησης είναι διαφορετική από την τυπική. Όταν κατατίθεται μία πρόταση για μία νέα δυνατότητα ή για τροποποίηση μίας που έχει ήδη προταθεί υπάρχει περίπτωση να υλοποιηθεί από κάποιους ή όλους τους κατασκευαστές με τον ίδιο ή διαφορετικό τρόπο. Όσο προχωράει η συζήτηση ο τρόπος με τον οποίο θα λειτουργεί αυτό το τμήμα της γλώσσας μπορεί να μεταβληθεί και να μεταβληθούν και οι υπάρχουσες υλοποιήσεις μέχρι να καταλήξει στην τελική της μορφή. Όταν γίνει αυτό τότε η δυνατότητα συνήθως υλοποιείται.

Η διαδικασία που περιγράφεται παραπάνω συνεπάγεται πως αρκετά στοιχεία της γλώσσας είναι ήδη υλοποιημένα και λειτουργούν, είτε με την τελική τους μορφή είτε με την ενδιάμεση που έχει υλοποιήσει κάθε διαφορετικός κατασκευαστής. Η ημερομηνία που το πρότυπο θα φτάσει την τελική του μορφή δεν μπορεί να υπολογιστεί με ακρίβεια, αφού διαφορετικά κομμάτια του βρίσκονται σε διαφορετικά στάδια ολοκλήρωσης και προχωράνε με διαφορετικό ρυθμό.



Εικόνα 2-4 Χρονοδιάγραμμα των διαφορετικών εκδόσεων κάθε προγράμματος περιήγησης, από το 2006 έως το 2011.

Επιπλέον κάθε διαφορετική έκδοση του ίδιου browser μπορεί να υποστηρίξει διαφορετικά στοιχεία σε διαφορετικό βαθμό ή να τα υλοποιεί με διαφορετικό τρόπο. Οι περισσότεροι σύγχρονοι browser αναβαθμίζονται συνεχώς, υιοθετώντας όλο και περισσότερα χαρακτηριστικά με κάθε νέα έκδοση.

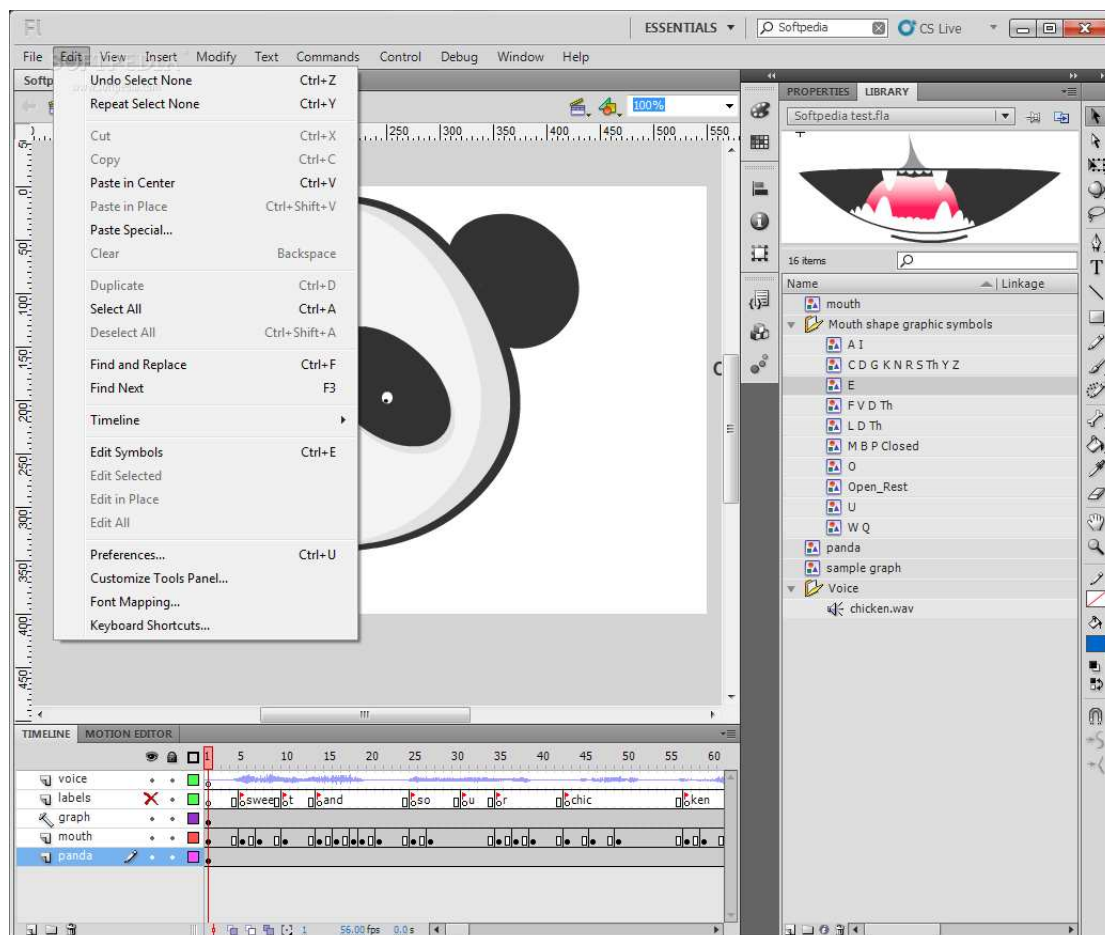
2.4 Περιορισμοί της HTML

Για να γίνει περισσότερο κατανοητό το σκεπτικό ανάπτυξης της HTML5 αξίζει να μελετήσουμε τα προβλήματα που υπάρχουν σήμερα για την κατασκευή σελίδων με τις προηγούμενες εκδόσεις της HTML.

Τα προβλήματα χωρίζονται σε δύο κατηγορίες. Η πρώτη έχει να κάνει με την έλλειψη χαρακτηριστικών από την HTML, όπως για παράδειγμα την έλλειψη animations ή την δυσκολία εμφάνισης αρκετά πολύπλοκων γραφικών.

2.4.1 Πληθώρα τρίτων τεχνολογιών

Για να αντιμετωπιστούν αυτές οι ελλείψεις αναπτύχθηκε μία πληθώρα τρίτων τεχνολογιών με σκοπό την ενσωμάτωσή τους σε μία σελίδα. Η πιο γνωστή τέτοια τεχνολογία είναι το Flash της εταιρείας Adobe, το οποίο χρησιμοποιείται κατά κόρων μεταξύ άλλων για την δημιουργία animations, παιχνιδιών και την προβολή βίντεο. Πέρα από το flash έχουν γνωρίσει μεγάλη εξάπλωση το πλαίσιο ανάπτυξης Silverlight από την εταιρεία Microsoft, το JavaFX που είναι βασισμένο στη γλώσσα Java και άλλες μικρότερες εφαρμογές.



Εικόνα 2-5 Το περιβάλλον ανάπτυξης εφαρμογών της τεχνολογίας Adobe Flash. Πρόκειται για το πιο διαδεδομένο πρόσθετο, κυρίως επειδή διευκολύνει την προβολή εντυπωσιακών γραφικών με συνέπεια σε όλους τους browsers. Η διάδοσή του ήταν παλιότερα τόσο μεγάλη ώστε αρκετές σελίδες να γράφονται εξ' ολοκλήρου σε αυτό.

Το πρόβλημα με τις παραπάνω τεχνολογίες είναι ότι για να λειτουργήσουν είναι απαραίτητο να υπάρχει εγκαταστημένο στον υπολογιστή του χρήστη, πέρα από το πρόγραμμα περιήγησης, και ένα επιπλέον πρόγραμμα που θα έχει ρυθμιστεί κατάλληλα για να συνεργάζεται με τον browser. Αυτό είναι προβληματικό από πολλές απόψεις.

Καθώς αυτά τα προγράμματα αναπτύσσονται από τρίτες εταιρείες δεν είναι εγγυημένο ότι θα υπάρχουν διαθέσιμα για όλες τις πλατφόρμες, όπως για παράδειγμα το Silverlight το οποίο ποτέ δεν κυκλοφόρησε για λειτουργικά συστήματα Linux. Ακόμη και αν είναι διαθέσιμα για μία πλατφόρμα δεν υπάρχει καμία εγγύηση ότι ο χρήστης θα θελήσει να τα εγκαταστήσει, είτε επειδή δεν θα έχει τις απαραίτητες γνώσεις είτε για άλλους λόγους. Για παράδειγμα το πρόγραμμα Flash Player που είναι υπεύθυνο για την αναπαραγωγή αντικειμένων τύπου Flash είναι γνωστό ότι έχει αρκετά κενά ασφαλείας, είναι πάρα πολύ απαιτητικό σε επεξεργαστική ισχύ και μνήμη και σε φορητές συσκευές προκαλεί πάρα πολύ μεγάλη κατανάλωση ενέργειας.

Αν ένας προγραμματιστής που κατασκευάζει μία σελίδα δεν είναι σίγουρος ότι κάθε χρήστης θα μπορεί να δει το ίδιο περιεχόμενο, αφού δεν υπάρχει κάποια εγγύηση πως θα έχει εγκαταστημένο το αντίστοιχο πρόσθετο, αυτόματα περιορίζει το κοινό της σελίδας του ή δεν χρησιμοποιεί την συγκεκριμένη τεχνολογία.

2.4.2 Αποκλίσεις των υλοποιήσεων από το πρότυπο

Ένα δεύτερο πρόβλημα που υπάρχει με τις σελίδες που είναι γραμμένες σε HTML είναι το γεγονός πως διαφορετικοί browsers μπορεί να τις εμφανίζουν με διαφορετικό τρόπο, είτε λόγω διαφορετικής ερμηνείας των προδιαγραφών είτε λόγω εσωτερικών προβλημάτων (bugs).

Αυτό έχει σαν αποτέλεσμα ο κώδικας που λειτουργεί σωστά σε ένα πρόγραμμα περιήγησης να μην εμφανίζεται σε κάποιο άλλο. Για την επίλυση του προβλήματος χρειάζεται να δοκιμάζεται ο κώδικας μίας σελίδας σε όλους τους browsers και να υλοποιείται διαφορετικά σε όσους δεν λειτουργού σωστά. Κάτι τέτοιο όμως αυξάνει κατακόρυφα το κόστος και τον χρόνο ανάπτυξης. Σαν αποτέλεσμα πολλές σελίδες γράφονται στοχεύοντας μόνο τις πιο πρόσφατες εκδόσεις των browsers και δεν λειτουργούν σωστά στις παλιότερες, και συνήθως παροτρύνουν τον χρήστη να αναβαθμίσει το πρόγραμμα περιήγησης, κάτι το οποίο όμως δεν μπορούν για διάφορους λόγους να κάνουν όλοι οι χρήστες.

Test	IE 5-5	IE 6	IE 7	IE 8	IE 9	IE10 pr2	FF 7.0 Win	FF 6.1 Mac	Saf 5.1 Win	Saf 5.1 Mac	Chrome 14 Win	Chrome 13 Mac	Opera 11.51 Win	Opera 11.51 Mac
<p>blur</p> <p>When an element loses the focus.</p> <ul style="list-style-type: none"> Firefox Mac, Safari, and Chrome sometimes don't support these events on links and/or form fields. See detail page for bug descriptions. 	yes	yes	yes	yes	yes	yes	yes	incomplete	incomplete	incomplete	incomplete	incomplete	yes	yes
<p>change</p> <p>When a form field value changes.</p> <ul style="list-style-type: none"> IE 5-8 has a serious bug in its handling of this event on checkboxes and radios. IE and Opera have a serious bug in their handling of this event when the user uses the keyboard to change a select. 	buggy	buggy	buggy	buggy	buggy	buggy	yes	yes	yes	yes	yes	yes	incomplete	incomplete
<p>click</p> <p>When a mousedown and mouseup event occur on the same element OR an element is activated by the keyboard.</p> <ul style="list-style-type: none"> All browsers but Safari have issues in preventing the default on form fields. Browsers judged Almost have two issues, the others (except Safari) one. 	almost	almost	almost	almost	almost	almost	almost	almost	almost	almost	almost	almost	almost	almost
<p>contextmenu</p> <p>When the user right-clicks to get the context menu.</p> <p>Preventing the default (i.e. preventing the context menu from appearing) is the whole point of this event.</p> <ul style="list-style-type: none"> IE8 always prevents the default when this event is handled on the document. 	yes	buggy	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes

Εικόνα 2-6 Ένας πίνακας συμβατότητας διαφόρων εκδόσεων των browsers σχετικά με το αν υλοποιούν σωστά το πρότυπο HTML. Η πλήρης έκδοση έχει πάνω από 150 ασυμβατότητες.

Ένα χαρακτηριστικό παράδειγμα κακής χρήσης των προτύπων παρουσιάστηκε κατά την έκδοση της HTML 3.0, η οποία ποτέ δεν επισημοποιήθηκε. Οι μεγάλοι κατασκευαστές browsers της εποχής ήταν οι εταιρείες Microsoft και Netscape, οι οποίες υιοθέτησαν διαφορετικά τμήματα της προδιαγραφής με αποτέλεσμα αρκετά πράγματα που λειτουργούσαν στα προϊόντα της μίας να μην λειτουργούν στα προϊόντα της άλλης.

Αυτός είναι ο λόγος που κατά την διαδικασία ανάπτυξης της HTML5 δίνεται αρκετά μεγάλο βάρος στην ακριβή διατύπωση της προδιαγραφής και στην παροχή διευκρινίσεων για όλες τις λεπτομέρειες που μπορεί να παρερμηνευτούν.

3 Παρουσίαση της HTML5

3.1 Εισαγωγή

Παρακάτω θα παρουσιάσουμε τις νέες δυνατότητες που δίνει η HTML5. Αρχικά πρέπει να γίνει κατανοητό πως δεν πρόκειται για μία νέα γλώσσα που σχεδιάζεται από την αρχή, αλλά βασίζεται πάνω στα προηγούμενα πρότυπα και πρέπει να λειτουργεί σε συνδυασμό με αυτά. Κατά συνέπεια δεν έχει σημασία μόνο η παρουσίαση των δυνατοτήτων όσο και το πώς οι νέες προδιαγραφές διαφέρουν από τις υπάρχουσες και πως μπορούν να χρησιμοποιηθούν σε συνδυασμό με αυτές ώστε οι σελίδες να λειτουργούν (μέχρι κάποιο βαθμό) και σε παλιότερους browsers.

3.2 Βασικά στοιχεία

Για την παρουσίαση των βασικών αλλαγών θα δούμε πώς πρέπει να τροποποιηθεί μία σελίδα της HTML 4.01 / XML ώστε να ακολουθεί το πρότυπο της HTML5.

3.2.1 Το στοιχείο DOCTYPE

Η πρώτη γραμμή κάθε σελίδας HTML ορίζει το είδος του εγγράφου. Σε μία παλιά σελίδα αυτό γράφεται ως:

```
<!DOCTYPE html  
PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

Η χρήση του doctype ξεκίνησε όταν κατά την ανάπτυξη του Internet Explorer 5 οι προγραμματιστές της εταιρείας Microsoft συνειδητοποίησαν πως καθώς οι έκδοση που έφτιαχναν ήταν πολύ πιο ακριβής στην υλοποίηση του προτύπου HTML σε σχέση με τις προηγούμενες, οι περισσότερες σελίδες, που είχαν γραφτεί ώστε να αντιμετωπίζουν τα προβληματικά αποτελέσματα των προηγούμενων εκδόσεων του browser, εμφανιζόντουσαν με λάθος τρόπο.

Η λύση που επινόησαν να διαβάζουν την πρώτη γραμμή της σελίδας, που είναι το doctype, το οποίο κατά κανόνα έλειπε από τις παλιότερες εκδόσεις. Όσες σελίδες δεν το είχαν εμφανιζόντουσαν με την χρήση των ελαττωμάτων που υπήρχαν μέχρι τότε, ενεργοποιώντας μία κατάσταση του λέγεται quirks mode. Οι νεότερες σελίδες, που είχαν γραφτεί για να υποστηρίξουν τα μέχρι τότε standards, χρησιμοποιούσαν το standards mode.

Όταν ο οργανισμός Mozilla δοκίμαζε την έκδοση 1.1 του προγράμματος περιήγησής του, το οποίο εφαρμόζε τα πρότυπα με ακόμη μεγαλύτερη ακρίβεια, ανακάλυψε ότι οι περισσότερες σελίδες εκμεταλλεύοντουσαν ελαττώματα στο standards mode του Internet Explorer, τα οποία ο Mozilla είχε διορθώσει. Έτσι δημιουργήθηκε και μία Τρίτη κατάσταση, το almost standards mode, το οποίο εξομοίωνε την προηγούμενη κατάσταση.

Σήμερα οι περισσότεροι browsers υποστηρίζουν αυτές τις τρεις καταστάσεις, δηλαδή το quirks mode, το almost standards mode και το standards mode, και υπάρχουν 15 περίπου doctype τα οποία καθορίζουν πιο από αυτά θα χρησιμοποιηθεί. Καθώς κατά την ανάπτυξη της HTML5 θεωρείται δεδομένο πώς όλοι οι browsers θα εφαρμόζουν αυστηρά το πρότυπο, η χρήση πολύπλοκων doctype είναι περιττή και τα παλιότερα έχουν αντικατασταθεί με ένα απλούστερο:

```
<!doctype html>
```

3.2.2 Το στοιχείο html

Κάθε αρχείο HTML ξεκινάει με το tag html, το οποίο περιέχει όλα τα υπόλοιπα. Σε παλιότερες εκδόσεις αυτό ήταν:

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
```

Αυτό το στοιχείο είναι έγκυρο με αυτή την μορφή, αλλά κάποιες από τις ιδιότητές του είναι περιττές. Συγκεκριμένα η ιδιότητα xmlns καθορίζει τον χώρο ονομάτων στον οποίο ανήκουν τα ονόματα των tags. Καθώς εξ' ορισμού όλα τα στοιχεία της HTML5 ανήκουν σε αυτόν τον χώρο ονομάτων δεν χρειάζεται να δηλώνεται κάθε φορά.

