



---

# **HTML 5.0 – XHTML 2.0**

---

## **ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

---

Εισηγήτριες: Τσούλη Δήμητρα 333  
Τύπα Μαρία 374

Επιβλέπων: Χριστοδούλου Σωτήριος

©

**2013**

## Περίληψη

Οι γλώσσες σήμανσης αρχικά δημιουργήθηκαν για την ανάγκη παρουσίασης ενός εγγράφου με διακριτό τρόπο, δίνοντας τις κατάλληλες οδηγίες για το πρόγραμμα παρουσίασης σχετικά με τον τρόπο απεικόνισης του εγγράφου. Δημιουργήθηκαν και εξελίχθηκαν για διαφορετικούς λόγους κάθε φορά (π.χ την αποστολή εγγράφων μέσω δικτύων , την περιγραφή ιδιαίτερων χαρακτηριστικών ηλεκτρονικών εγγράφων κλπ). Παρ' όλες τις διαφορές τους, όλες αποτελούνται κοινά χαρακτηριστικά αλλά και ετικέτες που κληρονομήθηκαν από υπεργλώσσες πρότυπα όπως η SGML.

Η HTML αναμφισβήτητα είναι η πιο δημοφιλής γλώσσα σήμανση η οποία υιοθέτησε αυτά τα χαρακτηριστικά και εξελίχθηκε μέσα από τις διάφορες εκδόσεις της στην τελευταία την HTML 4.01. Η εξέλιξη συνεχίζεται με την δημιουργία του νέου προτύπου της HTML 5.0 η οποία είναι η προσπάθεια μιας ομάδας κατασκευαστών στην αρχή του μετέπειτα του οργανισμού W3C. Περιέχει αρκετά νέα στοιχεία και δυνατότητες, ενώ και αυτή τη στιγμή που γράφεται υπάρχει μια διαρκής προσθήκη νέων χαρακτηριστικών

Η XHTML, ήρθε να καλύψει τα κενά που είχε αφήσει η HTML, προσθέτοντας πιο αυστηρούς κανόνες γραφής δίνοντας έναν «καθαρότερο» από σφάλματα κώδικα. Βασίστηκε στις προδιαγραφές της XML, ορίζοντας μια γλώσσα αποκλειστικά για την περιγραφή των εγγράφων και λιγότερο για την παρουσίαση. Η XHTML 2.0 αποτέλεσε επίσημο υπό ανάπτυξη πρότυπο του οργανισμού W3C, αλλά το 2009 εγκαταλείφθηκε και παρέμεινε προσχέδιο.

Η σύγκριση των δύο γλωσσών τόσο στα γλωσσικά στοιχεία, όσο και στις δυνατότητες δεν δίνει ξεκάθαρο αποτέλεσμα. Το αποτέλεσμα έχει να κάνει περισσότερο από την οπτική γωνία σύγκρισης καθώς η HTML 5.0 αποτελεί ένα υπερσύνολο της XHTML 2.0. Από την άλλη όμως η XHTML 2.0 είναι αυτό που εννοούμε όταν λέμε γλώσσα σήμανσης και όχι ένα ολοκληρωμένο διαδικτυακό περιβάλλον ανάπτυξης.

## **Abstract**

The markup languages were initially created by the need of presenting a document in a distinct way, giving appropriate instructions to the presentation program related to the document imaging mode. Each time these languages were created and developed for different reasons (eg sending documents through networks, describing special characteristics of electronic documents, etc.). Despite all their differences, they are all have common features and tags inherited from standards like SGML.

HTML is arguably the most popular markup language, which adopted these characteristics and evolved through several versions in the latest HTML 4.01. The evolution continues with the creation of a new standard of HTML 5.0 which is the effort of a group of manufacturers at the beginning of the subsequent organization of W3C. It contains several new elements and capabilities, while at the time of writing there is a constant addition of new features.

XHTML, came to fill the gaps that had left the HTML, adding more strict 'writing rules', giving a "cleaner" from coding errors. Based on the specifications of XML, by defining a language exclusively for describing documents and less about the presentation. The XHTML 2.0 became official standard under development of W3C organization, but in 2009 was abandoned and remained draft.

The comparison of the two languages both linguistic features, and capabilities gives no clear result. The result has more to do from the perspective of comparison as HTML 5.0 is a superset of XHTML 2.0. On the other hand, the XHTML 2.0 is what we mean, when we say markup language rather than an integrated web development environment.

Keywords: Markup Languages, HTML 5.0, XHTML 2.0

## Πίνακας Περιεχομένων

Κεφάλαιο 1- Εισαγωγή .....	9
1.1. Γλώσσες σήμανσης (Markup Languages) .....	9
1.2. Ιστορικά στοιχεία .....	9
1.3. Χαρακτηριστικά .....	12
1.3.1. Τύποι Σήμανσης .....	13
1.3.2. Ανεξαρτησία δεδομένων .....	16
1.3.3. Τύπος Εγγράφου .....	16
1.4. Χρήσεις .....	16
Κεφάλαιο 2- HTML 5 .....	18
2.1. Εισαγωγή .....	18
2.2. Περιγραφή – Εξέλιξη της HTML .....	18
2.3. Δομή της HTML5 .....	20
2.3.1. Σύνταξη .....	21
2.3.2. Μεταδεδομένα .....	22
2.3.3. Σενάρια - Αρχεία .....	23
2.3.4. Στοιχεία HTML5 .....	26
2.3.5. Ενότητες .....	27
2.3.6. Ομαδοποίηση περιεχομένου .....	28
2.3.7. Σημασιολογικά στοιχεία κειμένου .....	31
2.3.8. Επεξεργασμένο κείμενο .....	33
2.3.9. Ενσωματωμένο περιεχόμενο .....	33
2.3.10. Canvas .....	35
2.3.11. Φόρμες .....	39
2.3.12. Πίνακες .....	43
2.3.13. Σύνδεσμοι .....	45

2.4. Διεπαφές προγραμματισμού (APIs).....	47
2.4.1. Communication APIs.....	47
2.4.2. Διεπαφή συστήματος αρχείων (Filesystem APIs) .....	49
2.4.3. Geolocation API.....	51
Κεφάλαιο 3- XHTML 2.0.....	54
3.1. Εισαγωγή .....	54
3.2. Περιγραφή της XHTML .....	54
3.3. Ενότητες της XHTML 2.0 .....	56
3.3.1. Ενότητα εγγράφου .....	57
3.3.2. Διάρθρωση.....	57
3.3.3. Κείμενο .....	59
3.3.4. Υπερκείμενο.....	60
3.3.5. Λίστες.....	60
3.3.6. Ενότητα ιδιοτήτων πυρήνα .....	61
3.3.7. Πρόσβαση .....	61
3.3.8. Ιδιότητες ενσωματωμένου περιεχομένου .....	62
3.3.9. Εικόνες.....	62
3.3.10. Χάρτες Εικόνων .....	63
3.3.11. Πολυμέσα.....	63
3.3.12. Μεταπληροφορία .....	63
3.3.13. Φύλλα Στυλ.....	64
3.3.14. Πίνακες .....	65
3.3.15. Φόρμες .....	66
3.3.16. Αντικείμενα.....	68
3.3.17. XHTML 2.0 Events .....	69
3.3.18. XHTML 2.0 Handlers.....	69
3.3.19. XFrames.....	70

3.3.20. Άλλες ενότητες .....	71
Κεφάλαιο 4- Σύγκριση.....	72
4.1. Η φιλοσοφία πίσω από την XHTML V2 .....	72
4.2. Η φιλοσοφία πίσω από την HTML 5.....	73
4.3. Σημασιολογική Σήμανση .....	74
4.4. Στοιχεία γλωσσών σήμανσης.....	75
4.5. Λίστες πλοήγησης.....	81
4.6. XForms – Html Forms .....	81
4.7. Νέες δυνατότητες μόνο στην HTML 5.....	83
4.7.1. Web Application APIs .....	83
4.7.2. Ανίχνευση χρηστών .....	84
Κεφάλαιο 5- Υλοποίηση και σύγκριση .....	85
5.1. Τύπος εγγράφου.....	86
5.2. Κεφαλίδα.....	86
5.3. Κυρίως μέρος.....	87
5.3.1. Επικεφαλίδα .....	87
5.3.2. Πλοήγηση .....	88
5.3.3. Περιεχόμενο.....	90
5.3.4. Χάρτες Εικόνων.....	95
5.3.5. Υποσέλιδο.....	98
Συμπεράσματα .....	99
Βιβλιογραφία .....	104

## **Εικόνες**

Εικόνα 1-1 Ιστορική εξέλιξη στις κύριες γλώσσες σήμανσης.....	10
Εικόνα 2-1 Δέντρο DOM.....	21
Εικόνα 2-2 Javascript.....	25

Εικόνα 2-3 Εφέ σκιάς .....	26
Εικόνα 2-4 HTML 5 - Σχέδιο εγγράφου.....	28
Εικόνα 2-5 Χρήση στοιχείων κειμένου .....	30
Εικόνα 2-6 Αριθμημένες λίστες.....	30
Εικόνα 2-7 Περιγραφική λίστα.....	31
Εικόνα 2-8 Σημασιολογικά στοιχεία κειμένου .....	32
Εικόνα 2-9 Σύστημα συντεταγμένων canvas .....	37
Εικόνα 2-10 Παράδειγμα σχεδιασμού σε καμβά.....	38
Εικόνα 2-11 Φόρμα HTML5 .....	42
Εικόνα 2-12 - Φόρμες μηνύματα λάθους.....	42
Εικόνα 2-13 Αποτέλεσμα χρήσης href .....	46
Εικόνα 2-14 Αποτέλεσμα χρήσης της area, map .....	46
Εικόνα 2-15 Παράδειγμα εφαρμογής Filesystem API.....	50
Εικόνα 3-1 Μοντέλο MVC .....	67

## **Πίνακες**

Πίνακας 2-1 Νέα στοιχεία της HTML 5 .....	27
Πίνακας 3-1 Ιδιότητες Πυρήνα.....	61
Πίνακας 3-2 Στοιχεία και ιδιότητες αντικειμένων.....	68
Πίνακας 3-3 Σχέδιο Πλαισίων .....	70
Πίνακας 4-1 Στοιχεία HTML5 ^ XHTML 2.0.....	75

## **Σκοπός της εργασίας**

Η παρούσα εργασία, έχει σαν σκοπό την διερεύνηση των δυνατοτήτων των νέων εκδόσεων, γλωσσών του διαδικτύου, δηλαδή της HTML5 και της XHTML 2.0. Αρχικά θα γίνει αναφορά στις γλώσσες σήμανσης και στην ιστορική εξέλιξη τους. Θα περιγραφούν συνοπτικά τα κύρια χαρακτηριστικά, αλλά και η χρήση τους.

Στο δεύτερο κεφάλαιο θα αναλυθεί διεξοδικά η γλώσσα HTML5. Θα αναφερθούμε στα κυριότερα στοιχεία της, ιδιαίτερα δε σε εκείνα τα σημεία τα οποία αποτελούν τα καινούργια χαρακτηριστικά στη γλώσσα.

Στο τρίτο κεφάλαιο θα προσεγγίσουμε την XHTML 2.0, και θα παρουσιάσουμε μέσα από τις σημειώσεις της Κοινοπραξίας του Παγκόσμιου Ιστού τις ενότητες και τα χαρακτηριστικά που την απαρτίζουν.

Το τέταρτο κεφάλαιο αποτελεί μια σύγκριση μεταξύ των κοινών σημείων των δυο γλωσσών, αλλά και των διαφορών τους σε θεωρητικό επίπεδο, ενώ στο πέμπτο θα συγκρίνουμε τις δύο γλώσσες σε επίπεδο κώδικα. Τέλος θα προσπαθήσουμε να εξάγουμε συμπεράσματα, σχετικά με τα αποτελέσματα αυτής της σύγκρισης.



## Εισαγωγή

### Γλώσσες σήμανσης (Markup Languages)

Μια γλώσσα σήμανσης είναι ένα σύστημα για τον σχολιασμό και τη παρουσία ενός εγγράφου με διακριτό τρόπο. Η ονομασία προήρθε από τη διαδικασία αναθεώρησης εγγράφων από τους συντάκτες. Οι οδηγίες ή διορθώσεις επισημαίνονταν με μπλε μολύβι στα χειρόγραφα των συγγραφέων. Τέτοια παραδείγματα στοιχειοθεσίας συναντώνται στην troff<sup>1</sup> και LaTeX<sup>2</sup>, αλλά και στις ετικέτες σήμανσης αυτών των γλωσσών.

Η σήμανση καθοδηγεί το λογισμικό να προβεί στην εμφάνιση του κειμένου, σύμφωνα με τις οδηγίες που ενσωματώνονται. Μερικές γλώσσες σήμανσης, όπως η HTML, έχουν προκαθορισμένη σημασιολογική παρουσίαση, πράγμα που σημαίνει ότι οι προδιαγραφές τους ορίζουν, πως τα δεδομένα πρόκειται να παρουσιαστούν. Άλλες δεν υποστηρίζουν αυτή τη δυνατότητα όπως για παράδειγμα η XML.

Η HTML είναι μια ευρέως χρησιμοποιούμενη γλώσσα σήμανσης υπέρ – κειμένου, δηλαδή εγγράφων του διαδικτύου. Αποτελεί μια εκδοχή της SGML, παρ' όλο που δεν απόλυτα συμβατή με αυτή. Η SGML αποτελείται από κανόνες σήμανσης, οι οποίοι συναντώνται στη βιομηχανία των εκδόσεων μεταξύ των συγγραφέων, εκδοτών και εκτυπωτικών συστημάτων.

### Ιστορικά στοιχεία

Η πρώτη γνωστή δημόσια παρουσίαση μιας γλώσσας σήμανσης στην επιστήμη των υπολογιστών, έγινε από τον William W. Tunnicliffe σε ένα συνέδριο το 1967, αν και ο ίδιος ονόμασε την παρουσίαση ως «γενική κωδικοποίηση». Στη δεκαετία του 1970 ο Tunnicliffe πραγματοποίησε την ανάπτυξη ενός προτύπου που ονομάστηκε GenCode και εφαρμόστηκε στην εκδοτική βιομηχανία. Αργότερα, ήταν ο πρώτος πρόεδρος της επιτροπής του Διεθνούς Οργανισμού Τυποποίησης που δημιούργησε τη SGML, το πρώτο πρότυπο περιγραφικής γλώσσας σήμανσης.

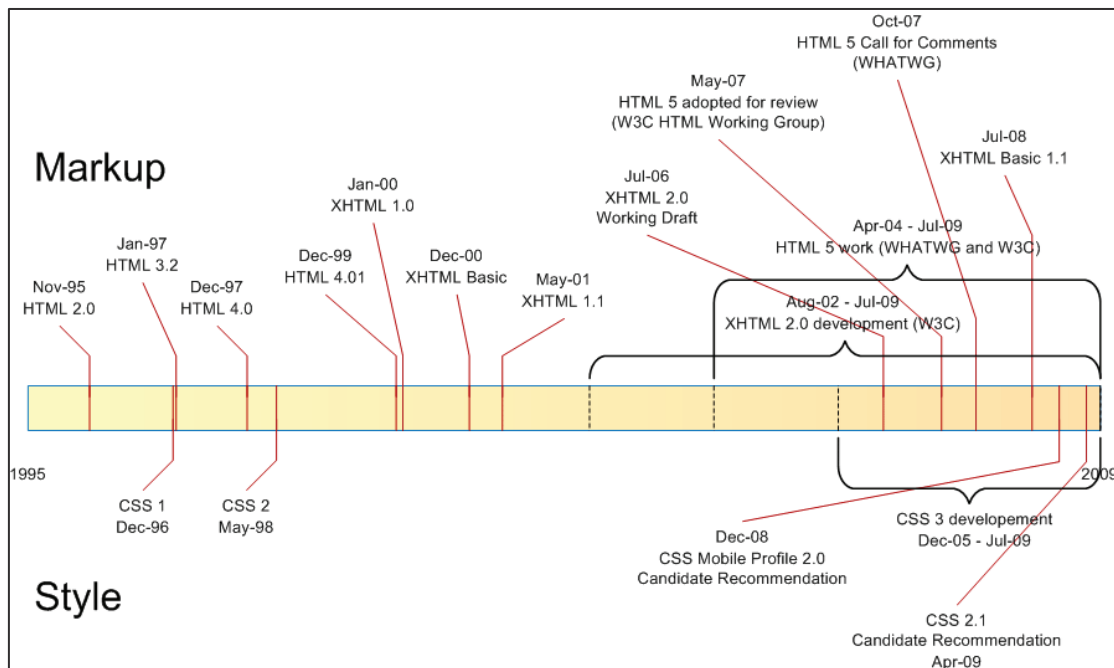
Παρ' όλα αυτά, ένας ερευνητής της IBM, ο Charles Goldfarb εμφανίζεται συχνότερα ως ο πατέρας των γλωσσών σήμανσης. Ο Goldfarb, ήρθε αντιμέτωπος με τη βασική ιδέα το 1969, όταν ανέπτυξε ένα σύστημα διαχείρισης εγγράφων για δικηγορικές

---

<sup>1</sup> Το Troff, είναι το κύριο συστατικό ενός συστήματος επεξεργασίας εγγράφων το οποίο αναπτύχθηκε από την AT&T για το λειτουργικό σύστημα Unix (Corderoy, n.d.)

<sup>2</sup> Το LaTeX είναι ένα σύστημα στοιχειοθεσίας. Περιλαμβάνει συστατικά για την δημιουργία τεχνικής και επιστημονικής τεκμηρίωσης (Anon., n.d.).

εταιρίες και βοήθησε στην ανάπτυξη της IBM GML αργότερα τον ίδιο χρόνο. Η GML πρωτοπαρουσιάστηκε το 1974.



Εικόνα 0-1 Ιστορική εξέλιξη στις κύριες γλώσσες σήμανσης

Το 1975 έγινε αρμόδιος σχεδίασης προϊόντων για την IBM στο ερευνητικό κέντρο της εταιρίας. Εκεί έπεισε τα στελέχη της IBM για την ανάπτυξη και εμπορική διάδοση της GML ως μέρος του προϊόντος «Document Composition Facility», που αφορούσε ένα σύστημα διαχείρισης εγγράφων.

Η SGML, βασίστηκε στη GML και GenCode και αναπτύχθηκε από τον Goldfarb το 1974. Ο ίδιος έγινε τελικά πρόεδρος της επιτροπής SGML, η οποία κυκλοφόρησε από το ISO, σαν ISO 8879 τον Οκτώβριο του 1986.

Ένα άλλο δημοφιλές πρότυπο σήμανσης είναι το TeX, το οποίο δημιουργήθηκε από τον Donald Knuth τη δεκαετία του 80. Το TeX πρόκειται για μια γλώσσα σήμανσης, η οποία περιγράφει ένα λεπτομερές σχέδιο του κειμένου, αλλά και των γραμματοσειρών και χρησιμοποιήθηκε κυρίως σε μαθηματικά βιβλία. Χρησιμοποιείται κυρίως στη σύγχρονη μορφή του LaTeX το οποίο αποτελεί το ηλεκτρονικό ανάλογο του TeX και χρησιμοποιείται για ακαδημαϊκούς λόγους (π.χ. εμφάνιση μαθηματικών τύπων σε κείμενα)

Η πρώτη γλώσσα που έκανε διακριτή τη διαφορά μεταξύ της δομής και παρουσίασης, ήταν η Scribe, η οποία αναπτύχθηκε από τον Brian Reid και

περιγράφηκε το 1980 στο διδακτορικό του. Η Scribe καινοτόμησε σε πολλούς τομείς, αλλά κυρίως παρουσίασε την ιδέα του διαχωρισμού των στυλ από την περιγραφή. Επηρέασε επίσης την ανάπτυξη της GML (αργότερα SGML).

Στις αρχές της δεκαετίας του 1980, παρουσιάστηκε η ιδέα, ότι η σήμανση θα έπρεπε να εστιάζει στη δομή ενός εγγράφου και να αφήνει την παρουσίαση σε μία άλλη διαδικασία. Αυτή η θεώρηση οδήγησε στην δημιουργία της Πρότυπης Γενικευμένης Γλώσσας Σήμανσης (SGML: Standard Generalized Markup Language), η οποία αποτέλεσε πρόγονο της HTML και του LaTeX.

Η SGML περιέγραψε τη σύνταξη για την συμπερίληψη της σήμανσης στα έγγραφα, καθορίζοντας ποιες ετικέτες (tags) επιτρέπονται αλλά και σε ποια θέση. Αυτό επέτρεψε στους δημιουργούς να υιοθετήσουν τις ετικέτες που ήθελαν, δημιουργώντας τις δικές τους γλώσσες σήμανσης. Υπό αυτό τον όρο, ουσιαστικά η SGML αποτέλεσε μια μετά-γλώσσα σήμανσης, από την οποία κληρονόμησαν χαρακτηριστικά οι επόμενες. Το 1986 η SGML κυκλοφόρησε σαν Διεθνές Πρότυπο, από τον Παγκόσμιο Οργανισμό Πιστοποίησης (ISO 8879).

Το 1991, η ανάγκη για μια απλή μορφή της SGML, οδήγησε τους Sir Tim Berners-Lee, να χρησιμοποιήσουν την SGML για να δημιουργήσουν την HTML. Η απλότητα αυτή, ίσως ήταν και ο σημαντικότερος παράγοντας επέκτασης της HTML, η οποία αποτελεί την πιο χρησιμοποιούμενη γλώσσα σήμανσης σήμερα.

Μία άλλη προσπάθεια πολύ αργότερα, τον Ιανουάριο του 2000 περιελάμβανε την δημιουργία μιας μετά – γλώσσας σήμανσης από το Παγκόσμιο Οργανισμό Ιστού την Εκτεταμένη Γλώσσα Σήμανσης (XML: eXtensible Marckup Language). Η XML αναπτύχθηκε από τη Κοινοπραξία του Παγκόσμιου Ιστού (W3C: World Wide Consortium), στα πλαίσια μια επιτροπής που Πρόεδρος ήταν ο Jon Bosak. Η κύρια επιδίωξη της XML, ήταν η απλούστευση της SGML. Παρέμεινε όμως μια μετά - γλώσσα όπως η SGML, παρέχοντας τη δυνατότητα στους χρήστες να δημιουργήσουν τις δικές τους ετικέτες, σύμφωνα με τις ανάγκες τους. Η XML χρησιμοποιήθηκε κυρίως επειδή διατήρησε τη συμβατότητα με την SGML και κάθε XML έγγραφο, μπορούσε να γραφεί με τέτοιο τρόπο ώστε να είναι και SGML έγγραφο.

Από τον Ιανουάριο του 2000 όλες οι προδιαγραφές της HTML βασίστηκαν στην XML, αντί για την SGML, χρησιμοποιώντας την ονομασία Εκτεταμένη Γλώσσα Σήμανσης Υπερκειμένου (XHTML: eXtended HyperText Markup Language). Οι

προδιαγραφές όριζαν ότι τα έγγραφα HTML, θα έπρεπε να είναι καλώς μορφοποιημένα με αυστηρότερο τρόπο και κανόνες (Πομπόρτσης, 2009).

Από τότε έχουν κυκλοφορήσει αρκετές εκδόσεις, τις οποίες θα δούμε αναλυτικότερα στα επόμενα κεφάλαια.

## Χαρακτηριστικά

Ένα κοινό χαρακτηριστικό πολλών γλωσσών σήμανσης, είναι η ανάμιξη του κειμένου ενός εγγράφου με οδηγίες σήμανσης που ενσωματώνονται στο ίδιο το αρχείο. Ο κώδικας που ακολουθεί αποτελεί ένα τέτοιο παράδειγμα.

```
<h1> Επικεφαλίδα </h1>  
<p>  
    Παράγραφος <i>Πλάγια Εγγραφή</i> κείμενο, κείμενο, κείμενο.  
</p>
```

Το κείμενο μεταξύ αυτών των οδηγιών είναι το πραγματικό κείμενο του εγγράφου. Οι κωδικοί **<h1>**, **<p>** είναι οδηγίες, για την μορφοποίηση και σημασιολογική σήμανση του κειμένου. Συγκεκριμένα ο κωδικός **<h1><h1>** χρησιμοποιείται για να δηλώσει ότι «αυτό είναι το πρώτο επίπεδο επικεφαλίδας», ενώ ο **<p></p>** για να δηλώσει την αρχή και το τέλος μιας παραγράφου. Η ετικέτα **<i> </i>** είναι ένα παράδειγμα σήμανσης παρουσίασης, γιατί χρησιμοποιείται για να καθορίσει ένα ιδιαίτερο χαρακτηριστικό στο κείμενο.

Οι ετικέτες αναφέρονται και σαν στοιχεία της γλώσσας και δεν ακολουθούν κάποιο κανόνα δημιουργίας, ούτε αυτός ο τρόπος περιγράφεται. Παρ' όλα αυτά οι δημιουργοί αυτών των συνόλων, φροντίζουν να υιοθετούν κατανοητά ονόματα για τα στοιχεία. Εντός του κειμένου κάθε στοιχείο θα πρέπει να εισάγεται με μια σήμανση αρχής και να κλείνει με μία σήμανση τέλους. Η σήμανση αρχής δηλώνεται με το όνομα του στοιχείου με αγκύλες ανοίγματος, ενώ η σήμανση τέλους αποτελείται από αγκύλες ανοίγματος και τον χαρακτήρα εγκοπής (slash).

Τα ιδιαίτερα χαρακτηριστικά των γλωσσών σήμανσης, ουσιαστικά αποτελούν στο μεγαλύτερο βαθμό χαρακτηριστικά της SGML. Αυτά αφορούν κυρίως:

- Το τύπο της σήμανσης.
- Τον καθορισμό του τύπου εγγράφου.
- Την ανεξαρτησία της γλώσσας από τα συστήματα εφαρμογής.

### 1.2.1. Τύποι Σήμανσης

Η SGML καθόρισε την περιγραφική σήμανση, κατά την οποία χρησιμοποιούνται κώδικες παροχής ονομάτων, για την κατηγοριοποίηση των τμημάτων του εγγράφου. Αυτοί οι κώδικες καθορίζουν για παράδειγμα, ότι η επόμενη καταγραφή είναι μια παράγραφος. Όμως στη πορεία οι απόγονοι της SGML καθόρισαν νέους τύπους ηλεκτρονικής σήμανσης όπως:

- Σήμανση παρουσίασης
- Διαρθρωτική σήμανση
- Διαδικαστική σήμανση
- Σημασιολογική σήμανση

#### 1.2.1.1. Σήμανση παρουσίασης

Παρόλο που οι προδιαγραφές της γλώσσας HTML καθορίζουν ότι η σήμανση μπορεί να χρησιμοποιηθεί για τη δημιουργία μιας ιστοσελίδας, αυτός που είναι υπεύθυνος για την επιλογή των στοιχείων της γλώσσας σήμανσης είναι ο ίδιος ο συγγραφέας.

Αν ο συγγραφέας χρησιμοποιεί λογισμικό που επιτρέπει την επεξεργασία ενός εγγράφου HTML οπτικά και όχι σε επίπεδο κώδικα, τότε είναι προτιμότερο να χρησιμοποιείται σήμανση παρουσίασης. Για παράδειγμα η εμφάνιση μιας επικεφαλίδας μπορεί να επιτευχθεί, μορφοποιώντας το κείμενο κατάλληλα (μεγαλύτερη γραμματοσειρά, έντονα γράμματα κλπ), παρά να εφαρμοστεί ένα στυλ επικεφαλίδας. Η σήμανση για μία τέτοια επικεφαλίδα θα μπορούσε να έμοιαζε:

```
<p><font size="+7" color="#000000"><b> Επικεφαλίδα </b></font></p>
```

Παρόμοια η δημιουργία μιας λίστας σε σήμανση παρουσίασης και σε σήμανση δομής θα ήταν:

Σήμανση	Κώδικας	Εμφάνιση
Παρουσίασης	<pre>&lt;p&gt; -Στοιχείο 1&lt;br/&gt; -Στοιχείο 2&lt;br/&gt; -Στοιχείο 3&lt;br/&gt; &lt;/p&gt;</pre>	<ul style="list-style-type: none"><li>- Στοιχείο 1</li><li>- Στοιχείο 2</li><li>- Στοιχείο 3</li></ul>

Δομής	<pre>&lt;ul&gt;   &lt;li&gt; Στοιχείο 1&lt;/li&gt;   &lt;li&gt; Στοιχείο 2&lt;/li&gt;   &lt;li&gt; Στοιχείο 3&lt;/li&gt; &lt;/ul&gt;</pre>	<ul style="list-style-type: none"> <li>• Στοιχείο 1</li> <li>• Στοιχείο 2</li> <li>• Στοιχείο 3</li> </ul>
-------	--	--

Η σήμανση παρουσίασης παρουσιάζει πολλά μειονεκτήματα όπως:

- Δεν είναι αποδοτική ως προς την συντήρηση του κώδικα. Για κάθε μελλοντική αλλαγή θα πρέπει να γίνεται επεξεργασία κάθε σελίδας.
- Δεν είναι φιλική ως προς τις μηχανές αναζήτησης κατά την ανίχνευση του site. Για παράδειγμα αν θέλουμε να εμφανίσουμε ένα τίτλο στην κορυφή της σελίδας ενός html εγγράφου, θα μπορούσαμε να χρησιμοποιήσουμε κάποια ετικέτα μορφοποίησης, όπως η `<b>` και να μεγαλώσουμε τη γραμματοσειρά. Με αυτή όμως τη προσέγγιση, ένας web crawler κατά την ανίχνευση της σελίδας δε θα καταλάβει τη διαφορά στο κείμενο. Αν όμως χρησιμοποιήσουμε τις διαθέσιμες ετικέτες `<H(x)>`, οι οποίες ορίζουν τίτλους ,τότε ο web crawler θα καταλάβει ποιος είναι ο τίτλος της σελίδας και σε τι αυτή αναφέρεται.
- Η υποστήριξη για τα στοιχεία που αφορούν τη σήμανση παρουσίασης, πολλές φορές εγκαταλείπεται από τις επόμενες εκδόσεις των γλωσσών σήμανσης, με αποτέλεσμα οι καινούργιοι φυλλομετρητές να μην μπορούν να εμφανίζουν αυτά τα στοιχεία σωστά.

### 1.2.1.2. Διαρθρωτική σήμανση

Η διαρθρωτική σήμανση χρησιμοποιείται για την περιγραφή της δομής ενός κειμένου. Η HTML5 χρησιμοποιεί δομικά στοιχεία όπως:

- **Header:** Κεφαλίδα της σελίδας
- **Head:** Τμήμα που περιέχει οδηγίες και μεταδεδομένα
- **Body:** Το κυρίως περιεχόμενο.
- **Footer:** Υποσέλιδο

Κάθε μία ενότητα από τις παραπάνω μπορεί να χρησιμοποιήσει δικές της εσωτερικές δομές οι οποίες χαρακτηρίζονται από στοιχεία όπως: λίστες, πίνακες, τμήματα div κλπ.