Τα υπόλοιπα δύο στοιχεία καθορίζουν την γλώσσα και είναι απομεινάρια από το πρότυπο XHTML. Δεν είναι πλέον αναγκαίο να καθορίζονται και τα δύο, και αρκεί η ιδιότητα lang για να περιγράψει την γλώσσα του αντικειμένου.

3.2.3 Το στοιχείο head

Το πρώτο tag που περιέχεται σε κάθε έγγραφο είναι το head, το οποίο περιέχει πληροφορίες για αυτό το έγγραφο. Αν και το ίδιο το head δεν έχει αλλάξει, τα περιεχόμενα του έχουν διαφοροποιηθεί. Έστω ότι έχουμε το παρακάτω τυπικό head:

```
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>My Weblog</title>
  <link rel="stylesheet" type="text/css" href="style-original.css" />
  <link rel="alternate" type="application/atom+xml" title="My Weblog feed"
href="/feed/" />
  <link rel="search" type="application/opensearchdescription+xml" title="My
Weblog search" href="opensearch.xml" />
</head>
```

3.2.3.1 Το στοιχείο meta

Η κωδικοποίηση χαρακτήρων δίνεται από τον κώδικα

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

Ο σκοπός αυτής της γραμμής ήταν να καθορίζει μία κεφαλίδα του προτύπου HTTP μέσω της οποίας ενημερώνει τον browser για την κωδικοποίηση. Εδώ δεν είναι πια απαραίτητο να καθορίζεται η κεφαλίδα, γιατί αρκεί μόνο το μέρος charset:

```
<meta charset="utf-8" >
```

Όλοι οι browsers υποστηρίζουν ήδη τη συγκεκριμένη μορφή σε περίπτωση που στην προηγούμενη μορφή ο χρήστης δεν είχε βάλει διπλά εισαγωγικά.

3.2.3.2 Τα στοιχεία link

Τα στοιχεία link καθορίζουν επιπλέον αρχεία τα οποία χρησιμοποιεί η σελίδα. Το πιο συνηθισμένο είναι για τα αρχεία CSS, τα οποία παλιότερα φορτώνονταν ως εξής:

```
<link rel="stylesheet" type="text/css" href="style-original.css" />
```

Αυτό έχει μετατραπεί σε:

```
<link rel="stylesheet" href="style-original.css" />
```

Ο καθορισμός του τύπου σε text/css δεν είναι απαραίτητος γιατί είναι η κύρια υπόθεση που κάνουν όλοι οι browsers. Χρειάζεται μόνο να καθοριστεί αν δεν πρόκειται για έγγραφο μορφοποίησης τύπου css, κάτι που μέχρι τώρα δεν υπάρχει.

3.2.4 Το στοιχείο body

Η μορφή του στοιχείου body δεν έχει αλλάξει αλλά μέσα σε αυτό μπορούν να μπουν αρκετά νέα στοιχεία, με στόχο να περιγράψουν καλύτερα την εννοιολογική δομή του εγγράφου. Ενώ σε παλιότερα έγγραφα τα στοιχεία ήταν div με διαφορετικές κλάσεις που περιέγραφαν το πώς θα φαίνονται, πλέον μπορούν να χρησιμοποιηθούν περισσότερων ειδών στοιχεία.

3.2.4.1 Το στοιχείο section

Το στοιχείο section αναπαριστά ένα τμήμα ενός εγγράφου ή μίας εφαρμογής. Πρόκειται για μία θεματική ενότητα, συνήθως με μία επικεφαλίδα. Παραδείγματα section θα μπορούσαν να είναι τα κεφάλαια ενός βιβλίου, τα αριθμημένα τμήματα μίας δημοσίευσης ή, στην αρχική σελίδα, τα κομμάτια για νέα προϊόντα, ανακοινώσεις κτλ.

3.2.4.2 Το στοιχείο nav

Το στοιχείο nav αναπαριστά ένα τμήμα μίας σελίδας με συνδέσμους σε άλλες σελίδες. Δεν αναφέρεται σε οποιαδήποτε ομάδα συνδέσμων, αλλά σε αυτούς που επιτρέπουν την πλοήγηση σε μία σελίδα, πχ. Στο κυρίως μενού ή σε έναν χώρο στο κάτω μέρος αρκετών σελίδων που περιέχει έναν συνοπτικό πίνακα με τις υπηρεσίες που παρέχονται. Έστω ο παρακάτω κώδικας που παρουσιάζει μία λίστα συνδέσμων:

```
<div id="nav">
  <ul>
    <li><a href="#">home</a></li>
    <li><a href="#">blog</a></li>
    <li><a href="#">about</a></li>
  </ul>
</div>
```

Με τη χρήση του στοιχείου nav καταλήγουμε στο:

```
<nav>
  <ul>
    <li><a href="#">home</a></li>
    <li><a href="#">blog</a></li>
    <li><a href="#">about</a></li>
  </ul>
</nav>
```

3.2.4.3 Το στοιχείο article

Το στοιχείο article αναπαριστά μία ενότητα μίας σελίδας που αποτελεί μία ανεξάρτητη σύνθεση η οποία θα μπορούσε να διανεμηθεί ή να επαναχρησιμοποιηθεί σαν απόσπασμα. Αυτό θα μπορούσε να είναι ένα άρθρο σε ένα περιοδικό ή μία εφημερίδα, μία δημοσίευση σε ένα forum ή ένα blog ή τα σχόλια των χρηστών.

3.2.4.4 Το στοιχείο aside

Το στοιχείο aside αναπαριστά ένα τμήμα περιεχομένου το οποίο είναι σχετικό με το περιεχόμενο που βρίσκεται πριν και μετά από αυτό αλλά δεν έχει άμεση σχέση με το κυρίως περιεχόμενο της σελίδας. Αυτό μπορεί να χρησιμοποιηθεί για την εμφάνιση τυπογραφικών στοιχείων όπως τα εισαγωγικά, για διαφημίσεις, για να ομαδοποιήσει στοιχεία τύπου nav και οτιδήποτε θεωρείται διαφορετικό από το κυρίως σώμα της σελίδας.

3.2.4.5 Το στοιχείο hgroup

Το στοιχείο hgroup αναπαριστά την επικεφαλίδα μίας ενότητας. Χρησιμοποιείται για την ομαδοποίηση των στοιχείων h1 με h6, τα οποία μπορούν να χρησιμοποιηθούν για μία επικεφαλίδα με πολλαπλά επίπεδα.

3.2.4.6 Το στοιχείο *header*

Το στοιχείο `header` αναπαριστά μία ομάδα εισαγωγικών στοιχείων ή βοηθημάτων πλοήγησης. Συνήθως περιέχει μία ομάδα στοιχείων μέσα σε ένα `hgroup`, αν και δεν είναι απαραίτητο. Επίσης μπορεί να περιέχει τα περιεχόμενα μίας σελίδας ή μία φόρμα αναζήτησης.

Ένα παράδειγμα είναι πως η επικεφαλίδα μίας σελίδας γράφεται συνήθως σαν:

```
<div id="header">
  <h1>My Weblog</h1>
  <p class="tagline">A lot of effort went into making this easy.</p>
</div>
```

Πλέον μπορεί να γραφτεί:

```
<header>
  <hgroup>
    <h1>My Weblog</h1>
    <h2>A lot of effort went into making this effortless.</h2>
  </hgroup>
</header>
```

Ο προηγούμενος τρόπος δεν είναι λάθος, αλλά με τα νέα στοιχεία φαίνεται καλύτερα ο ρόλος που παίζει κάθε τμήμα.

3.2.4.7 Το στοιχείο *footer*

Το στοιχείο `footer` περιέχει πληροφορίες σχετικά με το στοιχείο που το περιέχει. Τυπικά περιέχει πληροφορίες για τον δημιουργό του στοιχείου αυτού, στοιχεία για τα πνευματικά δικαιώματα κα. Επίσης μπορεί να περιέχει στοιχεία τύπου `section` σε περιπτώσεις που πρόκειται για αλφαβητικά ευρετήρια, έναν μακρύ επίλογο ή συμφωνίες χρήσης.

3.2.4.8 Το στοιχείο *time*

Αυτό το στοιχείο περιέχει μία ώρα ή μία ημερομηνία. Μπορεί να πρόκειται είτε για μία ώρα σε 24ωρη μορφή είτε για μία ακριβή ημερομηνία του Γρηγοριανού ημερολογίου με προαιρετικές πληροφορίες σχετικά με την ζώνη ώρας. Για παράδειγμα το παρακάτω στοιχείο κρατάει μία ημερομηνία:

```
<p class="post-date">October 22, 2009</p>
```

Αυτό μπορεί να αλλάξει σε:

```
<time datetime="2009-10-22" pubdate>October 22, 2009</time>
```

Με αυτό τον τρόπο περιέχεται και η πληροφορία για την ακριβή ημερομηνία, ενώ αν είναι απαραίτητη και η ώρα γίνεται:

```
<time datetime="2009-10-22T13:59:47-04:00" pubdate>
  October 22, 2009 1:59pm EDT
</time>
```

3.2.4.9 Το στοιχείο *mark*

Το στοιχείο `mark` αναπαριστά κείμενο το οποίο έχει χαρακτηριστεί για να δοθεί περισσότερο βάρος ή σαν αναφορά.

3.2.5 Αντιμετώπιση των νέων στοιχείων από παλιότερους browsers

Αν και τα νέα στοιχεία θα εμφανίζονται με τον επιθυμητό τρόπο σε σύγχρονους φυλλομετρητές, έχει σημασία το πώς θα τα χειριστούν προγράμματα παλιότερης τεχνολογίας. Συγκεκριμένα μας ενδιαφέρουν δύο παράμετροι, το πώς θα παρουσιαστεί το στοιχείο και τι στοιχεία θα θεωρηθεί ότι μπορεί να περιέχει.

Σχετικά με την εμφάνιση για κάθε στοιχείο θα θεωρηθεί πως είναι `inline`, κάτι το οποίο δεν είναι επιθυμητό. Για να αλλάξει αυτό απαιτείται ένας νέος κανόνας στο CSS, συγκεκριμένα:

```
article,aside,details,figcaption,figure,
footer,header,hgroup,menu,nav,section {
  display:block;
}
```

Οι εκδόσεις του Internet Explorer πριν από την 9^η δεν εφαρμόζουν κανένα είδος μορφοποίησης σε στοιχεία που δεν αναγνωρίζουν και δεν υπάρχει τρόπος για να τα εμφανίσουν.

3.3 Φόρμες

Η HTML5 φέρνει μεγάλες αλλαγές και στις φόρμες. Αν και πρόκειται για στοιχεία HTML τις αναλύουμε σε διαφορετική ενότητα καθώς το εύρος των αλλαγών είναι αρκετά μεγάλο. Η πλειοψηφία αυτών των αλλαγών περιλαμβάνονταν αρχικά στην προδιαγραφή Web Forms 2.0 αλλά στην πορεία συμπεριλήφθησαν στην HTML5.

Αρκετά από αυτά τα στοιχεία μπορούν να χρησιμοποιηθούν ήδη, αλλά σε περίπτωση που δεν υποστηρίζονται από το πρόγραμμα περιήγησης θα αγνοηθούν.

3.3.1 Κείμενο κράτησης θέσης

Το κείμενο κράτησης θέσης (placeholder text) είναι το κείμενο που εμφανίζεται μέσα σε ένα πεδίο κειμένου, που δίνει οδηγίες στον χρήστη για τις πληροφορίες που πρέπει να εισάγει στο πεδίο. Ένα πεδίο με τέτοιο κείμενο φαίνεται όπως παρακάτω.



Όταν ο χρήστης κάνει κλικ σε αυτό τότε το κείμενο χάνεται ώστε να μπορέσει να συμπληρώσει το πεδίο.



Ο κώδικας με τον οποίο παράγεται φαίνεται παρακάτω:

```
<form>
  <input name="q" placeholder="Go to a Website">
  <input type="submit" value="Search">
</form>
```

Αν ένας browser δεν υποστηρίζει την ιδιότητα απλά θα την αγνοήσει, οπότε μπορεί να χρησιμοποιηθεί με ασφάλεια. Η ιδιότητα placeholder υποστηρίζεται από τις παρακάτω εκδόσεις φυλλομετρητών:

Υποστήριξη Placeholder						
IE	Firefox	Safari	Chrome	Opera	iPhone	Android
10.0+	4.0+	4.0+	4.0+	11.0+	4.0+	2.1+

3.3.2 Αυτόματη εστίαση

Πολλές σελίδες χρησιμοποιούν javascript για να εστιάσουν αυτόματα σε ένα πεδίο, στο οποίο ο χρήστης μπορεί να γράψει αυτόματα, όπως για παράδειγμα μπορεί η αρχική σελίδα αναζήτησης του Google. Αν και αυτό μπορεί να είναι βολικό σε κάποιες περιπτώσεις, σε κάποιες άλλες μπορεί να είναι ενοχλητικό, όπως για παράδειγμα αν ο χρήστης θελήσει να κυλίσει την σελίδα προς τα κάτω πατώντας το κενό, αφού αντί να κυλίσει η σελίδα θα γράψει ένα κενό στο πεδίο. Επιπλέον αν ενώ φορτώνεται η σελίδα ο χρήστης επιλέξει ένα διαφορετικό πεδίο, με το τέλος της φόρτωσης θα επιλεγεί αυτόματα αυτό που έχει οριστεί.

Καθώς η αυτόματη εστίαση με javascript μπορεί να είναι ενοχλητική σε συγκεκριμένες περιπτώσεις η HTML5 εισάγει μία ιδιότητα autofocus στα πεδία μίας φόρμας. Με το που φορτώνεται ένα πεδίο που έχει αυτή την ιδιότητα το focus θα μεταφερθεί σε αυτό, κάνοντας το πιο εύχρηστο.

Για να εστιάζεται αυτόματα ένα πεδίο αρκεί ο παρακάτω κώδικας.


```
<form>
  <input name="q" autofocus>
  <input type="submit" value="Search">
</form>
```

Αν η ιδιότητα δεν υποστηρίζεται από έναν browser θα αγνοηθεί. Σε περίπτωση που αυτή η συμπεριφορά είναι αναγκαίο να υπάρχει μπορεί να χρησιμοποιηθεί σε συνδιασμό με την javascript ώστε να επιβάλλεται και σε παλιότερα προγράμματα περιήγησης ως εξής:

```
<form name="f">
  <input id="q" autofocus>
  <script>
    if (!("autofocus" in document.createElement("input"))) {
      document.getElementById("q").focus();
    }
  </script>
  <input type="submit" value="Go">
</form>
```

Η ιδιότητα autofocus υποστηρίζεται από τις παρακάτω εκδόσεις των προγραμμάτων περιήγησης.

Autofocus support						
IE	Firefox	Safari	Chrome	Opera	iPhone	Android
10.0+	4.0+	4.0+	3.0+	10.0+	.	3.0+

3.3.3 Πεδία διευθύνσεων email

Αν και όλα τα παλιότερα είδη πεδίων υποστηρίζονται στην HTML5 προστίθενται και κάποια νέα. Το πρώτο από αυτά είναι το πεδίο email, το οποίο ορίζεται ως εξής:

```
<form>
  <input type="email">
  <input type="submit" value="Go">
</form>
```

Αυτό δημιουργεί ένα απλό πεδίο κειμένου όπως φαίνεται παρακάτω.

Όμως αν ο χρήστης δοκιμάσει να εισάγει κείμενο που δεν είναι έγκυρη διεύθυνση email τότε εμφανίζεται σαν λάθος και πρέπει να το διορθώσει. Επιπλέον αν αγνοήσει το

λάθος και δοκιμάσει να παραδώσει την φόρμα δεν θα του επιτραπεί μέχρι να διορθώσει το λάθος του.

A screenshot of a web form. The input field contains the text "invalid" and is highlighted with a red border. To the right of the input field is a "Go" button. Below the input field, a message box displays the text "Please enter an email address."

Αυτό είναι εξαιρετικά χρήσιμο γιατί δεν είναι απαραίτητο να χρησιμοποιηθεί επιπλέον κώδικας javascript που μέχρι τώρα αναλάμβανε αυτή τη διαδικασία. Επιπλέον σε παλιότερες εκδόσεις το πεδίο email εμφανίζεται απλά σαν ένα πεδίο κειμένου.

Πέρα από αυτή την χρησιμότητα ένα άλλο πλεονέκτημα είναι ότι οι browsers των κινητών τηλεφώνων εμφανίζουν μία ειδική μορφή του πληκτρολογίου που είναι πιο κατάλληλη για την εισαγωγή διευθύνσεων ηλεκτρονικού ταχυδρομείου, όπως για παράδειγμα ο Safari για iOS.



3.3.4 Διευθύνσεις internet

Πολλές φορές σε ένα πεδίο ο χρήστης πρέπει να συμπληρώσει μία διεύθυνση URL. Στην HTML5 υπάρχει ένα νέο είδος πεδίου για αυτόν τον τύπο δεδομένων.

```
<form>
  <input type="url">
  <input type="submit" value="Go">
```

```
</form>
```

Όπως και στα πεδία email εμφανίζεται σαν κανονικό πεδίο κειμένου αλλά απαιτεί από τον χρήστη να εισάγει μία σωστή διεύθυνση URL.

Οι browsers των κινητών και σε αυτή την περίπτωση μπορούν να εμφανίσουν μία κατάλληλη μορφή του πληκτρολογίου.

3.3.5 Εισαγωγή αριθμών

Πολλές φορές είναι αναγκαίο να ζητηθεί από τον χρήστη να εισάγει νούμερα. Αυτή είναι μία διαδικασία κατά την οποία χρειάζεται να ελέγξει αν ο χρήστης έχει όντως εισάγει νούμερο και αν αυτό είναι έγκυρο. Αυτό μπορεί να διευκολυνθεί με την εισαγωγή ενός πεδίου number:

```
<form>
```

```
<input type="number"
  min="0"
  max="10"
  step="2"
  value="6">
<input type="submit" value="Go">
</form>
```

Όπως φαίνεται ένα πεδίο number έχει τρεις νέες ιδιότητες:

- Min, που καθορίζει την ελάχιστη επιτρεπτή τιμή.
- Max, που είναι η μέγιστη επιτρεπτή τιμή.
- Step, που είναι ο ρυθμός μεταβολής.

Αν ο φυλλομετρητής το υποστηρίζει θα εμφανιστεί ως εξής:

A screenshot of a web browser interface. It shows a single-line text input field containing the number '6'. To the right of the input field is a small square button with a vertical double-headed arrow, indicating a spinner control. Further to the right is a rectangular button with the text 'Go' inside it.

Έτσι δεν επιτρέπεται η εισαγωγή λάθος ή μη αριθμητικών τιμών.

Επιπλέον τα πεδία τύπο number έχουν κάποιες νέες συναρτήσεις javascript:

- input.stepUp(n), που εξομοιώνει ένα πάτημα του κουμπιού αύξησης.
- input.stepDown(n), που εξομοιώνει ένα πάτημα του κουμπιού μείωσης.
- input.valueAsNumber, που επιστρέφει την τρέχουσα τιμή σαν νούμερο αντί για string.

Τέλος, πάλι στα κινητά μπορεί να εμφανιστεί μία πιο εύχρηστη μορφή του πληκτρολογίου.



3.3.6 Εισαγωγή αριθμού με slider

Ένας διαφορετικός τρόπος εισαγωγής αριθμών είναι με την χρήση στοιχείων slider. Αυτό γίνεται με το στοιχείο range.

```
<form>
<input type="range"
  min="0"
  max="10"
  step="2"
  value="6">
  <input type="submit" value="Go">
</form>
```

Στους browsers που το υποστηρίζουν παράγει το παρακάτω πεδίο:



Οι ιδιότητες που δέχεται είναι ίδιες με αυτές του number.

3.3.7 Πεδίο επιλογής ημερομηνίας

Ένα άλλο στοιχείο που πολύ συχνά πρέπει να συμπληρώσουν οι χρήστες είναι μία ημερομηνία. Τέτοιου είδους στοιχεία εισάγονται με javascript και είναι αρκετά δύσκολο να γραφτούν και να ενσωματωθούν σωστά σε μία σελίδα. Αυτό πλέον αντιμετωπίζεται με το στοιχείο date:

```
<form>
  <input type="date">
  <input type="submit" value="Go">
</form>
```

Το αποτέλεσμα είναι το παρακάτω:

Δεν υποστηρίζεται μόνο ένας, αλλά 6 τύποι σχετικοί με ώρα και ημερομηνία, κάθε ένας από τους οποίους προσφέρει διαφορετική ακρίβεια:

- type="date"
- type="datetime"
- type="datetime-local"
- type="month"
- type="week"
- type="year"

Για παράδειγμα το πεδίο τύπου datetime επιτρέπει εισαγωγή ημερομηνίας και ώρας ως εξής:

Δυστυχώς αυτό το πεδίο υποστηρίζεται ακόμη από ελάχιστους browsers, με αποτέλεσμα να πρέπει να ελέγχεται η υποστήριξή του με javascript:

```
<form>
  <input type="date">
</form>
...
<script>
  var i = document.createElement("input");
  i.setAttribute("type", "date");
  if (i.type == "text") {
    // Insert datepicker toy replace the date field
  }
</script>
```

3.3.8 Πεδία αναζήτησης

Έχει δημιουργηθεί και ένα νέο πεδίο αναζήτησης. Η διαφορά του είναι πολύ μικρή και η κύρια χρησιμότητά του είναι η εμφάνισή του με τέτοια μορφή ώστε να μοιάζει με τα πεδία αναζήτησης του λειτουργικού συστήματος που χρησιμοποιεί ο χρήστης, ώστε να είναι άμεσα κατανοητό στον χρήστη πώς μπορεί να το χρησιμοποιήσει για αναζήτηση.

Εμφανίζεται με τον παρακάτω κώδικα:

```
<form>
  <input type="search">
  <input type="submit" value="Go">
</form>
```

Η εμφάνιση του πεδίου διαφέρει πολύ. Για παράδειγμα στον browser Safari, σε λειτουργικό σύστημα Mac OS X εμφανίζεται ως εξής:

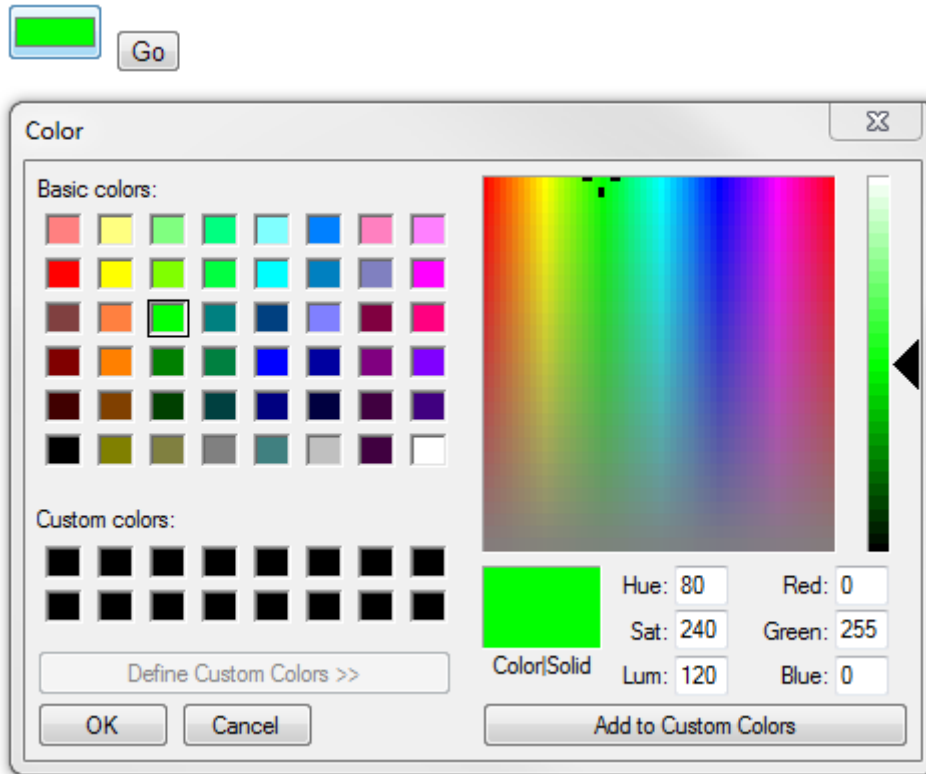


3.3.9 Πεδίο επιλογής χρώματος

Πρόκειται για ένα πεδίο τύπου color, το οποίο επιτρέπει στον χρήστη να διαλέξει εύκολα χρώμα. Παράγεται με αυτόν τον κώδικα:

```
<form>
  <input type="color" value="#00FF00">
  <input type="submit" value="Go">
</form>
```

Εμφανίζει ένα κατάλληλο παράθυρο επιλογής.



3.3.10 Έλεγχος στοιχείων φόρμας

Τα παρακάτω νέα πεδία αναγκάζουν τον χρήστη να εισάγει τιμές με μία συγκεκριμένη μορφή, χωρίς να μπορεί να βάλει λάθος τιμές. Ο πιο συχνά χρησιμοποιούμενος έλεγχος όμως είναι για το αν έχει συμπληρωθεί ένα πεδίο. Αυτό υποστηρίζεται πλέον εγγενώς με την ιδιότητα `required`.

```
<form>
  <input id="q" required>
  <input type="submit" value="Search">
</form>
```

Αυτό θα υποχρεώσει τον χρήστη να συμπληρώσει το πεδίο πριν παραδώσει την φόρμα.



Κάποιες φορές ίσως να είναι επιθυμητό να μην γίνεται έλεγχος των στοιχείων σε μία φόρμα, ώστε να γίνει με χρήση javascript με κάποιο τρόπο διαφορετικό από τον καθιερωμένο. Αυτό μπορεί να γίνει με την ιδιότητα `novalidate`.


```
<form novalidate>
  <input type="email" id="addr">
  <input type="submit" value="Subscribe">
</form>
```

3.4 Στοιχεία CSS

Στην HTML5 έχουν προστεθεί αρκετές νέες ιδιότητες στα στοιχεία CSS, τις οποίες θα αναλύσουμε παρακάτω. Αξίζει να σημειώσουμε πως οι περισσότερες ιδιότητες συνήθως υποστηρίζονται σε δύο στάδια. Σε πρώτη φάση όταν συμφωνηθεί ότι μία ιδιότητα θα μπει στο CSS συνήθως υποστηρίζεται σε μία έκδοση του browser (η οποία μπορεί να είναι beta) με ένα προσωρινό όνομα, που έχει σαν πρόθεμα κάτι σχετικό με την εταιρεία. Αφού γίνουν τουλάχιστον δύο διαφορετικές υλοποιήσεις συγκρίνονται μεταξύ τους για διαφορές και σε περίπτωση που υπάρχουν αποφασίζεται ποιος είναι ο σωστός τρόπος υλοποίησης. Έπειτα η νέα ιδιότητα μπαίνει στο πρότυπο και στις επόμενες εκδόσεις απορρίπτεται το πρόθεμα του κατασκευαστή.

Για παράδειγμα η σκιά ενός αντικειμένου μπορεί να εμφανίζεται με τις εξής ιδιότητες:

```
-webkit-box-shadow: rgba(0, 0, 128, 0.246094) 8px 6px 3px;
-moz-box-shadow: rgba(0, 0, 128, 0.246094) 8px 6px 3px;
box-shadow: rgba(0, 0, 128, 0.246094) 8px 6px 3px;
```

Οι δύο πρώτες είναι οι διαφορετικές υλοποιήσεις που έχουν γίνει για τους browser Chrome και Mozilla αντίστοιχα, ενώ η τρίτη είναι το όνομα της τελικής ιδιότητας.

Παρακάτω θα αναλύσουμε τα νέα στοιχεία.

3.4.1 Σκιά γραμμάτων

Υπάρχει δυνατότητα να σχεδιαστούν γράμματα που έχουν σκιά, κάτι που δεν ήταν δυνατόν παλιότερα. Αυτό γίνεται με την ιδιότητα text-shadow.

```
text-shadow: rgba(64, 64, 64, 0.496094) 6px 7px 5px;
```

Το αποτέλεσμα είναι αυτό:

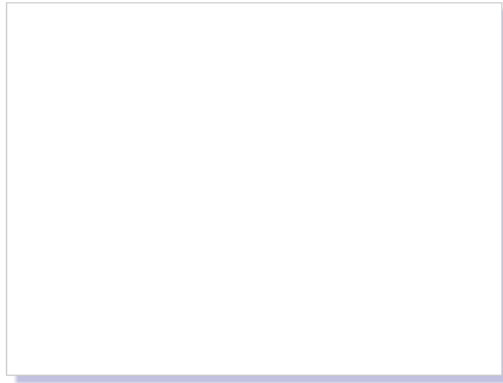
Shadows example

3.4.2 Σκιά γύρω από στοιχεία block

Υποστηρίζεται ο σχεδιασμός σκιάς γύρω από στοιχεία, κάτι που παλιότερα γινόταν αρκετά πολύπλοκα.

```
box-shadow: rgba(0, 0, 128, 0.246094) 8px 6px 3px;
```

Ο παραπάνω κώδικας παράγει αυτό το αποτέλεσμα:



3.4.3 Στρογγυλεμένες γωνίες

Ένα νέο στοιχείο είναι οι στρογγυλεμένες γωνίες στα πλαίσια με την ιδιότητα border-radius.

```
border-radius: 16px;
```

Ανάλογα με το μέγεθος το εφέ είναι λιγότερο ή περισσότερο έντονο.

50% round button style

Gmail button style

3.4.4 Στήλες

Το κείμενο, συνήθως μίας παραγράφου, μπορεί να εμφανιστεί σε δύο ή περισσότερες στήλες, ενώ μπορεί να οριστεί το κενό ανάμερά τους και κάποια διαχωριστική γραμμή.

```
column-count: 2;  
column-rule: 1px solid #bbb;  
column-gap: 2em;
```

Το παραγόμενο αποτέλεσμα φαίνεται παρακάτω:

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure

dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

3.4.5 Transitions

Οι μεταβάσεις ή transitions είναι ένας τρόπος ορισμού της ταχύτητας μεταβολής ενός μεγέθους. Αποτελούνται από τρία τμήματα τα οποία είναι το μέγεθος για το οποίο ορίζονται, τον χρόνο μέσα στον οποίο θα γίνεται η μεταβολή και τον ρυθμό μεταβολής. Τα μεγέθη μπορεί να είναι το μέγεθος, το χρώμα, η θέση ενός αντικειμένου κα.

Καθώς η HTML περιγράφει στατικές σελίδες, για να αλλάξει ένα μέγεθος μπορεί να γίνει μόνο μέσα από την χρήση ψευδοκλάσεων CSS, όπως οι hover, active κτλ. Όμως σε περίπτωση που ένα μέγεθος μεταβληθεί μέσω javascript θα εφαρμοστούν σε αυτό τα κατάλληλα transitions.

Ένα παράδειγμα είναι το παρακάτω:

```
#delay {  
  transition: font-size 4s linear;  
  font-size: 14px;  
}  
  
#delay:hover {  
  font-size: 36px;  
}
```

Αρχικά ορίζεται ότι το μέγεθος της ιδιότητας font-size θα μεταβάλλεται σε 4 δευτερόλεπτα και όχι στιγμιαία με γραμμικό ρυθμό μεταβολής. Έπειτα δηλώνεται ότι όταν ο κέρσορας του ποντικιού βρίσκεται από άνω του θα αλλάζει το μέγεθος, οπότε θα χρησιμοποιηθεί το κατάλληλο transition.

Οι διαθέσιμοι ρυθμοί μεταβολής που παρέχονται είναι οι linear, ease, ease-in, ease-out, ease-in-out.

3.4.6 Transformations

Μία άλλη αλλαγή είναι τα εφέ μετασχηματισμού, με τα οποία μπορεί να τροποποιηθεί ο τρόπος εμφάνισης ενός αντικειμένου. Χωρίζονται σε δύο κατηγορίες, τα δισδιάστατα και τα τρισδιάστατα. Η βασική μορφή τους είναι:

```
#rotated {  
  transform: translate(100px) rotate(20deg);  
  transform-origin: 60% 100%;  
}
```

Η ιδιότητα `transform` ορίζει τα εφέ που θα εφαρμοστούν και τις ιδιότητές τους, ενώ το `transform-origin` καθορίζει το σημείο του αντικειμένου το οποίο θα είναι το κέντρο γύρω από το οποίο θα εφαρμοστεί ο μετασχηματισμός.

Τα διαθέσιμα εφέ είναι τα εξής:

- Rotate
- Scale
- ScaleX
- ScaleY
- SkewX
- SkewY
- Translate
- TranslateX
- TranslateY
- Matrix

3.5 Η επιφάνεια ζωγραφικής

Η HTML5 έχει εισάγει ένα νέο στοιχείο ζωγραφικής, το `canvas`, το οποίο ορίζεται σαν μία επιφάνεια που χρησιμοποιείται για την παρουσίαση γραφικών ή εικόνων ανεξαρτήτου ανάλυσης. Πάνω σε αυτό μπορούν να ζωγραφιστούν νέα στοιχεία με χρήση `javascript`.

Αξίζει να σημειωθεί ότι η τρέχουσα έκδοση 9 του Internet Explorer δεν υποστηρίζει καθόλου το αντικείμενο `canvas`, αν και μπορεί να εξομοιωθεί με την χρήση `javascript`, με κάποιους περιορισμούς.

Για να εμφανιστεί ένα αντικείμενο `canvas` πρέπει να μπει σαν `tag` στην σελίδα και να του δοθούν διαστάσεις με ένα πλαίσιο για να φαίνονται τα όριά του.

```
<canvas width="300" height="300" style="border: 1px solid black"></canvas>
```

Αρχικά είναι μία διάφανη επιφάνεια. Καθώς χρησιμοποιείται μέσω `javascript` είναι βολικό να έχει ένα δικό το `id` για να είναι εύκολο να δηλωθεί σαν μεταβλητή.