### 1.2.1.3. Διαδικαστική σήμανση

Η σήμανση είναι ενσωματωμένη στο κείμενο και παρέχει οδηγίες για τα προγράμματα που πρόκειται να το επεξεργαστούν. Τέτοιες παραδείγματα, είναι η troff, LaTeX και PostScript. Ο επεξεργαστής σαρώνει το κείμενο από την αρχή έως το τέλος, ακολουθώντας τις οδηγίες που βρίσκει. Τα γενικά χαρακτηριστικά της διαδικαστικής σήμανσης είναι:

- Καθορίζει τον τρόπο που θα επεξεργαστεί το κείμενο.
- Παρέχει οδηγίες για τον τρόπο επεξεργασίας
- Ασχολείται με τη μορφοποίηση και παρουσίαση.
- Είναι αποτελεσματική καθώς δε χρειάζεται περεταίρω επεξεργασία.

### 1.2.1.4. Σημασιολογική σήμανση

Δεν υπάρχει επίσημη προδιαγραφή που να χαρακτηρίζεται ως σημασιολογική σήμανση. Τα χαρακτηριστικά όμως της HTML «πιέζουν» προς αυτή τη κατεύθυνση. Η σημασιολογική HTML αναφέρεται σε μια πρακτική, για τη δημιουργία εγγράφων, τα οποία θα περιέχουν μόνο την προτεινόμενη από το συγγραφέα έννοια, χωρίς καμία αναφορά για το πώς αυτή παρουσιάζεται (Πομπόρτσης, 2009).

Χρησιμοποιείται για τη σήμανση μερών του εγγράφου, με ετικέτες οι οποίες δηλώνουν την σημασία του κειμένου. Υπάρχει μια λεπτή γραμμή ανάμεσα στη σημασιολογική και στη διαθρωτική σήμανση. Για παράδειγμα οι επικεφαλίδες H(x) από τη μεριά του συγγραφέα είναι διαρθρωτική σήμανση αφού αποτελεί το τίτλο της σελίδας ο οποίος μπαίνει στην αρχή του κειμένου, για μια μηχανή αναζήτησης όμως έχει εννοιολογική σημασία γιατί αποτελεί το σήμα κατατεθέν του κειμένου. Τα χαρακτηριστικά της είναι:

- Αναγνώριση των λογικών στοιχείων ενός κειμένου.
- Δεν καθορίζει τις διαδικασίες που θα εφαρμοστούν στο κείμενο και για αυτό τον λόγο απαιτείται άλλη διαδικασία για αυτό. Για παράδειγμα περιέχει την πληροφορία για το μέγεθος του κειμένου, αλλά δεν επεξεργάζεται την αλλαγή.

### 1.2.2. Ανεξαρτησία δεδομένων

Η ανεξαρτησία δεδομένων, διασφαλίζει ότι τα έγγραφα που θα περιγραφούν με μία γλώσσα σήμανσης, θα μπορούν να μεταφέρονται από ένα περιβάλλον υλικού και λογισμικού σε ένα άλλο, χωρίς απώλεια πληροφορίας (Πομπόρτσας, 2009).

### 1.2.3. Τύπος Εγγράφου

Ένα άλλο χαρακτηριστικό είναι ο ορισμός τύπου εγγράφου (DTD: Document Type Definition). Ουσιαστικά ο τύπος εγγράφου χρησιμοποιείται για την κατάταξη των εγγράφων σε τύπους. Με αυτό τον τρόπο το πρόγραμμα που θα «διαβάσει» το έγγραφο, συνήθως ο συντακτικός αναλυτής, θα γνωρίζει τις διαδικασίες που θα πρέπει να ακολουθήσει για την επεξεργασία του.

## Χρήσεις

Αν και η ιδέα της γλώσσας σήμανσης ξεκίνησε με τα έγγραφα κειμένου, υπάρχει μια αυξανόμενη χρήση και σε άλλους τομείς που αφορούν την παρουσίαση διαφόρων τύπων πληροφορία όπως :

- Μουσικές λίστες
- Διανυσματικά γραφικά
- Διαδικτυακές υπηρεσίες
- Διεπιφάνειες χρήστη

Οι περισσότερες από αυτές τις εφαρμογές χρησιμοποιούν την XML επειδή όπως είδαμε είναι δυνατή η δημιουργία ετικετών ανάλογα με τις ανάγκες χρήσης. Ενδεικτικά μερικές από αυτές τις γλώσσες σήμανσης είναι:

- **BNML (Business Narrative Markup Language):** Χρησιμοποιείται για την σήμανση ηλεκτρονικών συμβολαίων.
- **BPML (Business Process Modeling Language):** Αποτελεί μια περιγραφική γλώσσα, για την παρουσίαση των δραστηριοτήτων μιας επιχείρησης.
- **CFML (ColdFusion Markup Language):** Είναι μια γλώσσα σεναρίων, για ανάπτυξη εφαρμογών web που εκτελείτε στην εικονική μηχανή της Java, στο πλαίσιο .NET, αλλά και στην μηχανή εφαρμογών της Google.
- **MathML (Mathematical Markup Language):** Χρησιμοποιείται για την περιγραφή μαθηματικών εξισώσεων.
- **CML (Chemical Markup Language):** Περιγράφει μοριακές δομές, αντιδράσεις, ανάλυση φάσματος, χημική κρυσταλλογραφία κλπ.



- **ABC:** Γλώσσα σήμανσης για την περιγραφή των μουσικών κειμένων (παρτιτούρες) σαν απλό κείμενο.
- **MediaWiki:** Αποτελεί «ελαφριά» έκδοση γλώσσας σήμανσης και χρησιμοποιείται για την δημιουργία σελίδων που διατηρούν γνωσιακές βάσεις.

Εκτός από τις παραπάνω υπάρχει ένας μεγάλος αριθμός γλωσσών σήμανσης<sup>3</sup>. Αυτές γενικότερα κατηγοριοποιούνται ανάλογα με την χρήση τους σε:

- XML
- Γενικής Χρήσης
- Εγγράφων
- «Ελαφριάς» χρήσης
- Διεπιφάνειας
- Γραφικών
- Υπηρεσιών WEB

---

<sup>3</sup> Μια πλήρης λίστα μπορεί να βρεθεί στην ηλεκτρονική διεύθυνση της Wikipedia: [http://en.wikipedia.org/wiki/List\\_of\\_markup\\_languages](http://en.wikipedia.org/wiki/List_of_markup_languages)

## HTML 5

### Εισαγωγή

Η HTML5 είναι το νέο πρότυπο για την HTML. Σχεδόν όλοι οι νέοι φυλλομετρητές υποστηρίζουν τα περισσότερα από τα καινούργια στοιχεία που εισάγει. Η HTML5 αναπτύσσεται με την υποστήριξη του Παγκόσμιου Συνεταιρισμού Διαδικτύου (W3C: World Wide Web Consortium) και της ομάδας Εφαρμοσμένης Τεχνολογίας Υπερκειμένου Διαδικτύου (WHATWG: Web Hypertext Application Technology Working Group).

Μερικά από τα καινούργια χαρακτηριστικά της HTML5 περιλαμβάνουν:

- Ανανεωμένα και πρόσθετα στοιχεία ετικετών.
- Μείωση της ανάγκης για εξωτερικά πρόσθετα (plugins).
- Καλύτερη διαχείριση σφαλμάτων.
- Προσανατολισμός στην σήμανση, ελαχιστοποιώντας την ανάγκη για συγγραφή σεναρίων (π.χ javascript).
- Ανεξαρτησία από τη συσκευή εφαρμογής (π.χ ένα html5 έγγραφο, θα μπορεί να φορτωθεί το ίδιο σε έναν υπολογιστή, αλλά και σε μια κινητή συσκευή).

### Περιγραφή – Εξέλιξη της HTML

Η γλώσσα σήμανσης υπερκειμένου HTML, εκδόθηκε γύρω στο 1990. Την περίοδο 1993, 1994 υπήρχε μόνο ένας φυλλομετρητής ο NCSA Mosaic και οι εξυπηρετητές WEB, ήταν ελάχιστοι. Από τότε ο όγκος του διαδικτύου έχει αυξηθεί εκθετικά, ιδιαίτερα με την έλευση των ευρυζωνικών συνδέσεων.

Η πρώτη έκδοση ήταν η 1.0 και αναπτύχθηκε από τους Tim Berners-Lee. Κατά τη διάρκεια της εργασίας του στο CERN, δούλευε σε πολλούς υπολογιστές και η εύρεση πληροφορίας σε όλους αυτούς ήταν μια χρονοβόρα διαδικασία. Σκέφτηκε λοιπόν, ότι θα πρέπει να υπάρχει ένας τρόπος, να μπορεί να μεταφέρεται αυτή η πληροφορία από τον έναν υπολογιστή στον άλλο. Αυτή η θεώρηση ενός συστήματος μεταφοράς της πληροφορίας, μέσω τεχνολογιών δικτύου και των αντίστοιχων πρωτοκόλλων, ήταν ουσιαστικά η αρχή της θεμελιώδους γλώσσας του διαδικτύου. Η αρχική έκδοση δεν είχε τις ίδιες δυνατότητες με αυτή που χρησιμοποιούμε σήμερα. Βασιζόταν βέβαια

στην ίδια λογική, αλλά ήταν ένα μικρό υποσύνολο, αν σκεφτούμε ότι αποτελούταν από 22 ετικέτες μόνο.

Η HTML 2.0 εμφανίστηκε το 1995. Δεν ήταν ιδιαίτερα διαφορετική από την 1.0, αλλά ήρθε να προσφέρει καινούργιες δυνατότητες.

Σε αυτό το σημείο το διαδίκτυο άρχισε να γίνεται ολοένα και πιο δημοφιλές και πολλοί θέλησαν να δημιουργήσουν τις δικές τους σελίδες. Για αυτό το λόγο χρειάστηκαν περισσότερες ετικέτες και χαρακτηριστικά για τον εμπλουτισμό αυτών των σελίδων. Παράλληλα ο φυλλομετρητής Netscape είχε κατακτήσει τα πρωτεία, και δημιουργήθηκαν ετικέτες που μπορούσαν να δουλέψουν μόνο σε αυτόν. Για όλους αυτούς τους λόγους παρουσιάστηκε ένα προσχέδιο της HTML 3.0. Όμως οι περισσότεροι φυλλομετρητές, δεν υιοθέτησαν όλα τα νέα χαρακτηριστικά για αυτό το λόγο η εξέλιξη του προσχεδίου εγκαταλείφθηκε.

Εφόσον εγκαταλείφθηκε η έκδοση 3.0 και οι ανάγκες των κατασκευαστών σελίδων μεγάλωνε, η ανάγκη για ένα πρότυπο ήταν επιτακτική. Το 1994 το W3C, σχηματίστηκε για την προτυποποίηση της HTML. Έτσι λοιπόν ξεκίνησε η έκδοση 3.2, η οποία αποτέλεσε τη νέα εξέλιξη της HTML. Η έκδοση 3.2 τελικά έγινε επίσημο πρότυπο τον Ιανουάριο του 1997 (Vossen & Hagemann, 2007).

Οι μεγαλύτερες όμως αλλαγές ήρθαν με την έκδοση 4.0, η οποία δημοσιεύθηκε ως σύσταση από το W3C. Ήταν σαφώς διαφορετική από την προηγούμενη και περιελάμβανε αρκετές ετικέτες και ιδιότητες που αφορούσαν το στυλ του κειμένου, αλλά και των συνδέσμων. Όμως αυτό ήταν ταυτόχρονα αρνητικό, γιατί οι κατασκευαστές σπαταλούσαν πολύ χρόνο για να ρυθμίζουν χειροκίνητα αυτές τις ιδιότητες σε όλες τις σελίδες ενός ιστοτόπου. Αυτό που κατανοήθηκε, ήταν ότι το στυλ και οι ιδιότητες, θα έπρεπε να διαχωριστούν και η HTML να προσανατολιστεί στην λογική οργάνωση του εγγράφου. Έτσι λοιπόν αυτή η ανάγκη γέννησε μια ξεχωριστή γλώσσα που ονομάστηκε Αλληλουχία Φύλλων Στυλ (CSS: Cascade Styling Sheets). Η HTML 4.0 όμως παρείχε και άλλα χαρακτηριστικά όπως σενάρια, παράθυρα, ενσωματωμένα αντικείμενα, περίπλοκους πίνακες, σύνθετες φόρμες και βελτιωμένα χαρακτηριστικά για άτομα με αναπηρίες. Τελικά έγινε επίσημο πρότυπο τον Απρίλιο του 1998.

Στη συνέχεια ακολούθησε η έκδοση 4.01, που ουσιαστικά ήταν μια αναθεώρηση της 4.0. Συμπεριλάμβανε την προσθήκη μερικών χαρακτηριστικών<sup>4</sup> όπως για παράδειγμα την προσθήκη της ιδιότητας **name** για την ετικέτα **<form>**. Ταυτόχρονα διορθώθηκαν κάποια λάθη της προηγούμενης έκδοσης.

Η HTML5 δεν αποτελεί μόνο την τελευταία έκδοση της HTML. Αποτελεί ένα σύνολο σχετικών τεχνολογιών για τη δημιουργία μοντέρνων σελίδων με πλούσιο περιεχόμενο.

Ο πυρήνας της HTML5, καθορίζει τα συστατικά που χρησιμοποιούνται για την σήμανση ενός κειμένου, με προσανατολισμό την σπουδαιότητα. Τα αρχεία CSS, από την άλλη επιτρέπουν τον έλεγχο της εμφάνισης του κειμένου σήμανσης, όπως αυτό δηλαδή θα παρουσιάζεται στο χρήστη. Η JavaScript, επιτρέπει τον χειρισμό των περιεχομένων σε ένα HTML κείμενο, ανταποκρινόμενο στην αλληλεπίδραση του χρήστη και εκμεταλλεύεται κάποια νέα χαρακτηριστικά της HTML5.

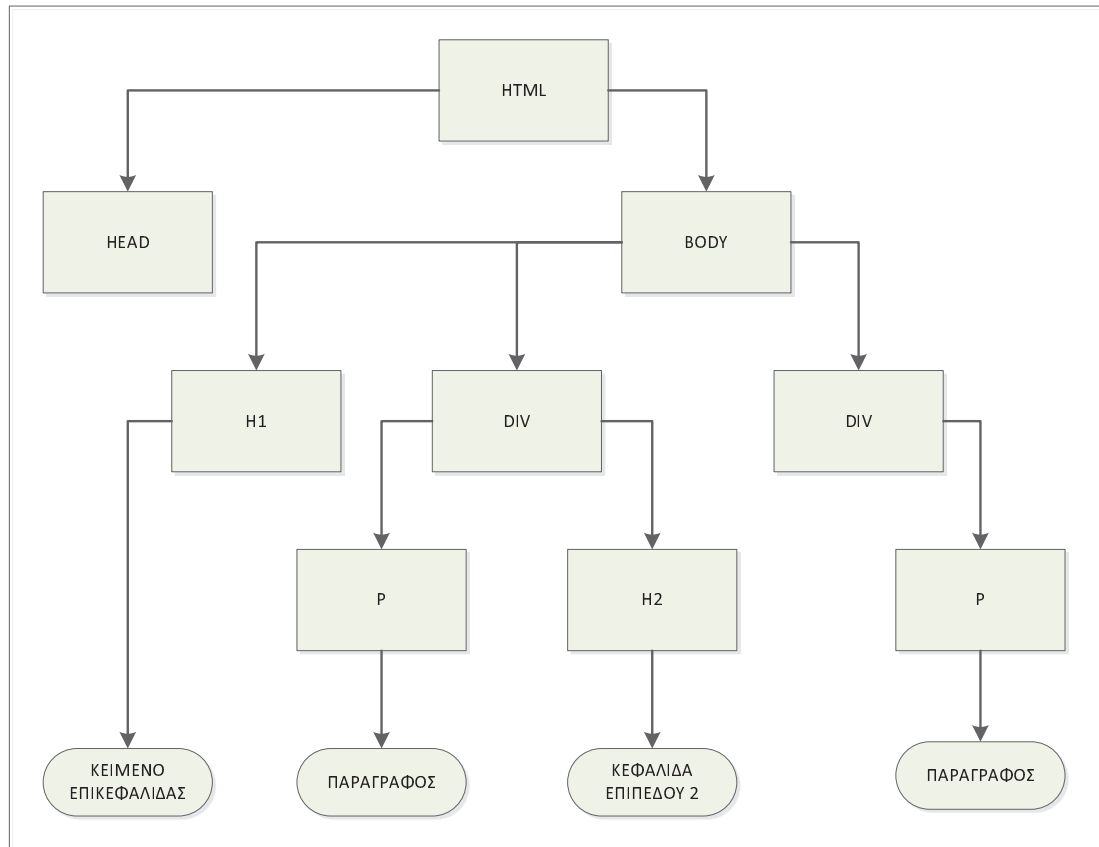
Επειδή η διαδικασία προτυποποίησης είναι μια χρονοβόρα διαδικασία, οι τεχνολογίες που αναφέρονται στο νέο πρότυπο, καθορίζονται μέσω ενός μεγάλου αριθμού μικρότερων σε μέγεθος προτύπων. Το θετικό από αυτή την προσέγγιση είναι ότι τα επί μέρους πρότυπα αναπτύσσονται ανεξάρτητα και πιο γρήγορα. Το αρνητικό όμως είναι ότι δεν είναι εύκολη η ταυτόχρονη παρακολούθηση όλων αυτών των προτύπων και πως σχετίζονται και επηρεάζονται το ένα από το άλλο (Freeman, 2011).

## Δομή της HTML5

Το μοντέλο αντικειμένων εγγράφου (DOM: Document Object Model), αναπαριστά όλα τα αντικείμενα σε μία HTML σελίδα. Είναι ανεξάρτητο από τη γλώσσα και τη πλατφόρμα και επιτρέπει την ενημέρωση του περιεχομένου και στυλ της σελίδας όταν αυτή «ανακτηθεί» από τον φυλλομετρητή. Το DOM είναι προσβάσιμο μέσω JavaScript. Για την ανάκτηση μιας HTML σελίδας, οι περισσότεροι φυλλομετρητές χρησιμοποιούν ένα εσωτερικό μοντέλο παρόμοιο με το DOM. Οι κόμβοι κάθε εγγράφου οργανώνονται σε δένδροειδή μορφή, το οποίο καλείται δέντρο DOM. Ο κορυφαίος κόμβος ονομάζεται αντικείμενο «Document». Κάθε έγγραφο HTML ή XML αναπαρίσταται από το αντικείμενο «Document» (McLaughlin, 2011).

---

<sup>4</sup> Στην διεύθυνση <http://www.w3.org/TR/REC-html40/appendix/changes.html>, υπάρχει αναλυτική λίστα με τις αναθεωρημένες αλλαγές και προσθήκες.



Εικόνα 0-1 Δέντρο DOM

Η διεύθυνση του εγγράφου είναι σε απόλυτη URL. Όταν το έγγραφο δημιουργείται μέσω κώδικα, π.χ ενός σεναρίου η διεύθυνση του είναι η ίδια με αυτή του σεναρίου.

### 2.2.1. Σύνταξη

Κάθε έγγραφο αποτελείται από τα στοιχεία που φαίνονται στον κώδικα που ακολουθεί.

```

<!DOCTYPEhtml>
<html xmlns="http://www.w3.org/2002/06/xhtml12" xml:lang="gr">
<head>
  <title>Τίτλος σελίδας</title>
</head>
<body id="theBody">
  <p>Μια παράγραφος</p>
</body>
  
```

Η πρώτη γραμμή σε κάθε έγγραφο είναι ένα ειδικός κώδικας, που λέγεται doctype. Ο ρόλος του είναι να χαρακτηρίζει ότι το έγγραφο που ακολουθεί είναι ένα έγγραφο HTML5. Σε σχέση με τις παλαιότερες εκδόσεις η περιγραφή του έχει απλοποιηθεί αρκετά. Μάλιστα ο μόνος λόγος που υπάρχει στην HTML5 είναι για λόγους συμβατότητας ως προς τους παλαιότερους φυλλομετρητές.

Το **html** είναι το αρχικό στοιχείο (root element) για όλες τις σελίδες. Με αυτό ορίζονται διάφορες ιδιότητες του εγγράφου, όπως η γλώσσα ή το είδος της σελίδας καθώς και άλλα χαρακτηριστικά. Το **head** ορίζει την επικεφαλίδα και μπορεί να περιέχει ένα τίτλο με την ετικέτα **title**, όπως φαίνεται στο κώδικα, ή μεταδεδομένα και λοιπές πληροφορίες. Το **body** περιλαμβάνει το περιεχόμενο της σελίδας και μπορεί να περιέχει κείμενο, εικόνες, γραφικά και μια σειρά ετικετών ή οδηγιών.

Τα στοιχεία που αποτελούν τον «κορμό» δηλαδή τα html, head και body είναι προαιρετικά και αυτό αποτελεί μια «χαλαρή συμπεριφορά» της HTML5. Η HTML5 επίσης αγνοεί τα κεφαλαία/μικρά συνεπώς οι ετικέτες `<p></p>` και `<P></P>` είναι ακριβώς οι ίδιες.

Οι επικεφαλίδες που δεν έχουν εμφωλευμένο περιεχόμενο, μπορούν να παραλείψουν την κάθετο ή να μπει αυτή και στις δύο άκρες. Συνεπώς οι ετικέτες `<br>`, `<br/>`, `</br>` είναι ταυτόσημες και έχουν σαν αποτέλεσμα την αλλαγή γραμμής.

Μία άλλη αλλαγή είναι ότι οι ιδιότητες των ετικετών δεν χρειάζονται πια εισαγωγικά, εκτός αν δηλώνεται κάποιος ειδικός χαρακτήρας. Για παράδειγμα η ετικέτα: `<img alt="Μία Εικόνα" src=image.jpg>`, χρησιμοποιεί εισαγωγικά στην ιδιότητα alt επειδή υπάρχει ο χαρακτήρας διάστημα, ενώ η ιδιότητα src δεν χρησιμοποιεί.

### 2.2.2. Μεταδεδομένα

Ενώ το στοιχείο **html** αποτελεί το ριζικό στοιχείο, τα επόμενα συστατικά της επικεφαλίδας αποτελούν τα μεταδεδομένα.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Τίτλος εγγράφου</title>
    <base href="http://www.example.com/index.html">
  </head>
  <body>
    <p>Visit the <a href="otherpage.html">otherpage</a>.</p>
  </body>
</html>
```

Τα μεταδεδομένα είναι:

- **head:** Η επικεφαλίδα
- **title:** Ο τίτλος της σελίδας
- **base:** Ορίζει την διεύθυνση του HTML εγγράφου έτσι ώστε να χρησιμοποιηθεί σαν απόλυτη αναφορά και οι υπόλοιπες να σχετιστούν αναφορικά με αυτή. Στο παράδειγμα του κώδικα, η αναφορά στην «otherpage.html», θα γίνει αφού προστεθεί η διεύθυνση βάσης, δηλαδή `http://www.example.com/otherpage.html`.
- **link:** Χρησιμοποιείται για να επιτρέψει την σύνδεση άλλων πόρων στο τρέχων έγγραφο. Για παράδειγμα με τη ετικέτα **link** μπορούμε να αναφερθούμε σε ένα CSS αρχείο, που βρίσκεται σε άλλο κατάλογο.
- **meta:** Ορίζει τις πληροφορίες σχετικά με το κείμενο κυρίως για της μηχανές αναζήτησης, όπως λέξεις κλειδιά περιγραφή κλπ. Το κείμενο αυτό δεν είναι ορατό στο τελικό χρήστη. Χρησιμοποιείται κυρίως για τεχνικές βελτιστοποίησης για μηχανές αναζήτησης (SEO: Search Engine Optimization).

### 2.2.3. Σενάρια - Αρχεία

Τα σενάρια χρησιμοποιούνται από τους κατασκευαστές για τη προσθήκη αλληλεπίδρασης στις σελίδες τους. Η ετικέτα **script** καλείται με το τύπο του σεναρίου που προστίθεται δυναμικά στη σελίδα. Μπορεί να είναι για παράδειγμα τύπου javascript. Η προσθήκη μπορεί να γίνει είτε με σύνδεσμο σε εξωτερικό αρχείο, είτε εσωτερικά του αρχείου που καλείται η ετικέτα **script**. Επίσης μια άλλη δυνατότητα είναι η εισαγωγή ομάδας δεδομένων. Σε αυτή την περίπτωση η ετικέτα είναι τύπου «text».

```
<script src="game-engine.js"></script>
<script type="text/x-game-map">
  .....U.....e
  o.....A.....e
  ....A...AAA.....e
  .A..AAA..AAAAA...e
</script>
```

### 2.2.3.1. Javascript

Η Javascript είναι μια γλώσσα που χρησιμοποιείται για την εκτέλεση κώδικα σε ιστοσελίδες. Μαζί με τη χρήση της τεχνολογίας AJAX, επιτρέπει στους δημιουργούς την κατασκευή πιο επαγγελματικών σελίδων με πλούσιο περιεχόμενο. Χαρακτηριστικό της Javascript, είναι ότι η επεξεργασία του κώδικα δεν πραγματοποιείται στον διακομιστή, αλλά τοπικά από το πρόγραμμα περιήγησης (Meyer, 2011).

Στο προσχέδιο της HTML5 δεν γίνεται κάποια σχετική αναφορά σχετικά με τη Javascript, όμως σύμφωνα με το W3C, σχεδιάζεται η αντικατάσταση του JavaScript APIs σε σχέση με την HTML4, XML1 και το επίπεδο 2 DOM εγγράφων. Ένα τυπικό σενάριο JavaScript φαίνεται στο κώδικα που ακολουθεί:

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript</h1>

<p id="demo"> Η javascript μπορεί να αντιδρά σε γεγονότα</p>

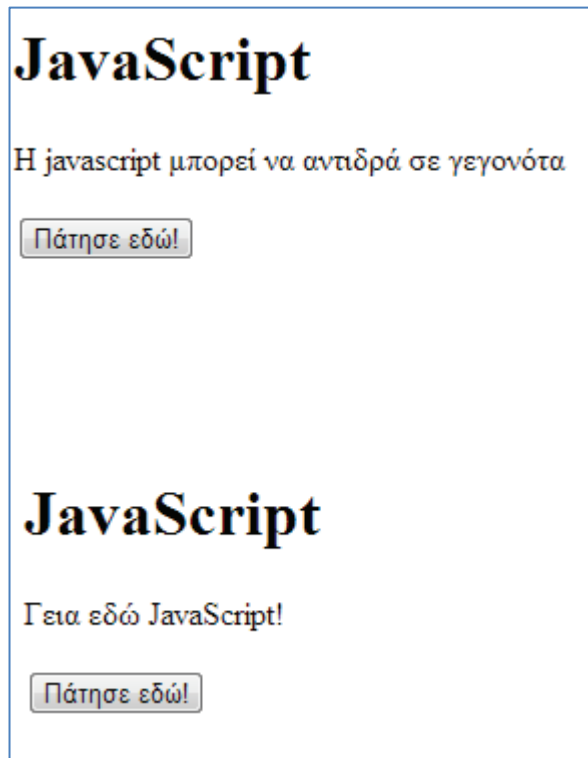
<script>
function myFunction(){
  document.getElementById("demo").innerHTML="Γεια εδώ JavaScript!";
}
</script>

<button type="button" onclick="myFunction()">Πάτησε εδώ!</button>

</body>
</html>
```

Το σενάριο εισάγεται με την ετικέτα `<script>`, ενώ θα μπορούσε να βρίσκεται σε ξεχωριστό αρχείο και να καλείται με σύνδεσμο. Αποτελείται από μία συνάρτηση η οποία καλείται όταν συμβεί κάποιο γεγονός. Στη περίπτωση όταν πατηθεί το κουμπί με τη περιγραφή «Πάτησε εδώ», τότε εμφανίζεται αντίστοιχα το μήνυμα «Γεια εδώ JavaScript».





Εικόνα 0-2 Javascript

### 2.2.3.2. Αρχεία στυλ CSS

Τα αρχεία αυτά χρησιμοποιούν τη γλώσσα CSS για να περιγράψουν την όψη και τη μορφοποίηση ενός εγγράφου, που έχει γραφτεί σε μία γλώσσα σήμανσης. Σχεδιάστηκε αρχικά για να διαχωρίσει το περιεχόμενο από την παρουσίαση και περιλαμβάνει στοιχεία για το σχέδιο του εγγράφου και την μορφοποίηση του κειμένου. Η τελευταία έκδοση είναι η CSS 3 και αποτελείται από ξεχωριστές ενότητες. Κάθε ενότητα προσθέτει νέες δυνατότητες ή επεκτείνει δυνατότητες όπως αυτές ορίστηκαν στην CSS 2. Συνοπτικά αυτές οι ενότητες περιλαμβάνουν:

- Επιλογείς (selectors)
- Μοντέλο κουτιού (box model)
- Παρασκήνια και περιθώρια
- Εφέ κειμένου
- 2D/3D μεταμορφώσεις
- Κίνηση
- Πολλαπλές στήλες
- Διεπιφάνεια χρήστη

Τα φύλλα στυλ, μπορούν να ενσωματωθούν σε ένα έγγραφο html, αλλά μπορούν να αποτελούν και ξεχωριστό αρχείο. Ο κώδικας που ακολουθεί αποτελεί ένα παράδειγμα εφαρμογής στυλ ενσωματωμένο στο έγγραφο HTML.

```
<!DOCTYPE html>
<html>
<head>
<style>
  h1{text-shadow: 5px 5px 5px #FF0000;}
</style>
</head>
<body>

<h1>Εφέ σκιάς στο κείμενο!</h1>

</body>
</html>
```

Η γραμμή `h1{text-shadow: 5px 5px 5px #FF0000;}` ορίζει το στυλ επικεφαλίδας h1 και προσθέτει σκιά κόκκινου χρώματος (#FF0000).



Εικόνα 0-3 Εφέ σκιάς

#### 2.2.4. Στοιχεία HTML5

Τα στοιχεία είναι οι ετικέτες που αφορούν την σήμανση ενός εγγράφου HTML5. Μπορούμε γενικότερα να τις κατατάξουμε σε δύο κατηγορίες. Ανάλογα με την εργασία που επιτελούν μπορούμε να τις χωρίσουμε σε διάφορες κατηγορίες, όπως κειμένου, φορμών, πινάκων κλπ. Οι περισσότερες ετικέτες προέρχονται από την HTML 4.01 και έχουν τον ίδιο ακριβώς σκοπό, ενώ κάποιες έχουν αποκτήσει και άλλες ιδιότητες.