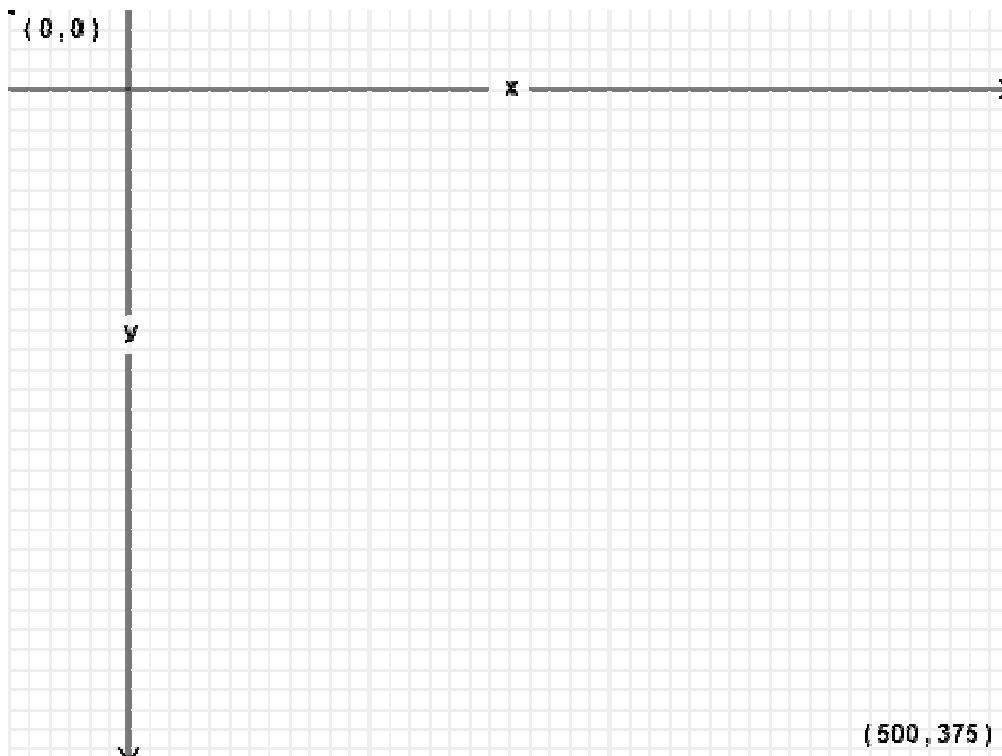
```
<canvas id="a" width="300" height="225" style="border: 1px solid black">
</canvas>
<script>
  var a_canvas = document.getElementById("a");
</script>
```

Η σχεδίαση απλών γεωμετρικών σχημάτων μπορεί να γίνει με απλές συναρτήσεις. Πχ αν στο γεγονός onclick οριστεί μία συνάρτηση μπορεί με κλικ σε αυτό να ζωγραφιστεί ένα σχήμα:

```
function draw_b() {  
  var b_canvas = document.getElementById("b");  
  var b_context = b_canvas.getContext("2d");  
  b_context.fillRect(50, 25, 150, 100);  
}
```

Η μέθοδος `getContext` επιστρέφει ένα γραφικό περιβάλλον, δηλαδή ένα αντικείμενο το οποίο μπορεί να χρησιμοποιηθεί για την εμφάνιση σχημάτων. Η παράμετρος `2d` δηλώνει ότι θέλουμε να ζωγραφίσουμε σε μία επιφάνεια 2 διαστάσεων, που είναι η μοναδική που έχει καθοριστεί αυτή τη στιγμή στο πρότυπο HTML5. Παρόλα αυτά στο πρότυπο έχει ανακοινωθεί ότι μελλοντικά θα υπάρξει υποστήριξη και για τρισδιάστατους χώρους, ενώ κάποιοι κατασκευαστές browsers έχουν ήδη ενσωματώσει κάποιες πειραματικές λειτουργίες.

Ξαναγυρνώντας στον παραπάνω κώδικα, η επόμενη γραμμή ζωγραφίζει ένα ορθογώνιο. Οι παράμετροι που παίρνει είναι 4, με τις δύο πρώτες να είναι οι συντεταγμένες x, y της πάνω αριστερά γωνίας και οι άλλες δύο να είναι το μήκος και το ύψος του ορθογωνίου. Το σύστημα συντεταγμένων ορίζεται ως εξής:



3.5.1 Σχεδιασμός μονοπατιών (Paths)

Το πιο απλό γεωμετρικό σχήμα που μπορεί να ζωγραφιστεί είναι μία γραμμή. Η διαδικασία αποτελείται από τρία βήματα.

Αρχικά χρειάζεται να χρησιμοποιηθεί η μέθοδος `moveTo` με παραμέτρους ένα ζεύγος συντεταγμένων ώστε να μετακινηθεί ένας νοητός δείκτης σε ένα συγκεκριμένο σημείο από το οποίο θα αρχίσει η σχεδίαση. Έπειτα χρησιμοποιείται η μέθοδος `lineTo` με παράμετρο ένα άλλο ζεύγος συντεταγμένων ώστε να δημιουργηθεί μία νέα γραμμή από το προηγούμενο σημείο σε αυτό.

Αυτή η γραμμή δεν είναι αρχικά ορατή. Για να γίνει ορατή χρειάζεται να οριστεί ένα χρώμα και μετά να ζωγραφιστεί. Αυτά γίνονται θέτοντας το χρώμα στην μεταβλητή `strokeStyle` του περιβάλλοντος γραφικών και έπειτα καλώντας την μέθοδο `stroke` η οποία ζωγραφίζει τη γραμμή με αυτό το χρώμα.

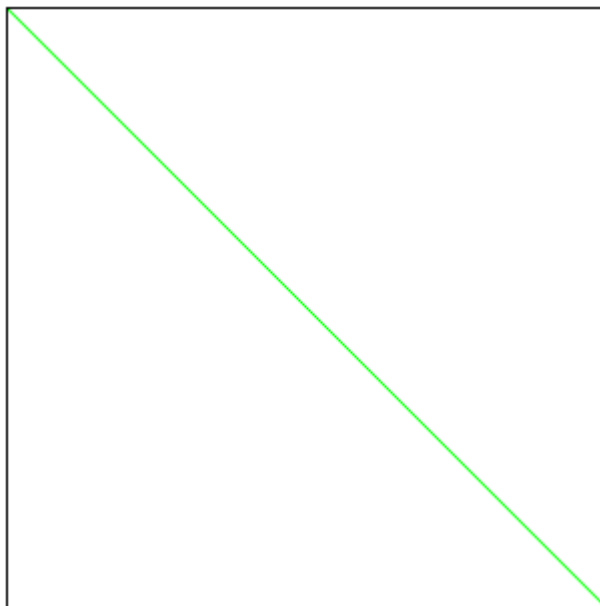
Άρα για τον σχεδιασμό μίας γραμμής χρειάζεται ο παρακάτω κώδικας:

```
<canvas id="a" width="300" height="300" style="border: 1px solid black"></canvas>
<script>
  var canvas = document.getElementById("a");
  var context = canvas.getContext("2d");

  context.moveTo(0, 0);
  context.lineTo(300, 300);

  context.strokeStyle = "#00ff00";
  context.stroke();
</script>
```

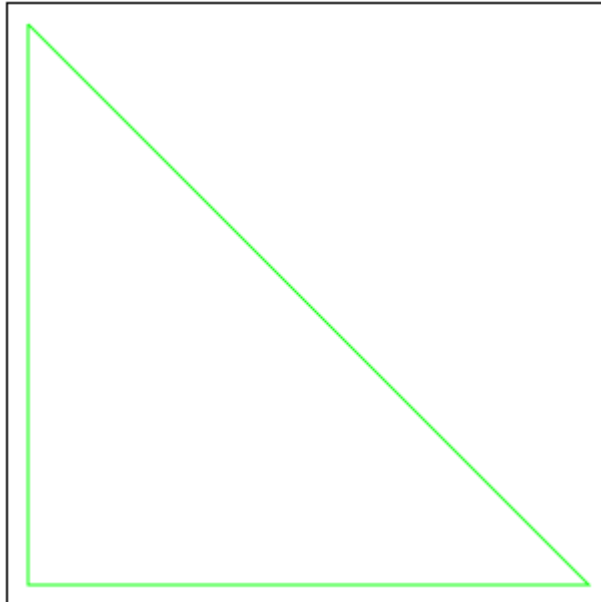
Το αποτέλεσμα είναι αυτό:



Αν θέλουμε ένα τρίγωνο πλαίσιο έχουμε αυτό:

```
var canvas = document.getElementById("a");  
var context = canvas.getContext("2d");  
  
context.moveTo(10, 10);  
context.lineTo(290, 290);  
context.lineTo(10, 290);  
context.lineTo(10, 10);  
  
context.strokeStyle = "#00ff00";  
context.stroke();
```

Το παραπάνω βγάζει αυτό το αποτέλεσμα:



Προφανώς όλες οι τιμές μπορούν να τροποποιηθούν υπολογιστικά και να χρησιμοποιηθούν με περισσότερο πολύπλοκο τρόπο ώστε να ζωγραφιστούν αρκετά πιο λεπτομερή σχήματα.

3.5.2 Σχεδιασμός κειμένου

Πέρα από απλές γραμμές μπορεί να ζωγραφιστεί και κείμενο με τέτοιο τρόπο ώστε να μην επηρεάζεται από τις ιδιότητες που χρησιμοποιούνται στο CSS. Το γραφικό περιβάλλον έχει τις παρακάτω ιδιότητες που επηρεάζουν την εμφάνιση του κειμένου:

- font που είναι αντίστοιχο με την εντολή font του CSS, και υποστηρίζει τις παραμέτρους font style, font variant, font weight, font size, line height, και font family.
- textAlign που ελέγχει την στοίχιση του κειμένου. Μπορεί να πάρει τις τιμές start, end, left, right, και center.

- `textBaseline` που ελέγχει την κάθετη θέση που θα έχει το κείμενο σε σχέση με το αρχικό σημείο. Παίρνει τιμές `top`, `hanging`, `middle`, `alphabetic`, `ideographic`, ή `bottom`.

Για να ζωγραφίσουμε ένα αλφαριθμητικό με διαφορετικές γραμματοσειρές χρειάζεται ο παρακάτω κώδικας.

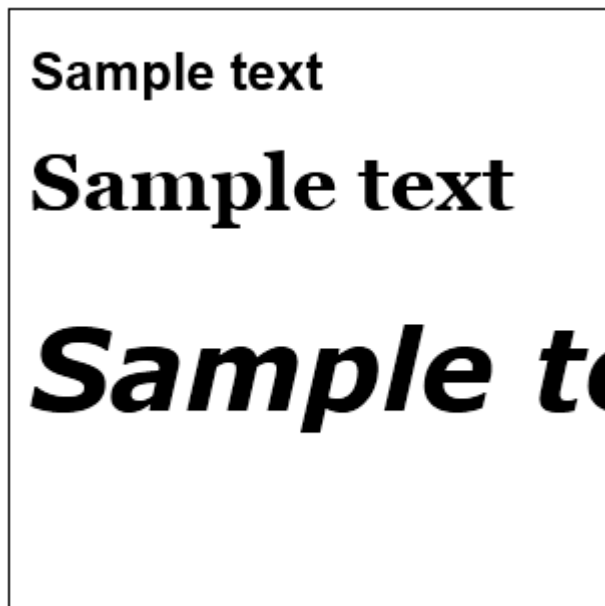
```
var canvas = document.getElementById("a");
var context = canvas.getContext("2d");

context.font = "bold 26px Arial";
context.fillText("Sample text", 10, 40);

context.font = "bold 40px Georgia";
context.fillText("Sample text", 10, 100);

context.font = "bold italic 56px Verdana";
context.fillText("Sample text", 10, 200);
```

Αυτό παράγει τα εξής:



3.5.3 Σχεδιασμός gradient

Παραπάνω είδαμε πως γίνεται να ζωγραφιστεί ένα ορθογώνιο ή μία γραμμή με ένα χρώμα. Όμως δεν είναι απαραίτητο να χρησιμοποιηθεί μόνο ένα χρώμα για τα σχήματα, καθώς γίνεται να χρησιμοποιηθούν `gradients`, δηλαδή να εμφανίζεται μία μετάβαση που ξεκινάει από ένα συγκεκριμένο χρώμα και σταδιακά να μετατρέπεται σε ένα άλλο.

Το δισδιάστατο περιβάλλον σχεδίασης που μελετήσαμε προηγουμένως παρέχει δύο συναρτήσεις για την δημιουργία `gradients`. Η πρώτη είναι η `createLinearGradient(x0, y0, x1, y1)`, η οποία ζωγραφίζει ένα `gradient` κατά μήκος της γραμμής που ξεκινάει από το

σημείο που ορίζεται από το (x0, y0) και καταλήγει στο σημείο (x1, y1). Αυτή η γραμμή ορίζει την κατεύθυνση κατά την οποία θα αλλάζει το χρώμα.

Εναλλακτικά παρέχεται και η `createRadialGradient(x0, y0, r0, x1, y1, r1)`, η οποία ζωγραφίζει κατά μήκος ενός κώνου που περιγράφεται από δύο κύκλους. Ο πρώτος κύκλος έχει σαν κέντρο το (x0, y0) και ακτίνα r0, ενώ ο δεύτερος έχει σαν κέντρο το (x1, y1) και ακτίνα r1.

Οι δύο παραπάνω συναρτήσεις περιγράφουν απλά τη φορά κατά την οποία θα αλλάζει το χρώμα αλλά δεν προσδιορίζουν τα χρώματα που θα χρησιμοποιηθούν. Αυτό γίνεται με την `addColorStop`, που παίρνει δύο παραμέτρους. Η πρώτη είναι ένας δεκαδικός που έχει τιμή από 0 έως 1 και περιγράφει το σημείο στο `gradient` που θα εμφανίζεται το χρώμα, ενώ η δεύτερη είναι το ίδιο το χρώμα σαν αλφαριθμητικό.

Για την χρήση δήλωση του `gradient` είναι απαραίτητο αρχικά να δηλωθεί χρησιμοποιώντας τα παραπάνω. Για παράδειγμα ένα `gradient` που θα έχει φορά από αριστερά προς τα δεξιά και θα ξεκινάει από μπλε, θα γίνεται σταδιακά λευκό μέχρι τη μέση και μετά θα γίνεται μαύρο ορίζεται με τις παρακάτω εντολές:

```
var gradient = context.createLinearGradient(0, 0, 300, 0);
gradient.addColorStop(0, "#0000ff");
gradient.addColorStop(0.5, "#ffffff");
gradient.addColorStop(1, "#000000");
```

Ο παραπάνω κώδικας δεν εμφανίζει κάτι, γιατί ένα `gradient` από μόνο του δεν είναι ένα σχήμα αλλά μοιάζει περισσότερο με ένα χρώμα το οποίο χρησιμοποιείται στη ζωγραφική άλλων σχημάτων. Για εμφανιστεί ένα ορθογώνιο που το χρησιμοποιεί πρέπει να γίνει η μεταβλητή `fillStyle` ίση με αυτό:

```
context.fillStyle = gradient;
```

Έπειτα μπορεί να χρησιμοποιηθεί στη ζωγραφική. Ο πλήρης κώδικας είναι:

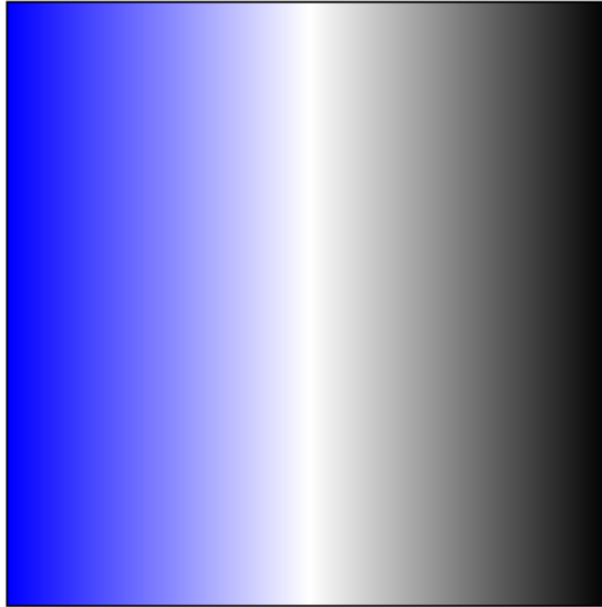
```
<canvas id="a" width="300" height="300" style="border: 1px solid black">
</canvas>
<script>
var canvas = document.getElementById("a");
var context = canvas.getContext("2d");

var gradient = context.createLinearGradient(0, 0, 300, 0);
gradient.addColorStop(0, "#0000ff");
gradient.addColorStop(0.5, "#ffffff");
gradient.addColorStop(1, "#000000");

context.fillStyle = gradient;
```

```
context.fillRect(0, 0, 300, 300);  
</script>
```

Η εικόνα που παράγεται είναι η παρακάτω:



3.5.4 Σχεδιασμός εικόνων

Αν και οι εικόνες υποστηρίζονται εγγενώς από την HTML με το στοιχείο `img`, μπορεί να είναι επιθυμητή η σχεδιάσή τους σε ένα αντικείμενο `canvas` ώστε μετά να μπορούν να τροποποιηθούν. Η σχεδίαση μίας εικόνας μπορεί να γίνει με την μέθοδο `drawImage`, η οποία μπορεί να πάρει τρεις διαφορετικές ομάδες παραμέτρων.

Αρχικά πρέπει να δημιουργηθεί η εικόνα. Αν και μπορεί να χρησιμοποιηθεί οποιαδήποτε εικόνα αποτελεί μέρος του εγγράφου, συνήθως η δημιουργία της είναι επιθυμητό να γίνει με τέτοιο τρόπο ώστε να μην εμφανίζεται στη σελίδα. Αυτό επιτυγχάνεται με το παρακάτω:

```
var image = new Image();  
image.src = "robot.png";
```

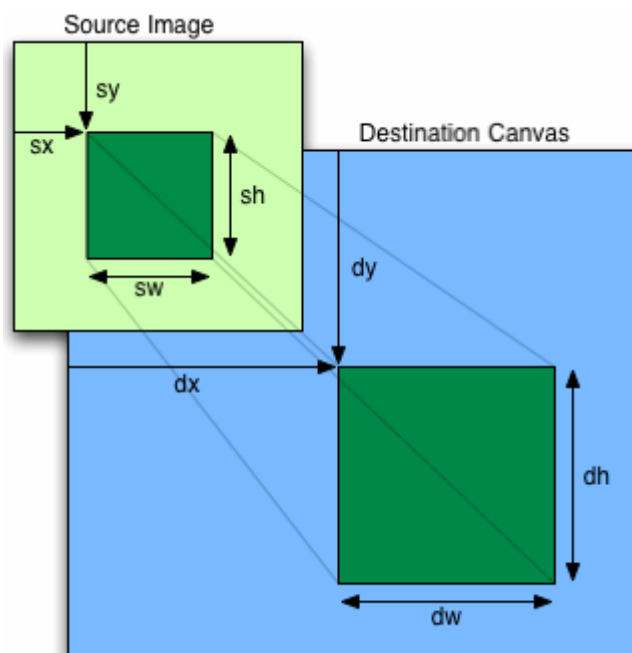
Έπειτα πρέπει να χρησιμοποιηθεί η έκδοση της `drawImage` που χρειάζεται. Οι εκδόσεις είναι οι εξής:

- `drawImage(image, dx, dy)`: Αυτή η έκδοση παίρνει για παράμετρο μία εικόνα και την σχεδιάζει με τέτοιο τρόπο ώστε η επάνω αριστερή γωνία της να απέχει `dx` pixels

από την αριστερή πλευρά του αντικειμένου canvas και dy κάτω από την επάνω γραμμή.

- `drawImage(image, dx, dy, dw, dh)`: Εδώ οι μεταβλητές `image`, `dx`, `dy` κάνουν ότι και παραπάνω. Τα `dw` και `dh` είναι το ύψος και το μήκος που θα έχει η εικόνα που θα σχεδιαστεί. Η εικόνα δεν θα κοπεί αλλά θα παραμορφωθεί ώστε να έχει το μέγεθος που δώθηκε.
- `drawImage(image, sx, sy, sw, sh, dx, dy, dw, dh)`: Οι παράμετροι `image`, `dx`, `dy`, `dw`, `dh` κάνουν ότι και προηγουμένως. Οι νέες παράμετροι ορίζουν ένα ορθογώνιο που είναι το τμήμα της εικόνας που θα ζωγραφιστεί. Τα `sx`, `sy`, είναι η απόσταση από την επάνω αριστερή γωνία ενώ τα `sw`, `sh` είναι οι διαστάσεις του.

Η προδιαγραφή της HTML5 έχει ένα επεξηγηματικό σχήμα που παρουσιάζει γραφικά το τι κάνει η κάθε παράμετρος.



Εικόνα 3-1 Επεξήγηση των παραμέτρων της συνάρτησης `drawImage`.

Ένα παράδειγμα χρήσης δίνεται από τον παρακάτω κώδικα που ζωγραφίζει την ίδια εικόνα δύο φορές, μία κανονικά και μία συμπιεσμένη.

```
<canvas id="a" width="300" height="300" style="border: 1px solid black">
</canvas>
<script>
  var canvas = document.getElementById("a");
  var context = canvas.getContext("2d");

  var image = new Image();
  image.src = "robot.png";

  context.drawImage(image, 0, 0);
  context.drawImage(image, 256, 0, 44, 256);
</script>
```

Το αποτέλεσμα είναι το παρακάτω:



3.6 Υποστήριξη εικόνων SVG

Τα αρχεία εικόνων SVG (Scalable Vector Graphics) είναι μία διανυσματική μορφή γραφικών, τα οποία έχουν αρκετά πλεονεκτήματα σε σχέση με τις παραδοσιακές μορφές αρχείων εικόνας (raster). Ο λόγος είναι ότι τα κλασικά αρχεία εικόνας περιγράφουν μία εικόνα σαν ένα σύνολο από pixels. Αυτό σημαίνει ότι μία εικόνα έχει κατασκευαστεί για να έχει ένα συγκεκριμένο μέγεθος. Τυχών αλλαγές στο μέγεθός της σημαίνουν ότι πρέπει να προστεθεί πληροφορία που δεν υπάρχει ή να αφαιρεθεί πληροφορία.

Το φαινόμενο φαίνεται ξεκάθαρα στο αμέσως προηγούμενο σχήμα όπου έχει ζωγραφιστεί μία εικόνα στο κανονικό της μέγεθος (αριστερά) και πιο μικρή (δεξιά). Η δεξιά εικόνα έχει χάσει σε ακρίβεια και φαίνεται να έχει τετραγωνισμένες ατέλειες. Καθώς όταν σχεδιάζεται μία σελίδα δεν μπορεί να αποφασιστεί η ανάλυση της οθόνης στην οποία θα προβληθεί είναι σχεδόν σίγουρο ότι σε αρκετές περιπτώσεις θα υπάρξει μεγέθυνση ή σμίκρυνση των στοιχείων της.

Τα αρχεία SVG αποθηκεύουν τα περιεχόμενά τους όχι σαν pixel αλλά σαν μαθηματικές περιγραφές γεωμετρικών σχημάτων. Αυτό σημαίνει ότι η σχεδίαση γίνεται κάθε φορά από το πρόγραμμα προβολής για το κατάλληλο μέγεθος, με αποτέλεσμα να μην υπάρχουν παραμορφώσεις. Επιπλέον εξαιτίας αυτού καταλαμβάνουν πολύ λιγότερο χώρο σε σχέση με τις άλλες εικόνες και μπορούν να διαβαστούν και από τον χρήστη, καθώς η μορφή τους είναι μορφή αρχείων XML.

Το μειονέκτημα είναι πως συνήθως η δημιουργία τους είναι επίπονη διαδικασία, καθώς είναι εξαιρετικά δύσκολη ή και αδύνατη η μετατροπή μίας εικόνας σε αυτή τη μορφή και τις περισσότερες φορές πρέπει να ξαναγίνει από την αρχή.

Αν και το format SVG προϋπήρχε σαν πρότυπο ανεξάρτητα από την HTML5, υιοθετείτε σταδιακά για τυποποίηση σε αυτή. Αυτή τη στιγμή οι περισσότεροι browsers μπορούν να προβάλλουν αρχεία SVG αλλά διαφέρουν στο βαθμό των χαρακτηριστικών τους που υποστηρίζουν.

3.7 WebGL

Το WebGL (Web Graphics Library) είναι μία βιβλιοθήκη της javascript η οποία επιτρέπει τη σχεδίαση δισδιάστατων και τρισδιάστατων γραφικών με τα οποία μπορεί να αλληλεπιδράσει ο χρήστης. Αν και αυτή η δυνατότητα υπήρχε και παλιότερα με τη χρήση επιπλέον πρόσθετων, πλέον καθορίζεται ο τρόπος με τον οποίο πρέπει να είναι ενσωματωμένη στους browsers.

Το μεγαλύτερο πρόβλημα με τα τρισδιάστατα γραφικά είναι πως για την παρουσίαση όλων εκτός από τις απλούστερες εικόνες απαιτείται πάρα πολύ μεγάλη υπολογιστική ισχύς. Παλιότερες υλοποιήσεις χρησιμοποιούσαν τον επεξεργαστή του συστήματος για τους αναγκαίους υπολογισμούς, μία λύση που παρέχει πολύ χαμηλές επιδόσεις. Πλέον ο κώδικας για τους shaders, δηλαδή για τους υπολογισμούς που καθορίζουν την εμφάνιση και την υφή των αντικειμένων, εκτελείται χρησιμοποιώντας την κάρτα γραφικών του συστήματος, με αποτέλεσμα να παράγονται πολύ καλύτερα αποτελέσματα.

Η υποστήριξη της τεχνολογίας βρίσκεται σε πειραματικό βαθμό και παρόλο που υπάρχει σε όλους τους δημοφιλείς browsers είναι πολύ ρευστή και αλλάζει συνεχώς.

3.8 Αναπαραγωγή βίντεο

Η ενσωμάτωση βίντεο στις σελίδες HTML αποτελεί κοινό τρόπο εδώ και πολλά χρόνια, όμως πάντοτε η προβολή του γινόταν με κάποια Τρίτη τεχνολογία, συνήθως με τη χρήση Flash, Quicktime ή RealPlayer. Βέβαια σε μία πλατφόρμα που δεν υποστηρίζει τις παραπάνω τεχνολογίες, όπως κάποια κινητά τηλέφωνα, η αναπαραγωγή είναι προβληματική.

Στην HTML5 έχει εισαχθεί ένα νέο στοιχείο video, το οποίο εκ πρώτης όψεως φαίνεται να υποστηρίζεται από όλους τους σύγχρονους browser.

Υποστήριξη στοιχείου <video>						
IE	Firefox	Safari	Chrome	Opera	iPhone	Android
9.0+	3.5+	3.0+	3.0+	10.5+	1.0+	2.0+

Όμως η υποστήριξη που παρέχεται είναι προβληματική κυρίως λόγω της έντονης διαφωνίας που έχει ξεσπάσει για τον τρόπο υλοποίησής του. Για να εξηγήσουμε την φύση της διαφωνίας είναι απαραίτητο να αναλύσουμε συνοπτικά την δομή ενός αρχείου βίντεο.

Ένα αρχείο βίντεο αποτελείται από τρία βασικά συστατικά. Το πρώτο είναι μία ή περισσότερες ροές βίντεο, όπου κάθε ροή είναι μία αλληλουχία εικόνων (ένα αρχείο μπορεί να έχει πολλές ροές επειδή πχ μπορεί να έχει τα ίδια πλάνα από διαφορετικές κάμερες). Το δεύτερο στοιχείο είναι καμία (αν δεν έχει ήχο), μία ή περισσότερες ροές ήχου (που μπορεί να είναι σε διαφορετικές γλώσσες η κάθε μία). Τέλος υπάρχει και το αρχείο φορέας, που είναι η μορφή που έχει το τελικό αρχείο και περιέχει πληροφορίες σχετικά με τις ροές βίντεο και ήχου που περιέχει και πως αυτές είναι οργανωμένες μέσα στο αρχείο.

Για να λειτουργήσει το στοιχείο βίντεο πρέπει να συμφωνηθεί πως όλοι οι browsers θα περιέχουν τα κατάλληλα προγράμματα που θα μπορούν να διαβάσουν το αρχείο που τους στέλνεται και να αποκωδικοποιήσουν τα περιεχόμενά του. Η διαφωνία έγκειται στην επιλογή αυτών των προγραμμάτων (που ονομάζονται codecs) και συγκεκριμένα στο ποιες μορφές θα υποστηρίζονται. Καθώς οι codecs πολλές φορές ανήκουν σε άλλες εταιρείες ή αποτελούν πνευματική ιδιοκτησία άλλων εταιρειών δεν μπορούν ελεύθερα να ενσωματωθούν σε όλα τα προγράμματα περιήγησης και εκεί βρίσκεται η ρίζα του προβλήματος.

Αυτή τη στιγμή σχεδόν κάθε εξυπηρετητής υποστηρίζει την αναπαραγωγή διαφορετικών ροών βίντεο, ήχου και αρχείων – φορέων. Επιπλέον κάποιοι από αυτούς μπορούν εύκολα να αναπαράγουν οποιοδήποτε τύπο αρχείου υποστηρίζει το λειτουργικό σύστημα, ενώ κάποιοι άλλοι χρησιμοποιούν μόνο τα ενσωματωμένα format τους, κάνοντας την κατάσταση χαοτική.

Ένας συγκεντρωτικός πίνακας φαίνεται παρακάτω αν και σχεδόν σίγουρα θα μεταβληθεί μελλοντικά.

<video> element support						
IE	Firefox	Safari	Chrome	Opera	iPhone	Android
9.0+	3.5+	3.0+	3.0+	10.5+	1.0+	2.0+

Ένα επιπλέον πρόβλημα είναι πως τα βίντεο που υπάρχουν ήδη στο internet έχουν τεράστιο όγκο, και ακόμη και σε περίπτωση που υπάρξει συμφωνία για τα υποστηριζόμενα πρότυπα η μετατροπή τους θα είναι μία τραγικά δύσκολη διαδικασία.

Παρόλα αυτά έχει επέλθει τουλάχιστον συμφωνία στη μορφή του tag η οποία είναι η εξής:

```
<video src="pr6.webm"></video>
```

Η ιδιότητα src καθορίζει την διεύθυνση από την οποία θα φορτωθεί το βίντεο. Επιπλέον υποστηρίζονται οι εξής ιδιότητες:

- Width: το μήκος του βίντεο
- Height: το ύψος του βίντεο
- Controls: καθορίζει αν θα εμφανιστούν τα κουμπιά ελέγχου.

- Preload: δηλώνει ότι το βίντεο θα αρχίσει να φορτώνεται με τη φόρτωση της σελίδας και όχι όταν πατηθεί το πλήκτρο έναρξης αναπαραγωγής.
- Autoplay: δίνει την οδηγία να αρχίσει η αναπαραγωγή με την φόρτωση της σελίδας.

Η πλήρη μορφή λοιπόν είναι:

```
<video src="pr6.webm" width="320" height="240" preload="none"></video>
```

3.9 Εύρεση θέσης

Η εύρεση της θέσης ενός χρήστη ονομάζεται Geolocation. Μπορεί να γίνει με διαφορετικούς τρόπους, όπως με ανάλυση της διεύθυνσης IP με την οποία ο χρήστης είναι συνδεδεμένος στο δίκτυο, μέσω του ασύρματου δικτύου που χρησιμοποιεί, μέσω της θέσης του πύργου κινητής τηλεφωνίας στην οποία είναι συνδεδεμένων ή και με την χρήση ειδικών συσκευών.

Το πρότυπο HTML5 καθορίζει το Geolocation API, που είναι οι προδιαγραφές των αντικειμένων javascript που καθορίζουν αυτά τα στοιχεία. Το πρώτο πράγμα που ξεκαθαρίζεται είναι ότι η αποστολή των στοιχείων είναι προαιρετική και γίνεται μόνο με την άδεια του χρήστη. Κάθε πρόγραμμα περιήγησης πρέπει να ρωτήσει τον χρήστη πριν αποκαλύψει αυτά τα στοιχεία στην σελίδα που τα ζήτησε.



Εικόνα 3-2 Το πλαίσιο διαλόγου με το οποίο ο Firefox ρωτάει τον χρήστη αν θέλει να αποκαλύψει τη θέση του στη σελίδα που επισκέπτεται.

Αν και το παραπάνω πλαίσιο φαίνεται αρκετά απλό ο τρόπος που εμφανίζεται και οι επιλογές που δίνει είναι αυστηρά καθορισμένες.

Για να πάρει μία σελίδα την τοποθεσία του χρήστη χρειάζεται να καλέσει την μέθοδο `getCurrentPosition` ως εξής:

```
<script>  
  navigator.geolocation.getCurrentPosition(callback);  
</script>
```

Η μέθοδος επιστρέφει αμέσως και δεν επιστρέφει κανένα αντικείμενο. Παίρνει για παράμετρο μία συνάρτηση. Αυτό γίνεται γιατί η κλήση της κάνει το παραπάνω πλαίσιο διαλόγου να εμφανιστεί. Αν ο χρήστης απαντήσει καταφατικά στην ερώτηση τότε καλείται η συνάρτηση με παράμετρο ένα αντικείμενο που περιέχει τις πληροφορίες, πχ.:

```
function callback(position) {
    var latitude = position.coords.latitude;
    var longitude = position.coords.longitude;
    // ...
}
navigator.geolocation.getCurrentPosition(callback);
```

Το αντικείμενο είναι τύπου position και περιέχει τα παρακάτω πεδία:

Αντικείμενο Position		
Ιδιότητα	Τύπος	Παρατηρήσεις
<code>coords.latitude</code>	double	decimal degrees
<code>coords.longitude</code>	double	decimal degrees
<code>coords.altitude</code>	double or null	meters above the reference ellipsoid
<code>coords.accuracy</code>	double	meters
<code>coords.altitudeAccuracy</code>	double or null	meters
<code>coords.heading</code>	double or null	degrees clockwise from true north
<code>coords.speed</code>	double or null	meters/second
<code>timestamp</code>	DOMTimeStamp	like a <code>Date()</code> object

Οι πληροφορίες που υπάρχουν εγγυημένα είναι οι συντεταγμένες latitude και longitude, καθώς και η ακρίβεια της μέτρησης accuracy.