Τα νέα στοιχεία της HTML5 παρουσιάζονται στον πίνακα που ακολουθεί, ανά κατηγορία χρήσης.

Πίνακας 0-1 Νέα στοιχεία της HTML 5

Κατηγορία	Στοιχεία
Στοιχεία για τη δόμηση της σελίδας	<article>, <aside>, <figcaption>, <figure>, <footer>, <header>, <hgroup>, <nav>, <section>, <details>, <summary>
Στοιχεία για το κείμενο	<mark>, <time>, <wbr>
Φόρμες και διαδραστικότητα	<datalist>, <keygen>, <meter>, <progress>, <command>, <menu>, <output>
Πολυμέσα και πρόσθετα	<audio>, <video>, <source>, <embed>
Canvas	<canvas>
Γλωσσική υποστήριξη	<bdo>, <rp>, <rt>, <ruby>

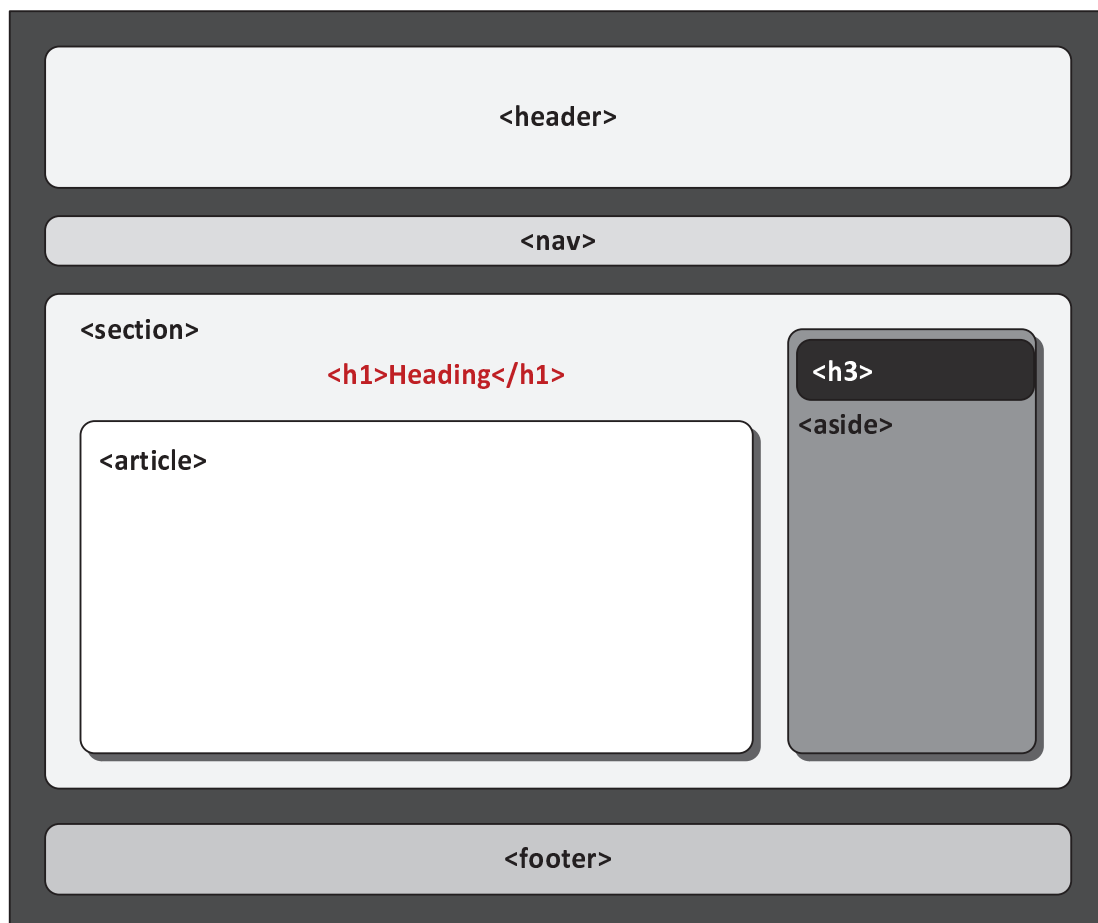
### 2.2.5. Ενότητες

Οι ενότητες έχουν να κάνουν με το σχέδιο που θα έχει η σελίδα. Αφορούν δηλαδή το επίπεδο της παρουσίασης, αλλά και της εσωτερικής δομής. Οι αντίστοιχες ενότητες χαρακτηρίζονται από τις ετικέτες:

- **body**: Αυτή η ενότητα μπορεί να ανταποκρίνεται σε διάφορα γεγονότα που συμβαίνουν και με τους αντίστοιχους χειριστές να δημιουργούνται οι αντίστοιχες ενέργειες.
- **article**: Αναπαριστά μία σύνθεση σε ένα έγγραφο, η οποία μπορεί να είναι ένα άρθρο, ένα σχόλιο, ένα νήμα ή κάποιο άλλο ανεξάρτητο στοιχείο του περιεχομένου.
- **section**: Ορίζει μία γενική ενότητα στο κείμενο. Χρησιμοποιείται κυρίως για την θεματική κατηγοριοποίηση του περιεχομένου.
- **nav**: Χρησιμοποιείται για τον ορισμό τμημάτων που έχουν εξωτερικούς ή εσωτερικούς συνδέσμους δηλαδή για πλοήγηση.
- **aside**: Είναι καινούργια ετικέτα και χρησιμοποιείται για την παράθεση κειμένου σχετικού με το κυρίως κείμενο, και βρίσκεται συνήθως σε πλευρική θέση.

- **hgroup**: Αποτελεί το συστατικό για των ορισμό των επικεφαλίδων μιας ενότητας. Η τάξη της επικεφαλίδας οριοθετείται μεταξύ h1-h6, με h1 το υψηλότερο επίπεδο και h6 το χαμηλότερο.
- **header**: Καθορίζει την επικεφαλίδα του εγγράφου.
- **footer**: Καθορίζει το υποσέλιδο του εγγράφου.
- **address**: Αποτελεί τα στοιχεία επικοινωνίας ανά ενότητα. Αν αυτό περιέχεται στο body τότε αντιστοιχεί για όλο το έγγραφο. Μπορεί όμως να οριστεί μέσα σε ένα section, οπότε ισχύει μόνο για αυτό.

Η εικόνα που ακολουθεί δείχνει ένα τυπικό σχέδιο, το οποίο έχει δημιουργηθεί μέσω των αντίστοιχων ετικετών, που αφορούν τις ενότητες.



Εικόνα 0-4 HTML 5 - Σχέδιο εγγράφου

### 2.2.6. Ομαδοποίηση περιεχομένου

Σε αυτή την κατηγορία ανήκουν οι ετικέτες που αφορούν την οργάνωση τη παρουσίαση και τη διάταξη του κειμένου. Μπορούμε γενικότερα να τις κατατάξουμε

σε δύο κατηγορίες. Στις ετικέτες που αφορούν γενικότερα τον τρόπο παρουσίασης του κειμένου, σε ετικέτες που αφορούν τον τρόπο οργάνωσης και σε ετικέτες που αφορούν λίστες.

### 2.2.6.1. Οργάνωση κειμένου

Οι αντίστοιχες ετικέτες αφορούν τον τρόπο που παρουσιάζεται το κείμενο στο τελικό χρήστη και είναι:

- **p**: Αναπαριστά μια παράγραφο κειμένου.
- **hr**: Χωρίζει θεματικά δύο ή περισσότερες παραγράφους.
- **pre**: Χρησιμοποιείται για την εμφάνιση κειμένου με ασυνήθιστη εμφάνιση, όπως για παράδειγμα κείμενο κώδικα.
- **blockquote**: Εμφανίζει μια ενότητα μέσα εισαγωγικά. Χρησιμοποιείται κυρίως για να καθορίσει ότι ένα τμήμα που αναφέρεται από άλλη πηγή.

Ο κώδικας που ακολουθεί δείχνει ένα παράδειγμα χρήσης των στοιχείων. Στο παράδειγμα επίσης φαίνεται ότι η χρήση του `blockquote` έχει αλλάξει σε σχέση με την HTML 4.01 που η χρήση του ήταν για την παράθεση ενός αποσπάσματος.

```
<body>
```

```
<p>
```

Αυτή είναι μια παράγραφος η οποία έχει δύο θεματικές ενότητες.

Η πρώτη θεματική ενότητα αφορά τη γλώσσα προγραμματισμού c.

Ο κώδικας που χρησιμοποιείται για την εκτύπωση του "Hello World" σε c είναι:

```
<pre>
```

```
printf("Hello World");
```

```
</pre>
```

```
<hr/>
```

Η δεύτερη θεματική ενότητα αφορά την γλώσσα προγραμματισμού c++. Ο αντίστοιχος κώδικας σε c++ είναι:

```
<pre>
```

```
std::cout << "Hello World";
```

```
</pre>
```

```
</p>
```

```
<blockquote>
```

Σύμφωνα με τον ιστότοπο <http://www.java2s.com/Code/C/CatalogC.htm>, η printf μπορεί να χρησιμοποιηθεί και σε ένα πρόγραμμα c++!

```
</blockquote>
```

Το παράδειγμα έχει σαν έξοδο την εικόνα που ακολουθεί:

Αυτή είναι μια παράγραφος η οποία έχει δύο θεματικές ενότητες. Η πρώτη θεματική ενότητα αφορά τη γλώσσα προγραμματισμού c. Ο κώδικας που χρησιμοποιείται για την εκτύπωση του "Hello World" σε c είναι:

```
printf ("Hello World");
```

Η δεύτερη θεματική ενότητα αφορά την γλώσσα προγραμματισμού c++. Ο αντίστοιχος κώδικας σε c++ είναι:

```
std::cout << "Hello World";
```

Σύμφωνα με τον ιστότοπο <http://www.java2s.com/Code/C/CatalogC.htm>, η printf μπορεί να χρησιμοποιηθεί και σε ένα πρόγραμμα c++!

Εικόνα 0-5 Χρήση στοιχείων κειμένου

### 2.2.6.2. Λίστες

Οι λίστες περιέχουν ένα ή περισσότερα στοιχεία. Υπάρχουν οι αριθμημένες λίστες που δηλώνονται με την ετικέτα **ol** και οι μη αριθμημένες λίστες που δηλώνονται με την ετικέτα **ul**. Αντίστοιχα τα στοιχεία των λιστών περιέχονται στην ετικέτα **li**. Το παράδειγμα που ακολουθεί

```
<ol>
  <li>Στοιχείο 1</li>
  <li>Στοιχείο 2</li>
  <li>Στοιχείο 3</li>
  <li>Στοιχείο 4</li>
</ol>
```

έχει σαν αποτέλεσμα:

1. Στοιχείο 1
2. Στοιχείο 2
3. Στοιχείο 3
4. Στοιχείο 4

Εικόνα 0-6 Αριθμημένες λίστες

Εκτός από τις παραπάνω λίστες υπάρχουν και οι περιγραφικές λίστες. Αυτές ορίζονται με την ετικέτα **dl**. Οι ενότητες αυτών των λιστών αντίστοιχα με την ετικέτα

```
<dl>
  <dt>Επικεφαλίδα 1</dt>
  <dl>Στοιχείο 1.1</dl>
  <dl>Στοιχείο 1.2</dl>
  <dt>Επικεφαλίδα 2</dt>
  <dl>Στοιχείο 2.1</dl>
  <dl>Στοιχείο 2.1</dl>
</dl>
```

εμφανίζει:

Επικεφαλίδα 1
Στοιχείο 1.1
Στοιχείο 1.2
Επικεφαλίδα 2
Στοιχείο 2.1
Στοιχείο 2.1

Εικόνα 0-7 Περιγραφική λίστα

### 2.2.6.3. Άλλα στοιχεία

Στα στοιχεία ομαδοποίησης ανήκουν άλλες δύο ετικέτες. Η μία είναι η **div** που χρησιμοποιείται για τον ορισμό ενός τμήματος στο έγγραφο. Χρησιμεύει για το ορισμό ενός κοινού στυλ, στο τμήμα που ορίζει. Η άλλη είναι η **figure** που χρησιμοποιείται για την ομαδοποίηση διαφορετικών στοιχείων, όπως εικόνες, διαγράμματα, κείμενο, κώδικα κλπ. Ο τίτλος της ενότητας μπορεί να οριστεί επίσης με την ετικέτα **figcaption**.

### 2.2.7. Σημασιολογικά στοιχεία κειμένου

Τα στοιχεία αυτά, προσδίδουν την ανάλογη σημασία στο κείμενο σε σχέση πάντα με την ετικέτα που θα χρησιμοποιηθεί. Αυτές είναι:

- **code**: Εμφανίζει το κείμενο έτσι ώστε να μοιάζει με κώδικα μιας γλώσσας προγραμματισμού
- **em**: Εμφανίζει το κείμενο με πλάγιους και κάπως αραιά γραμμένους χαρακτήρες
- **dfn**: υποδηλώνει έναν όρο
- **samp**: Προσδιορίζει αποτέλεσμα ενός κώδικα
- **kbd**: υποδεικνύει το κείμενο που πρέπει να πληκτρολογήσει ο χρήστης
- **var**: μια μεταβλητή του κώδικα μιας γλώσσας προγραμματισμού
- **cite**: Παραθέτει μια αναφορά

Η χρήση και η έξοδος, των σημασιολογικών στοιχείων φαίνεται στο παράδειγμα που ακολουθεί.

```

<!DOCTYPE HTML>
<html lang="el">
<head>
<meta charset="utf-8" />
</head>
<body>
<p>
    Αναλύεται το ακόλουθο παράδειγμα κώδικα:
</p>
<hr/>
<code>
    char s[]=('H','e','l','l','o',' ','W','o','r','l','d');
    printf("%s",s);
</code>
<hr/>
    Οι μεταβλητή <var>s</var>, είναι ένας πίνακας τύπου χαρακτήρα.
<hr/>
<em>
    Η ανάθεση της τιμής του αλφαριθμητικού μπορεί να γίνει απ' ευθείας με απόδοση
    όλων των στοιχείων κατά τη δήλωση του πίνακα!!
</em>
<hr/>
<dfn>
    Το %s, δηλώνει ότι η τιμή που θα εκτυπωθεί είναι αλφαριθμητικό.
</dfn>
<hr/>
    Το αποτέλεσμα του κώδικα θα είναι:
<samp>
    Hello World
</samp>
<hr/>
<cite title="HTML & XHTML: The Definitive Guide. Published by O'Reilly Media, Inc.; fifth
edition (August 1, 2002)">
HTML & XHTML: The Definitive Guide. Published by O'Reilly Media, Inc.; fifth edition (August
1, 2002)</cite>

```

Αναλύεται το ακόλουθο παράδειγμα κώδικα:
<code>char s[]=('H','e','l','l','o',' ','W','o','r','l','d'); printf("%s",s);</code>
Οι μεταβλητή <i>s</i> , είναι ένας πίνακας τύπου χαρακτήρα.
<i>Η ανάθεση της τιμής του αλφαριθμητικού μπορεί να γίνει απ' ευθείας με απόδοση όλων των στοιχείων κατά τη δήλωση του πίνακα!!</i>
<i>Το %s, δηλώνει ότι η τιμή που θα εκτυπωθεί είναι αλφαριθμητικό.</i>
Το αποτέλεσμα του κώδικα θα είναι: Hello World
<i>HTML &amp; XHTML: The Definitive Guide. Published by O'Reilly Media, Inc.; fifth edition (August 1, 2002)</i>

Εικόνα 0-8 Σημασιολογικά στοιχεία κειμένου



Παρατηρούμε ότι ενώ μπορούμε να παραθέσουμε μια αναφορά με τη `blockquote`, η `cite` προσδίδει σημασιολογικό χαρακτήρα στην αναφορά και έτσι θα αντιμετωπιστεί από τις μηχανές αναζήτησης.

### 2.2.8. Επεξεργασμένο κείμενο

Οι ετικέτες που χαρακτηρίζουν την κατάσταση ενός κειμένου είναι η **ins** και η **del**. Χρησιμοποιούνται για να περιγράψουν αλλαγές στο κείμενο. Η πρώτη χαρακτηρίζει το κείμενο προστιθέμενο και το εμφανίζει υπογραμμισμένο. Η δεύτερη το χαρακτηρίζει διορθωμένο και το εμφανίζει υπογεγραμμένο.

### 2.2.9. Ενσωματωμένο περιεχόμενο

Περιεχόμενο εκτός κειμένου, όπως εικόνες, βίντεο, αντικείμενα, μπορούν να ενσωματωθούν με την χρήση των αντίστοιχων ετικετών που είναι:

#### 2.2.9.1. Εισαγωγή εικόνας

Η ετικέτα `img` χρησιμοποιείται για την εισαγωγή εικόνας σε μια σελίδα. Οι ιδιότητες που υποστηρίζει η ετικέτα είναι:

- **alt**: Εναλλακτικό κείμενο. Αν δεν βρεθεί η εικόνα τότε χρησιμοποιείται αυτό το κείμενο.
- **src**: Τοποθεσία της εικόνας.
- **width** , **height**: Το πλάτος και το ύψος της εικόνας.

Αντίστοιχα ο κώδικας για την εισαγωγή μιας εικόνας σε μια σελίδα είναι:

```

```

#### 2.2.9.2. Εισαγωγή εσωτερικού πλαισίου .

Ο κώδικας:

```
<iframe src="somepage.htm" sandbox=""></iframe>
```

χρησιμοποιείται για την ενσωμάτωση ενός εγγράφου `html` σε ένα άλλο. Υποστηρίζει την ιδιότητα `sandbox`, που είναι νέα στην `HTML 5` και παίρνει τις τιμές: `""`, `allow-same-origin`, `allow-top-navigation`, `allow-forms` και `allow-scripts`, οι οποίες καθορίζουν το επίπεδο αλληλεπίδρασης, αλλά και ασφάλειας μεταξύ του πλαισίου, και του αρχικού εγγράφου.

### 2.2.9.3. Ενσωμάτωση περιεχομένου

Οι ετικέτες αυτής της ομάδας επιτρέπουν την ενσωμάτωση εξωτερικού περιεχομένου όπως: εφαρμογή, αντικείμενο, βίντεο ή αρχείο ήχου. Αυτές είναι:

- **embed**: Επιτρέπει την ενσωμάτωση μιας εξωτερικής εφαρμογής.
- **object**: Ενσωματώνει ένα εξωτερικό αντικείμενο. Για τον ορισμό των παραμέτρων με τις οποίες θα κληθεί το αντικείμενο χρησιμοποιείται η param.
- **video**: Νέα ετικέτα στην HTML5, η οποία χρησιμοποιείται για την ενσωμάτωση ενός βίντεο.
- **audio**: Νέα ετικέτα στην HTML5, η οποία χρησιμοποιείται για την ενσωμάτωση ενός αρχείου ήχου.
- **map**: Δεν ενσωματώνει κάποιο περιεχόμενο, αλλά χρησιμοποιείται για να ορίσει μια περιοχή σε μια εικόνα, στην οποία θα υπάρχουν σύνδεσμοι. Για τον καθορισμό των συντεταγμένων που θα έχει ο κάθε σύνδεσμος χρησιμοποιείται η ετικέτα **area**.

Οι ετικέτες video, audio χρησιμοποιούν την ιδιότητα source, με την οποία μπορούν να οριστούν εναλλακτικοί πόροι για συγκεκριμένα αντικείμενα. Για παράδειγμα μπορεί να ορίσει τους codecs που θα χρησιμοποιηθούν για την αναπαραγωγή ενός αρχείου ήχου, όπως φαίνεται και στον κώδικα που ακολουθεί.

```
<audio controls>
<source src="example.ogg" type="audio/ogg">
<source src="example.mp3" type="audio/mpeg">
Ο περιηγητής δεν υποστηρίζει αυτό τον τύπο.
</audio>
```

Στη περίπτωση αυτή αν δεν υπάρχει κανένας από τους δύο κωδικοποιητές ήχου (ogg, mp3) στο σύστημα του χρήστη τότε θα εμφανιστεί το μήνυμα ότι ο περιηγητής δεν υποστηρίζει αυτό το τύπο αρχείου.

Εδώ επίσης θα πρέπει να αναφέρουμε ότι η HTML5 έχει μεθόδους DOM, ιδιότητες και συμβάντα για τα στοιχεία video, audio. Αυτοί οι μέθοδοι περιλαμβάνουν την: αναπαραγωγή, παύση, εναλλαγή και οι ιδιότητες αντίστοιχα: την διάρκεια, την ένταση κλπ.

Η προσπέλαση αυτών των μεθόδων μπορεί να γίνει μέσω του DOM με τη χρήση Javascript. Ένα τέτοιο παράδειγμα αποτελεί ο κώδικας που ακολουθεί, και επιδεικνύει την χρήση των μεθόδων play(), pause().

```

<!DOCTYPE html>
<html>
<body>

<div style="text-align:center">
<button onclick="playPause()">Play/Pause</button>
<br>
<video id="video1" width="420">
  <source src="mov_bbb.mp4" type="video/mp4">
  <source src="mov_bbb.ogv" type="video/ogg">
  Δεν υποστηρίζεται HTML5 video.
</video>
</div>

<script>
var myVideo=document.getElementById("video1");

function playPause()
{
  if (myVideo.paused)
    myVideo.play();
  else
    myVideo.pause();
}
</script>
</body>
</html>

```

### 2.2.10. Canvas

Ένα άλλο νέο στοιχείο ενσωμάτωσης περιεχομένου, ορίζεται με το στοιχείο **canvas**. Αυτή η ετικέτα δημιουργεί μια ορθογώνια περιοχή που μπορεί να χρησιμοποιηθεί για τον σχεδιασμό γραφικών όπως κύκλους, ορθογώνια και διάφορα σχέδια. Επίσης μπορούν να δημιουργηθούν μια σειρά από οπτικά εφέ, αλλά και εφέ μετασχηματισμού.

Ο καμβάς είναι μια περιοχή γραφικών άμεσης λειτουργίας, η οποία μπορεί να ελεγχθεί με τη χρήση JavaScript. Ο όρος άμεση λειτουργία αναφέρεται στον τρόπο με τον οποίο ο καμβάς επανασχεδιάζει από την αρχή κάθε πλαίσιο της οθόνης (frame), χρησιμοποιώντας API κλήσεις μέσω της JavaScript. Ο δημιουργός θα πρέπει να ρυθμίζει τη οθόνη πριν τον σχεδιασμό κάθε πλαισίου, έτσι ώστε να εμφανίζονται τα σωστά εικονοστοιχεία. Αυτή η τεχνολογία διαφέρει από τις άλλες όπως Flash, Silverlight που λειτουργούν σε κατάσταση έκτακτης λειτουργίας. Σε αυτή την λειτουργία, μία λίστα με αντικείμενα διατηρούνται από τη μηχανή γραφικών και αυτά εμφανίζονται σύμφωνα με τις ιδιότητες που έχουν (θέση x,y, χρώμα, διαφάνεια κλπ). Αυτό έχει σαν πλεονέκτημα ο δημιουργός να αποφεύγει τις χαμηλού επιπέδου

λειτουργίες, από την άλλη όμως περιορίζει τον λεπτομερή χειρισμό αυτών των αντικειμένων (Fulton & Fulton, 2011).

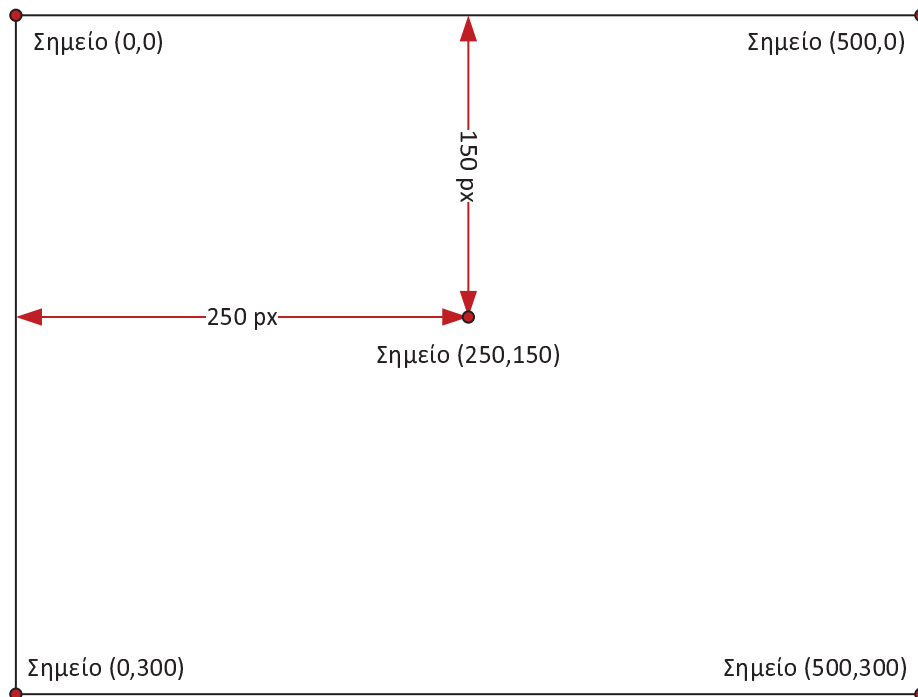
Ο βασικός καμβάς στην HTML5 περιλαμβάνει ένα 2D πλαίσιο που επιτρέπει την δημιουργία ποικίλων σχημάτων, την εξομάλυνση κειμένου (render text) και την εμφάνιση εικόνων απ' ευθείας στη καθορισμένη περιοχή. Υποστηρίζεται επίσης μια σειρά λειτουργιών όπως εφαρμογή χρώματος, περιστροφές, διαφάνειες, έλεγχο εικονοστοιχείων.

Εκτός όμως από την εμφάνιση και εξομάλυνση, μια σειρά λειτουργιών όπως απόκριση σε γεγονότα εισόδων του χρήστη, χρονοιστές, αντικείμενα, κλάσεις, υποστήριξη ήχου, μαθηματικές συναρτήσεις επιτρέπουν τη δημιουργία ολοκληρωμένων εφαρμογών.

Για την εμφάνιση σχεδίων ή εφαρμογών σε canvas, χρειάζεται η δημιουργία JavaScript κώδικα. Πρώτα απ' όλα θα πρέπει να χρησιμοποιηθεί η `document.getElementById("drawingCanvas")`, αφού η μέθοδος `getElementById()` χρησιμοποιείται για την προσπέλαση οποιουδήποτε στοιχείου της σελίδας. Στη συνέχεια ο κώδικας χρησιμοποιεί την μέθοδο `getContext()`, για να ανακτήσει ένα δυσδιάστατο περιεχόμενο σχεδίασης με την εντολή `var context = canvas.getContext("2d");`

Κάθε canvas ξεκινά με ένα άδειο παραλληλόγραμμο, σε μια θέση στη σελίδα που θα ορίσουμε. Μπορούμε στη συνέχεια να διαχειριστούμε αυτό το αντικείμενο σε όποιο σημείο θελήσουμε: για παράδειγμα, όταν η σελίδα φορτώνεται, όταν ο επισκέπτης πατήσει κάποιο κουμπί ή με κάποιο άλλο γεγονός.

Ο canvas που δημιουργείται χρησιμοποιεί ένα σύστημα συντεταγμένων για τον καθορισμό των σημείων του περιεχομένου. Το σύστημα αυτό παρουσιάζεται στην επόμενη εικόνα, όπου φαίνονται τα σημεία σε μία ορθογώνια περιοχή μεγέθους 500x300 pixel (MacDonald, 2011).



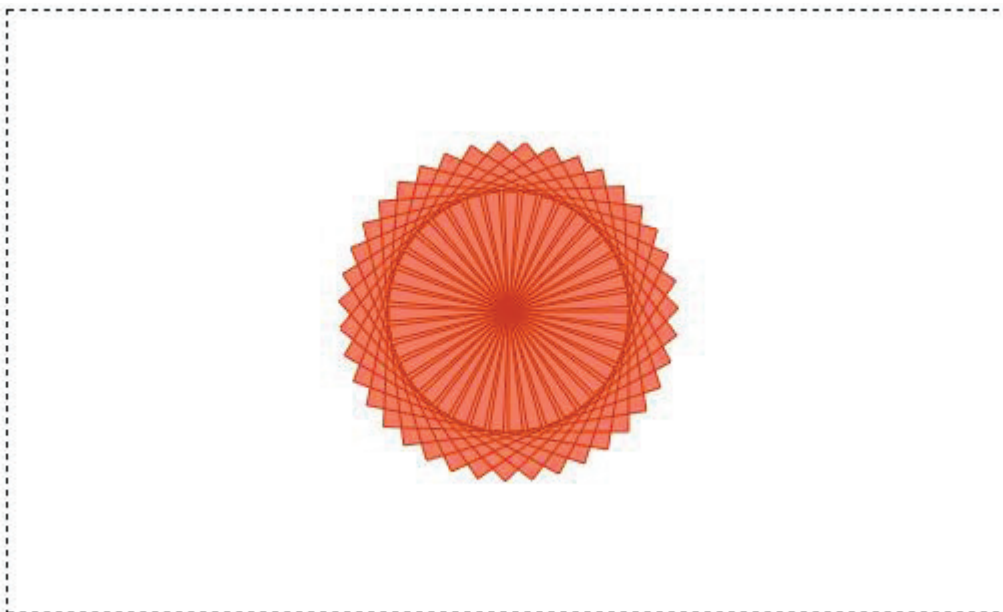
Εικόνα 0-9 Σύστημα συντεταγμένων canvas

Τα στοιχεία βασικής σχεδίασης σε ένα καμβά είναι:

- **Ευθείες γραμμές:** Το πιο απλό σχέδιο είναι οι ευθείες γραμμές. Χρειάζονται τρεις ενέργειες για αυτό. Η μέθοδος `moveTo()` που μετακινεί το δείκτη στο σημείο από το οποίο θα ξεκινήσει η γραμμή, τη μέθοδο `lineTo()`, για την σχεδίαση μιας γραμμής από το σημείο που βρισκόμαστε στο σημείο που ορίζεται στη `lineTo()` και τέλος η `stroke()` για την εμφάνιση της γραμμής.
- **Μονοπάτια και σχήματα:** Για τα σχήματα, χρησιμοποιείται πάλι η μέθοδος `lineTo()`, με τη μόνη διαφορά ότι ορίζεται ένα μονοπάτι από γραμμές για τον σχεδιασμό. Η δημιουργία σχημάτων θα μπορούσε να γίνει και χωρίς μονοπάτι, αλλά με αυτό τον τρόπο δεν υπάρχει δυνατότητα «ορισμού» του σχήματος και περαιτέρω επεξεργασίας, όπως για παράδειγμα ο χρωματισμός του. Το μονοπάτι ξεκινά με την μέθοδο `beginPath()` και κλείνει αντίστοιχα με την `closePath()`.
- **Καμπύλες γραμμές:** Ο σχεδιασμός καμπύλων σχημάτων, μπορεί να επιτευχθεί με τη χρήση των: `arc()`, `arcTo()`, `bezierCurveTo()` και `quadraticCurveTo()`. Άλλες αυτές οι μέθοδοι, ζωγραφίζουν καμπύλες γραμμές με διαφορετικό τρόπο χρησιμοποιώντας μαθηματικούς υπολογισμούς.