3.10 Τοπική αποθήκευση

Η επίμονη τοπική αποθήκευση (persistent local storage) είναι η αποθήκευση δεδομένων τα οποία συνεχίζουν να υπάρχουν στον υπολογιστή του χρήστη ακόμη και όταν αυτός φύγει από μία σελίδα ή κλείσει το πρόγραμμα περιήγησης. Είναι μία από τις περιοχές που οι τυπικές εφαρμογές πλεονεκτούν σε σχέση με τις σελίδες, καθώς μπορούν να αποθηκεύσουν τα δεδομένα τους σε αρχεία και να τα ξαναφορτώσουν όταν ξεκινήσουν.

Μία ιστοσελίδα μπορεί να χρησιμοποιήσει cookies για να αποθηκεύσει πληροφορίες αλλά αυτά μπορούν να έχουν περιορισμένο μέγεθος (μέχρι 4KB) και στέλνονται κάθε φορά που ζητείται μία σελίδα στον server, δημιουργώντας τον κίνδυνο υποκλοπής και καθυστερώντας την σύνδεση. Έχουν ήδη γίνει προσπάθειες προς αυτή την κατεύθυνση παλιότερα αλλά καμία από αυτές δεν έχει γνωρίσει αποδοχή γιατί δεν αποτελούν μέρος κοινώς αποδεκτών προτύπων.

Το τμήμα της HTML5 που ασχολείται με την τοπική αποθήκευση λέγεται Web Storage και επιτρέπει την αποθήκευση ζευγών ονομάτων – τιμών στον υπολογιστή του χρήστη τα οποία δεν στέλνονται στον server της σελίδας.

Οι τιμές αποθηκεύονται σαν αλφαριθμητικά και αποθηκεύονται με την μέθοδο `setItem`, που παίρνει για παραμέτρους το κλειδί και την τιμή του. Η ανάκληση γίνεται με την μέθοδο `getItem` που παίρνει το κλειδί.

```
var str = "Some data";
localStorage.setItem("key1", str);
// ...
str = localStorage.getItem("key1");
```

Όπως και όλα τα αντικείμενα της javascript είναι δυνατή η χρήση του αντικειμένου `localStorage` σαν να είναι πίνακας.

```
var str = "Some data";
localStorage["key1"] = str;
// ...
str = localStorage["key1"];
```

Επιπλέον παρέχονται οι συναρτήσεις `removeItem` για την διαγραφή ενός αντικειμένου, `clear` για το άδειασμα όλων των δεδομένων και `key` για επιστροφή ενός συγκεκριμένου κλειδιού. Το πλήθος των αποθηκευμένων αντικειμένων δίνεται από την ιδιότητα `length`.

```
localStorage.removeItem("key1"); // delete key1
localStorage.clear(); // delete all data
FirstKey = localStorage.key(0); // get the 1st key
var size = localStorage.length;
```

Στην πρόταση του W3C για το Web Storage προτείνονται τα 5MB σαν όριο για το μέγεθος των δεδομένων που μπορεί να αποθηκεύσει κάθε σελίδα και αυτό ακολουθείται από όλους τους φυλλομετρητές. Σε περίπτωση που το όριο ξεπεραστεί τότε η τρέχουσα λειτουργία σταματάει και εμφανίζεται ένα λάθος τύπου `QUOTA_EXCEEDED_ERR`.

Παρόλο που το όριο των 5MB είναι μία πρόταση και δεν είναι δεσμευτικό, αυτή τη στιγμή δεν παρέχεται η δυνατότητα ερώτησης του χρήστη ώστε μία σελίδα να χρησιμοποιήσει περισσότερο χώρο στο δίσκο.

3.11 Web SQL Database

Η χρήση ζευγαριών κλειδιού – τιμής για αποθήκευση μπορεί να είναι περιοριστική για πιο πολύπλοκες εφαρμογές, καθώς η οργάνωση μεγάλου όγκου δεδομένων και αναζήτηση σε αυτά μπορεί να γίνει αρκετά δύσκολη διαδικασία. Επιπλέον δεν υπάρχει κανένας αυτόματος τρόπος ώστε να οριστούν σχέσεις ανάμεσα στα αποθηκευμένα δεδομένα.

Για την κάλυψη πιο προχωρημένων αναγκών αποθήκευσης έχει προταθεί το πρότυπο Web SQL Database, το οποίο, όπως λέει και το όνομά του, δίνει την δυνατότητα πρόσβασης σε μία βάση δεδομένων μέσω javascript. Ο κώδικας με τον οποίο ο χρήστης θα δημιουργούσε μία βάση και έναν πίνακα σε αυτή φαίνεται παρακάτω.

```
openDatabase('documents', '1.0', 'Local document storage', 5*1024*1024,
function (db) {
  db.changeVersion('', '1.0', function (t) {
    t.executeSql('CREATE TABLE docids (id, name)');
  }, error);
});
```

Το βασικό στοιχείο είναι η μέθοδος executeSql μέσω της οποίας μπορεί να εκτελεστεί οποιαδήποτε εντολή SQL με τα αποτελέσματα να επιστρέφονται σαν ένα πίνακα. Το παραπάνω έχει ήδη υλοποιηθεί σε μερικούς browsers.

Υποστήριξη Web SQL Database						
IE	Firefox	Safari	Chrome	Opera	iPhone	Android
.	.	4.0+	4.0+	10.5+	3.0+	2.0+

Όμως η συγκεκριμένη τεχνολογία αντιμετωπίζει ένα πολύ μεγάλο πρόβλημα. Αυτό πηγάζει από το γεγονός ότι η γλώσσα SQL δεν υλοποιείται ακριβώς με τον ίδιο τρόπο σε όλες τις βάσεις δεδομένων. Καθώς δεν υπάρχει ένα ακριβές πρότυπο είναι απαραίτητο να οριστεί ακριβώς η διάλεκτος της SQL που θα χρησιμοποιηθεί.

Όμως όλες οι μέχρι τώρα υλοποιήσεις έχουν γίνει χρησιμοποιώντας εσωτερικά την βάση δεδομένων Sqlite. Αυτό σημαίνει ότι καθώς δεν υπάρχει κάποια άλλη υλοποίηση βάση της οποίας να εντοπιστούν διαφορές και να γίνουν συγκρίσεις, το W3C έχει σταματήσει την ανάπτυξη. Ο λόγος είναι πως ένα πρότυπο δεν μπορεί να περιέχει τις προδιαγραφές ενός συγκεκριμένου προγράμματος, αλλά οι προδιαγραφές για την SQL αυτή τη στιγμή απλά αναφέρονται στις προδιαγραφές της Sqlite.

Επιπλέον οι δύο μεγάλοι κατασκευαστές browsers Microsoft και Mozilla αντιτίθενται στην συγκεκριμένη τεχνολογία, θεωρώντας την ακατάλληλη για τις ανάγκες του διαδικτύου. Αντίθετα προωθείται το πρότυπο IndexedDb, το οποίο δεν χρησιμοποιεί την γλώσσα SQL αλλά η σύνταξή του βασίζεται στην javascript.

3.12 Σελίδες χωρίς σύνδεση

Μία νέα δυνατότητα που καθορίζεται στην HTML5 είναι η δυνατότητα δημιουργίας μίας σελίδας η οποία να μπορεί να λειτουργήσει ακόμη και αν ο χρήστης δεν έχει σύνδεση στο διαδίκτυο, αφού προφανώς θα έχει ήδη κατεβάσει την σελίδα.

Αυτό γίνεται με την δήλωση μίας λίστας με διευθύνσεις οι οποίες αποτελούν τα στοιχεία τα οποία μπορούν να επαναχρησιμοποιηθούν με ασφάλεια ώστε η εφαρμογή να λειτουργήσει εκτός σύνδεσης. Καθώς κάθε browser αποθηκεύει στην μνήμη cache τα αρχεία της σελίδας που χρησιμοποιεί, μπορεί από την παραπάνω λίστα να ξέρει πια από αυτά είναι ασφαλές να χρησιμοποιηθούν όταν η σύνδεση χαθεί.

Αυτά ορίζονται σε ένα αρχείο που δηλώνεται στο στοιχείο html της σελίδας.

```
<!doctype HTML>  
<html manifest="/cache.manifest">  
<body>  
...  
</body>  
</html>
```

Ο παραπάνω κώδικας λέει στον browser να ανοίξει το αρχείο cache.manifest. Αυτό περιέχει τέσσερα τμήματα:

- Πάντα πρέπει να ξεκινάει με τη γραμμή CACHE MANIFEST
- Το τμήμα NETWORK καθορίζει πια αρχεία της σελίδας πρέπει να φορτωθούν από το δίκτυο σε κάθε περίπτωση.
- Το τμήμα CACHE καθορίζει τα αρχεία που μπορούν να αποθηκευτούν και να επαναχρησιμοποιηθούν.
- Το τμήμα FALLBACK που καθορίζει κάποια εναλλακτικά αρχεία να χρησιμοποιηθούν σε περίπτωση που δεν υπάρχει σύνδεση.

3.13 WebRTC

Το WebRTC (Web Real Time Communications) είναι ένα API της Javascript που επιτρέπει την μεταφορά δεδομένων σε πραγματικό χρόνο, κάνοντας εφικτή την δημιουργία εφαρμογών όπως για παράδειγμα μετάδοση βίντεο και φωνής, συνομιλίες με ήχο και βίντεο.

Αυτή τη στιγμή έχει υλοποιηθεί το τμήμα του μέσω του οποίου η σελίδα μπορεί να αποκτήσει πρόσβαση στο μικρόφωνο ή στην κάμερα του υπολογιστή. Αν και υπάρχει μία υλοποίηση της εταιρείας Google για την μεταφορά δεδομένων σε άλλους υπολογιστές χωρίς να μεσολαβεί ένας server, βρίσκεται υπό συζήτηση και δεν έχει εγκριθεί από τους υπόλοιπους φορείς.

4 Παρουσίαση εφαρμογής

4.1 Γενικά

Για την παρουσίαση των δυνατοτήτων της HTML5 που είδαμε παραπάνω δημιουργήσαμε μία εφαρμογή, η οποία με τη σειρά της είναι γραμμένη στην γλώσσα HTML5. Σαν βάση για την παρουσίαση χρησιμοποιήθηκε η εφαρμογή Reveal.js, που επιτρέπει την δημιουργία παρουσιάσεων βασισμένες σε διαφάνειες. Κάθε διαφάνεια είναι γραμμένη και αυτή με τη σειρά της σε HTML και περιέχει το κείμενο και τυχών παραδείγματα.

Καθώς αρκετά από τα στοιχεία της HTML5 που παρουσιάζονται δεν έχουν ακόμη υιοθετηθεί, δεν υποστηρίζονται σε όλους τους browsers. Για αυτόν τον λόγο επιλέχθηκε ο Google Chrome, καθώς υποστηρίζει έναν μεγάλο αριθμό από τις νέες ιδιότητες.

Επιπλέον επιλέξαμε να δείξουμε τον τρόπο με τον οποίο πολλές από τις ιδιότητες μπορούν να χρησιμοποιηθούν σε πολύπλοκες εφαρμογές, χρησιμοποιώντας κάποια από τα Chrome Experiments. Αυτά είναι μία βιβλιοθήκη παραδειγμάτων που δημιουργήθηκε από την εταιρεία Google με σκοπό να κινήσει το ενδιαφέρον για τις νέες δυνατότητες της HTML5, στην οποία οι χρήστες μπορούν να υποβάλλουν τις εφαρμογές τους.

Παρόλο που τα περισσότερα από τα στοιχεία μπορούν να λειτουργήσουν σωστά σε όλους τους browsers, για κάποια από αυτά δεν ήταν δυνατόν. Για την παρουσίαση επιλέχθηκε η χρήση του Google Chrome.

4.2 Βιβλιοθήκες και εγκατάσταση

Η βιβλιοθήκη που χρησιμοποιήσαμε για την παρουσίαση είναι το reveal.js, μία αρκετά εντυπωσιακή παρουσίαση που έχει κατασκευαστεί από τον Hakim El Hattab. Για την χρήση της χρειάζεται να την κατεβάσουμε από τη διεύθυνση <https://github.com/hakimel/reveal.js> και να αποσυμπιέσουμε τα περιεχόμενα του αρχείου σε έναν φάκελο.

css	27/11/2012 14:45	File folder
images	29/11/2012 14:24	File folder
js	27/11/2012 14:45	File folder
lib	27/11/2012 14:45	File folder
plugin	27/11/2012 14:45	File folder
grunt.js	2 KB 22/11/2012 13:25	JScript Script File
index.html	8 KB 3/12/2012 13:52	Firefox HTML Doc...
LICENSE	2 KB 22/11/2012 13:25	File
package.json	1 KB 22/11/2012 13:25	JSON File
README.md	14 KB 22/11/2012 13:25	MD File

Εικόνα 4-1 Τα αρχεία του reveal.js

Από τα αρχεία που δίνονται μας ενδιαφέρει το index.html, το οποίο περιέχει τον κώδικα της παρουσίασης. Καθώς αυτό που δίνεται είναι μία έτοιμη παρουσίαση χρειάζεται να την σβήσουμε, ώστε να μείνουμε με τον αρχικό κώδικα.

```

<!doctype html>
<html lang="en">

  <head>
    <meta charset="utf-8">

    <title>Παρουσίαση των δυνατοτήτων της HTML5</title>

    <meta name="apple-mobile-web-app-capable" content="yes" />
    <meta name="apple-mobile-web-app-status-bar-style" content="black-
translucent" />

    <link rel="stylesheet" href="css/reveal.min.css">
    <link rel="stylesheet" href="css/theme/default.css" id="theme">

    <link rel="stylesheet" href="lib/css/zenburn.css">

    <!--[if lt IE 9]>
    <script src="lib/js/html5shiv.js"></script>
    <![endif]-->
  </head>

  <body>

    <div class="reveal">

      <!-- Any section element inside of this container is displayed
as a slide -->
      <div class="slides">
        </div>

    </div>

    <script src="lib/js/head.min.js"></script>
    <script src="js/reveal.min.js"></script>

    <script>
      Reveal.initialize({
        controls: true,
        progress: true,
        history: true,
        center: false,

        theme: 'night',
        transition: Reveal.getQueryHash().transition || 'default',

        dependencies: [
          { src: 'lib/js/classList.js', condition: function() {
return !document.body.classList; } },
          { src: 'plugin/markdown/showdown.js', condition:
function() { return !!document.querySelector( '[data-markdown]' ); } },
          { src: 'plugin/markdown/markdown.js', condition:

```

```
function() { return !!document.querySelector( '[data-markdown]' ); } },
    { src: 'plugin/highlight/highlight.js', async: true,
callback: function() { hljs.initHighlightingOnLoad(); } },
    { src: 'plugin/zoom-js/zoom.js', async: true,
condition: function() { return !!document.body.classList; } },
    { src: 'plugin/notes/notes.js', async: true,
condition: function() { return !!document.body.classList; } }
  ]
});

</script>

</body>
</html>
```

Η αρχή και το τέλος της σελίδας περιέχουν τις ρυθμίσεις των επιπλέον βιβλιοθηκών που χρησιμοποιούνται από το reveal.js. Μας ενδιαφέρει το div slides.

4.3 Δημιουργία διαφανειών

Κάθε διαφάνεια είναι ένα στοιχείο section. Για την εμφάνιση του κειμένου χρειάζεται να εισάγουμε τα κατάλληλα στοιχεία HTML. Τα απλά στοιχεία κειμένου που χρησιμοποιούνται είναι τα h1, h2, p, ul και li.

4.3.1 Διαφάνειες για την HTML

Ο κώδικας για την πρώτη διαφάνεια είναι:


```
<div class="slides">

  <section>
    <h1>Παρουσίαση της HTML5</h1>
    
    <h3>Πτυχιακή εργασία Πέτρου Δαλακούρα</h3>
    <p>
      <small><a href="http://www.epp.teiher.gr">Τμήμα εφαρμοσμένης
πληροφορικής και Πολυμέσων</a>, ΤΕΙ Ηρακλείου.</small>
    </p>
  </section>

</div>
```

Αυτός παράγει:

Παρουσίαση της HTML5



Πτυχιακή εργασία Πέτρου Δαλακούρα

Τμήμα εφαρμοσμένης πληροφορικής και Πολυμέσων, ΤΕΙ Ηρακλείου.

Στις επόμενες διαφάνειες δίνονται κάποιες βασικές πληροφορίες για το που θα κινηθεί η παρουσίαση.

```
<section>
  <h2>Μειονεκτήματα της HTML</h2>
  <p>
    Η γλώσσα HTML δημιουργήθηκε για την παρουσίαση απλών σελίδων
    κειμένου. Όμως με το περασμα του
    χρόνου έγινε φανερό ότι είχε αρκετούς περιορισμούς.
  </p>
  <ul>
    <li>Είναι ακατάλληλη για την δημιουργία πολύπλοκων
    εφαρμογών</li>
    <li>Οι δυνατότητες μορφοποίησης περιεχομένου που προσφέρει δεν
    ανταποκρίνονται στις σύγχρονες ανάγκες</li>
    <li>Διαφορετικοί browsers έχουν μικρές αλλά αισθητές διαφορές
    στον τρόπο που εμφανίζουν συγκεκριμένες επιλογές</li>
  </ul>
</section>

<section>
  <h2>Αλλαγές στην HTML5</h2>
  <p>
    Το πρότυπο HTML5 δημιουργήθηκε για να καλύψει τις νέες ανάγκες
    που έχουν προκύψει,
    ενσωματώνοντας αλλαγές σε όλες τις πλευρές της ανάπτυξης μίας
    ιστοσελίδας.
  </p>
  <ul>
    <li>Προστέθηκαν νέα στοιχεία στην HTML</li>
    <li>Νέες ιδιότητες έχουν προστεθεί στα CSS</li>
    <li>Η γλώσσα Javascript έχει εμπλουτιστεί με νέες
```

```

</li>
</ul>
<p>
    Παρακάτω θα παρουσιάσουμε τις αλλαγές σε αυτούς τους τομείς.
</p>
</section>

<section>
    <h2>Τομείς που θα καλυφθούν</h2>
    <p>
        Δεν είναι δυνατόν να καλυφθούν όλες οι αλλαγές.
    </p>

    <p>
        <ul>
            <li>Το πρότυπο HTML5 περιέχει πάνω από 150 αλλαγές</li>
            <li>Αυτές χωρίζονται σε περίπου 25 επιμέρους πρότυπα</li>
            <li>Κάθε ένα από αυτά βρίσκεται σε διαφορετικό στάδιο
                υλοποίησης</li>
        </ul>
    </p>

    <p>
        Θα μελετήσουμε τους τομείς που βρίσκονται πιο κοντά στην
        καθολική αποδοχή και καθιέρωσή τους σαν πρότυπο.
    </p>
</section>

```

Όπως φαίνεται για να εμφανίσουμε λίστες με σημεία χρησιμοποιούμε στοιχεία li μέσα σε λίστες ul, όπως πχ παρακάτω.

Αλλαγές στην HTML5

Το πρότυπο HTML5 δημιουργήθηκε για να καλύψει τις νέες ανάγκες που έχουν προκύψει, ενσωματώνοντας αλλαγές σε όλες τις πλευρές της ανάπτυξης μίας ιστοσελίδας.

- Προστέθηκαν νέα στοιχεία στην HTML
 - Νέες ιδιότητες έχουν προστεθεί στα CSS
 - Η γλώσσα Javascript έχει εμπλουτιστεί με νέες λειτουργίες.
- Παρακάτω θα παρουσιάσουμε τις αλλαγές σε αυτούς τους τομείς.

Για την ενσωμάτωση κώδικα χρειάζονται στοιχεία pre μέσα στα οποία εισάγεται κώδικας με tag τύπου code. Η βιβλιοθήκη highlight.js αναλαμβάνει να χρωματίσει κατάλληλα τον κώδικα.

```
<section>
  <h2>Semantic markup 1/2</h2>
  <p>
    Η περιγραφή της δομής μίας σύγχρονης σελίδας γίνεται με στοιχεία
    div, κάθε ένα έχει την δική του class.
    Μέχρι τώρα η δομή μίας τυπικής σελίδας ήταν αυτή:
  </p>
  <pre><code class="html" contenteditable style="font-size: 18px; margin-
  top: 20px;">
  &lt;!DOCTYPE html&gt;
  &lt;html&gt;
    &lt;body&gt;
      &lt;div class="header"&gt;
        &lt;h1&gt;Page Title&lt;/h1&gt;
        &lt;h2&gt;Page Subtitle&lt;/h2&gt;
      &lt;/div&gt;
      &lt;div class="nav"&gt;&lt;/div&gt;
      &lt;div id="content"&gt;
        &lt;div class="article"&gt;&lt;/div&gt;
      &lt;/div&gt;
      &lt;div class="footer"&gt;&lt;/div&gt;
    &lt;/body&gt;
  &lt;/html&gt;
  </code></pre>
</section>
```

Στο συγκεκριμένο παράδειγμα που περιλαμβάνει κώδικα HTML χρειάζεται να αντικαταστήσουμε τα στοιχεία που ανοίγουν και κλείνουν τα tags με τους αντίστοιχους κωδικούς χαρακτήρων < και > ώστε να μην χαλάσει ο κώδικας της σελίδας.

Semantic markup 1/2

Η περιγραφή της δομής μίας σύγχρονης σελίδας γίνεται με στοιχεία div, κάθε ένα έχει την δική του class. Μέχρι τώρα η δομή μίας τυπικής σελίδας ήταν αυτή:

```
<!DOCTYPE html>
<html>
  <body>
    <div class="header">
      <h1>Page Title</h1>
      <h2>Page Subtitle</h2>
    </div>
    <div class="nav"></div>
    <div id="content">
      <div class="article"></div>
    </div>
    <div class="footer"></div>
  </body>
</html>
```

Μετά εισάγουμε την επόμενη διαφάνεια που περιέχει τον κώδικα που χρησιμοποιείται στην HTML5 με τον ίδιο τρόπο.

```
<section>
  <h2>Semantic markup 2/2</h2>
  <p>
    Πλέον έχουν εισαχθεί νέα στοιχεία που έχουν την συμπεριφορά των div
    αλλά κάνουν πιο εύκολη την
    ανάλυση μίας σελίδας.
  </p>
  <pre><code class="html" contenteditable style="font-size: 18px; margin-top: 20px;">
    &lt;!DOCTYPE html&gt;
    &lt;html&gt;
    &lt;body&gt;
      &lt;header&gt;
        &lt;h1&gt;Page Title&lt;/h1&gt;
        &lt;h2&gt;Page Subtitle&lt;/h2&gt;
      &lt;/header&gt;

      &lt;nav&gt;&lt;/nav&gt;

      &lt;article&gt;
        &lt;section&gt;&lt;/section&gt;
      &lt;/article&gt;

      &lt;footer&gt;&lt;/footer&gt;

    &lt;/body&gt;
    &lt;/html&gt;
  </code></pre>
</section>
```

Για τις φόρμες εισάγουμε μία δοκιμαστική φόρμα που δεν κάνει τίποτα και από κάτω τον κώδικά της. Η διαφάνεια για το email είναι η παρακάτω:

```
<section>
  <h2>Φόρμες</h2>
  <h3>Στοιχείο email</h3>
  <p>
    Το στοιχείο email φαίνεται σαν απλό πεδίο κειμένου αλλά δεν
    επιστρέφει την εισαγωγή κειμένου
    που δεν είναι έγκυρη διεύθυνση email.
  </p>

  <p>
    Στην παρακάτω φόρμα δεν μπορεί να πατηθεί το Go αν δεν συμπληρωθεί
    η διεύθυνση email.
  </p>

  <form>
    <input type="email">
    <input type="submit" value="Go">
  </form>

  <pre><code class="html" contenteditable style="font-size: 18px; margin-top:
  20px;">
    &lt;form&gt;
      &lt;input type=&quot;email&quot;&gt;
      &lt;input type=&quot;submit&quot; value=&quot;Go&quot;&gt;
    &lt;/form&gt;
  </code></pre>
</section>
```

Ομοίως και η διαφάνεια για το URL.

```
<section>
  <h2>Φόρμες</h2>
  <h3>Στοιχείο url</h3>
  <p>
    Όπως και το στοιχείο email, τα πεδία τύπου URL φαίνονται σαν
    κανονικά πεδία κειμένου
    αλλά είναι απαραίτητο να περιέχουν έγκυρες διευθύνσεις.
  </p>

  <p>
    Στην παρακάτω φόρμα δεν μπορεί να πατηθεί το Go αν δεν συμπληρωθεί
    η διεύθυνση url.
  </p>

  <form>
    <input type="url">
```

```

<input type="submit" value="Go">
</form>

<pre><code class="html" contenteditable style="font-size: 18px; margin-top:
20px;">
  &lt;form&gt;
    &lt;input type=&quot;url&quot;&gt;
    &lt;input type=&quot;submit&quot; value=&quot;Go&quot;&gt;
  &lt;/form&gt;

</code></pre>
</section>

```

Το αποτέλεσμα είναι:

Φόρμες Στοιχείο url

Όπως και το στοιχείο email, τα πεδία τύπου URL φαίνονται σαν κανονικά πεδία κειμένου αλλά είναι απαραίτητο να περιέχουν έγκυρες διευθύνσεις.

Στην παρακάτω φόρμα δεν μπορεί να πατηθεί το Go αν δεν συμπληρωθεί η διεύθυνση url.

```

<form>
  <input type="url">
  <input type="submit" value="Go">
</form>

```

Η διαφάνεια για την ιδιότητα placeholder:

```

<section>
  <h2>Φόρμες</h2>
  <h3>Placeholder</h3>
  <p>
    Κάθε πεδίο κειμένου μπορεί να περιέχει ένα κείμενο που θα δείχνει
    στον χρήστη τι θα γράψει.
  </p>
  <p>
    Το κείμενο εξαφανίζεται όταν γραφτεί κάτι στο πεδίο.
  </p>
  <form>
    <input type="text" placeholder="Your text here...">

```

```

    <input type="submit" value="Go">
  </form>

<pre><code class="html" style="font-size: 18px; margin-top: 20px;">
  &lt;form&gt;
    &lt;input type=&quot;text&quot; placeholder=&quot;Your text
here...&quot;&gt;
    &lt;input type=&quot;submit&quot; value=&quot;Go&quot;&gt;
  &lt;/form&gt;

</code></pre>
</section>

```

Η διαφάνεια για τα πεδία αριθμών:

```

<section>
  <h2>Φόρμες</h2>
  <h3>Εισαγωγή αριθμών</h3>
  <p>
    Είναι εφικτή η εισαγωγή αριθμών και ο ορισμός αποδεκτών ορίων.
  </p>

  <form>
    <input type="number" min="0" max="100" step="10">
    <input type="submit" value="Go">
  </form>

<pre><code class="html" style="font-size: 18px; margin-top: 20px;">
  &lt;form&gt;
    &lt;input type=&quot;number&quot; min=&quot;0&quot; max=&quot;100&quot;
step=&quot;10&quot;&gt;
    &lt;input type=&quot;submit&quot; value=&quot;Go&quot;&gt;
  &lt;/form&gt;

</code></pre>
</section>

```

Η διαφάνεια για τους sliders:

```

<section>
  <h2>Φόρμες</h2>
  <h3>Εισαγωγή αριθμών με slider</h3>
  <p>
    Εναλλακτικά μπορούν να εισαχθούν νούμερα με slider.
  </p>

  <form>
    <input type="range" min="0" max="100" step="10">
    <input type="submit" value="Go">
  </form>

<pre><code class="html" style="font-size: 18px; margin-top: 20px;">

```

```

<form>
  <input type="range" min="0" max="100"
step="10">
  <input type="submit" value="Go">
</form>

</code></pre>
</section>

```

Το πεδίο επιλογής χρώματος:

```

<section>
  <h2>Φόρμες</h2>
  <h3>Επιλογή χρώματος</h3>
  <p>
    Ένα νέο πεδίο επιτρέπει την επιλογή χρώματος από ένα κατάλληλο
πλαίσιο διαλόγου.
  </p>

  <form>
    <input type="color">
    <input type="submit" value="Go">
  </form>

<pre><code class="html" style="font-size: 18px; margin-top: 20px;">
  <form>
    <input type="color">
    <input type="submit" value="Go">
  </form>

</code></pre>
</section>

```

Τέλος η διαφάνεια επιλογής ημερομηνίας

```

<section>
  <h2>Φόρμες</h2>
  <h3>Επιλογή ημερομηνίας</h3>
  <p>
    Η ημερομηνία είναι ένας από τους πιο πολύπλοκους τύπους δεδομένων
που χρειάζεται να εισάγει
    ο χρήστης. Πλέον υποστηρίζεται εγγενώς.
  </p>

  <form>
    <input type="date">
    <input type="submit" value="Go">
  </form>

<pre><code class="html" style="font-size: 18px; margin-top: 20px;">
  <form>
    <input type="date">

```

```

    &lt;input type=&quot;submit&quot; value=&quot;Go&quot;&gt;
  &lt;/form&gt;

</code></pre>
</section>

```

Η διαφάνεια για το novalidate:

```

<section>
  <h2>Φόρμες</h2>
  <h3>Παράληψη επιβεβαίωση στοιχείων φόρμας</h3>
  <p>
    Αν για κάποιο λόγο δεν είναι επιθυμητή η επιβεβαίωση των στοιχείων
    μίας φόρμας τότε μπορεί να μπει η
    ιδιότητα novalidate.
  </p>

  <pre><code class="html" style="font-size: 18px; margin-top: 20px;">
    &lt;form novalidate&gt;
      &lt;input type=&quot;email&quot;&gt;
      &lt;input type=&quot;submit&quot; value=&quot;Go&quot;&gt;
    &lt;/form&gt;
  </code></pre>
</section>

```

4.3.2 Διαφάνειες για τις νέες ιδιότητες CSS

Έπειτα προχωράμε στην παρουσίαση των νέων στοιχείων CSS.

```

<section>
  <h2>Στοιχεία CSS</h2>
  <h3>Σκιά γύρω από αντικείμενα και στρογγυλεμένες γωνίες</h3>

  <pre><code style="font-size: 18px; margin-top: 20px;">
    box-shadow: 8px 6px 15px #f00;
    border-radius: 10px;

  </code></pre>

  <div style="background-color: #77F; border: 5px solid #FF0; width: 400px;
  height: 200px; margin: auto; box-shadow: 8px 6px 15px #f00;border-radius:
  10px;"></div>

</section>

```

Καθώς η χρήση CSS δεν είναι πρακτική οι ιδιότητες προσδιορίζονται στο πεδίο style του στοιχείου.

```

<section>
  <h2>Στοιχεία CSS</h2>
  <h3>Σκιά γύρω από αντικείμενα και στρογγυλεμένες γωνίες</h3>

  <pre><code style="font-size: 18px; margin-top: 20px;">
    box-shadow: 8px 6px 15px #f00;
    border-radius: 10px;

  </code></pre>

  <div style="background-color: #77F; border: 5px solid #FF0; width: 400px;
  height: 200px; margin: auto; box-shadow: 8px 6px 15px #f00;border-radius:
  10px;"></div>

</section>

```

Μετά δείχνουμε την σκιά που μπορεί να έχουν τα γράμματα.

```

<section>
  <h2>Στοιχεία CSS</h2>
  <h3>Σκιά γύρω από τα γράμματα.</h3>

  <pre><code style="font-size: 18px; margin-top: 20px;">
    text-shadow: 10px 10px #f00;
    text-shadow: 10px 10px 20px #f00;

  </code></pre>

  <h3 style="text-shadow: 10px 10px #ff0000;">Πρώτο παράδειγμα</h3>
  <h3 style="text-shadow: 10px 10px 20px #ff0000;">Δεύτερο
  παράδειγμα</h3>

</section>

```

Μετά παρουσιάζονται τα CSS transformations.

```

<section>
<h2>CSS tranformations</h2>
<p>
Η εφαρμογή δισδιάστατων και τρισδιάστατων εφέ είναι εφικτή με την χρήση του
CSS.
</p>

<pre style="float:left;width: 400px; border: solid red; background-
color:#999; -webkit-transform: rotate(45deg); -webkit-transform-origin: 10%
50%;">
transform: rotate(45deg);
transform-origin: 10% 50%;
</pre>

```



```

<pre style="float:left;margin-top:200px;width: 450px;border: solid red;
background-color:#999; -webkit-transform: rotate(90deg); -webkit-transform-
origin: 50% 50%;">

transform: rotate(90deg);
transform-origin: 50% 50%;
</pre>
<div class="clear"></div>
</section>

```

Τέλος δείχνουμε τα transitions.

```

<section>
  <h2>Στοιχεία CSS</h2>
  <h3>Transitions.</h3>
  <p>
    Τα transition επιτρέπουν την σταδιακή μεταβολή ενός μεγέθους. Εδώ
    όταν το ποντίκι περνάει
    πάνω από το κείμενο αλλάζει σταδιακά το χρώμα του.
  <pre><code style="font-size: 18px; margin-top: 20px;">
  #a1 { -webkit-transition: color 2s linear;}
  #a1:hover { color: #F00;}
  </code></pre>
  <style>
    #a1 { -webkit-transition: color 2s linear;}
    #a1:hover { color: #F00;}
  </style>

  <h1 id="a1">Παράδειγμα</h3>
</section>