Αντίστοιχες μέθοδοι χρησιμοποιούνται για άλλες παραμέτρους όπως:

- **Χρωματισμός:** Για τον χρωματισμό της γραμμής που χαράσσεται χρησιμοποιείται η μέθοδος `strokeStyle()`, η οποία ορίζει ένα χρώμα μορφής `rgb`. Αντίστοιχα για το «γέμισμα» ενός σχήματος χρησιμοποιείται η `fillStyle()` σε συνδυασμό με την `fill()`.
- **Διαφάνεια:** Για την διαφάνεια ενός σχήματος, χρησιμοποιείται πάλι η `fillStyle()`, αλλά το χρώμα δεν ορίζεται με την `rgb`, αλλά με την `rgba` που παίρνει τέσσερις παραμέτρους. Οι τρεις κοινές με την `rgb` και αφορούν την ποσόστωση των τριών βασικών χρωμάτων, ενώ η τέταρτη αφορά το άλφα κανάλι και παίρνει τιμές από 0..1.
- **Μετασχηματισμός:** Με τον μετασχηματισμό, ουσιαστικά αλλάζουμε το σύστημα συντεταγμένων του καμβά. Ο μετασχηματισμός χρησιμοποιείται σε περίπτωση που θέλουμε να αποδώσουμε κίνηση σε ένα σχέδιο, χωρίς να το επανασχεδιάζουμε κάθε φορά σε καινούργια θέση. Ειδικά αν το σχέδιο αποτελείται από πολλά σχήματα αυτό είναι ιδιαίτερα βολικό. Για τον μετασχηματισμό χρησιμοποιείται η μέθοδος `translate()`.



Εικόνα 0-10 Παράδειγμα σχεδιασμού σε καμβά

Η εικόνα 2-10 είναι ένα σχήμα που έχει προκύψει από τον ακόλουθο κώδικα που περιέχει τα στοιχεία που αναφέραμε. Ο κώδικας χρησιμοποιεί τη μέθοδο `rotate`, για την περιστροφή των συντεταγμένων κάθε φορά που σχεδιάζεται ένα τετράγωνο. Εφόσον μια πλήρη περιστροφή (ένας κύκλος) ορίζεται σε  $2 \times \pi$ , αντίστοιχα σε κάθε

πέρασμα του βρόγχου περιστρέφουμε το σύστημα κατά  $2 \times \pi / \text{copies}$ , όπου  $\text{copies}$  ο αριθμός των τετραγώνων.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Canvas</title>
<style>
canvas {border: 1px dashed black;}
</style>
<script>
window.onload = function() {
  var canvas = document.getElementById("drawingCanvas");
  var context = canvas.getContext("2d"); context.translate(250, 150);
  // Draw 10 squares.
  var copies = 40;
  for (var i=1; i<copies; i++) {
    context.rotate(2 * Math.PI * 1/(copies-1));
    // Draw the square.
    context.rect(0, 0, 60, 60);
    context.strokeStyle = "rgb(205,55,0)";
    context.fillStyle = "rgba(255,36,0,0.6)";
  }
  context.fill();
  context.stroke();
};
</script>
</head>
<body>
<canvas id="drawingCanvas" width="500" height="300"></canvas>
</body>
</html>
```

### 2.2.11. Φόρμες

Μια φόρμα HTML, αποτελείται από στοιχεία εισόδου όπως: πεδία κειμένου, κουμπιά επιλογής, λίστες, κουμπιά αποστολής στοιχείων αλλά και άλλα. Οι φόρμες χρησιμοποιούνται για την αποστολή στοιχείων στον εξυπηρετητή, για αποθήκευση και μελλοντική επεξεργασία.

Οι φόρμες αποτελούν στοιχείο της HTML, από το ξεκίνημά της. Πολλά εργαλεία τρίτων κατασκευαστών όπως το JQuery, προσπάθησαν να δώσουν στις φόρμες μια πιο επαγγελματική όψη. Στην HTML5 όμως προστέθηκαν μερικά χαρακτηριστικά για τη δημιουργία πιο επαγγελματικών φορμών (Harris, 2011).

Τα αντίστοιχα στοιχεία που χρησιμοποιούνται στις φόρμες είναι:

- **datalist:** Επιτρέπει σε έναν προγραμματιστή να επισυνάψει μια λίστα από προτεινόμενες τιμές σε μια λίστα κειμένου. Μόλις ξεκινήσει η πληκτρολόγηση τότε εμφανίζεται μια αναδιπλούμενη λίστα με τις αντίστοιχες προτάσεις και ο χρήστης μπορεί να επιλέξει κάποια από αυτές.
- **fieldset:** Χρησιμοποιείται για την ομαδοποίηση των στοιχείων μιας φόρμας. Για την προσθήκη τίτλου στα ομαδοποιημένα στοιχεία της φόρμας χρησιμοποιείται η ετικέτα **<legend>**.
- **keygen:** Δημιουργεί ένα κλειδί, με το οποίο γίνεται κωδικοποίηση των δεδομένων για να μεταφερθούν με ασφάλεια στον εξυπηρετητή.
- **label:** Χρησιμοποιείται για την αντιστοίχιση ενός στοιχείου HTML σε ένα στοιχείο φόρμας. Με αυτό τον τρόπο μπορούν να εφαρμοστούν στυλ εμφάνισης στο στοιχείο της φόρμας μέσω ενός CSS αρχείου.
- **meter:** Με την ετικέτα αυτή μπορεί να οριοθετηθεί ένα αριθμητικό πεδίο. Για την οριοθέτηση χρησιμοποιεί ιδιότητες όπως η μέγιστη (max) και η ελάχιστη (min) τιμή. Επίσης μπορεί να οριστεί μια βέλτιστη τιμή (optimum).
- **output:** Περιγράφει μια ενότητα η οποία μπορεί να αλλαχτεί με τη χρήση ενός σεναρίου, όπως javascript. Αν για παράδειγμα αλλάξει η τιμή ενός πεδίου, τότε μπορεί να εμφανίζεται ένα μήνυμα δίπλα στο πεδίο ότι η τιμή άλλαξε.
- **progress:** Μετράει την πρόοδο που έχει γίνει στη συμπλήρωση μιας φόρμας. Όταν ένα στοιχείο συμπληρώνεται, τότε μπορείτε να καλείται ένα σενάριο javascript και να αυξάνει η ιδιότητα value ανάλογα με το ποσοστό ολοκλήρωσης.

Μαζί με τα νέα στοιχεία, δημιουργήθηκαν ή επαναπροσδιορίστηκαν νέες και παλιές ιδιότητες όπως:

- **autofocus:** Μπορεί να εφαρμοστεί σε οποιοδήποτε στοιχείο της φόρμας, το οποίο θα είναι το πρώτο στοιχείο που θα επικεντρωθεί η είσοδος για τον χρήστη. Συνήθως αφορά το πρώτο στοιχείο σε μια φόρμα.
- **pattern:** Χρησιμοποιεί μια κανονική έκφραση για να ταιριάζει την είσοδο του χρήστη. Επίσης πρέπει να οριστεί και ένα τίτλος ο οποίος αποτελεί μια συμβουλή προς τον χρήστη (tip).
- **placeholder:** Χρησιμοποιείται αντί του **<label>**, για να χαρακτηρίσει ένα πεδίο κειμένου. Όταν κάποιος εστιάσει στο πεδίο τότε το κείμενο του placeholder εξαφανίζεται.



- **required:** Ορίζει αν είναι υποχρεωτική η συμπλήρωση ενός πεδίου.

Στην HTML5 έχουν οριστεί και μια σειρά τύπων για τα πεδία μιας φόρμας όπως:

- **color:** Επιτρέπει στον χρήστη να επιλέξει ένα χρώμα, σε δεκαεξαδική μορφή.
- **date:** Ορίζει ότι η τιμή θα είναι τύπου ημερομηνίας.
- **datetime:** Η τιμή θα είναι σε μορφή ημερομηνίας και ώρας.
- **time:** Αντίστοιχα θα πρέπει να γίνει εισαγωγή ώρας.
- **email:** Το πεδίο αφορά την ηλεκτρονική διεύθυνση ενός χρήστη.
- **number.** Το πεδίο είναι αριθμητικό
- **search:** Στο πεδίο αυτό θα εισαχθεί ένα αλφαριθμητικό για αναζήτηση.
- **tel:** Αφορά τηλεφωνικό αριθμό.
- **url:** Αποτελεί διαδικτυακή διεύθυνση.

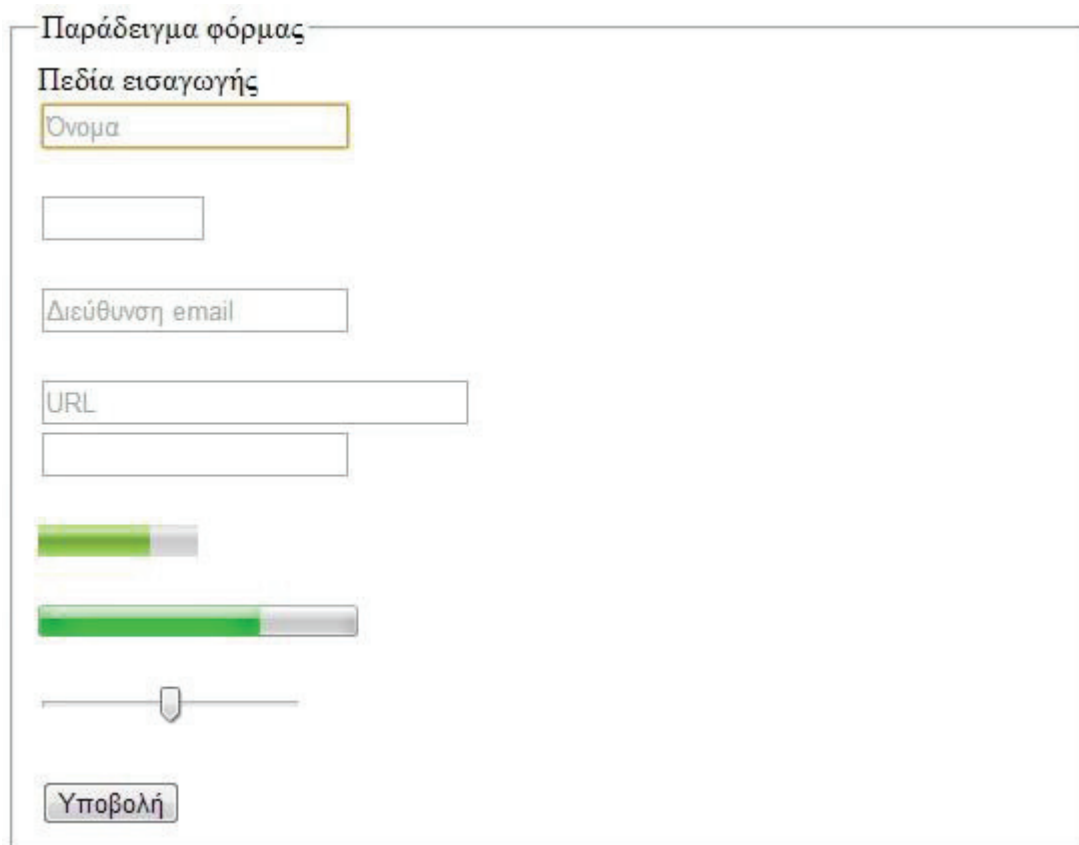
Ένα απλό παράδειγμα δημιουργίας μιας φόρμας μας δίνει ο κώδικας που ακολουθεί, ενώ τα αντίστοιχα αποτελέσματα φαίνονται στην εικόνα 2-3.

```

<!DOCTYPE HTML>
<html lang = "el">
<head>
<meta charset = "UTF-8" />
</head>
<body>
<form action = "">
<fieldset>
<legend>Παράδειγμα φόρμας</legend>
<caption>Πεδία εισαγωγής</caption><br>
<input type="text" size="20" autofocus required placeholder="Όνομα"/><br><br>
<input type="text" size="8" pattern="\d{3}-\d{4}" title="ddd-dddd"/><br><br>
<input type="email" size="20" name="email" required placeholder="Διεύθυνση
email"/><br><br>
<input type="url" size="30" name="url" placeholder="URL"/><br>
<input type="text" name="choices" list="data-list">
<datalist id="data-list">
<option value="Επιλογή 1">
<option value="Επιλογή 2">
<option value="Επιλογή 3">
<option value="Επιλογή 4">
<option value="Επιλογή 5">
</datalist><br><br>
<meter min="0" max="10" value="7"></meter><br><br>
<progress max="100" value="70">30%</progress><br><br>
<input type="range" id="range" name="range"><br><br>
<input type="submit"/>
</fieldset>
</form>
</body>
</html>

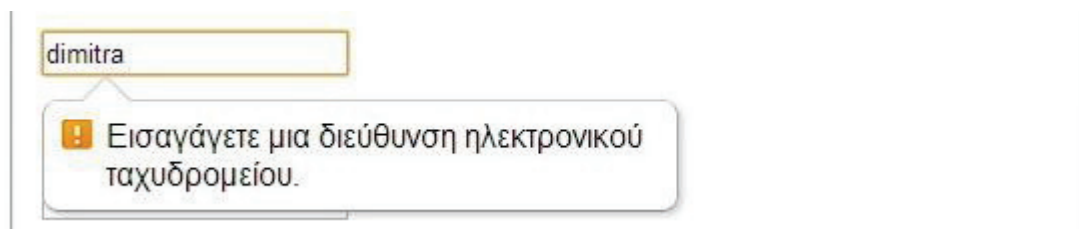
```

Η πρώτη γραμμή «form action» καθορίζει τον τρόπο με τον οποίο θα αποσταλούν τα δεδομένα στον εξυπηρετητή (POST ή GET). Επίσης η φόρμα μας έχει εστιάσει στο πεδίο όνομα, επειδή έχει χρησιμοποιηθεί η ιδιότητα autofocus. Μετά τα πεδία έχουμε δημιουργήσει αντίστοιχα αριθμητικό πεδίο (meter), μπάρα προόδου (progress) και πεδίο ορίου (range).



Εικόνα 0-11 Φόρμα HTML5

Τα πεδία που ακολουθούν έχουν δηλωθεί “required”, συνεπώς αν δεν συμπληρωθούν θα βγει το αντίστοιχο μήνυμα λάθους, αν βέβαια υποστηρίζει το λογισμικό πλοήγησης αυτή την ιδιότητα. Βλέπουμε ότι στην περίπτωση που πληκτρολογήσουμε λάθος διεύθυνση email, η υποβολή της φόρμας δεν γίνεται και εμφανίζεται αντίστοιχο μήνυμα προειδοποίησης χωρίς να γράψουμε καθόλου κώδικα.



Εικόνα 0-12 - Φόρμες μηνύματα λάθους

### 2.2.12. Πίνακες

Το μοντέλο πίνακα στην HTML γενικότερα αποτελείται από τα στοιχεία του πίνακα και τις ιδιότητες αυτών των στοιχείων. Ο πίνακας αποτελείται από κελιά στοιχισμένα σε ένα δυσδιάστατο πλέγμα με θέσεις και συντεταγμένες (x,y). Το πλέγμα είναι πεπερασμένο και μπορεί να είναι άδειο ή να έχει μία ή περισσότερες θέσεις. Τα κελιά μπορούν να είναι κελιά δεδομένων είτε κελιά επικεφαλίδων. Τα κελιά δεδομένων αντιστοιχούν στα στοιχεία **td** και τα κελιά επικεφαλίδων στα στοιχεία **th**.

Μια γραμμή είναι ένα σύνολο θέσεων για μια συγκεκριμένη τιμή του y. Οι γραμμές αντιστοιχούν σε **tr** στοιχεία. Μία στήλη είναι ένα σύνολο θέσεων για μια συγκεκριμένη τιμή του x. Αντίστοιχα μπορούν να ορισθούν και οι ομάδες γραμμών και στηλών. Άλλες ετικέτες που αφορούν τους πίνακες είναι:

- **thead**: Ορίζουμε επικεφαλίδα
- **tfoot**: Ορίζουμε υποσέλιδο
- **tbody**: ορίζουμε τις γραμμές που αποτελούν το κυρίως περιεχόμενο του πίνακα.
- **caption**: Ορίζει μια περιγραφή για τον πίνακα

Εξίσου σημαντικές όμως είναι και οι ιδιότητες των **td** και **th** στοιχείων όπως η **colspan** της **th** που χρησιμοποιείται για την οριζόντια συγχώνευση κελιών και αντίστοιχα η **rowspan** της **td** για την κατακόρυφη συγχώνευση κελιών.

Η χρήση των πινάκων γενικότερα μπορεί να κατηγοριοποιηθεί σε δύο κατηγορίες:

Η πρώτη κατηγορία που θέλει τον πίνακα, να χρησιμοποιείται ως στοιχείο παρουσίασης, δηλαδή ενός τρόπου στοιχειοθέτησης των δεδομένων. Βέβαια σε αυτή τη περίπτωση είναι προτιμότερο να χρησιμοποιείται η ετικέτα `div`, γιατί με αυτό τον τρόπο η δομή ανακτάται γρηγορότερα από το σύστημα πλοήγησης. Η χρήση του πίνακα αντί της `div`, γίνεται χρησιμοποιώντας μόνο τις ετικέτες `<table>`, `<tr>` και `<td>`.

Αντίθετα η δεύτερη κατηγορία χρησιμοποιεί και άλλα στοιχεία σήμανσης για την παρουσίαση δεδομένων στον τελικό χρήστη.

Ο κώδικας που ακολουθεί

```

<table> <caption>Παράδειγμα HTML5 πίνακα</caption>
  <thead>
    <tr>
      <th>Στήλη Α</th>
      <th>Στήλη Β</th>
      <th>Στήλη Γ</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th scope="rowgroup" colspan="3">Πρώτη ομάδα</th>
    </tr>
    <tr>
      <th>Γραμμή 1</th>
      <td>Κελί Β1 </td>
      <td>Κελί Γ1 </td>
    </tr>
    <tr>
      <th>Γραμμή 2</th>
      <td>Κελί Β2</td>
      <td>Κελί Γ2</td>
    </tr>
  </tbody>
  <tbody>
    <tr>
      <th scope="rowgroup" colspan="3">Δεύτερη ομάδα</th>
    </tr>
    <tr>
      <th>Γραμμή 1</th>
      <td>Κελί Β1</td>
      <td>Κελί Γ1</td>
    </tr>
    <tr>
      <th>Γραμμή 2</th>
      <td>Κελί Β2</td>
      <td>Κελί Γ2</td>
    </tr>
  </tbody>
</table>

```

μας δίνει τον πίνακα

Παράδειγμα HTML5 πίνακα		
Στήλη Α	Στήλη Β	Στήλη C
Πρώτη ομάδα		
Γραμμή 1	Κελί Β1	Κελί Γ1
Γραμμή 2	Κελί Β2	Κελί Γ2
Δεύτερη ομάδα		
Γραμμή 1	Κελί Β1	Κελί Γ1
Γραμμή 2	Κελί Β2	Κελί Γ2

### 2.2.13. Σύνδεσμοι

Οι σύνδεσμοι σχηματίζονται με τα στοιχεία `a`, `area` και `link` και αναπαριστούν μια σύνδεση μεταξύ δύο πόρων.

Με την ετικέτα `<a>` δημιουργείται ένας σύνδεσμος προς μια άλλη σελίδα, μέσω της ιδιότητας `href`. Αντίστοιχα με την ετικέτα `<area>` ορίζεται μια περιοχή σε μια εικόνα ως σύνδεσμος σε συνδυασμό με την ετικέτα `map`. Με την `<link>` ορίζεται η σχέση μεταξύ του τρέχοντος εγγράφου και ενός εξωτερικού αρχείου. Χρησιμοποιείται κυρίως όπως είδαμε για την προσθήκη κάποιου αρχείου CSS ή Javascript.

Οι ιδιότητες των συνδέσμων φαίνονται στον πίνακα που ακολουθεί:

Ιδιότητα	Τιμή	Περιγραφή
<b>href</b>	URL	Ορίζει το URL του εγγράφου
<b>hreflang</b>	κωδικός γλώσσας	Ορίζει την γλώσσα του εγγράφου που δείχνει το URL της ιδιότητας <b>href</b>
<b>media</b>	all, braille, print, projection, screen, speech	Ορίζει την συσκευή (οθόνη, εκτυπωτής, προτζέκτορας, ομιλία) στην οποία θα εμφανιστεί το περιεχόμενο του εγγράφου
<b>rel</b>	alternate, archives, author, first, help, icon, index, last, license, next pingback, prefetch prev, search, stylesheet, sidebar, tag, up	Ορίζει την σχέση μεταξύ του τρέχον εγγράφου με το έγγραφο που δείχνει το URL της ιδιότητας <b>href</b>
<b>sizes</b>	αριθμός any	Ορίζει τα μεγέθη των εξωτερικών πηγών. Χρησιμοποιείται μόνο όταν <b>rel="icon"</b>
<b>type</b>	τύπος αρχείου, όπως: text/css text/javascript image/gif	Ορίζει τον τύπο του περιεχομένου του εγγράφου που δείχνει το URL της ιδιότητας <b>href</b>

Τα αντίστοιχα παραδείγματα για τις ετικέτες παρουσιάζονται στον κώδικα που ακολουθεί:

```
<!DOCTYPE HTML>
<html lang="el">
<head>
  <meta charset="utf-8" />
  <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
  Σύνδεσμος στο in.gr: <br /><a href="http://www.in.gr" target="_blank">in.gr σύνδεσμος</a>
</body>
</html>
```

Στην τέταρτη γραμμή ορίζεται το αρχείο στυλ για την εμφάνιση. Ενώ στο body ορίζουμε έναν υπερσύνδεσμο σε έναν εξωτερικό ιστότοπο. Το αποτέλεσμα που λαμβάνουμε είναι:

Σύνδεσμος προς το [in.gr](#)  
[in.gr Σύνδεσμος](#)

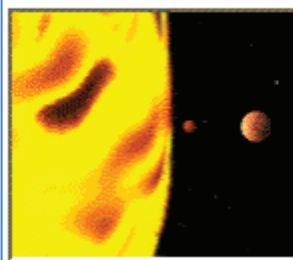
Εικόνα 0-13 Αποτέλεσμα χρήσης href

Αντίστοιχα η ετικέτα area χρησιμοποιείται ως εξής:

```
<!DOCTYPE HTML>
<html lang="el">
<head>
  <meta charset="utf-8" />
  <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
  <p>Επέλεξε έναν από τα τρία ουράνια σώματα:</p>
  
  <map name="planetmap">
    <area shape="rect" coords="0,0,82,126" alt="Sun" href="sun.htm" />
    <area shape="circle" coords="90,58,3" alt="Mercury" href="mercur.htm" />
    <area shape="circle" coords="124,58,8" alt="Venus" href="venus.htm" />
  </map>
</body>
</html>
```

Το αποτέλεσμα θα είναι η εικόνα που ακολουθεί, η οποία έχει τμηματοποιηθεί με την χρήση των ιδιοτήτων shape, coords. Έτσι η αν ο χρήστης πατήσει στο δεξιό ουράνιο σώμα θα ακολουθήσει τον σύνδεσμο που περιγράφεται από την αναφορά «venus.htm».

Επέλεξε έναν από τα τρία ουράνια σώματα:



Εικόνα 0-14 Αποτέλεσμα χρήσης της area, map

## Διεπαφές προγραμματισμού (APIs)

Η διεπαφή προγραμματισμού εφαρμογών, ή αλλιώς διασύνδεση προγραμματισμού εφαρμογών (APIs: Application Programming Interface), παρέχει τις απαραίτητες βιβλιοθήκες προκειμένου να εκτελούνται λειτουργίες ή ανταλλαγή δεδομένων. Αυτές οι βιβλιοθήκες αποτελούν την διεπαφή για τις κλήσεις συστήματος με τέτοιο τρόπο, ώστε να αποκρύπτονται οι χαμηλού επιπέδου λεπτομέρειες από τον προγραμματιστή, με στόχο την ασφάλεια και την προστασία του συστήματος.

Η HTML5 έχει πρόσβαση σε αντίστοιχες διεπαφές μέσω javascript. Τέτοιες διεπαφές είναι:

- Communication APIs
- FileSystem APIs
- Geolocation APIs

### 2.3.1. Communication APIs

Όλες οι εφαρμογές WEB, που εκτελούνται, επηρεάζονται από έναν μηχανισμό ασφαλείας του φυλλομετρητή. Αυτό σημαίνει ότι ο φυλλομετρητής δεν επιτρέπει την εκτέλεση ύποπτου κώδικα που θα μπορούσε να βλάψει το τοπικό μηχάνημα. Για αυτό τον λόγο άλλωστε, οι εφαρμογές WEB, δεν μπορούν να δημιουργήσουν, τροποποιήσουν ή να διαβάσουν αρχεία από το τοπικό σύστημα. Επίσης δεν μπορούν να έχουν πρόσβαση ή να εκτελέσουν σενάρια σε σελίδες ή τοποθεσίες που δεν χρησιμοποιούν το ίδιο πρωτόκολλο, αριθμό πόρτα και όνομα τομέα (Freeman, 2011).

Αυτό ο περιορισμός στην HTML5 μπορεί να αρθεί με συγκεκριμένες τεχνικές όπως:

#### 2.3.1.1. XHR & XHR2 με CORS

Το XHR είναι ένα API, που υποστηρίζει σύγχρονες αιτήσεις, δηλαδή η εκτέλεση του κώδικα γίνεται μετά από την απάντηση του καλούντα. Το XHR2 υποστηρίζει επιπλέον τον χειρισμό της επιφόρτωσης (upload) και της μεταφόρτωσης (download) μέσω ενός συμβάντος προόδου. Το XHR2 με CORS (CORS: Cross-Origin Resource Sharing), επιτρέπει να γίνει το αίτημα μόνο όταν ο εξυπηρετητής το επιτρέπει.

Το κομμάτι του κώδικα για την πλευρά του πελάτη (client) είναι:

```
var client = new XMLHttpRequest();
client.onreadystatechange = function () {
  if (this.readyState == 4 && this.status == 200) {
    alert(Δοκιμή xhr2 με cors: ' + this.responseText);
  }
};
client.open('GET', 'http://example.com/cors.php');
client.send();
```

Αν ο εξυπηρετητής απαντήσει με μία επικεφαλίδα CORS τότε το μήνυμα θα μεταδοθεί κανονικά. Βέβαια για να δοθεί η απάντηση θα πρέπει να υπάρχει και ο αντίστοιχος κώδικας στον εξυπηρετητή, δηλαδή στην διεύθυνση <http://www.example.com/cors.php>

```
<?php
header('Access-Control-Allow-Origin: *');
?>
@απάντηση
```

### 2.3.1.2. Web Messaging

Σε αυτή την περίπτωση χρησιμοποιείται ένα παλαιότερο API, το `postMessage` με το οποίο επιτρέπεται η αποστολή αλφαριθμητικών (μηνυμάτων), σε διαφορετικούς τομείς και προελεύσεις (domains and origins).

Για παράδειγμα μία σελίδα, που δέχεται γεγονότα `onmessage`, μπορεί να περιλαμβάνει τον ακόλουθο κώδικα:

```
window.onmessage = function (event) {
  if (event.origin == 'mytrustedsite.com') {
    alert('my trusted site said: ' + event.data);
  }
};
```

Με την χρήση ενός `<iframe>`, που περιέχει κώδικα και χρησιμοποιεί τον `<iframe>` κόμβο DOM (DOM node), μπορούμε να θέσουμε στο `<iframe>`:

```
iframe.contentWindow.postMessage("γεια εκεί", location.host);
```

Αυτό μας δίνει την δυνατότητα όπως είδαμε αποστολής του αλφαριθμητικού “γεια εκεί”, ανάμεσα σε τομείς εμπιστοσύνης.

### 2.3.1.3. Web Sockets

Χρησιμοποιείται για την αποστολή μηνυμάτων προς και από τον εξυπηρετητή. Η επικοινωνία μπορεί να γίνει ανάμεσα σε διαφορετικούς τομείς.



Με την χρήση των Web Sockets, μπορούν οι απλές εφαρμογές να εγκατασταθούν σε έναν εξυπηρετητή, ενώ σε έναν άλλο να χρησιμοποιηθούν άλλες πιο απαιτητικές υπηρεσίες όπως για παράδειγμα πολυμέσα ροής (streaming).

Η αποστολή μηνυμάτων γίνεται μόνο στην περίπτωση που οι υποδοχές (sockets)<sup>5</sup>, είναι ανοιχτές. Το μοντέλο επικοινωνίας μοιάζει ως εξής:

```
var ws = new WebSocket('ws://somesite.com/updates');
ws.onmessage = function (event) {
  alert(event.data);
};
ws.onopen = function () {
  ws.send('yay! we connected!');
};
```

#### 2.3.1.4. Server Sent Events

Επιτρέπει την αποστολή μηνυμάτων από τον εξυπηρετητή προς τον πελάτη και όχι το αντίστροφο. Ο πελάτης μπορεί μόνο να «ακούει» μηνύματα. Αυτό το API, χρησιμοποιεί το μοντέλο onmessage. Κατασκευάζεται χρησιμοποιώντας το αντικείμενο EventSource (Sharp, 2011).

Η δημιουργία κάποιου γεγονότος μπορεί να γίνει ως εξής:

```
var es = new EventSource('/sse');
es.onopen = function () {
  console.log('opened stream');
};

es.onmessage = function (event) {
  console.log('new message: ' + event.data);
};
```

Το SSE (Server Sent Event), συνδέεται αυτόματα, κατά την δημιουργία του αντικειμένου και ενεργοποιεί το γεγονός onopen.

#### 2.3.2. Διεπαφή συστήματος αρχείων (Filesystem APIs)

Το HTML5 σύστημα αρχείων API, δίνει τη δυνατότητα στις εφαρμογές να γράφουν και να αποθηκεύουν αρχεία μέσω της Javascript. Οι δυνατότητες που προσφέρονται μέσω της διασύνδεσης είναι:

---

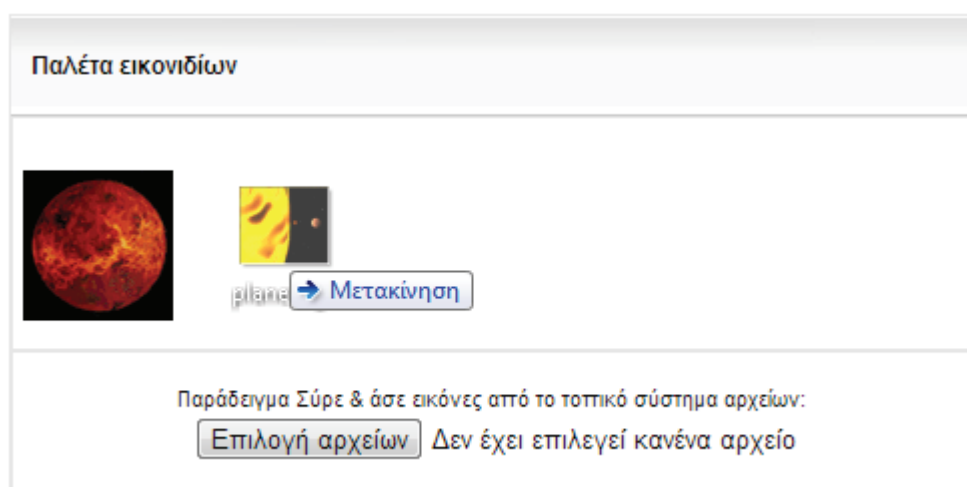
<sup>5</sup> Socket είναι το ένα άκρο, από έναν επικοινωνιακό δίαυλο διπλής κατεύθυνσης, μεταξύ δύο προγραμμάτων που εκτελούνται στο δίκτυο. Περιλαμβάνει το πρωτόκολλο, την διεύθυνση και τον αριθμό θύρας του άκρου (Anon., 2013).

- Ανάγνωση και διαχείριση αρχείων με τη χρήση των File/Blob, FileList, FileReader.
- Δημιουργία και εγγραφή με τις εντολές BlobBuilder, FileWriter.
- Πρόσβαση σε καταλόγους και συστήματα αρχείων με DirectoryReader, FileEntry/DirectoryEntry, LocalFileSystem.

Ένα παράδειγμα εφαρμογής είναι το ακόλουθο (Leah, 2013). Ουσιαστικά χρησιμοποιείται η λειτουργία σύρε & άσε, για την επιλογή αρχείων από το τοπικό σύστημα αρχείων.

```
function imagesSelected(myFiles) {
  for (var i = 0, f; f = myFiles[i]; i++) {
    var imageReader = new FileReader();
    imageReader.onload = (function(aFile) {
      return function(e) {
        var span = document.createElement('span');
        span.innerHTML = [''].join("");
        document.getElementById('thumbs').insertBefore(span, null);
      };
    })(f);
    imageReader.readAsDataURL(f);
  }
}

function dropIt(e) {
  imagesSelected(e.dataTransfer.files);
  e.stopPropagation();
  e.preventDefault();
}
```



Εικόνα 0-15 Παράδειγμα εφαρμογής Filesystem API

Οι επιλογές αποθήκευσης του API, αφορούν μόνο την πλευρά του πελάτη. Μια τυπική διαδικτυακή εφαρμογή, μπορεί να χρησιμοποιήσει χώρο αποθήκευσης είτε προσωρινά είτε μόνιμα.

Η προσωρινή αποθήκευση μπορεί να πραγματοποιηθεί εύκολα χωρίς στην συγκατάθεση του χρήστη. Επίσης δεν απαιτείται η εκχώρηση ειδικών δικαιωμάτων. Η αποθήκευση αυτού του τύπου συνίσταται για χρήση κρυφής μνήμης (cache).

Η μόνιμη αποθήκευση χρησιμοποιείται για την αποθήκευση δεδομένων με τέτοιο τρόπο ώστε αυτά να είναι διαχειρίσιμα και για τις επόμενες προσβάσεις. Μία τέτοια λειτουργία είναι η αποθήκευση cookies, που διασφαλίζει ότι ο ιστότοπος «θυμάται» τις προτιμήσεις του χρήστη. Βέβαια ακόμα και η μόνιμη αποθήκευση μπορεί να διαγραφεί από τον χρήστη (Bidelman, 2011).

Σε αντίθεση με την προσωρινή αποθήκευση ισχύουν:

- Ο φυλλομετρητής ζητάει την άδεια του χρήστη για να αποθηκεύσει τα δεδομένα.
- Το μέγεθος των δεδομένων είναι σταθερό και δεν μπορεί να ζητηθεί έξτρα χώρος.

### 2.3.3. Geolocation API

Το Geolocation API, χρησιμοποιείται για τον προσδιορισμό της γεωγραφικής θέσης ενός χρήστη. Αποτελεί υψηλού επιπέδου API, συνεπώς ο χρήστης δεν χρειάζεται να ξέρει λεπτομέρειες για τον τρόπο που γίνεται ο εντοπισμός.

Το Geolocaton API, υποστηρίζεται σχεδόν από όλους τους μοντέρνους φυλλομετρητές τόσο σε υπολογιστές, αλλά και σε κινητές συσκευές. Ο εντοπισμός μπορεί να επιτευχθεί μέσω πολλών τρόπων, με το καθένα να έχει συγκεκριμένη ακρίβεια ως προς τον ακριβή προσδιορισμό. Οι τρόποι αυτοί συνοπτικά είναι:

- **IP Διεύθυνση:** Είναι δυνατός ο εντοπισμός της θέσης μέσω της διεύθυνση IP του χρήστη. Κάθε χώρα, περιοχή, πόλη έχει συγκεκριμένο εύρος IP διευθύνσεων. Συνεπώς κάποιος μπορεί να προσδιορίσει τη θέση ενός χρήστη, μέσω αυτού του τρόπου, όχι όμως με μεγάλη ακρίβεια.

- **Σήμα Wi-Fi:** Σε συνδυασμό με τον εντοπισμό μέσω IP διεύθυνσης μπορεί να δώσει μεγαλύτερη ακρίβεια. Στηρίζεται στο γεγονός, ότι τα Wi-Fi δίκτυα έχουν ευρετηριαστεί από αντίστοιχες υπηρεσίες και βάση αυτής της πληροφορίας είναι γνωστό το στίγμα τους. Έτσι από την ένταση του σήματος, το όνομα του δικτύου και το χρόνο ανίχνευσης μπορεί να υπολογιστεί η θέση του χρήστη από την γνωστή τοποθεσία.
- **Σήμα GPS:** Είναι η μέθοδος με την υψηλότερη ακρίβεια, χρησιμοποιείται όμως κυρίως από κινητές συσκευές. Χρησιμοποιεί ένα σύστημα δορυφόρων για τον εντοπισμό της θέσης ενός σημείου στη γη, χρησιμοποιώντας τριγωνισμό.

Οι τρεις βασικές συναρτήσεις που χρησιμοποιεί το API είναι:

- **getCurrentPosition:** Χρησιμοποιείται για να προσδιορίσει τις γεωγραφικές συντεταγμένες του χρήστη. Παίρνει δύο ορίσματα: το ένα για την επιτυχή αναζήτηση (locationSuccess) και το άλλο για την περίπτωση αποτυχίας (locationFail). Η συνάρτηση επιστρέφει ένα αντικείμενο θέσης, μέσω του οποίου μπορούμε να λάβουμε αρκετές πληροφορίες σε σχέση με τη θέση όπως: γεωγραφικό μήκος/πλάτος/ύψος, ακρίβεια, ταχύτητα, ώρα.
- **watchPosition:** Παρακολουθεί τις συντεταγμένες του χρήστη σε πραγματικό χρόνο. Επιστρέφει ένα αναγνωριστικό χρήστη (ID)
- **clearWatch:** Σταματάει την παρακολούθηση της θέσης του χρήστη. Παίρνει σαν παράμετρο το αναγνωριστικό που έχει επιστραφεί με τη watchPosition.

Χρησιμοποιώντας αντίστοιχα τις παραπάνω εντολές μπορούμε να προσδιορίσουμε τη γεωγραφική μας θέση με τον κάτωθι τρόπο:

```

if (navigator.geolocation) {
  var timeoutVal = 10 * 1000 * 1000;
  navigator.geolocation.getCurrentPosition(
    displayPosition,
    displayError,
    { enableHighAccuracy: true, timeout: timeoutVal, maximumAge: 0 }
  );
}
else {
  alert("Δεν υποστηρίζεται από το σύστημα πλοήγησης");
}

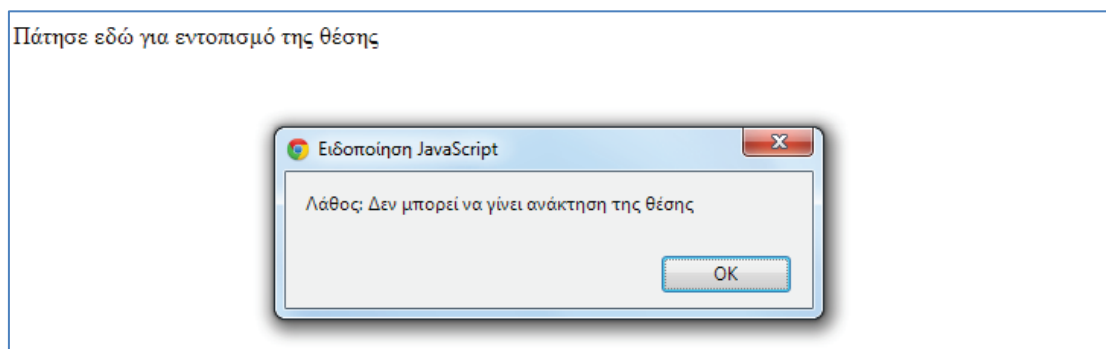
function displayPosition(position) {
  alert("Γεωγ.Πλάτος: " + position.coords.latitude + ", Γεωγ.Μήκος: " + position.coords.longitude);
}

```

Αντίστοιχα η συνάρτηση `displayError` ορίζεται ως εξής:

```
function displayError(error) {  
  var errors = {  
    1: 'Δεν επιτρέπεται η πρόσβαση',  
    2: 'Δεν μπορεί να γίνει ανάκτηση της θέσης',  
    3: 'Το χρονικό όριο του αιτήματος εξαντλήθηκε'  
  };  
  alert("Λάθος: " + errors[error.code]);  
}
```

Στην περίπτωση τη δική μας δεν ήταν δυνατός ο εντοπισμός της θέσης, αφού στον υπολογιστή δεν είχαμε κάποιο gps η wifi δέκτη.



Εικόνα 0-16 Εντοπισμός θέσης

## XHTML 2.0

### Εισαγωγή

Η XHTML ανήκει στην οικογένεια των XML γλωσσών σήμανσης. Ουσιαστικά αποτέλεσε μία αναδιατύπωση της HTML. Αποτελείται από χαρακτηριστικά και ενότητες τα οποία επεκτείνουν την HTML και συγκεκριμένα την έκδοση 4.0. Υιοθέτησε χαρακτηριστικά από την XML, αλλά και από την HTML με σκοπό την αυστηρότερη και καθαρότερη παραγωγή κώδικα (Duckett, 2010).

### Περιγραφή της XHTML

Η XHTML αναπτύχθηκε για να υπηρετήσει ένα σημαντικό σκοπό στο διαδίκτυο. Πολλές σελίδες περιείχαν «κακό» κώδικα HTML. Αυτό συνέβαινε για λόγους όπως ετικέτες που δεν έκλειναν, η έκλειναν με λάθος σειρά, αλλά και ιδιότητες οι οποίες δεν ήταν σωστές. Η XHTML δημιουργήθηκε για να αντιμετωπίσει αυτά τα λάθη. Ορίζοντας κανόνες που θα έπρεπε να ακολουθήσει η HTML, όπως όλες οι ετικέτες θα πρέπει να κλείνουν και μάλιστα εμφωλιασμένες η μία στην άλλη με τη σωστή σειρά, κατάφερε να βελτιώσει τον κώδικα HTML. Αυτή η βελτίωση παρήγαγε ένα καθαρότερο κώδικα με καλύτερα οπτικά αποτελέσματα για τους επισκέπτες των δικτυακών τόπων.

Η πρώτη έκδοση της XHTML, XHTML 1.0 προτάθηκε από το W3C στις 26 Ιανουαρίου του 2000. Δεν διέφερε πολύ από την HTML 4.01, αλλά πρόσθεσε μια σειρά νέων κανόνων σύνταξης και ετικετών που θα έπρεπε να τηρούνται από τους κατασκευαστές. Ο συνδυασμός του σκοπού της HTML (εμφάνισης περιεχομένου) και της XML (σήμανση σωστή με κανόνες) είχε σαν αποτέλεσμα η XHTML να γίνει αρκετά πρακτική στη δημιουργία σελίδων. Η έκδοση αυτή περιείχε τρεις τύπους εγγράφων που αντιστοιχούν στους διαφορετικούς τύπους της HTML 4.01, σύμφωνα με τη μορφή DTDs<sup>6</sup> που ήταν:

- **XHTML 1.0 Strict:** Περιλαμβάνει στοιχεία και ιδιότητες που δεν έχουν χαρακτηριστεί υπό κατάργηση στις προδιαγραφές της HTML 4.01.
- **XHTML 1.0 Transitional:** Περιλαμβάνει τα στοιχεία παρουσίασης όπως (center, font και strike) που δεν περιλαμβάνονται στην strict έκδοση.
- **XHTML 1.0 Frameset:** Επιτρέπει τον καθορισμό εγγράφων με πλαίσια.

---

<sup>6</sup> (DTD: Document Type Definition) Αποτελεί ένα σύνολο δηλώσεων σήμανσης που καθορίζουν το τύπο του εγγράφου.

Η XHTML 1.1 εισήγαγε την έννοια των ενοτήτων (modules). Στην XHTML μια ενότητα είναι ένα σύνολο σχετιζόμενων στοιχείων. Για παράδειγμα η ενότητα «Forms», περιέχει συστατικά για την δημιουργία φορμών.

Σκοπός αυτής της τμηματοποίησης ήταν ο διαχωρισμός των στοιχείων HTML σε ομάδες σχετικών στοιχείων. Αντί λοιπόν να χρησιμοποιείται ολόκληρο το σύνολο της γλώσσας, οι κατασκευαστές μπορούν να διαλέξουν ποιες ενότητες θα χρησιμοποιήσουν, ανάλογα με την πλατφόρμα ανάπτυξης.

Ενδιάμεσα παρουσιάστηκαν κάποιες ειδικές εκδόσεις της XHTML όπως:

- **XHTML Print:** Σχεδιάστηκε για την εκτύπωση εγγράφων από κινητές συσκευές σε χαμηλού κόστους εκτυπωτές που δεν έχουν μνήμη αποθήκευσης και χρησιμοποιούνται για απ' ευθείας εκτύπωση (Anon., 2010). Έγινε πρότυπο τον Σεπτέμβριο του 2006.
- **XHTML Mobile:** Σχεδιάστηκε από μία ομάδα υπηρεσιών για κινητές συσκευές (ομα: Open Mobile Alliance) και αποτελεί μια γλώσσα σήμανσης κυρίως για κινητά τηλέφωνα, γενικότερα δε για συσκευές με περιορισμένους πόρους (Anon., 2001).

Η ομάδα εργασίας XHTML 2, θεώρησε την δημιουργία μιας νέας γλώσσας βασισμένη στην HTML 1.1 με την ονομασία XHTML 1.2. Παρ' όλα αυτά ποτέ δεν προχώρησε η υλοποίηση και η έκδοση αυτή παρέμεινε σαν μια ιδέα.

Μεταξύ του Αυγούστου του 2002 και του Ιουλίου του 2006, το W3C κυκλοφόρησε οκτώ προσχέδια της XHTML 2.0, μιας νέας έκδοσης XHTML η οποία επανασχεδιάστηκε με σκοπό να είναι μια ξεκάθαρη έκδοση, χωρίς την υποστήριξη συμβατότητας προς τα πίσω. Αυτή η έλλειψη συμβατότητας, προκάλεσε μεγάλη διαμάχη στην κοινότητα των κατασκευαστών. Ένα ένατο προσχέδιο της XHTML 2.0, αναμενόταν να εμφανιστεί το 2009, όμως στις 2 Ιουλίου του 2009, το W3C αποφάσισε να μην προχωρήσει σε περαιτέρω ανάπτυξη. Έτσι η XHTML 2.0 και τα σχετικά κείμενα κυκλοφόρησαν από το W3C σαν σημειώσεις τον Δεκέμβριο του 2010.

Η XHTML 2.0 υιοθέτησε νέα χαρακτηριστικά την HTML οικογένεια γλωσσών σήμανσης (Πομπόρτσης, 2009):

- Οι HTML φόρμες αντικαταστάθηκαν από της XForms που είναι μια προδιαγραφή εισόδου του χρήστη με βάση την XML και η οποία επιτρέπει στις φόρμες να εμφανίζονται κατάλληλα σε διαφορετικές διατάξεις απόδοσης.
- Τα HTML πλαίσια αντικαταστάθηκαν από τα XFrames.
- Τα DOM γεγονότα, αντικαταστάθηκαν από τα XML γεγονότα, τα οποία χρησιμοποιούν το Μοντέλο Αντικειμένου Εγγράφου (DOM-Document Object Model).
- Ένας νέος τύπος στοιχείου `ul` συμπεριλήφθηκε για να ορίσει ένα κατάλογο, ως ένα κατάλογο πλοήγησης.

Παρ' όλο τη μη υιοθέτηση τελικά της XHTML 2.0 για πρότυπο, αξίζει να αναφερθούμε και να εντοπίσουμε μειονεκτήματα ή πλεονεκτήματα σε σχέση με τον ανταγωνιστή του.

### **Ενότητες της XHTML 2.0**

Ο XHTML 2.0 τύπος εγγράφου, είναι ένας πλήρης λειτουργικός τύπος με ισχυρή σημασιολογία. Αποτελείται από ενότητες, με τη κάθε ενότητα να αποτελείται από σύνολα ετικετών και ιδιοτήτων. Υπάρχουν δύο ειδών ενότητες. Αυτές που έχουν αναπτυχθεί εσωτερικά από το W3C και είναι:

- Ενότητα εγγράφου
- Ενότητα διάρθρωσης
- Ενότητα κειμένου
- Ενότητα λίστας
- Ενότητα ιδιοτήτων πυρήνα
- Ενότητα ιδιοτήτων υπερκειμένου
- Ενότητα διεθνοποίησης
- Ενότητα αμφίδρομου κειμένου
- Ενότητα σύνταξης
- Ενότητα ιδιοτήτων ενσωμάτωσης
- Ενότητα εικόνων χαρτών
- Ενότητα πολυμέσων
- Ενότητα μεταδεδομένων
- Ενότητα αντικειμένου
- Ενότητα φύλλου στυλ



- Ενότητα πινάκων

Οι αντίστοιχες ενότητες που έχουν αναπτυχθεί από τρίτους είναι:

- Ενότητα σχολίων Ruby
- XForms 2.0
- Ενότητα ιδιοτήτων μεταδεδομένων
- Ενότητα γεγονότων
- Ενότητα χειριστών
- Ενότητα σεναρίων
- Σύνολα ετικετών διεθνοποίησης

### 3.2.1. Ενότητα εγγράφου

Αυτή η ενότητα περιγράφει τη κύρια δομή ενός εγγράφου XHTML μέσω μιας ιεραρχημένης σειράς συστατικών. Η δομή αυτή περιγράφεται στον κώδικα που ακολουθεί:

```
<html xmlns="http://www.w3.org/2002/06/xhtml2" xml:lang="gr">
<head>
  <title>Τίτλος σελίδας</title>
</head>
<body id="theBody">
  <p>Μια παράγραφος</p>
</body>
```

Το **html** είναι το αρχικό στοιχείο για όλες τις σελίδες. Με αυτό ορίζονται διάφορες ιδιότητες του εγγράφου, όπως η γλώσσα ή το είδος της σελίδας καθώς και άλλα χαρακτηριστικά. Το **head** ορίζει την επικεφαλίδα και μπορεί να περιέχει ένα τίτλο με την ετικέτα **title** όπως φαίνεται στο κώδικα. Το **body** περιλαμβάνει το περιεχόμενο της σελίδας και μπορεί να περιέχει κείμενο, εικόνες, γραφικά και μια σειρά ετικετών ή οδηγιών.

### 3.2.2. Διάρθρωση

Η ενότητα αυτή περιγράφει όλα τα στοιχεία, ιδιότητες και μοντέλα που χρησιμοποιούνται για την διάρθρωση ενός εγγράφου. Τα στοιχεία αυτά είναι:

- **address**: Χρησιμοποιείται για την παροχή στοιχείων επικοινωνίας για τον συγγραφέα ενός εγγράφου είτε για μέρος του εγγράφου.

```
<address href="mailto:info@example.com">Πληροφορίες</address>
```

- **blockquote**: Επισημαίνει ότι το κείμενο που περιλαμβάνεται αποτελεί ένα μπλοκ με κώδικα.

```
<blockquote class="C++"
  cout << "Γεια σου.\n" <<endl;
</blockquote>
```

- **blockquote**: Ορίζει ένα μπλοκ κειμένου σε παράθεση.

```
<blockquote cite="http://www.example.com/index.html"
<p> Αυτή η παράγραφος θα είναι σε παράθεση και θα ξεχωρίζει από τις άλλες.</p>
</blockquote>
```

- **div**: Το στοιχείο div σε συνδυασμό με τις ιδιότητες τα @id, @class και @role, προσφέρει ένα γενικό μηχανισμό για την διάρθρωση του εγγράφου. Συνήθως χρησιμοποιείται σε συνδυασμό με φύλλα στυλ.

```
<div class="post">
<p>.....</p>
</div>
```

- **section**: Σε συνεργασία με το στοιχείο h, προσφέρει ένα μηχανισμό δόμησης των εγγράφων σε τμήματα.
- **h**: Περιγράφει εν συντομία το θέμα της ενότητας που εισάγει με την έννοια του τίτλου.

```
<h> Επικεφαλίδα 1ο επιπέδου </h>
<p>.....</p>
<section>
  <h> Επικεφαλίδα 2ο επιπέδου </h>
  <p>.....</p>
</section>
  <h>Επικεφαλίδα 3ο επιπέδου </h>
</section>
</section>
```

- **p**: Αναπαριστά μια παράγραφο.

```
<p>Αυτή είναι μία παράγραφος</p>
```

- **pre**: Καθορίζει ότι τα κενά που περιέχονται στο κείμενο έχουν σημασιολογικό ενδιαφέρον. Δηλαδή ο κώδικας που ακολουθεί θα εμφανίσει στον

```
<pre> Κείμενο με κενά</pre>
```

φυλλομετρητή το «Κείμενο με κενά» αντί για «Κείμενο με κενά» αν δεν είχε χρησιμοποιηθεί.

- **separator**: Διαχωρίζει τμήματα του κειμένου με την προσθήκη π.χ μιας κενής γραμμής.

```
<p>Γραμμή 1</p>
<separator/>
</p>Γραμμή 2</p>
```

### 3.2.3. Κείμενο

Αυτή η ενότητα καθορίζει όλα τα βασικά στοιχεία κειμένου, καθώς και τις ιδιότητές του. Αυτά τα στοιχεία μπορούν να ενσωματωθούν σε άλλα στοιχεία (inline mode), καθορίζοντας τον τρόπο παρουσίασης. Αν χρησιμοποιηθούν μόνα τους αποτελούν σημασιολογικό στοιχείο του εγγράφου.

Τα στοιχεία είναι:

- **abbr**: Χαρακτηρίζει το κείμενο ως ακρωνύμιο.

```
<abbr> title="Δημόσια Επιχείρηση Ηλεκτρισμού"> Δ.Ε.Η </abbr>
```

- **br**: Εισαγωγή αλλαγή γραμμής.

```
<p>Γραμμή 1<br/>
Γραμμή 2
</p>
```

- **cite**: Περιέχει μια αναφορά σε άλλες πηγές.

```
<cite cite="http://www.example.com/books/book.html">The Two Towers</cite>
```

- **code**: Ορίζει ένα τμήμα κώδικα σε μία πρόταση.

```
<p> Η εντολή αύξησης μιας αέρας μεταβλητής κατά ένα στη C++ <code> i = i + 1
</code> μπορεί να πραγματοποιηθεί με την εντολή <code> i++ </code>. </p>
```

- **dfn**: Περιέχει την περιγραφή ενός όρου

```
<p> Ένα <dfn id="def-acronym">Ακρωνύμιο</dfn> είναι μια λέξη που σχηματίζεται από
τα αρχικά μια ομάδας λέξεων</p>
```

- **em**: Δίνει έμφαση στο κείμενο. Υποστηρίζονται αντίστοιχα εμφωλιασμένα em. Τα εμφωλιασμένα em στοιχεία φανερώνουν μεγαλύτερη σπουδαιότητα.

```
Η προσφορά ισχύει <em> έως 31/12/2013</em>
```

- **kbd**: Καθορίζει ότι η είσοδος δεδομένων θα πραγματοποιηθεί από τον χρήστη

Πληκτρολογήστε `<kbd>ONOMA</kbd>`

- **span**: Το στοιχείο span, σε συνδυασμό με τα @id, @class, @role, προσφέρει ένα γενικό μηχανισμό διάρθρωσης στα έγγραφα.

`<p>Το κείμενο από εδώ <span class="note">αποτελεί μία σημείωση</span></p>`

- **strong**: Καταδεικνύει μεγαλύτερη σπουδαιότητα στο περιεχόμενο.

Την `<strong>`Δευτέρα`</strong>`θα γίνει συνάντηση.

- **sub**: Το κείμενο θα παρουσιαστεί σαν δείκτης.

X`<sub>`ij`</sub>`

- **sup**: Το κείμενο θα παρουσιαστεί σαν εκθέτης.

E=mc `<sup>`2`</sup>`

- **var**: Δείχνει ότι το κείμενο αποτελεί μια μεταβλητή προγράμματος.

Η παράμετρος `<var>`ncols`</var>` επιστρέφει τον αριθμό των γραμμών.

### 3.2.4. Υπερκείμενο

Η ενότητα υπερκειμένου, παρέχει ένα μόνο στοιχείο που έχει χρησιμοποιηθεί στην HTML, για να καθορίσει συνδέσμους υπερκειμένου σε άλλες πηγές. Το στοιχείο αυτό είναι το a και καθορίζει μια άγκυρα.

`<a href="http://www.example.gr">`Example Site`</a>`

### 3.2.5. Λίστες

Η XHTML προσφέρει αρκετούς μηχανισμούς για τη δημιουργία λιστών. Οι λίστες μπορεί να περιλαμβάνουν.

- Αταξινόμητη πληροφορία
- Ταξινομημένη πληροφορία
- Ορισμούς.

Η αταξινόμητες λίστες προσδιορίζονται με το στοιχείο **ul**, ενώ οι ταξινομημένες με το **ol**, όμως και οι δύο τύποι λίστες χρησιμοποιούν το στοιχείο **li** για τον καθορισμό των στοιχείων τους.

```
<ul>
  <li>Στοιχείο 1</li>
  <li>Στοιχείο 2</li>
  <li>Στοιχείο 3</li>
  <li>Στοιχείο 4</li>
</ul>
```

Οι λίστες ορισμού, χρησιμοποιούν το στοιχείο **dl**. Ο ορισμός αποτελείται από το στοιχείο **dt**, ενώ η περιγραφή με το **dd**.

```
<dl>
  <dt>Ορισμός 1</dt>
  <dd>Περιγραφή ορισμού 1</dd>
  <dt>Ορισμός 2</dt>
  <dd>Περιγραφή ορισμού 2</dd>
</dl>
```

### 3.2.6. Ενότητα ιδιοτήτων πυρήνα

Σε αυτή την ενότητα καθορίζονται οι ενσωματωμένες ιδιότητες της XHTML 2.0 και παρουσιάζονται στον πίνακα που ακολουθεί:

Πίνακας 0-1 Ιδιότητες Πυρήνα

Ιδιότητα	Περιγραφή	Χρήση
<b>@class</b>	Αναθέτει ένα ή περισσότερα ονόματα κλάσης σε ένα στοιχείο.	Για τον ορισμό στυλ μέσω φύλλων στυλ.
<b>@id</b>	Αναθέτει ένα μοναδικό αναγνωριστικό στο στοιχείο.	Για τον ορισμό στυλ, σαν στόχος για συνδέσμους, σαν όνομα ενός αντικειμένου
<b>@layout</b>	Καθορίζει αν ο χαρακτήρας κενό έχει σημασία για το κείμενο.	
<b>@title</b>	Καθορίζει την μετά-πληροφορία ενός στοιχείου	Κυρίως για φιλικότητα ως προς τις μηχανές αναζήτησης (SEO).

### 3.2.7. Πρόσβαση

Το στοιχείο πρόσβασης εκχωρεί μια σχέση (χαρτογραφεί), μεταξύ κλειδιών, ή γεγονότων σε ένα έγγραφο. Η ενεργοποίηση αυτής της σχέσης χρησιμοποιείται για

την εστίαση στα στοιχεία που παίρνουν μέρος στη σχέση ή για να πυροδοτήσει κάποιο γεγονός.

Ένα στοιχείο πρόσβασης πρέπει να έχει είτε ρόλο (*targetrole*), είτε μια καθορισμένη ιδιότητα (*targetid attribute*). Το στοιχείο πρόσβασης επιτρέπει σε ένα συγγραφέα να καθορίσει μια σχέση μεταξύ κάποιων στοιχείων σε ένα έγγραφο.

Οι ιδιότητες ενός στοιχείου πρόσβασης είναι:

- **activate**: Παίρνει τις τιμές TRUE:FALSE και αντίστοιχα ενεργοποιεί ή απενεργοποιεί τη σχέση.
- **key**: Παρέχει ένα ή περισσότερα κλειδιά για την χαρτογράφηση της σχέσης.
- **order**: Καθορίζει την σειρά με την οποία θα προσπελαστούν τα στοιχεία της σχέσης.
- **targetid**: Καθορίζει ένα ή περισσότερα αναγνωριστικά σχετικά με τα στοιχεία.

### 3.2.8. Ιδιότητες ενσωματωμένου περιεχομένου

Αυτή η συλλογή ιδιοτήτων, προκαλεί την ενσωμάτωση ενός απομακρυσμένου πόρου, στο έγγραφο, στη θέση του περιεχομένου του στοιχείου. Αν η πρόσβαση στον απομακρυσμένο πόρο αποτύχει, τότε επεξεργάζεται μόνο το περιεχόμενο του στοιχείου.

Τα χαρακτηριστικά αυτής της συλλογής είναι:

- **encoding**: Καθορίζει την επιτρεπτή κωδικοποίηση του εξωτερικού πόρου (π.χ. “utf-8”, “utf-16” κλπ).
- **src**: Αναφέρεται στη τοποθεσία του εξωτερικού πόρου για τα περιεχόμενα του στοιχείου.
- **srctype**: Αυτή η ιδιότητα καθορίζει τον επιτρεπτό τύπο του πόρου που αναφέρεται στην ιδιότητα src (π.χ image/png, image/jpeg κλπ).

### 3.2.9. Εικόνες

Η ενότητα παρέχει βασικές οδηγίες για την ενσωμάτωση εικόνων. Η ενσωμάτωση παρέχεται με το στοιχείο `img`. Με το στοιχείο αυτό μπορούν να ενσωματωθούν χαρακτηριστικά, όπως π.χ το `@src`. Το παράδειγμα που ακολουθεί έχει σαν αποτέλεσμα την εμφάνιση του λογότυπου. Στη περίπτωση που δεν υπάρχει το λογότυπο, τότε στη θέση του εμφανίζεται το κείμενο.