```

Έπειτα παρουσιάζουμε το στοιχείο canvas.

```

<section>
  <h2>Canvas</h2>
  <p>
    Το στοιχείο canvas είναι μία επιφάνεια ζωγραφικής πάνω στην οποία
    μπορούν
    να ζωγραφιστούν απλά και πολύπλοκα γραφικά.
  </p>
  <p>
    Η εισαγωγή του σκοπεύει να καταστήσει εφικτή την δημιουργία
    παιχνιδιών και animation.
  </p>
</section>

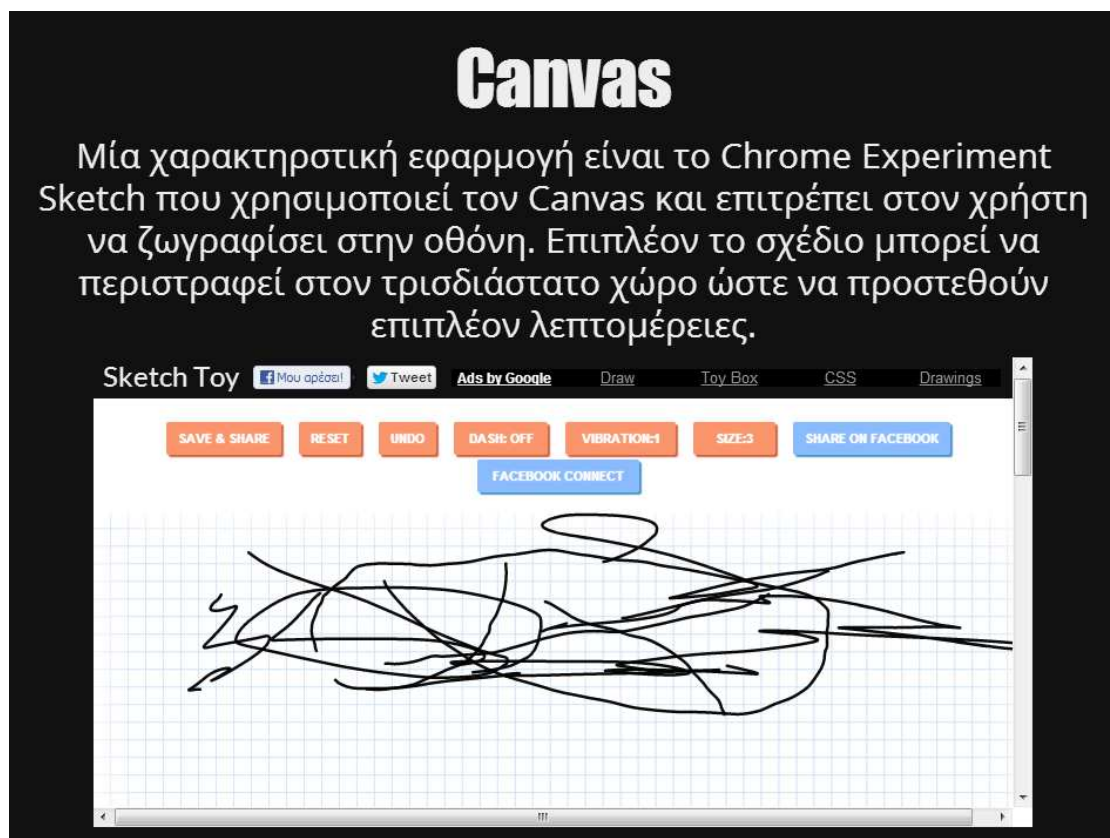
```

4.3.3 Διαφάνειες με Chrome Experiments

Σαν παραδείγματα δείχνουμε κάποια πειράματα από τα Chrome Experiments.

```
<section>
  <h2>Canvas</h2>
  <p>
    Μία χαρακτηριστική εφαρμογή είναι το Chrome Experiment Sketch που
    χρησιμοποιεί τον Canvas και
    επιτρέπει στον χρήστη να ζωγραφίσει στην οθόνη. Επιπλέον το σχέδιο
    μπορεί να περιστραφεί στον
    τρισδιάστατο χώρο ώστε να προστεθούν επιπλέον λεπτομέρειες.
  </p>
  <iframe src="http://hakim.se/experiments/html5/sketch/" width="800"
  height="400"></iframe>
</section>
```

Η χρήση του iframe δίνει το παρακάτω αποτέλεσμα:



Η διαφάνεια για το WebGL:

```
<section>
```

```

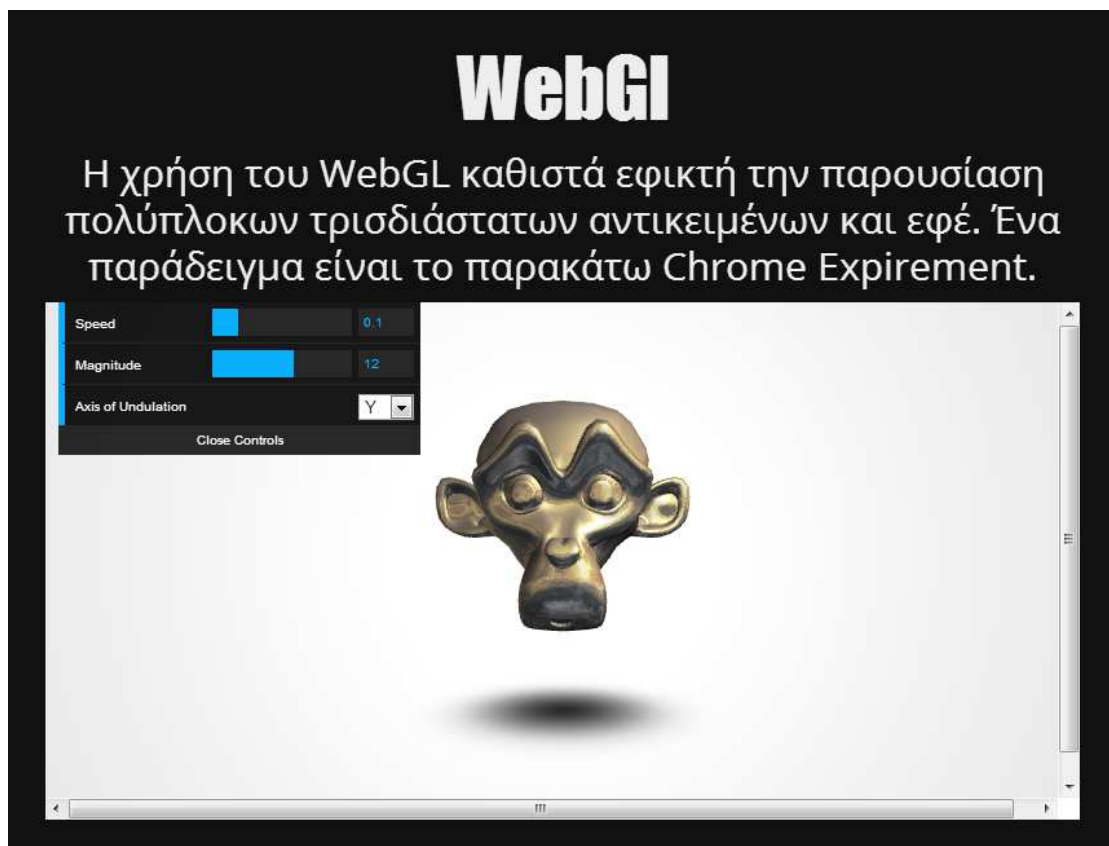
<h2>WebGL</h2>
<p>
Η χρήση του WebGL καθιστά εφικτή την παρουσίαση πολύπλοκων τρισδιάστατων
αντικειμένων και εφέ.
Ένα παράδειγμα είναι το παρακάτω Chrome Experiment.
</p>

<iframe src="http://lab.aerotwist.com/webgl/undulating-monkey" width="800"
height="400"></iframe>

<div class="clear"></div>
</section>

```

Έτσι παράγεται αυτή η διαφάνεια:



Μετά παρουσιάζονται κάποιες εφαρμογές.

```

<section>
<h2>Εφαρμογές</h2>
<p>
Πλέον έχουν αρχίσει να αναπτύσσονται πλήρεις εφαρμογές με αρκετές
δυνατότητες.
</p>

<iframe src="http://mrdoob.com/projects/code-editor/" width="800"

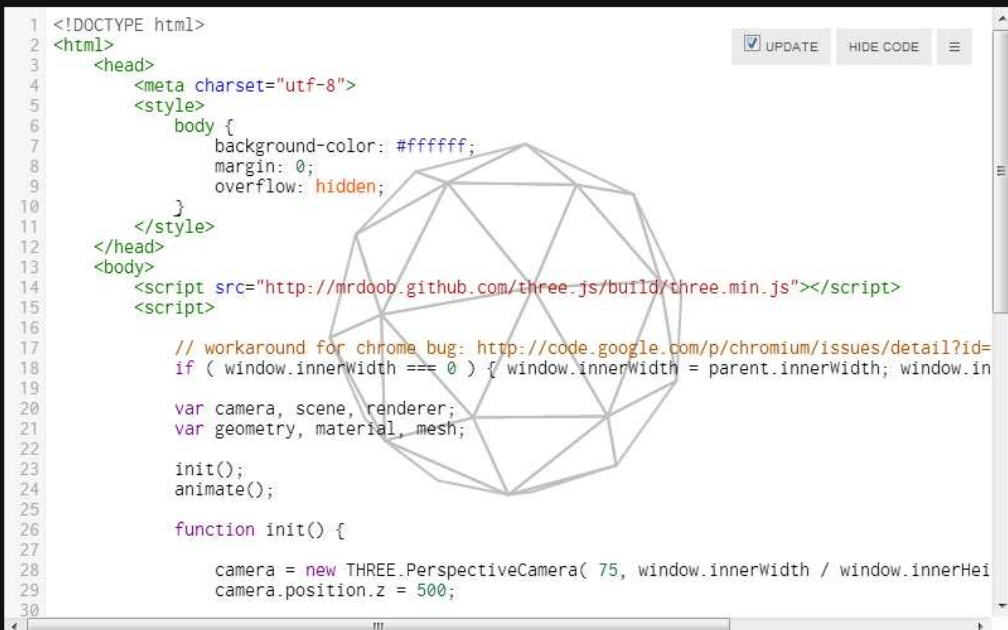
```

```
height="500"></iframe>
<div class="clear"></div>
</section>
```

Η εφαρμογή ενσωματώνεται στην σελίδα ως εξής:

Εφαρμογές

Πλέον έχουν αρχίζει να αναπτύσσονται πλήρεις εφαρμογές με αρκετές δυνατότητες.



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <style>
6       body {
7         background-color: #ffffff;
8         margin: 0;
9         overflow: hidden;
10    }
11  </style>
12 </head>
13 <body>
14   <script src="http://mrdoob.github.com/three.js/build/three.min.js"></script>
15   <script>
16
17     // workaround for chrome bug: http://code.google.com/p/chromium/issues/detail?id=
18     if ( window.innerWidth === 0 ) { window.innerWidth = parent.innerWidth; window.in
19
20     var camera, scene, renderer;
21     var geometry, material, mesh;
22
23     init();
24     animate();
25
26     function init() {
27
28       camera = new THREE.PerspectiveCamera( 75, window.innerWidth / window.innerHei
29       camera.position.z = 500;
30
```

5 Συμπεράσματα

Το πρώτο συμπέρασμα στο οποίο καταλήγει κάποιος μετά την ενασχόληση με την HTML5 είναι ότι πλέον έχει ξεφύγει κατά πολύ από τους στόχους που είχε κληθεί αρχικά να καλύψει η γλώσσα HTML. Σε αντίθεση με την απλή δημιουργία σελίδων, για η οποία έχουν συμπεριληφθεί αρκετές νέες δυνατότητες μεγαλύτερο βάρος έχει δοθεί σε αρκετά προχωρημένες λειτουργίες οι οποίες μέχρι τώρα υπήρχαν σε παραδοσιακές γλώσσες προγραμματισμού. Η ίδια η γλώσσα στοχεύει να διευκολύνει την ανάπτυξη εφαρμογών και όχι απλά την παρουσίασή τους.

Βέβαια αυτό συνεπάγεται ότι αρκετά μεγάλο μέρος από αυτές τις λειτουργίες βρίσκεται σε πολύ πρώιμο στάδιο, με πολύ λίγες από αυτές να έχουν υλοποιηθεί έστω και πειραματικά. Κάποιες άλλες, κυρίως αυτές που αναπτύσσουν τις υπάρχουσες δυνατότητες έχουν οριστικοποιηθεί και υποστηρίζονται σε μεγάλο βαθμό από σχεδόν όλους τους browsers.

Η προσεκτική διαδικασία υιοθέτησης και ελέγχουν κάθε νέα λειτουργίας σημαίνει ότι το συνολικό πρότυπο θα αργήσει αρκετά χρόνια να παγιωθεί. Όμως αυτή η διαδικασία εγγυάται και την ακριβή υιοθέτησή του από όλους τους κατασκευαστές, με αποτέλεσμα να έχει ήδη πετύχει τον κυριότερο στόχο του, δηλαδή την καθολική αποδοχή.

Βιβλιογραφία

- hakimel / reveal.js*. (n.d.). Retrieved from github: <https://github.com/hakimel/reveal.js>
- Adobe Flash*. (n.d.). Retrieved from Wikipedia: http://en.wikipedia.org/wiki/Adobe_Flash
- Berners-Lee, T. (March 1989, May 1990). *Information Management: A Proposal*.
- Berners-Lee, T. (n.d.). *Re: SGML/HTML docs, X Browser*. Retrieved from www-talk mailing list post: <http://lists.w3.org/Archives/Public/www-talk/1991NovDec/0020.html>
- Berners-Lee, T., & Connelly, D. (November 1995). *"RFC 1866 - Hypertext Markup Language - 2.0*.
- Beyond HTML5: Database APIs and the Road to IndexedDB*. (n.d.). Retrieved from mozilla.org: <http://hacks.mozilla.org/2010/06/beyond-html5-database-apis-and-the-road-to-indexeddb/>
- Choosing a Language Tag*. (n.d.). Retrieved from World Wide Web Consortium.: <http://www.w3.org/International/questions/qa-choosing-language-tags>
- Form Fields*. (n.d.). Retrieved from HTML5 ROCKS: http://playground.html5rocks.com/#form_fields
- FOX News: No Flash on the iPhone? Apple's Steve Jobs Finally Explains Why*. (n.d.). Retrieved from Fox News: <http://www.foxnews.com/scitech/2010/04/29/flash-iphone-apples-steve-jobs-finally-explains/>
- HTML*. (n.d.). Retrieved from Wikipedia: <http://en.wikipedia.org/wiki/HTML>
- HTML 4 Errata*. (n.d.). Retrieved from World Wide Web Consortium: <http://www.w3.org/MarkUp/html4-updates/errata>
- HTML 4.0 Specification*. (n.d.). Retrieved from World Wide Web Consortium: <http://www.w3.org/TR/1998/REC-html40-19980424/>
- HTML Living Standard - Common infrastructure*. (n.d.). Retrieved from WHATWG: <http://www.whatwg.org/specs/web-apps/current-work/multipage/infrastructure.html#xml>
- HTML Living Standard - Offline Web applications*. (n.d.). Retrieved from WHATWG: <http://www.whatwg.org/specs/web-apps/current-work/multipage/offline.html>
- HTML Living Standard - States of the type attribute*. (n.d.). Retrieved from WHATWG: <http://www.whatwg.org/specs/web-apps/current-work/multipage/states-of-the-type-attribute.html#e-mail-state-%28type=>
- HTML5*. (n.d.). Retrieved from Wikipedia: <http://en.wikipedia.org/wiki/HTML5>

HTML5 Differences from HTML4. (n.d.). Retrieved from World Wide Web Consortium: <http://www.w3.org/TR/2011/WD-html5-diff-20110405/>

Introduction to Web Storage. (n.d.). Retrieved from MSDN: <http://msdn.microsoft.com/en-us/library/cc197062%28VS.85%29.aspx>

New elements in HTML5: Structure and semantics. (n.d.). Retrieved from IBM Developer Works: <http://www.ibm.com/developerworks/library/x-html5/?ca=dgr-Inxw01NewHTML>

Pilgrim, M. (n.d.). *Dive Into HTML5.* Retrieved from Dive Into HTML5: <http://diveintohtml5.info/>

Reveal.js. (n.d.). Retrieved from lab.hakim.se: <http://lab.hakim.se/reveal-js/#/>

Scalable Vector Graphics in HTML5. (n.d.). Retrieved from IBM Developer Works: <http://www.ibm.com/developerworks/library/wa-scalable/>

Sidwell, A. (n.d.). *review of "The root element" subsection.* Retrieved from World Wide Web Consortium: <http://lists.w3.org/Archives/Public/public-html/2007Jul/0550.html>

Sivonen, H. (n.d.). *Activating Browser Modes with Doctype.* Retrieved from Hsivonen.iki.fi: <http://hsivonen.iki.fi/doctype/>

Using CSS transforms. (n.d.). Retrieved from MDN: https://developer.mozilla.org/en/docs/CSS/Using_CSS_transforms

Using CSS transitions. (n.d.). Retrieved from MDN: https://developer.mozilla.org/en-US/docs/CSS/Using_CSS_transitions

Web SQL Database. (n.d.). Retrieved from World Wide Web Consortium: <http://dev.w3.org/html5/webdatabase/>

Web Storage. (n.d.). Retrieved from World Wide Web Consortium : <http://dev.w3.org/html5/webstorage/>

WebGL. (n.d.). Retrieved from Wikipedia: <http://en.wikipedia.org/wiki/WebGL>

WebGL - OpenGL ES 2.0 for the Web. (n.d.). Retrieved from Khronos Group: <http://www.khronos.org/webgl/>

When Will HTML5 Be Finished? (n.d.). Retrieved from WHAT Working Group: http://wiki.whatwg.org/wiki/FAQ#What.27s_this_I_hear_about_2022.3F

XHTML 1.0 – Differences with HTML 4. (n.d.). Retrieved from World Wide Web Consortium: <http://www.w3.org/TR/xhtml1/diffs.html#h-4.4>

XHTML 1.0: The Extensible HyperText Markup Language (Second Edition). (n.d.). Retrieved from World Wide Web Consortium: <http://www.w3.org/TR/xhtml1/>

XHTML 1.1 – Module-based XHTML – Second Edition. (n.d.). Retrieved from World Wide Web Consortium: <http://www.w3.org/TR/xhtml11/>

XHTML 2 Working Group Expected to Stop Work End of 2009, W3C to Increase Resources on HTML5. (n.d.). Retrieved from World Wide Web Consortium: <http://www.w3.org/News/2009#item119>