```
W3C</img>
```

### 3.2.10. Χάρτες Εικόνων

Με τους χάρτες ορίζεται μια συλλογή χαρακτηριστικών που προσδιορίζουν μια ενσωματωμένη εικόνα, με τέτοιο τρόπο, έτσι ώστε διάφορα τμήματα της εικόνας να είναι υπερ-σύνδεσμοι. Τα χαρακτηριστικά αυτής της συλλογής είναι:

- **usemap**: Αντιστοιχίζει ένα χάρτη εικόνας με το στοιχείο ul (unordered list). Η τιμή του usemap, πρέπει να ταιριάζει με την τιμή του χαρακτηριστικού @id, ενός στοιχείου ul που περιέχει ένα ή περισσότερα στοιχεία li με ιδιότητες @shape και @coords.
- **ismap**: Δείχνει ότι ο σχετικός ενσωματωμένος πόρος, θα πρέπει να αντιμετωπιστεί σαν ένας χάρτης εικόνας στη πλευρά του διακομιστή. Όταν επιλεγθεί ένα τμήμα του, οι συντεταγμένες μαζί με το επιλεγμένο στοιχείο στέλνονται στο διακομιστή όπου βρίσκεται το έγγραφο.
- **shape**: Μπορεί να πάρει τιμές από ένα σύνολο (default|rect|circle|poly) και καθορίζει μια περιοχή, η οποία αποτελεί τμήμα του χάρτη εικόνας.
- **Cords**: Καθορίζει τη θέση και το σχήμα μιας περιοχής. Ο αριθμός και η σειρά των τιμών εξαρτάται από την τιμή της ιδιότητας @shape (π.χ **rect**: left-x, top-y, right-x, bottom-y|**circle**:center-x, **center**-y, radius.). Οι συντεταγμένες ορίζονται σε σχέση με την αριστερή κορυφή του αντικειμένου.

### 3.2.11. Πολυμέσα

Αυτή η ενότητα καθορίζει την ιδιότητα media. Η τιμή αυτής της ιδιότητας αποτελείται από μία περιγραφική λίστα διαχωρισμένη με κόμματα. Όταν η τιμή της ιδιότητας ταιριάζει με τα τρέχοντα διαθέσιμα μέσα επεξεργασίας, τότε το στοιχείο θεωρείται ενεργό και η γίνεται αντίστοιχα η επεξεργασία, αλλιώς αγνοείται.

```
<style src="style.css" type="text/css" media="screen" />  
<span src="photo.jpg" media="screen">Photo Title</span>
```

### 3.2.12. Μεταπληροφορία

Η μεταπληροφορία καθορίζει στοιχεία που επιτρέπουν τον ορισμό σχέσεων. Οι σχέσεις αυτές αφορούν:

- Το ίδιο το έγγραφο

- Στοιχεία εξωτερικά του εγγράφου
- Άλλα στοιχεία που αφορούν μεταδεδομένα του εγγράφου.

Τα στοιχεία της ενότητας είναι:

- **link**: Καθορίζει ένα σύνδεσμο. Ο σύνδεσμος μεταφέρει πληροφορίες για τις σχέσεις στους πράκτορες (φυλλομετρητές) με διάφορους τρόπους (π.χ με ένα μενού επιλογών drop-down που αποτελείται από links). Οι πράκτορες θα πρέπει να επιτρέπουν την ενεργοποίηση αυτών των συνδέσμων

```
<head>
<title>Chapter 2</title>
<link rel="index" href="../index.html"/>
</head>
```

- **meta**: Καθορίζει τις ιδιότητες ενός εγγράφου (π.χ. συγγραφέας, ημερομηνία λήξης, λέξεις κλειδιά κλπ). Κάθε μετά-στοιχείο καθορίζεται από το ζεύγος ιδιότητα/τιμή. Η ακόλουθη δήλωση αφορά τον συγγραφέα του εγγράφου.

```
<meta property="author">Steven Pemberton</meta>
```

Τόσο ο σύνδεσμοι, όσο και η μεταπληροφορία, αποτελούν σημαντικό κομμάτι του εγγράφου για τις μηχανές αναζήτησης. Οι σύνδεσμοι παρέχουν, εναλλακτικές εκδόσεις του εγγράφου, αλλά και ένα σύστημα πλοήγησης σε άλλα έγγραφα. Από την άλλη μέσω της μεταπληροφορίας, καθορίζονται λέξεις κλειδιά, με σκοπό την βελτίωση των αποτελεσμάτων αναζήτησης.

### 3.2.13. Φύλλα Στυλ

Για την ενσωμάτωση φύλλων στυλ, χρησιμοποιείται το στοιχείο style. Αντίστοιχα οι ιδιότητες του στοιχείου είναι:

- **disabled**: Αν τεθεί “disabled”, τότε απενεργοποιείται το φύλλο στυλ.
- **media**: Καθορίζει το τύπο του μέσου που αφορά το στυλ.

Τα φύλλα στυλ, μπορούν είτε να ενσωματώνονται σε ένα XHTML2.0 έγγραφο, είτε να παραπέμπουν σε εξωτερικό αρχείο στυλ. Τα παραδείγματα που ακολουθούν δείχνουν τις δύο προσεγγίσεις.

```
<style type="text/css">
  h1 {border-width: thin; border-style: solid; text-align: center}
</style>
```



```
<style src="style.css" type="text/css" media="screen" />
```

### 3.2.14. Πίνακες

Οι πίνακες παρέχουν τα κατάλληλα στοιχεία για την ενσωμάτωση δεδομένων σε στήλες και γραμμές. Τα στοιχεία των πινάκων για την XHTML 2.0 είναι:

- **table**: Είναι το αρχικό στοιχείο δήλωση του πίνακα και περιέχει όλα τα άλλα στοιχεία καθώς και το περιεχόμενο.
- **summary**: Προσθέτει μια περιγραφή του σκοπού και της δομής του πίνακα για μη οπτική χρήση από τα προγράμματα πελάτη, όπως εκφώνηση ή σύστημα Braille.
- **col**: Επιτρέπει στους συγγραφείς να μοιράζονται ιδιότητες μεταξύ διαφορετικών στηλών, χωρίς να ομαδοποιούν τις στήλες. Μπορεί δηλαδή οι στήλες μεταξύ τους να έχουν διαφορετικό χρώμα ανάλογα με τα δεδομένα που έχουν.
- **colgroup**: Επιτρέπει στους συγγραφείς να δημιουργούν δομημένες υποδιαίρεσεις σε ένα πίνακα. Ουσιαστικά ομαδοποιεί τις στήλες σε λογικές ενότητες.
- **thead, tfoot, tbody**: Δημιουργεί μία ομάδα γραμμών στην επικεφαλίδα, στο υποσέλιδο ή στο κυρίως σώμα αντίστοιχα. Κάθε ομάδα πρέπει να περιλαμβάνει τουλάχιστον μία γραμμή.
- **tr**: Δημιουργία γραμμής
- **td**: Δημιουργία κελιού
- **th**: Δημιουργία του κελιού που περιέχει την επικεφαλίδα του πίνακα

Οι ιδιότητες των στοιχείων των πινάκων είναι:

- **span**: Αφορά τα στοιχεία `col`, `colgroup` και καθορίζει τον αριθμό των στηλών που ομαδοποιούνται.
- **abbr**: Χρησιμοποιείται για να παρέχει μια συντομογραφία των περιεχομένων των κελιών.

- **axis:** Μπορεί να χρησιμοποιηθεί για να τοποθετήσει τα κελιά σε λογικές κατηγορίες.
- **scope:** Καθορίζει το σύνολο των δεδομένων των κελιών για το οποίο η τρέχουσα επικεφαλίδα παρέχει πληροφορίες.
- **headers:** Καθορίζει μια λίστα κελιών που αποτελούν τις επικεφαλίδες για στήλες ή γραμμές.

### 3.2.15. Φόρμες

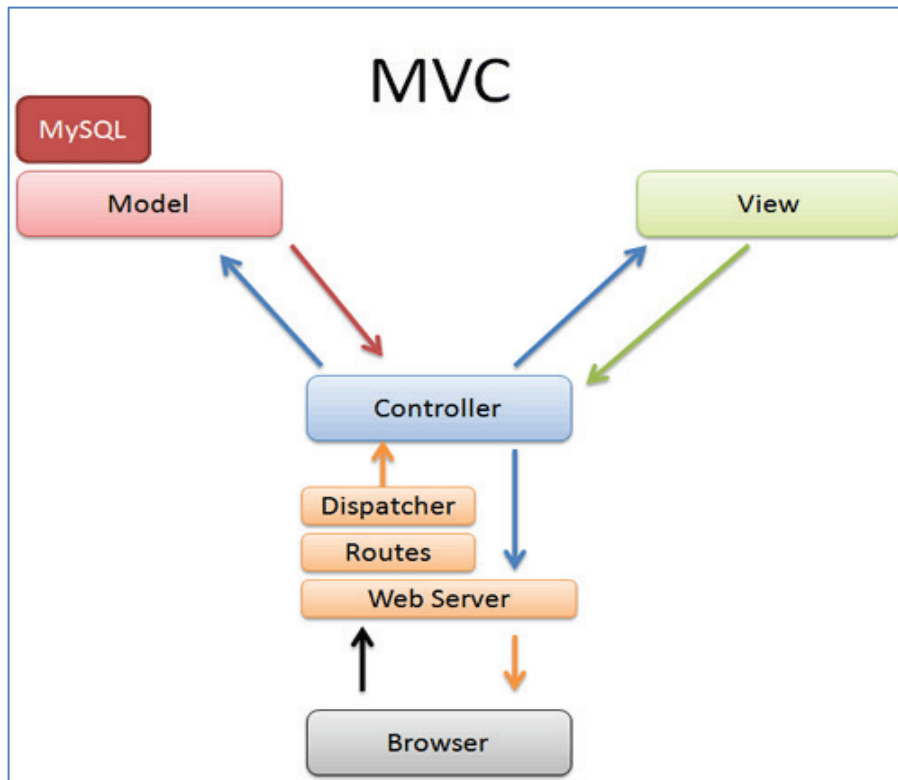
Οι φόρμες που χρησιμοποιούνται από την XHTML 2.0 καλούνται XForms και αναπτύσσονται ξεχωριστά. Οι φόρμες χρησιμοποιούν την XML και το μοντέλο που περιγράφεται σε αυτή τόσο για την παρουσίαση, όσο και για το χειρισμό των δεδομένων. Σχεδιάστηκαν να αντικαταστήσουν τις HTML φόρμες και μπορούν να χρησιμοποιηθούν και από τις δύο γλώσσες σήμανσης. Οι XForms ακολουθούν το μοντέλο MVC.

Το μοντέλο MVC αποτελείται όπως φαίνεται από την εικόνα 1 από τις εξής ενότητες:

- **Model:** Αναπαριστά τα δεδομένα της εφαρμογής
- **View:** Παράγει μια παρουσίαση των δεδομένων του μοντέλου
- **Controller:** Διαχειρίζεται και κατευθύνει τα αιτήματα των χρηστών

Η διαδικασία που πραγματοποιείται όταν δημιουργείται ένα αίτημα, δηλαδή η ανάκτηση μιας σελίδας συνοπτικά είναι:

- Ο φυλλομετρητής στέλνει ένα αίτημα για την ανάκτηση μιας σελίδας.
- Ο εξυπηρετητής (Web Server) λαμβάνει το αίτημα και το δρομολογεί.
- Στη συνέχεια ο controller δίνει τις κατάλληλες εντολές, για να ανακτηθεί η αντίστοιχη πληροφορία, μέσω του model.
- Το model αναλαμβάνει να προσπελάσει την βάση για την ανάκτηση της πληροφορίας.
- Τέλος το view είναι υπεύθυνο για την παρουσίαση της ανακτημένης πληροφορίας στην κατάλληλη μορφή



Εικόνα 0-1 Μοντέλο MVC

(πηγή: Intermediate Rails: Understanding Models, Views and Controllers, 2011)

Τα στοιχεία μιας φόρμας XForm, δεν υποστηρίζουν άμεσα τον έλεγχο κατά την καταχώριση. Παρ' όλα αυτά όμως μπορούν να χρησιμοποιηθούν κανονικές εκφράσεις (regular expressions), για τον έλεγχο των δεδομένων κατά την είσοδο. Η έκφραση αυτή ορίζεται μέσω της ιδιότητας pattern. Έτσι για παράδειγμα η είσοδος μιας ηλεκτρονικής διεύθυνσης ορίζεται με τον κώδικα που ακολουθεί.

```
<xs:simpleType name="email">
  <xs:restriction base="xsd:string">
    <xs:pattern value="([A-Za-z0-9!#-'\*\+\-\/=\?\^\_`\{-~}]+)(\.[A-Za-z0-9!#-'\*\+\-\/=\?\^\_`\{-~}]+)*@[A-Za-z0-9!#-'\*\+\-\/=\?\^\_`\{-~}]+(\.[A-Za-z0-9!#-'\*\+\-\/=\?\^\_`\{-~}]+)*" />
  </xs:restriction>
</xs:simpleType>
```

Αντίστοιχα με την χρήση διαφορετικών εκφράσεων, μπορούμε να ορίσουμε αντίστοιχους ελέγχους για άλλες μορφές, όπως πιστωτικές κάρτες, ημερομηνία, αριθμούς κλπ.

Η έκδοση που αναπτύσσεται αυτή τη στιγμή από το W3, είναι η XForms 2.0 που προσθέτει αρκετά νέα χαρακτηριστικά. Η παρουσίαση των πολυάρθρων

χαρακτηριστικών, ξεφεύγει από το σκοπό της εργασίας, όπου θα αναφερθούμε συνοπτικά στα κυριότερα χαρακτηριστικά, τα οποία είναι:

- Δομή εγγράφου Xforms
- Εκφράσεις
- Τύποι δεδομένων
- Ιδιότητες μοντέλου
- Ενότητα υποβολής
- Χειριστές ελέγχου (form controls)
- Χειριστές ενεργειών (actions)

### 3.2.16. Αντικείμενα

Η συμπερίληψη αντικειμένου σε ένα έγγραφο XHTML2 περιγράφεται σε αυτή την ενότητα. Τα ενσωματωμένα αντικείμενα μπορεί να είναι εικόνες, πολυμέσα, ακόμη όμως και εκτελέσιμος κώδικας. Η ενότητα αντικειμένου υποστηρίζει τα εξής στοιχεία και ιδιότητες:

Πίνακας 0-2 Στοιχεία και ιδιότητες αντικειμένων

Στοιχείο	Περιγραφή	Ιδιότητες
<b>object</b>	Το αντικείμενο	archive: καθορίζει μία λίστα που περιέχει τους πόρους που είναι σχετικοί με το αντικείμενο
		content-length: Το μέγεθος του αντικειμένου για την επεξεργασία των δεδομένων από τον χειριστή του αντικειμένου
		declare: Δηλώνει μόνο το αντικείμενο. Αυτό θα χρησιμοποιηθεί αν κληθεί μέσω ενός γεγονότος χρήστη.
<b>Param</b>	Παράμετροι αντικειμένου	name: Καθορίζει το όνομα της παραμέτρου.
		Value: Καθορίζει τη τιμή της παραμέτρου
		valuetype: Καθορίζει το τύπο της παραμέτρου (data ref object).
<b>Standby</b>	Καθορίζει ένα μήνυμα που θα εμφανίζεται όσο το αντικείμενο φορτώνεται για εκτέλεση.	

### 3.2.17. XHTML 2.0 Events

Σε αυτή την ενότητα καθορίζεται τα είδη και ο τύπος των συμβάντων σε ένα XHTML 2.0 έγγραφο. Η συλλογή αυτή περιέχει το στοιχείο listener που υποστηρίζει τις εξής ιδιότητες:

- **event**: Καθορίζει τον τύπο συμβάντος για το οποίο δεσμεύεται ο listener
- **observer**: Καθορίζει το αναγνωριστικό του συμβάντος για το οποίο δεσμεύεται ο listener.
- **targeted**: Καθορίζει το αναγνωριστικό του στοιχείου που προκάλεσε το συμβάν.
- **handler**: Περιέχει την αναφορά του πόρου που καθορίζει την ενέργεια που θα πρέπει να γίνει.
- **phase**: Καθορίζει αν το listener, θα ενεργοποιηθεί βάσει κάπου άλλου συμβάντος.
- **propagate**: Αν συμβαίνουν διαδοχικά συμβάντα τότε μπορεί αυτά είτε να απενεργοποιηθούν, είτε να ενεργοποιηθούν ξανά με τις οδηγίες stop και continue αντίστοιχα.
- **defaultAction**: Καθορίζει αν μετά την επεξεργασία των συμβάντων θα εκτελεστεί μια προκαθορισμένη ενέργεια ή όχι.

### 3.2.18. XHTML 2.0 Handlers

Αντίστοιχα με την ενότητα των συμβάντων, η ενότητα των χειριστών, καθορίζει στοιχεία τα οποία χρησιμοποιούνται για την παροχή πληροφοριών σε γεγονότα, κυρίως σε γλώσσες σεναρίου. Αυτά είναι:

- **action**: Χρησιμοποιείται για την ομαδοποίηση χειριστών για ένα συμβάν.
- **script**: Περιέχει τις αναφορές σεναρίου, για την ενεργοποίηση ενός ή περισσότερων χειριστών συμβάντων για ένα έγγραφο, μέσω της γλώσσας σεναρίου που χρησιμοποιείται.
- **dispatchEvent**: Πυροδοτεί την εκτέλεση του συμβάντος.
- **addEventListener**: Επιτρέπει την δέσμευση ενός listener για το συγκεκριμένο γεγονός.
- **removeEventListener**: Αποδεσμεύει τον listener.
- **stopPropagation**: Σταματάει την εκτέλεση ενός γεγονότος.

### 3.2.19. XFrames

Τα XFrames βασίζονται στην XML μορφοποίηση για τον συνδυασμό και οργάνωση εγγράφων διαδικτύου μέσω της χρήσης πλαισίων. Τα XFrames μπορούν να χρησιμοποιηθούν για την δημιουργία περιεχομένου που μπορεί να κυλάει, ενώ τα άλλα μέρη να είναι σταθερά (π.χ. το κυρίως περιεχόμενο να κυλάει, ενώ ένα μενού επιλογών στα αριστερά να είναι σταθερό). Οι αλλαγές που έχουν πραγματοποιηθεί στην γενιάς των XFrames που υλοποιείται στην XHTML 2.0, έχουν να κάνουν κυρίως, με προβλήματα που υπήρχαν στο παραδοσιακό σύστημα πλαισίου.

Τα συστατικά που αναφέρονται στο προσχέδιο του W3 είναι:

- **frames:** Το στοιχείο frames, σχηματίζει τον υποδοχέα για το υπό σύνθεση έγγραφο.
- **head:** Περιέχει μεταδεδομένα που αφορούν το πλαίσιο.
- **title:** Το τίτλο του πλαισίου.
- **style:** Περιέχει πληροφορίες σχετικά με το στιλιστικές οδηγίες που αφορούν την παρουσίαση του πλαισίου.
- **group:** Καθορίζει μια σειρά από ομάδες πλαισίων ή από απλά πλαίσια. Τόσο οι ομάδες όσο και τα μοναδικά πλαίσια τοποθετούνται μαζί, σύμφωνα με το στυλ που έχει εφαρμοσθεί σε αυτά.

και οι αντίστοιχες ιδιότητες

- **compose:** Παίρνει τις τιμές vertical, horiztal, single και παρέχει οδηγίες στον πελάτη του χρήση για την σύνθεση των πλαισίων στην οπτική διάταξη.
- **xml-id:** Αντιστοιχεί ένα αναγνωριστικό σε ένα στοιχείο του πλαισίου.
- **class:** Αυτή η ιδιότητα, εκχωρεί μία κλάση ή ένα σύνολο κλάσεων σε ένα στοιχείο του πλαισίου.
- **title:** Παρέχει πληροφορίες για το στοιχείο που θα χρησιμοποιηθεί.

Τα δύο παραδείγματα που ακολουθούν δείχνουν αντίστοιχες διατάξεις πλαισίων, καθώς και τον κώδικα που θα πρέπει να χρησιμοποιηθεί.

Πίνακας 0-3 Σχέδιο Πλαισίων

Κώδικας	Σχέδιο (layout)
---------	-----------------

<pre>&lt;group compose="horizontal"&gt;   &lt;group compose="vertical"&gt;     &lt;frame/&gt;     &lt;frame/&gt;   &lt;/group&gt;   &lt;frame/&gt; &lt;/group&gt;</pre>		

<pre>&lt;group compose="vertical"&gt;   &lt;group compose="horizontal"&gt;     &lt;frame/&gt;     &lt;frame/&gt;   &lt;/group&gt;   &lt;frame/&gt; &lt;/group&gt;</pre>		

### 3.2.20. Άλλες ενότητες

Υπάρχουν αντίστοιχα με τις ενότητες που εξετάσαμε και άλλες ενότητες που αφορούν την XHTML 2.0. Αυτές κυρίως εκτελούν συμπληρωματικό έργο, υποστηρίζοντας άλλες ενότητες. Αυτές είναι:

- Scripting:
- Legacy
- Ruby
- Role
- I18N
- Bi-Directional
- XF

## Σύγκριση

Η μεταβατική περίοδος στο διαδίκτυο, όπου οι μεγάλοι ιστότοποι, είχαν φτάσει τις τεχνολογίες στα όρια τους έφερε την επιτακτική ανάγκη ανάπτυξης νέων πρωτοτύπων. Ενώ το W3C ξεκίνησε την νέα έκδοση της XHTML 2.0, παράλληλα κάποιοι ανεξάρτητοι κατασκευαστές αποφάσισαν ότι χρειάζεται κάτι πιο ολοκληρωμένο και προχώρησαν στην ανάπτυξη της HTML V5. Ενώ τα δύο πρωτόκολλα εμφανίζουν κάποιες ομοιότητες, που αφορούν κυρίως τα στοιχεία που χρησιμοποιούνται, από την άλλη οι διαφορές είναι σημαντικές τόσο ως προς την φιλοσοφία, αλλά και σε άλλες παραμέτρους.

### Η φιλοσοφία πίσω από την XHTML V2

Η πρώτη και σημαντικότερη σχεδιαστική φιλοσοφία πίσω από την XHTML 2.0 είναι ο διαχωρισμός μεταξύ του περιεχομένου και της παρουσίασης, βελτιώνοντας μερικά χαρακτηριστικά της HTML V4 και της XHTML V1. Για παράδειγμα, η ενσωματωμένη υποστήριξη για τον καθορισμό μιας αναφοράς εικόνας σε κάθε στοιχείο μιας μη διαταγμένης λίστας. Η παλιά ετικέτα `img src`, αντικαταστάθηκε από την ελεύθερη επιλογή καθορισμού του `src` σε οποιοδήποτε στοιχείο και όχι μόνο σε συνδυασμό με το `img`. Με αυτή την αλλαγή, το φύλα `στυλ CSS` απλοποιούνται και εναλλακτικές συσκευές μπορούν εύκολα να αντικαταστήσουν την εικόνα με κείμενο για λόγους οικονομίας πόρων.

Πολύς όμως χρόνος είχε σπαταληθεί για την αλληλεπίδραση των HTML φορμών με τους εξυπηρετητές, χρησιμοποιώντας πολλές γραμμές κώδικα Javascript. Οι φόρμες υπάκουαν στον περιορισμό του μονοδιάστατου ζευγαριού κλειδιού-τιμής. Από την άλλη η υλοποίηση κώδικα Javascript, απαιτεί μεγάλη προσπάθεια, η οποία καταλήγει στο κάλαθο των αχρήστων, όταν υπάρχουν εναλλακτικές διεπιφάνειες όπως για παράδειγμα συσκευές κειμένου-σε-ομιλία (text-to-speech devices).

Παράλληλα με την τμηματική προσέγγιση (modular), η XHTML 2.0 αντικαθιστά τις HTML φόρμες με την ενότητα XForms, προσθέτοντας υποστήριξη για κοινά προβλήματα όπως: λογική αλληλεπίδραση χρήστη-φόρμας, κανόνες επικύρωσης και υπολογισμούς χωρίς την παρουσία ούτε μιας γραμμής κώδικα. Επίσης η τεχνική χρησιμοποιεί πλούσιες XML δομές, αντί τα ζευγάρια κλειδιών-τιμών, επιτρέποντας φωλιασμένες υπό-φόρμες και επαναλαμβανόμενα στοιχεία.



Επιπρόσθετα στις XForms, αρκετά άλλα στοιχεία προσαρμόστηκαν σε δικές τους ενότητες και προδιαγραφές, έτσι ώστε να επαναχρησιμοποιηθούν για άλλους σκοπούς, όπως XML γεγονότα, XFrames, και Ruby.

Με τον διαχωρισμό της παρουσίασης, η προγραμματιστικότητα (programmability), επίσης διαχωρίστηκε από το πρότυπο. Οι ιδιότητες αλληλεπίδρασης όπως π.χ onClick, αντικαταστάθηκαν από την ενότητα των XML Γεγονότων. Επειδή οι προδιαγραφές αυτών των γεγονότων ορίστηκαν για αυτό τον σκοπό, η ενότητα αυτή παρέχει ένα ισχυρό σύνολο εργαλείων για τον χειρισμό της αλληλεπίδρασης με το χρήστη.

Συνοψίζοντας τις καινοτομίες της XHTML V2, θα λέγαμε ότι η βασική φιλοσοφία είναι ο διαχωρισμός των απαιτήσεων. Κάθε απαίτηση δεν αποτελεί πλέον ένα δεύτερο χαρακτηριστικό της HTML, αλλά ένα πρωταρχικό στόχο για τον ορισμό μιας νέας προδιαγραφής. Ως εκ τούτου, οι νέες προδιαγραφές είναι βελτιστοποιημένες για την πλευρά του δημιουργού. Ένας έξυπνος προγραμματιστής που διαθέτει τεχνικές δεξιότητες για να εκμεταλλευτεί το σύνολο των εργαλείων είναι σε θέση να πετύχει εντυπωσιακά αποτελέσματα.

## **Η φιλοσοφία πίσω από την HTML 5**

Η φιλοσοφία «αυτό που βλέπεις είναι αυτό που παίρνεις» (WHATWG: What you see is What you Get), προσεγγίζεται περισσότερο πρακτικά από ποτέ στο σχεδιασμό της HTML5. Αντί της αφηρημένης προσέγγισης της XHTML 2.0, η ομάδα ανάπτυξης άρχισε να τεκμηριώνει τη πραγματική συμπεριφορά των προγραμμάτων περιήγησης, η οποία διέφερε από τις προδιαγραφές που είχε θέσει εξ' αρχής το W3C. Χρησιμοποιώντας την ανάλυση ως θεμέλιο, η ομάδα διερευνά πως η HTML χρησιμοποιείται στην πράξη. Με αυτή την φιλοσοφία, η ομάδα του W3C, πρότεινε χαρακτηριστικά, τα οποία θεωρεί ότι θα διευκολύνει τον μέσο Web κατασκευαστή.

Η κατασκευή Web εφαρμογών έχει απλοποιηθεί με την προσθήκη χαρακτηριστικών όπως πλέγματα δεδομένων, μενού και εργαλειοθήκες. Η χρήση αυτών των νέων στοιχείων οδηγεί σε απλοποίηση κατά την κατασκευή και τη συγγραφή λιγότερου Javascript κώδικα.

Οι προδιαγραφές της HTML 5 δεν περιορίζονται μόνο στην παρουσίαση στοιχείων και ιδιοτήτων. Καθορίζουν ένα Javascript Api για εξειδικευμένους σκοπούς όπως η επεξεργασία των κειμένων και η χρήση «σύρε κι άσε» (drag n drop) αλληλεπίδρασης.

Αυτή η προσέγγιση είναι διαμετρικά αντίθετη με την λογική των ενοτήτων που χρησιμοποιεί η XHTML 2.0. Απλοποιεί τις διαδικασίες για τους δημιουργούς σελίδων, αλλά αυξάνει το μέγεθος του ορισμού των προδιαγραφών σημαντικά. Αυτός είναι κυρίως ο λόγος για τον οποίο οι προδιαγραφές ορίζονται σε μικρότερες ομάδες όπως αναφέραμε στο 2<sup>ο</sup> κεφάλαιο.

Η HTML 5 είναι περισσότερο σχετική με την HTML 4, παρά απ' ότι η XHTML V2 με την XHTML V1. Η μετάβαση είναι ομαλότερη και είναι ευκολότερο για έναν HTML V4 κατασκευαστή να μπορέσει να κατανοήσει την καινούργια έκδοση. Ακόμη και τα νέα χαρακτηριστικά ακολουθούν την ίδια λογική με τα παλαιά.

Οι τρέχουσες διαδικτυακές εφαρμογές βασίζονται σε ασύγχρονη τεχνολογία (Javascript + XML (Ajax) για αλληλεπίδραση με τον εξυπηρετητή. Η HTML 5 αναγνωρίζει την σπουδαιότητα αυτή και παρέχει αρκετούς τρόπους επικοινωνίας σε ένα δίκτυο και αποστολής μηνυμάτων σε διαφορετικούς τομείς χωρίς περιορισμούς σε θέματα ασφάλειας.

Η κύρια φιλοσοφία πίσω από την HTML 5 είναι η επέκταση της HTML V4 με πρακτικά χαρακτηριστικά, που ένας μέσος δημιουργός σελίδων Web αναζητά. Επίσης απλοποιεί την τεχνολογία όπως αυτή παρουσιάζεται στην HTML V4, επεκτείνοντάς όμως τις δυνατότητές της σε σημαντικό βαθμό.

### **Σημασιολογική Σήμανση**

Οι δύο ομάδες ανάπτυξης, έχουν σαφώς υιοθετήσει την σημασιολογική σήμανση, επιτρέποντας στους προγραμματιστές την ενσωμάτωση μεταδεδομένων στα έγγραφά τους. Όπως με τις φόρμες η ομάδα της XHTML 2, έχει υιοθετήσει μια πιο εξελιγμένη τεχνολογία, ενώ από την άλλη η ομάδα HTML 5 έχει αφήσει τα πράγματα απλά. Στην XHTML 2, τα μεταδεδομένα μπορούν να ενσωματωθούν, χρησιμοποιώντας αρκετές ιδιότητες από την αντίστοιχη ενότητα. Συγκεκριμένα η ιδιότητα “role”, προορίζεται για να περιγράψει το νόημα ενός στοιχείου στο περιεχόμενο ενός εγγράφου. Συνεπώς ένα στοιχείο μπορεί να αποτελεί πραγματικό περιεχόμενο το οποίο εμφανίζεται στο πρόγραμμα πελάτη, ή να είναι απλή μεταπληροφορία.

Η ομάδα ανάπτυξης της HTML, έχει ακολουθήσει μια διαφορετική προσέγγιση, υπερφορτώνοντας την ιδιότητα “class”, με ένα καθορισμένο σύνολο δεσμευμένων κλάσεων που αναπαριστούν διάφορους τύπους δεδομένων. Αυτή τη στιγμή η προδιαγραφή παραθέτει επτά δεσμευμένες κλάσεις: copyright, error, example, issue,

note, search και warning. Η υπερφόρτωση της ιδιότητας “class”, μπορεί να οδηγήσει σε σύγχυση, καθώς οι πελάτες (φυλλομετρητές), μπορεί να αναπαράγουν αυτή την πληροφορία διαφορετικά.

### Στοιχεία γλωσσών σήμανσης

Ένας μεγάλος αριθμός στοιχείων έχουν απομακρυνθεί τόσο από την XHTML V2, όσο και στην HTML5, επειδή θεωρήθηκαν στοιχεία που αφορούσαν αποκλειστικά την παρουσίαση, ενώ η πρακτική που ακολουθήθηκε θέλει την παρουσίαση να ορίζεται στα φύλλα στυλ.

Τα έγγραφα HTML 5 και XHTML 2.0 δεν μπορούν να υποστηρίξουν τέτοια στοιχεία όπως: basefont, bi, font, s, strike, tt και u. Η XHTML 2.0 επίσης απομάκρυνε τα στοιχεία small, b, i και hr, ενώ η HTML5 τα επανακαθορίζει, έτσι ώστε να μην αφορούν την παρουσίαση. Στην XHTML 2.0, το hr στοιχείο αντικαστάθηκε με το separator, σε μια προσπάθεια να αποφευχθεί η σύγχυση, καθώς το στοιχείο hr, που ορίζεται ως οριζόντιος κανόνας (horizontal rule), δεν ανταποκρίνεται σε αυτή την ερμηνεία.

Ο πίνακας που ακολουθεί εμφανίζει τα στοιχεία που υποστηρίζονται από τις δύο γλώσσες, τα κοινά τους στοιχεία, καθώς και την περιγραφή:

Πίνακας 0-1 Στοιχεία HTML5 ^ XHTML 2.0

Στοιχείο	HTML 5	XHTML 2
a	✓	✓
abbr	✓	✓
access	✗	✓
acronym	✗	✗
action	✗	✓
address	✓	✓
applet	✗	✗
area	✓	✗
article	✓	✗
aside	✓	✗
audio	✓	✗
b	✓	✗
base	✓	✗
basefont	✗	✗
bdo	✓	✗
big	✗	✗

blockcode	X	✓
blockquote	✓	✓
body	✓	✓
br	✓	X
button	✓	X
canvas	✓	X
caption	✓	✓
center	X	X
cite	✓	✓
code	✓	✓
col	✓	✓
colgroup	✓	✓
command	✓	X
datagrid	✓	X
datalist	✓	X
dd	✓	✓
del	✓	X
delete	X	✓
details	✓	X
dfn	✓	✓
di	X	✓
dialog	✓	X
dir	X	X
dispatch	X	✓
div	✓	✓
dl	✓	✓
dt	✓	✓
em	✓	✓
embed	✓	X
ev:listener	X	✓
event-source	✓	X
fieldset	✓	X
figure	✓	X
font	✓	X
footer	✓	X
form	✓	X
frame	X	X
frameset	X	X

group	X	✓
h	X	✓
h1	✓	X
h2	✓	X
h3	✓	X
h4	✓	X
h5	✓	X
h6	✓	X
handler	X	✓
head	✓	✓
header	✓	X
hr	✓	X
html	✓	✓
i	✓	X
iframe	✓	X
img	✓	✓
input	✓	✓
ins	✓	X
insert	X	✓
isindex	X	X
kbd	✓	✓
l	X	✓
label	✓	✓
legend	✓	X
li	✓	✓
link	✓	✓
load	X	✓
m	✓	X
map	✓	X
menu	✓	X
message	X	✓
meta	✓	✓
meter	✓	X
model	X	✓
nav	✓	X
nl	X	✓
noframes	X	X
noscript	✓	X

object	✓	✓
ol	✓	✓
optgroup	✓	✗
option	✓	✗
output	✓	✓
p	✓	✓
param	✓	✓
pre	✓	✓
progress	✓	✗
q	✓	✓
range	✗	✓
rb	✗	✓
rbc	✗	✓
rebuild	✗	✓
recalculate	✗	✓
refresh	✗	✓
repeat	✗	✓
reset	✗	✓
revalidate	✗	✓
rp	✗	✓
rt	✗	✓
rtc	✗	✓
ruby	✗	✓
s	✗	✗
samp	✓	✓
script	✓	✗
secret	✗	✓
section	✓	✓
select1	✗	✓
select	✓	✓
send	✗	✓
separator	✗	✓
setfocus	✗	✓
setindex	✗	✓
setvalue	✗	✓
small	✓	✗
source	✓	✗
span	✓	✓
standby	✗	✓

strike	X	X
strong	✓	✓
style	✓	✓
sub	✓	✓
submit	X	✓
summary	X	✓
sup	✓	✓
switch	X	✓
table	✓	✓
tbody	✓	✓
td	✓	✓
textarea	✓	✓
tfoot	✓	✓
th	✓	✓
thead	✓	✓
time	✓	X
title	✓	✓
tr	✓	✓
trigger	X	✓
tt	X	X
u	X	X
ul	✓	✓
upload	X	✓
var	✓	✓
video	✓	X

Αρκετά νέα στοιχεία έχουν προστεθεί στην HTML 5, αλλά δεν είναι διαθέσιμα στην XHTML 2.0 όπως φαίνεται στο πίνακα. Επιπρόσθετα νέα στοιχεία έχουν δοθεί, για την σημασιολογική σήμανση του εγγράφου όπως: section, article, header, footer και aside.

Αντίστοιχα υπάρχουν νέα στοιχεία που έχουν παρουσιαστεί μόνο στην XHTML V2. Ιδιαίτερα όμως αξίζει να αναφερθούνε στις δυνατότητες των νέων χαρακτηριστικών, όπως για παράδειγμα ότι κάθε στοιχείο μπορεί να είναι ένας υπερσύνδεσμος. Η ιδιότητα “href” μπορεί να χρησιμοποιηθεί σε κάθε στοιχείο. Με αυτή την αλλαγή το στοιχείο “a” δεν είναι πια απαραίτητο και έχει αποσυρθεί.

Επίσης κάθε στοιχείο μπορεί να είναι εικόνα ή κάποιος άλλος πόρος. Πλέον στην XHTML V2, το στοιχείο img έχει καταργηθεί. Η ιδέα είναι ότι όλες οι εικόνες

αποτελούν “ένα μακρύ περιεχόμενο”. Ορίζοντας την ιδιότητα `src` οποιοδήποτε στοιχείου, αναφέρεται στο πρόγραμμα πελάτη (φυλλομετρητή), να χρησιμοποιήσει τον αντίστοιχο πόρο, που μπορεί να είναι μια εικόνα είτε ένα αρχείο βίντεο. Αν ο πόρος δεν είναι διαθέσιμος χρησιμοποιείται το στοιχείο.

Το στοιχείο “`br`”, το οποίο χρησιμοποιείται για την εισαγωγή γραμμών, έχει αποσυρθεί από την XHTML 2.0. Το νέο στοιχείο “`l`” το έχει αντικαταστήσει και αναπαριστά μια γραμμή κειμένου.

Επίσης τα νέα στοιχεία “`h`” και “`section`” έχουν αντικαταστήσει τα αριθμημένα στοιχεία “`h1 – h6`”, ενώ αντίθετα στην HTML 5 έχουν διατηρηθεί. Ο στόχος ήταν ακριβής αναπαράσταση της ιεραρχικής δομής ενός εγγράφου, η οποία θα πρέπει να δίνεται μέσω εμφωλευμένων τμημάτων.

Γενικότερα η ομάδα ανάπτυξης της XHTML V2, επικεντρώθηκε στην δημιουργία μιας πιο γενικής, απλοποιημένης γλώσσας. Για αυτό το λόγο προστέθηκαν αρκετά ειδικά στοιχεία, για την αναπαράσταση των διαφορετικών τύπων περιεχομένου.

Κάποιες γενικότερες διαφορές ανά στοιχείο είναι:

- **html**: Η XHTML2 περιέχει την ιδιότητα `version` και `xmlns:base`, ενώ στην HTML5 δεν υπάρχουν και η αντίστοιχη δήλωση έχει απλοποιηθεί πολύ.
- **head**: Η XHTML2 υποστηρίζει την ιδιότητα `profile`, ενώ η HTML5 όχι και ο λόγος που καταργήθηκε ήταν επειδή δεν ήταν συχνή η χρήση της.
- **blockcode**: Η HTML5 δεν υποστηρίζει αυτό το στοιχείο. Αντί για αυτό έχει τα `pre`, `code`, τα οποία εξασφαλίζουν συμβατότητα προς τα πίσω ενώ το `blockcode` όχι.
- **separator**. Η HTML5 δεν έχει `separator` στοιχείο, αλλά το `<hr>`, που κάνει το ίδιο πράγμα.
- **edit**: Το στοιχείο `edit` της XHTML2, χρησιμοποιώντας τις ιδιότητες `deleted`, `inserted` δείχνει αν το περιεχόμενο έχει αλλάξει. Τα αντίστοιχα στοιχεία της HTML5 είναι το `del`, `ins`.
- **dialog**. Υπάρχει μόνο στην HTML5 και αναπαριστά ένα διάλογο.



## Λίστες πλοήγησης

Οι λίστες πλοήγησης παρουσιάστηκαν πρώτη φορά και την HTML 5, και στην XHTML 2. Στην XHTML 2.0, χρησιμοποιείται το στοιχείο `nl`. Οι λίστες πλοήγησης πρέπει να ξεκινούν με ένα αρχικό στοιχείο, το οποίο καθορίζει το τίτλο λίστας. Ακολουθώντας το τίτλο, ένα ή περισσότερα `li` στοιχεία χρησιμοποιούνται για τα στοιχεία της λίστας. Επίσης στην XHTML 2.0, μια νέα δυνατότητα είναι η χρήση υπερσυνδέσμων ως στοιχεία της λίστας με τη χρήση της ιδιότητας `href`. Με το συνδυασμό αυτών των χαρακτηριστικών παράγεται μια απλή, ελαφριά δομή σήμανσης.

```
<nl>
<label>Category</label>
<li href="">Όλα</li>
<li href="">Νέα</li>
<li href="">Βίντεο</li>
<li href="">Εικόνες</li>
</nl>
```

Στην HTML5, ένα νέο στοιχείο `nav` έχει προστεθεί για αυτό τον σκοπό. Δυστυχώς το `nav` δεν μπορεί να είναι στοιχείο λίστας, συνεπώς δεν μπορεί να περιέχει στοιχεία `li` για την λογική οργάνωση συνδέσμων. Επίσης οι ετικέτες αγκύρωσης (anchor tags), είναι απαραίτητοι για την δημιουργία υπερσυνδέσμων στην HTML5. Όλα αυτά τα στοιχεία δεν καθιστούν την σήμανση πλοήγησης τόσο ελκυστική.

```
<nav>
<h1>Category</h1>
<ul>
<li><a href="">Όλα</a></li>
<li><a href="">Νέα</a></li>
<li><a href="">Βίντεο</a></li>
<li><a href="">Εικόνες</a></li>
</ul>
</nav>
```

## XForms – Html Forms

Στην περίπτωση των φορμών και οι δύο προδιαγραφές έχουν νέα χαρακτηριστικά, για την δημιουργία πιο ισχυρών φορμών με λιγότερη συγγραφή κώδικα σεναρίου (javascript). Στην XHTML 2.0, οι φόρμες `html` δεν υποστηρίζονται και έχουν

υιοθετηθεί οι XForms. Η ομάδα ανάπτυξης της XHTML 2.0 δεν αναπτύσσει το συγκεκριμένο πρότυπο, αλλά αναφέρεται με λεπτομέρεια σε αυτό. Για την διευκόλυνση της επαναχρησιμοποίησης, οι XForms διαχωρίζουν τα δεδομένα που συλλέγονται από την παρουσίαση και τον έλεγχο.

Σε αντίθεση με τις αρχικές φόρμες HTML, οι δημιουργοί των XForms έχουν χρησιμοποιήσει την προσέγγιση μοντέλο-όψη-έλεγχος (MVC: Model View Controller). Όπως είδαμε αυτό το μοντέλο αποτελείται από ένα ή περισσότερα μοντέλα που περιγράφουν τα δεδομένα, συνθήκες που αφορούν τα δεδομένα και καταχωρίσεις. Η όψη περιγράφει ποια πεδία ελέγχου θα εμφανίζονται στις φόρμες και πώς αυτά ομαδοποιούνται και σε τι δεδομένα αποσκοπούν. Τα φύλλα στυλ χρησιμοποιούνται για να περιγράψουν την μορφή της φόρμας.

Ένα έγγραφο XForms μπορεί να είναι τόσο απλό, όπως μια HTML φόρμα (με το να καθορίσει κάποιος το στοιχείο αποστολής στο τμήμα του μοντέλου και τοποθετώντας τα πεδία ελέγχου στο σώμα του εγγράφου), όμως οι XForms έχουν αρκετά εξειδικευμένα χαρακτηριστικά. Για παράδειγμα, νέα δεδομένα μπορούν να ανακτηθούν και να χρησιμοποιηθούν για την ενημέρωση της φόρμας, όσο αυτή εκτελείται με τη χρήση XMLHttpRequest/AJAX χωρίς χρήση κώδικα. Ο συντάκτης επίσης της φόρμας μπορεί να επικυρώσει τον τύπο των δεδομένων που έχει καταχωρίσει ο χρήστης με αυτά που περιγράφονται στο XML σχήμα, απαιτώντας συγκεκριμένα δεδομένα, απενεργοποιώντας πεδία ελέγχου ή αλλάζοντας την μορφή της φόρμας κατά την εκτέλεση βάσει ορισμένων συνθηκών.

Τόσο οι φόρμες HTML όσο και οι XForms, μπορούν να χρησιμοποιήσουν ποικίλα μη-XML πρωτόκολλα αποστολής όπως: multipart/form-data, application/x-www-form-urlencoded, αλλά οι XForms μπορούν να στείλουν δεδομένα στον εξυπηρετητή σε μορφή XML. Ουσιαστικά αποτελούν από μόνες τους ένα διάλογο XML, για αυτό το λόγο μπορούν να δημιουργηθούν αυτόματα από XHTML 2.0 έγγραφα. Μπορούν να δημιουργήσουν σχήματα XML, αλλά και να μετατραπούν σε XHTML φόρμες.

Η HTML 5, διατηρεί της γνωστές φόρμες HTML, αλλά προσθέτει αρκετούς νέους τύπου για την απλοποίηση της ανάπτυξης και την βελτίωση της χρηστικότητας. Στην HTML 5, παρουσιάζονται νέα στοιχεία εισόδου για διευθύνσεις email, urls, ημερομηνίες, ώρα και αριθμητικών δεδομένων. Αυτό επιτρέπει στους δημιουργούς να παρέχουν πιο εξελιγμένα περιβάλλοντα (π.χ. επιλογείς ημερομηνιών, date pickers), να

ενσωματώνουν δεδομένα από άλλες πηγές (π.χ. από το outlook) και να επικυρώνουν την είσοδο του χρήστη πριν την αποθήκευση, χωρίς την χρήση κώδικα. Επιπρόσθετα ο χρόνος εκμάθησης των html φορμών είναι σαφώς μικρότερος, καθώς αποτελεί γνωστό αντικείμενο με πρόσθετες δυνατότητες.

## Νέες δυνατότητες μόνο στην HTML 5

Υπάρχουν αρκετές νέες δυνατότητες και χαρακτηριστικά στην HTML 5, που δεν υπάρχουν στην XHTML 2.0. Αυτές είναι:

### 4.6.1. Web Application APIs

Η HTML 5 παρουσιάζει αρκετά APIs, που βελτιώνουν δραματικά την ανάπτυξη εφαρμογών πελάτη (client-side). Αυτά τα APIs είναι αυτά που κάνουν την HTML 5 όχι μόνο μια γλώσσα σήμανσης, αλλά και ένα εργαλείο ανάπτυξης ισχυρών διαδικτυακών εφαρμογών. Τα συγκεκριμένα APIs, αναπτύσσονται ξεχωριστά από το γενικό πρότυπο, συνεπώς μπορεί να μην υιοθετηθούν τελικά όλα. Τα νέα APIs είναι:

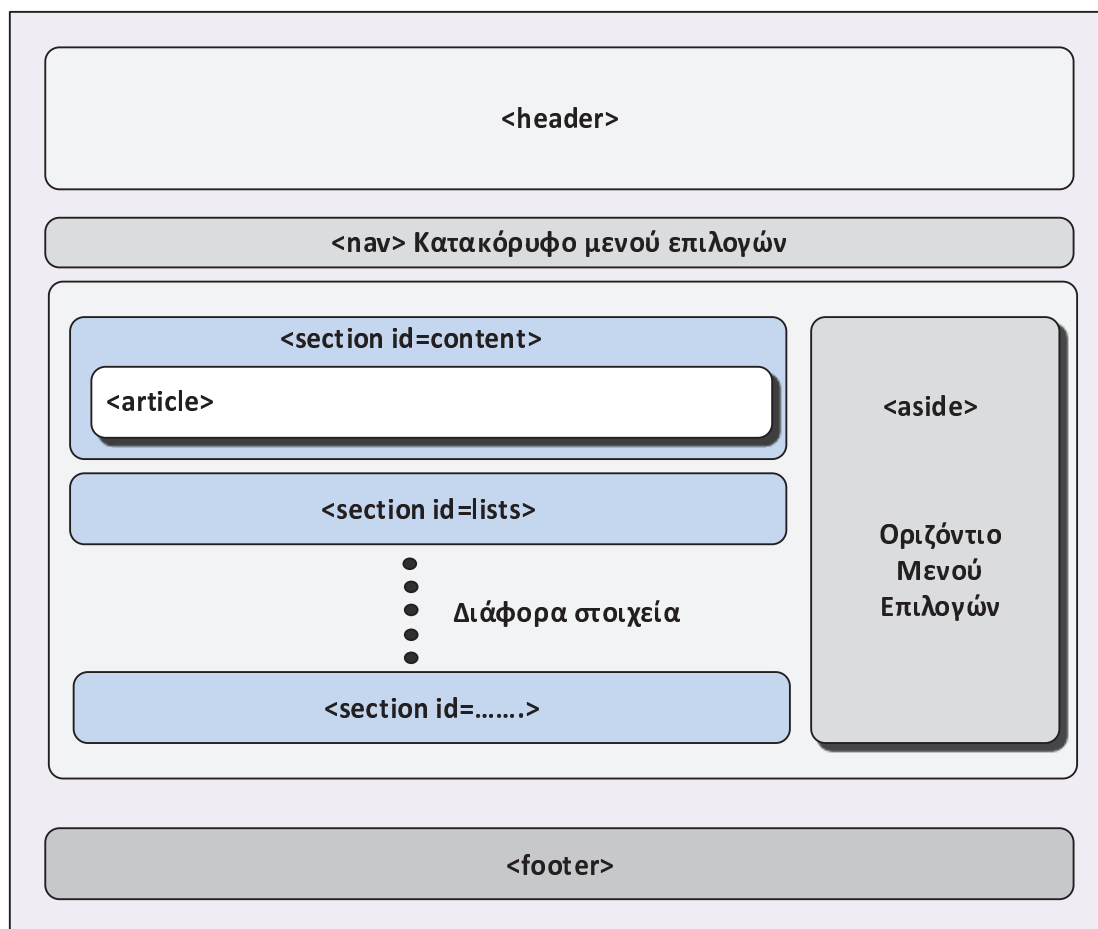
- Ένα 2D API σχεδιασμού και το αντίστοιχο στοιχείο “canvas”
- Ένα API αναπαραγωγής ήχου και βίντεο, με υποστήριξη πολλών τύπων αρχείων, τα οποία μπορούν να χρησιμοποιηθούν με τα νέα στοιχεία βίντεο και ήχου.
- Ένα API λειτουργίας σε αποσύνδεση (offline) (παρόμοιο με το Google Gears), που επιτρέπει την εκτέλεση κάποιας εφαρμογής χωρίς κάποιος να βρίσκεται συνδεδεμένος στο διαδίκτυο.
- Ένα API, που επιτρέπει στις εφαρμογές WEB, να εγγράφονται από μόνες τους για ορισμένα πρωτόκολλα.
- Ένα API επεξεργασίας, που μπορεί να χρησιμοποιηθεί σε συνδυασμό με το χαρακτηριστικό “global”.
- Ένα API που επιτρέπει την λειτουργία σύρε-και-άσε (drag n drop).
- API δικτύου, για την επικοινωνία μέσω της χρήσης TCP.
- Ένα API που επιτρέπει στις εφαρμογές να διαχειριστούν το ιστορικό του φυλλομετρητή.
- APIs για την δημιουργία μηνυμάτων, ανάμεσα στον πελάτη και στον εξυπηρετητή, αλλά και ανάμεσα σε διαφορετικούς τομείς.

#### **4.6.2. Ανίχνευση χρηστών**

Η καινούργια ιδιότητα ring μπορεί να χρησιμοποιηθεί στα στοιχεία “a” και “area” για την ανίχνευση των επισκεπτών ενός ιστοτόπου. Έτσι αν ο επισκέπτης ακολουθήσει συγκεκριμένους υπερσυνδέσμους, αυτό μπορεί να καταγραφεί μέσω αυτής της ιδιότητας.

## Υλοποίηση και σύγκριση

Για την σύγκριση σε επίπεδο κώδικα, θα δημιουργήσουμε μία σελίδα, η οποία θα επιδεικνύει τα κοινά στοιχεία των δύο γλωσσών. Η σελίδα θα αποτελείται από μία επικεφαλίδα, ένα υποσέλιδο, ένα μενού επιλογών σε κατακόρυφη διάταξη και ένα μενού επιλογών σε οριζόντια διάταξη. Ο κύριος κορμός της σελίδας θα περιέχει ενότητες με τα αντίστοιχα στοιχεία των δύο γλωσσών. Το σχέδιο παρουσιάζεται στην εικόνα που ακολουθεί.



Εικόνα 2 - Σχέδιο σελίδας για σύγκριση

Στο κύριο μέρος της σελίδας, θα υπάρχουν οι αντίστοιχες ενότητες (sections), που θα αποτελέσουν την σύγκριση των στοιχείων ανά ομάδα: λίστες, σημασιολογικά στοιχεία, φόρμες, πίνακες, καθώς και άλλα στοιχεία.

Στη παρούσα φάση θα μας απασχολήσει μόνο το σχέδιο της σελίδας και οι ετικέτες ενώ η στυλιστική διαμόρφωση της παρουσίασης θα γίνει μέσω CSS, για να μπορούμε να δούμε διαφορές μόνο στο κώδικα.

## Τύπος εγγράφου

Η πρώτη γραμμή σε κάθε έγγραφο καλείται doctype. Χρησιμοποιείται για να πληροφορήσει σχετικά με τον τύπο περιεχομένου σε οποιοδήποτε ενδιαφερόμενο (π.χ φυλλομετρητή, μηχανή αναζήτησης). Αντίστοιχα για τις δύο γλώσσες έχουμε:

### HTML 5.0

```
<!DOCTYPE html>
```

### XHTML 2.0

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 2.0//EL"
```

Στον κώδικα δήλωσης του doctype, παρατηρούμε ότι η HTML5 παρέχει μια απλοϊκή οδηγία που δείχνει ότι αυτή η σελίδα είναι html. Γενικότερα το doctype, όπως έχουμε αναφέρει, παραμένει για ιστορικούς κυρίως λόγους, οπότε η απλοποίηση που έχει επιτευχθεί στην HTML 5.0 διευκολύνει αρκετά.

## Κεφαλίδα

Η κεφαλίδα περιέχει τα μεταδεδομένα της σελίδας μας και είναι αντίστοιχα:

### HTML 5.0

```
<head>  
  <title>HTML 5 Κώδικας</title>  
  <meta name="keywords" content="html5, www, xhtml2 V html5" />  
  <meta name="description" content="Κώδικας HTML 5.0" />  
  <link rel="stylesheet" type="text/css" href="style.css" />  
</head>
```

### XHTML 2.0

```
<head>  
  <meta about="" property="title">XHTML 2.0 Κώδικας</meta>  
  <meta name="author" content="Dimitra Tsoulh"></meta>  
  <meta name="keywords" content="xhtml2, www, xhtml2 V html5"></meta>  
  <meta name="description" content="Κώδικας XHTML 2.0 "></meta>  
  <link rel='stylesheet' type="text/css" href="style.css"> </link>  
</head>
```

Βλέπουμε ότι στην HTML 5 τα μεταδεδομένα δεν χρειάζονται ετικέτες κλεισίματος. Αντίθετα η σήμανση στην XHTML 2 είναι πιο ξεκάθαρη αφού είναι διακριτός ο τρόπος που κλείνουν οι ετικέτες. Επίσης αυτό το χαρακτηριστικό, επιτρέπει σε έναν φυλλομετρητή να ανακτήσει πιο γρήγορα τη σελίδα, καθώς δεν χρειάζεται κάθε φορά να εκτελεί συνθήκες για τον έλεγχο των εντολών. Μία άλλη παρατήρηση, είναι ότι στην XHTML 2.0 ο τίτλος της σελίδας ορθά υπολογίζεται ως μεταπληροφορία και αυτός είναι ο λόγος που ορίζεται ως μεταδεδομένο με την ετικέτα meta και την ιδιότητα about.

## Κυρίως μέρος

Στο κύριο μέρος θα τοποθετήσουμε τον κώδικα που αφορά την δομή και σε ορισμένες περιπτώσεις την εμφάνιση της σελίδας μας.

### 5.2.1. Επικεφαλίδα

Η επικεφαλίδα τοποθετείται στην κορυφή της σελίδας και συνήθως περιέχει ένα λογότυπο ή μία ένδειξη για το περιεχόμενο. Ο ορισμός επικεφαλίδας γίνεται ως:

#### HTML 5.0

```
<header>
  <hgroup>
    <h1><a href=#>HTML5 Version</a></h1>
    <h2>Σύγκριση HTML5 v XHTML2</h2>
  </hgroup>
  <nav id=global>
    <ul>
      <li><a href="#">ΑΡΧΙΚΗ ΣΕΛΙΔΑ</a></li>
      <li><a href="#">ΣΧΕΤΙΚΑ</a></li>
    </ul>
  </nav>
</header>
```

#### XHTML 2.0

```
<h id="PageHeader">XHTML 2.0 Version</h>
```

Στην HTML 5 έχουμε χρησιμοποιήσει την λογική της ετικέτας header, για να φτιάξουμε μια επικεφαλίδα. Η header ανάλογα με το επίπεδο που δηλώνεται, μπορεί να χρησιμοποιηθεί σαν τίτλος – επικεφαλίδα στο συγκεκριμένο επίπεδο

ομαδοποίησης. Εφόσον στην περίπτωση μας δηλώνεται μετά το κύριο μέρος, αποτελεί επικεφαλίδα όλης της σελίδας. Από την άλλη η XHTML 2.0 δεν υποστηρίζει τη συγκεκριμένη ετικέτα. Χρησιμοποιεί όμως την ετικέτα **<h>**, η οποία προσφέρεται για τη δημιουργία επικεφαλίδων ανά επίπεδα ομαδοποίησης. Κατά τον ίδιο τρόπο με την HTML 5, αφού βρίσκεται μετά το **<body>** αποτελεί πρώτο επίπεδο ομαδοποίησης, άρα επικεφαλίδα για την πρώτη σελίδα. Σαφώς η δήλωση της XHTML 2.0 είναι μινιμαλιστική και χρησιμοποιεί λιγότερες ετικέτες.

### 5.2.2. Πλοήγηση

Στην σελίδα, έχουμε επιλέξει δύο τρόπους πλοήγησης: ένα οριζόντιο μενού και ένα κατακόρυφο στα αριστερά της σελίδας.

Το οριζόντιο μενού επιλογών είναι:

#### HTML 5.0

```
<header>
  <hgroup>
    <h1><a href=#>HTML5 Version</a></h1>
    <h2>Σύγκριση HTML5 v XHTML2</h2>
  </hgroup>
  <nav id=global>
    <ul>
      <li><a href="#">ΑΡΧΙΚΗ ΣΕΛΙΔΑ</a></li>
      <li><a href="#">ΣΧΕΤΙΚΑ</a></li>
    </ul>
  </nav>
</header>
```

#### XHTML 2.0

```
<div id="navigation">
  <nl>
    <li href="#">ΑΡΧΙΚΗ ΣΕΛΙΔΑ</li>
    <li href="#">ΣΧΕΤΙΚΑ</li>
  </nl>
</div>
```



Αντίστοιχα το κάθετο μενού επιλογών ανά περίπτωση είναι:

## HTML 5.0

```
<aside>
  <nav>
    <h1>Επιλογές</h1>
    <ul class=links>
      <li><a href="#">Αρχική Σελίδα</a></li>
      <li><a href="#">Ενότητα Κειμένου</a></li>
      <li><a href="#">Ενότητα Εικόνας</a></li>
      <li><a href="#">Ενότητα Φορμών</a></li>
      <li><a href="#">Ενότητα Πινάκων</a></li>
      <li><a href="#">Ενότητα Υπερσυνδέσμων</a></li>
    </ul>
  </nav>
</aside>
```

## XHTML 2.0

```
<div id="sidebar">
  <nl>
    <name>Επιλογές</name>
    <li href="#"> Αρχική Σελίδα</li>
    <li href="#"> Ενότητα Κειμένου</li>
    <li href="#"> Ενότητα Εικόνας</li>
    <li href="#"> Ενότητα Φορμών</li>
    <li href="#"> Ενότητα Πινάκων</li>
    <li href="#"> Ενότητα Υπερσυνδέσμων</li>
  </nl>
</div>
```

Στην HTML5 ορίσαμε το οριζόντιο μενού επιλογών να είναι στην περιοχή της επικεφαλίδας. Οι επιλογές δεν αντιστοιχούν σε κάποιο σύνδεσμο, αφού εξυπηρετούν λόγους επίδειξης κώδικα.

Βλέπουμε ότι και στις δύο περιπτώσεις για την δημιουργία των επιλογών χρησιμοποιούνται οι λίστες. Στην περίπτωση της HTML 5, υπάρχει το νέο στοιχείο `<nav>`, που όμως δεν αποτελεί στοιχείο λίστας, όπως ήδη έχουν αναφέρει στο κεφάλαιο 4, σε αντίθεση με το νέο στοιχείο `<nl>` της XHTML 2, που είναι στοιχείο λίστας. Συνεπώς στην HTML 5, χρειάζεται να προσθέσουμε ένα ακόμη στοιχείο, που

στην προκειμένη περίπτωση είναι το `<ul>`. Επιπρόσθετα στο κάθετο μενού, βλέπουμε ότι για την HTML 5, χρησιμοποιούμε το στοιχείο `<aside>`, για να στοιχίσουμε το μενού στα αριστερά. Αυτό είναι ένα παράδειγμα σήμανσης, που αφορά εν μέρει την σημασιολογία, αλλά που όμως χρησιμοποιείται και ως σήμανση παρουσίασης. Η χρήση της ετικέτας `<div>`, στην XHTML 2.0, δίνει τη δυνατότητα για τη συγγραφή καθαρότερου κώδικα, αφήνοντας την παρουσίαση σε άλλους μεθόδους, όπως για παράδειγμα τα φύλλα στυλ (CSS). Τέλος είναι εμφανής η δυνατότητα στις λίστες της XHTML 2, να περιέχουν υπερσυνδέσμους ενσωματωμένους στην λίστα, με αποτέλεσμα έναν συνοπτικότερο τρόπο συγγραφής κώδικα.

### 5.2.3. Περιεχόμενο

Το περιεχόμενο της σελίδας μας, αποτελείται από ενότητες που αφορούν την παρουσίαση των κοινών στοιχείων των δύο προτύπων.

#### 5.2.3.1. Εισαγωγικό κείμενο

Το εισαγωγικό κείμενο αφορά, μία παράγραφο που αναφέρεται γενικά στο σκοπό της σελίδας μας και είναι ενημερωτικού χαρακτήρα.

## HTML 5.0

```
<article>
  <header>
    <h1>Σχετικά</h1>
  </header>
  <p>
    Η σελίδα αυτή παρουσιάζει μερικά παραδείγματα πηγαίου κώδικα σε δύο πρότυπα.
    Σκοπός είναι η σύγκριση του κώδικα υλοποίησης ανάμεσα στην HTML5 και την XHTML 2.0
    Η σύγκριση αφορά τα κοινά στοιχεία των δύο γλωσσών, ενώ τα στοιχεία της HTML 5.0 που
    δεν υπάρχουν στην XHTML 2.0 και αντίστοιχα δεν παρουσιάζονται
  </p>
  <footer>
    <em>Πτυχιακή Εργασία:</em> <strong>Σύγκριση HTML5 v XHTML2</strong>
    <span class=newLine><em>Ετικέτες:</em>
    <a class=tags href=#>html5</a>
    <a class=tags href=#>xhtml2</a>
  </footer>
</article>
```

## XHTML 2.0

```
<section id="intro">
  <h>Σχετικά</h>
  <p>
    Η σελίδα αυτή παρουσιάζει μερικά παραδείγματα πηγαίου κώδικα σε δύο πρότυπα.
    Σκοπός είναι η σύγκριση του κώδικα υλοποίησης ανάμεσα στην HTML5 και την XHTML 2.0
    Η σύγκριση αφορά τα κοινά στοιχεία των δύο γλωσσών, ενώ τα στοιχεία της HTML 5.0 που
    δεν υπάρχουν στην XHTML 2.0 και αντίστοιχα δεν παρουσιάζονται
  </p>
  <section id="footer">
    <em>Πτυχιακή Εργασία:</em> <strong>Σύγκριση HTML5 v XHTML2</strong>
    <span class=newLine><em>Ετικέτες:</em>
    <a class=tags href=#>html5</a>
    <a class=tags href=#>xhtml2</a>
  </section>
</section>
```

Εδώ βλέπουμε ότι η HTML 5, έχει τη δυνατότητα ενσωμάτωσης επικεφαλίδας και υποσέλιδου σε κάθε τμήμα / ομάδα (section, article, aside). Αντίθετα η XHTML 2, δεν υποστηρίζει υποσέλιδα ή επικεφαλίδες μέσα στις ενότητες, αλλά μπορούμε να τις δημιουργήσουμε με εμφωλευμένα sections και τη χρήση φύλλων στυλ.

### 5.2.3.2. Πολυμέσα

Η ενσωμάτωση πολυμέσων, δηλαδή αρχείων ήχου, βίντεο, αρχείων γίνεται ως εξής:

## HTML 5.0

```
<article>
  <header>
    <h1>Πολυμέσα</h1>
  </header>
  <h2>Παραδείγματα ενσωμάτωσης πολυμέσων</h2>
  <h3>Ενσωμάτωση εικόνας</h3>
  
  <h3>Ενσωμάτωση βίντεο</h3>
  <video width="320" height="240" controls>
    <source src="movie.mp4" type="video/mp4">
    Δεν υποστηρίζεται ο codec!!
  </video>
</article>
```

## XHTML 2.0

```
<section id="media">
  <h>Πολυμέσα</h>
  <section id="image">
    <h>Ενσωμάτωση εικόνας</h>
    <object>
      src=picture.jpg
      srctype="image/jpg"
      xml:base="http://www.example.com/images/">
      Δεν υποστηρίζεται ο codec και δε βρέθηκε καμία εικόνα
    </object>
  </section>
  <section id="video">
    <h>Ενσωμάτωση βίντεο</h>
    <object>
      src="movie.mp4"
      srctype="video/mp4"
      xml:base="http://www.example.com/videos/">
      <object>
        src=picture.jpg
        srctype="image/jpg"
        xml:base="http://www.example.com/images/">
        Δεν υποστηρίζεται ο codec και δε βρέθηκε καμία εικόνα
      </object>
    </object>
  </section>
</section>
```

Παρατηρούμε ότι στην HTML 5, υπάρχουν ξεχωριστές ετικέτες για το κάθε τύπο πολυμέσου, ενώ υποστηρίζεται και η ετικέτα **<object>**, για να εξασφαλίσει τη συμβατότητα προς τα πίσω. Η XHTML 2, έχει μόνο το στοιχείο **<object>** και καθορίζει το τύπο του αρχείου μέσα από τις ιδιότητες της ετικέτας. Στην ενσωμάτωση βίντεο που αφορά τον κώδικα XHTML 2, παρατηρούμε επίσης ότι έχουμε την δυνατότητα εμφωλευμένων αντικειμένων. Έτσι στην περίπτωση που για παράδειγμα δεν υποστηρίζεται ο codec του βίντεο, τότε εμφανίζεται μία εικόνα, ενώ αν δεν υπάρχει ούτε η εικόνα εμφανίζεται εναλλακτικό κείμενο. Η δυνατότητα αυτή υποστηρίζεται στην HTML 5, με απλούστερο τρόπο, απλά δηλώνοντας κατά σειρά τα αρχεία αντικατάστασης.

### 5.2.3.3. Φόρμες

Οι φόρμες αποτελούν σημαντική, ενότητα καθώς μέσα από αυτές επιτυγχάνεται η εισαγωγή στοιχείων. Η HTML5, έχει ενσωματώσει ενότητα φορμών, ενώ η XHTML 2 στηρίζεται στην ενότητα XForms 2.0, η οποία υλοποιείται από τρίτους. Μια φόρμα XForm, μπορεί να ενσωματωθεί σε μία σελίδα HTML 5.

#### HTML 5.0

```
<article>
  <header>
    <h1>Φόρμες HTML 5</h1>
  </header>
  <h2>Παράδειγμα φόρμας</h2>
  <form action = "post">
    <fieldset>
      <legend>Φόρμα εισαγωγής στοιχείων</legend>
      <caption>Πεδία εισαγωγής</caption>
      <input type="text" size="20" autofocus required placeholder="Όνομα"/>
      <input type="text" size="8" pattern="\d{3}-\d{4}" title="ddd-dddd"/>
      <input type="email" size="20" name="email" required placeholder="Διεύθυνση email"/>
      <input type="url" size="30" name="url" placeholder="URL"/>
      <input type="text" name="choices" list="data-list">
      <datalist id="data-list">
        <option value="Επιλογή 1">
        <option value="Επιλογή 2">
        <option value="Επιλογή 3">
        <option value="Επιλογή 4">
        <option value="Επιλογή 5">
      </datalist>
      <meter min="0" max="10" value="7"></meter>
      <progress max="100" value="70">30%</progress>
      <input type="range" id="range" name="range">
      <input type="submit"/>
    </fieldset>
  </form>
</article>
```

## XHTML 2.0

```
<head>
  <title>Παράδειγμα Φόρμας XForm</title>
  <xforms:model id="data">
    <xforms:instance xmlns="simple-data">
      <name></name>
      <phone></phone>
      <address></address>
      <url></url>
      <choices>
        <item name="Επιλογή 1"/>
        <item name="Επιλογή 2"/>
      </choices>
    </xforms:instance>
  <xforms:submission id="submit_data"
    action="http://example.com/submit" method="post" id="submit"/>
</head>
<body>
  <input ref="name">
    <label>Όνομα</label>
  </input>
  <input ref="phone">
    <label>Τηλέφωνο</label>
  </input>
  <input ref="address">
    <label>Διεύθυνση</label>
  </input>
  <select ref="choices">
    <label>Επιλογές</label>
    <item>
      <label>Επιλογή 1</label>
      <value>choice1</value>
    </item>
    <item>
      <label>Επιλογή 2</label>
      <value>choice2</value>
    </item>
  </select>
  <submit submission="submit">
    <label>Αποστολή</label>
  </submit>
</body>
```

Η κύρια διαφορά, κατά τη σύγκριση των XForms με αυτές τις φόρμες HTML, είναι το γεγονός ότι οι πρώτες χρησιμοποιούν την σύνταξη της XML. Μία άλλη διαφορά των XForms είναι ο διαχωρισμός του ελέγχου των δεδομένων, από τον σχεδιασμό και την εμφάνιση. Το μοντέλο καθορίζεται στην επικεφαλίδα της σελίδας, ενώ η εμφάνιση των δεδομένων στο κύριο μέρος. Μια άλλη διαφορά είναι, ότι ενώ οι

XForms έχουν σχεδιαστεί για την XHTML, δεν αποτελούν μέρος της γλώσσας, συνεπώς μπορούν να χρησιμοποιηθούν και σε άλλες γλώσσες σήμανσης.

#### 5.2.3.4. Πίνακες

Οι ετικέτες των πινάκων δεν έχουν κάποια διαφοροποίηση. Έτσι ο κώδικας που ακολουθεί ισχύει και για τις δύο γλώσσες. Αυτό που διαφοροποιείται είναι κυρίως, ιδιότητες που πηγάζουν από άλλα στοιχεία των γλωσσών, όπως για παράδειγμα η “href”, που μπορεί να χρησιμοποιηθεί σε οποιοδήποτε στοιχείο, συνεπώς και στο <td>.

#### HTML 5.0, XHTML 2.0

```
<table>
  <tbody>
    <tr>
      <td>1</td>
      <td rowspan="2">2</td>
      <td>3</td>
    </tr>
    <tr>
      <td>4</td>
      <td>6</td>
    </tr>
    <tr>
      <td>7</td>
      <td>8</td>
      <td>9</td>
    </tr>
  </tbody>
</table>
```

#### 5.2.4. Χάρτες Εικόνων

Οι χάρτες εικόνων μας δίνουν τη δυνατότητα χρήσης υπερ-συνδέσμων, βάσει συντεταγμένων και ορίων. Στο κώδικά μας, έχουμε επιλέξει μια εικόνα με τον ήλιο και τους πλανήτες Ερμή, Αφροδίτη. Επιλέγοντας είτε έναν πλανήτη, είτε τον ήλιο ο χρήστης οδηγείται σε μια αντίστοιχη σελίδα.

## HTML 5.0

```
<p>Πατήστε στον πλανήτη για περισσότερες πληροφορίες:</p>

<map name="planetmap">
  <area shape="rect" coords="0,0,82,126" alt="Ηλιος" href="sun.html">
  <area shape="circle" coords="90,58,3" alt="Ερμής" href="mercur.html">
  <area shape="circle" coords="124,58,8" alt="Αφροδίτη" href="venus.html">
</map>
```

## HTML 5.0, XHTML 2.0

```
<ul id="map1" role="navigation" src="planets.gif" srctype="image/png" usemap="#map1">
  <caption>Πλοήγηση στους Πλανήτες:</caption>
  <li href="sun.html" shape="rect" coords="0,0,82,126">
    Ηλιος
  </li>
  <li href="mercury.html" shape="circle" coords="90,58,3">
    Ερμής
  </li>
  <li href="venus.html" shape="circle" coords="124,58,8">
    Αφροδίτη
  </li>
</ul>
```

Παρατηρούμε ότι στην HTML 5, η χαρτογράφηση μιας εικόνας, γίνεται με τον συνδυασμό τριών στοιχείων: το **<img>** για τον ορισμό της εικόνας, το **<map>** για τον καθορισμό του «αντικειμένου» **<map>** και **<area>** για τον καθορισμό των ορίων των επιλογών. Αντίθετα η XHTML 2 δεν χρησιμοποιεί κάποιο πρόσθετο στοιχείο, αλλά εκμεταλλεύεται το γεγονός, ότι ένα στοιχείο μπορεί να αλλάξει ρόλους μέσω της ιδιότητας **@role**. Συνεπώς αναφερόμαστε στην εικόνα μέσω της ιδιότητας **@src** μιας αταξινομήτης λίστας και τα όρια καθορίζονται μέσα από τα στοιχεία της λίστας, χρησιμοποιώντας την ιδιότητα **@href**, **@shape** και **@coords**.

### 5.2.5. Υπέρ – Σύνδεσμοι

Οι υπέρ-σύνδεσμοι, χρησιμοποιούνται για πολλούς λόγους σε ένα web έγγραφο. Μπορεί να χρησιμοποιηθούν συμπληρώνοντας το σύστημα πλοήγησης, προσφέροντας έτσι έναν γρήγορο τρόπο μετάβασης στις εσωτερικές σελίδες ενός ιστοτόπου. Εξάλλου και το μενού πλοήγησης αποτελείται από υπέρ-συνδέσμους.



Όμως προσφέρουν και μια πολύτιμη υπηρεσία, που σαν αντίκτυπο έχει την αύξηση της δημοτικότητας μιας σελίδας στα αποτελέσματα των μηχανών αναζήτησης. Αυτοί οι υπερ-σύνδεσμοι καλούνται backlinks και αποτελούν συνδέσμους με λέξεις κλειδιά προς εξωτερικές σελίδες.

## HTML 5.0

```
<article id="elemHyperlinks">
  <header>
    <h1>Υπέρ-Συνδέσμοι</h1>
  </header>
  <h2>Παραδείγματα Υπέρ-Συνδέσμων</h2>
  <h3>Απλός σύνδεσμος</h3>
  Σύνδεσμος στο in.gr:
  <a href="http://www.in.gr" target="_blank" >in.gr σύνδεσμος</a>
  <hr/>
  <h3>Εικόνα με υπερσύνδεσμο</h3>
  <a href="http://www.google.com">
  
  <hr/>
</article>
```

## XHTML 2.0

```
<article id="elemHyperlinks">
  <section>
    <h>Υπέρ-Συνδέσμοι</h>
  </section>
  <h>Παραδείγματα Υπέρ-Συνδέσμων</h>
  <section>
    <h>Απλός σύνδεσμος</h>
    Σύνδεσμος στο in.gr:
    <a href="http://www.in.gr" target="_blank" >in.gr σύνδεσμος</a>
  </section>
  <seperator/>
  <section>
    <h>Εικόνα με υπερσύνδεσμο</h>
    Σύνδεσμος στο Google</img>
  </section>
  <seperator/>
</section>
</article>
```

Η ουσιαστική διαφορά των υπέρ-συνδέσμων ανάμεσα στις δύο γλώσσες είναι ότι στην XHTML 2.0, όπως ήδη έχουμε πει οποιοδήποτε στοιχείο μπορεί να είναι υπερ-

σύνδεσμος και αυτό προσφέρει ένα συγκριτικά μεγαλύτερο πλεονέκτημα για τη συγγραφή καθαρότερου κώδικα.

### 5.2.6. Υποσέλιδο

Το υποσέλιδο, μπαίνει στο τέλος μίας σελίδας και συνήθως περιέχει πληροφορίες για τα πνευματικά δικαιώματα του ιστοτόπου.

#### HTML 5.0

```
<footer class=clearfix>
  <h1>Υποσέλιδο</h1>
  <p class=copyright><strong>Πτυχιακή εργασία στην HTML5 & XHTML2.</strong> (2013)</p>
</footer>
```

#### XHTML 2.0

```
<section id="footer">
<h>Υποσέλιδο</h>
  <p class=copyright><strong>Πτυχιακή εργασία στην HTML5 & XHTML2.</strong> (2013)</p>
</section>
```

Η λογική του υποσέλιδου για τις δύο γλώσσες, είναι η ίδια με αυτή της επικεφαλίδας. Η HTML 5 χρησιμοποιεί ξεχωριστή ετικέτα **<footer>**, ενώ η XHTML 2 το αντιμετωπίζει σαν μια ενότητα της σελίδας. Ο ορισμός ξεχωριστής ετικέτας, αποτελεί σημασιολογικό πλεονέκτημα, αφού μία μηχανή αναζήτησης μπορεί να καταλάβει ότι το συγκεκριμένο τμήμα αποτελεί πράγματι το υποσέλιδο.

## Συμπεράσματα

Η εξέλιξη της τεχνολογίας στο διαδίκτυο, οδήγησε στην ανάγκη ανάπτυξης νέων προτύπων. Ο πόλεμος αυτή τη φορά όμως δεν αφορούσε τους φυλλομετρητές, αλλά τις τεχνολογίες σήμανσης. Τα τέλη του 2009 ήταν ενεργά δύο πρότυπα τα οποία φερόντουσαν ως οι διάδοχοι της κλασικής HTML και μάλιστα με διαφορές τόσο στον κώδικα (στοιχεία, ιδιότητες), όσο και στην φιλοσοφία τους.

Όπως είδαμε λόγω της ραγδαίας εξέλιξης του διαδικτύου, κάποιοι κατασκευαστές (λογισμικού πλοήγησης κυρίως), άρχισαν να προσθέτουν επιπλέον ετικέτες και ιδιότητες στην HTML, καθιστώντας την από αυστηρή γλώσσα σήμανσης σε γλώσσα παρουσίασης. Ιδιαίτερα δε ο πόλεμος ανάμεσα στους φυλλομετρητές ιστούς επιδείνωσε την κατάσταση η οποία ξέφυγε από κάθε έλεγχο. Έτσι μετά τις εκδόσεις V2.0, V3.0, της HTML παρουσιάστηκε η ανάγκη προτυποποίησης και μια ομάδα αποτελούμενη κυρίως από κατασκευαστές πήραν την κατάσταση στα χέρια τους και μέσω του W3C άρχισε η προτυποποίηση για τις επόμενες εκδόσεις.

Ουσιαστικά αυτή η περίοδος ήταν που κυριάρχησε η HTML 4.01 ως νεότερη έκδοση γλώσσας σήμανσης για το διαδίκτυο, ενώ κάποιοι άλλοι κατασκευαστές άρχισαν να χρησιμοποιούν την XML που ήταν μια αναδιατύπωση της HTML 4.0.

Η εξέλιξη όμως της τεχνολογία οδήγησε στην ανάγκη κατασκευής νέων προτύπων και έτσι το επίσημο W3C ξεκινάει το προσχέδιο της XHTML2, ενώ ταυτόχρονα ανεξάρτητοι κατασκευαστές δημιουργούν μία κοινότητα ανάπτυξης και ξεκινούν το προσχέδιο της HTML5 το οποίο παραδίδεται σε κάποια στιγμή στο W3C.

Στην αρχή και οι δύο προτάσεις έδειχναν πολλά υποσχόμενες και υιοθέτησαν αρκετά χαρακτηριστικά για να λύσουν κοινά κατασκευαστικά προβλήματα. Η HTML5 όμως είχε ένα συγκριτικό πλεονέκτημα, το οποίο ήταν η υποστήριξη των κατασκευαστών, αφού από αυτούς ξεκίνησε. Ήδη τα περισσότερα στοιχεία της HTML5 άρχισαν να ενσωματώνονται στα συστήματα πλοήγησης των εταιριών και πολλές νέες σελίδες υιοθέτησαν τα νέα στοιχεία του προτύπου. Αυτό είχε σαν αποτέλεσμα από τη μία, τη γρήγορη ανάπτυξη του προτύπου και την υιοθέτηση όλων και περισσότερων χαρακτηριστικών, ενώ από την άλλη επισκιάστηκε η ανάπτυξη της XHTML 2.0. Αυτός ήταν και ο κύριος λόγος που τελικά το XHTML 2.0 πρωτόκολλο εγκαταλείφτηκε και παρέμεινε προσχέδιο. Ουσιαστικά δεν ήταν η σύγκριση

καθεαυτή, η οποία ανέδειξε τον νικητή, αλλά η δημοτικότητα που εξασφάλισε από την αρχή η HTML 5.0.

Στο σύνολο των νέων γλωσσών η σύγκριση φαίνεται να είναι άνιση, καθώς η XHTML 2.0 αποτελεί ένα υποσύνολο από πλευράς ετικετών, αλλά και λειτουργιών σε σχέση με την HTML 5.0.

Θα μπορούσαμε να πούμε ότι η HTML 5.0 είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης για το διαδίκτυο σε συνδυασμό με τη χρήση JavaScript και CSS3. Επιπροσθέτως παρέχει αρκετές ευκολίες στους κατασκευαστές για ποικίλες υλοποιήσεις από τις πιο απλές έως τις πιο περίπλοκες συν το γεγονός ότι διατηρεί την συμβατότητα προς τα πίσω.

Από την άλλη η XHTML 2.0, φαίνεται να υπερέχει σαν καθαρή γλώσσα σήμανσης, αφού δίνει την δυνατότητα για παραγωγή πιο αυστηρού και δομημένου κώδικα, διαχωρίζοντας την σήμανση από την παρουσίαση.

Τέλος μπορούμε να πούμε ότι ουσιαστικά δεν μπορούμε να κρίνουμε με σιγουριά ποιος θα ήταν καλύτερος τρόπος ανάπτυξης, καθώς οι δύο γλώσσες είχαν από την αρχική τους ανάπτυξη, διαφορετικούς στόχους. Μπορούμε όμως με σιγουριά να πούμε ότι σύμφωνα με τα σημερινά δεδομένα, νικητής είναι η HTML 5.0, αφού είναι η μόνη που συνεχίζει να αναπτύσσεται και θα αποτελέσει το νέο πρότυπο στις γλώσσες σήμανσης.

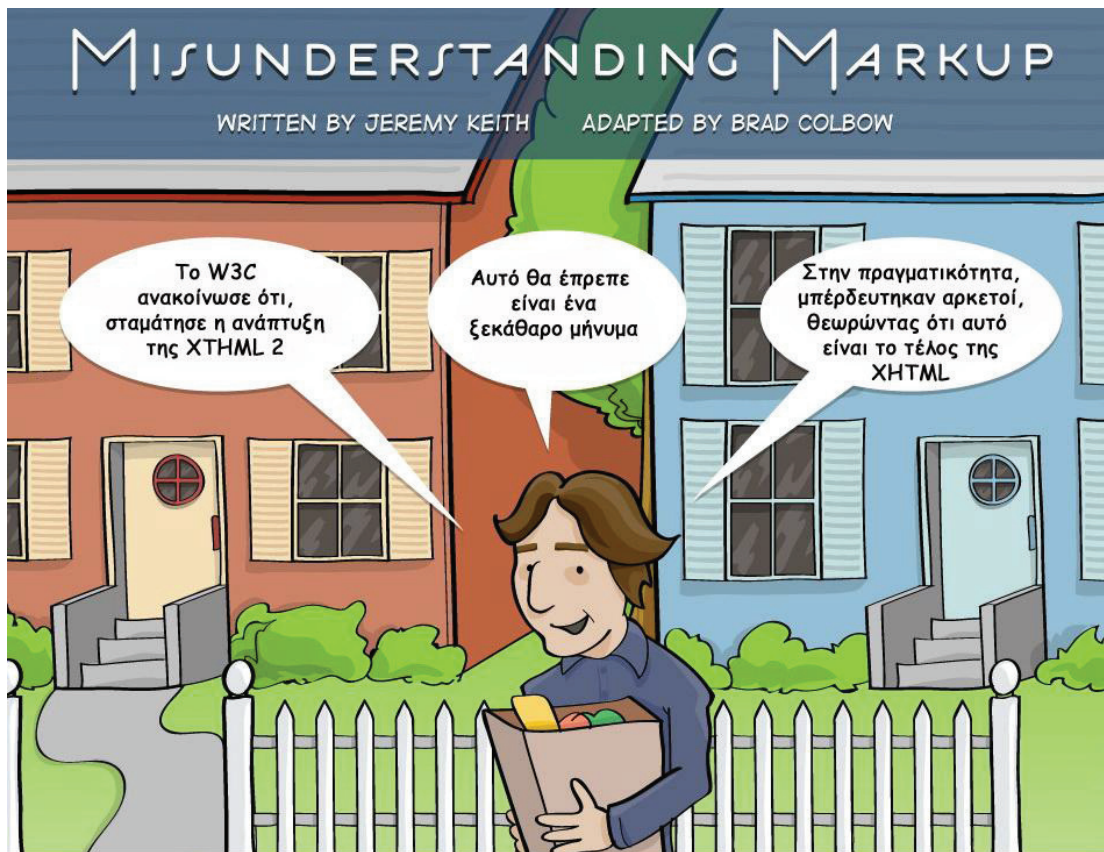
Θα μπορούσαμε να ελπίζουμε ότι η HTML5 στο τελικό στάδιο ανάπτυξης, θα μπορούσε να υιοθετήσει μια πιο αυστηρή σύνταξη όπως η XHTML 2.0, δίνοντας έμφαση περισσότερο στη σήμανση. Έτσι στοιχεία όπως `<nl>`, `<separator>` και ιδιότητες όπως `href`, `@role` θα έδιναν στην HTML 5 μια πιο ξεκάθαρη μορφή γλώσσας σήμανσης. Ακόμα όμως και στο επίπεδο οργάνωσης ενός εγγράφου, οι εμφωλευμένες ενότητες της XHTML 2.0 θα ήταν σαφώς ένας καλύτερος τρόπος δόμησης.

Κλείνοντας την πτυχιακή μας εργασία παραθέτουμε ένα κόμικ, που προσεγγίζει συνοπτικά το θέμα που αναπτύξαμε, δηλαδή την σύγκριση των δύο προτύπων.

# MISUNDERSTANDING MARKUP

WRITTEN BY JEREMY KEITH

ADAPTED BY BRAD COLBOW







Πάρε αυτή την γραμμή

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

Αντικατέστησε με:

```
<!DOCTYPE html>
```



## Βιβλιογραφία

- Anon., 2001. *XHTML Mobile Profile*. [Ηλεκτρονικό]  
Available at: <http://www.openmobilealliance.org/tech/affiliates/wap/wap-277-xhtmlmp-20011029-a.pdf>  
[Πρόσβαση 09 11 2012].
- Anon., 2010. *XHTML - Print*. [Ηλεκτρονικό]  
Available at: <http://www.w3.org/TR/xhtml-print/xhtml-print.pdf>  
[Πρόσβαση 09 11 2012].
- Anon., 2013. *Network socket*. [Ηλεκτρονικό]  
Available at: [http://en.wikipedia.org/wiki/Network\\_socket](http://en.wikipedia.org/wiki/Network_socket)  
[Πρόσβαση 01 15 2013].
- Anon., n.d. *LaTeX - A document preparation system*. [Ηλεκτρονικό]  
Available at: <http://www.latex-project.org>  
[Πρόσβαση 03 11 2012].
- Bidelman, E., 2011. *Using the HTML5 Filesystem API*. 1η Έκδοση επιμ. Sebastopol: O'Reilly.
- Corderoy, R., n.d. *The Text Processor for Typesetters*. [Ηλεκτρονικό]  
Available at: <http://www.troff.org>  
[Πρόσβαση 03 11 2012].
- Duckett, J., 2010. *Beginning HTML, XHTML, CSS and JavaScript*. Indianapolis: Wiley Publishing, Inc.
- Freeman, A., 2011. *The Definitive Guide to HTML 5*. 1η Έκδοση επιμ. s.l.:Apress.
- Fulton, S. & Fulton, J., 2011. *HTML5 Canvas*. Sebastopol: O'Reilly.
- Fulton, S. & Fulton, J., 2011. *HTML5 Canvas*. Sebastopol: O'Reilly.
- Harris, A., 2011. Part 4: New and modified form elements. Στο: *HTML5 for dummies*. New Jersey: Willey Publishing, Inc, pp. 58-70.
- Leah, B., 2013. *HTML5 - code example of File API - drag and drop hard drive files to a webpage!*. [Ηλεκτρονικό]  
Available at:



[https://www.ibm.com/developerworks/mydeveloperworks/blogs/bobleah/entry/html5\\_code\\_example\\_of\\_file\\_api\\_drag\\_drop\\_hard\\_drive\\_files\\_to\\_a\\_webpage28?lang=en](https://www.ibm.com/developerworks/mydeveloperworks/blogs/bobleah/entry/html5_code_example_of_file_api_drag_drop_hard_drive_files_to_a_webpage28?lang=en)

[Πρόσβαση 19 01 2013].

- MacDonald, M., 2011. *HTML5 The Missing Manual*. Cambridge: O'Reilly.
- McLaughlin, B., 2011. *What is HTML 5*. s.l.:O'Reilly.
- Meyer, J., 2011. *HTML5 and Javascript Projects*. New York: APress.
- Sharp, R., 2011. *Methods of communication*. [Ηλεκτρονικό] Available at: <http://html5doctor.com/methods-of-communication/> [Πρόσβαση 01 12 2012].
- Sharp, R., n.d. *HTML5 Doctor*. [Ηλεκτρονικό] Available at: <http://html5doctor.com/methods-of-communication/> [Πρόσβαση 15 01 2013].
- Vossen, G. & Hagemann, S., 2007. Chapter 1 - A Brief History of the Web. Στο: *Unleashing Web 2.0*. s.l.:Morgan Kaufmann, pp. 1-68.
- Πομπόρτσος, Α., 2009. *Εισαγωγή στις γλώσσες σήμανσης*. 1η Έκδοση επιμ. Θεσσαλονίκη: Τζιόλα.