

Τμήμα
Μηχανικών
Πληροφορικής τ.ε.
Τεχνολογικό Εκπαιδευτικό Ίδρυμα
Δυτικής Ελλάδας

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

*Ποιοτικά χαρακτηριστικά, μετρικές και
εργαλεία για CSS3*

*Quality characteristics, metrics and tools for
CSS3*

ΚΑΝΤΕΛΤΖΗΣ ΑΘΑΝΑΣΙΟΣ

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ : ΧΡΙΣΤΟΔΟΥΛΟΥ ΣΩΤΗΡΗΣ
Καθηγητής Εφαρμογών

ΑΝΤΙΠΡΙΟ 2014



Τ.Ε.Ι. Δυτικής Ελλάδας
Τμήμα Μηχανικών Πληροφορικής Τ.Ε

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή

Αντίρριο, Ημερομηνία

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. Ονοματεπώνυμο, Υπογραφή
2. Ονοματεπώνυμο, Υπογραφή
3. Ονοματεπώνυμο, Υπογραφή

Περίληψη

Στην παρούσα πτυχιακή εργασία, μελετήσαμε τα *Διαδοχικά Φύλλα Στυλ – Cascading Style Sheets (CSS)* και επικεντρωθήκαμε στις νέες ιδιότητες που εισήγαγε η 3^η έκδοση. Συγκεκριμένα στο 1^ο κεφάλαιο κάναμε μια ιστορική αναφορά της εξέλιξης της CSS αφού πρώτα κατανοήσαμε τον τρόπο που εμφανίζεται το περιεχόμενο μιας ιστοσελίδας στον Ιστό. Στην συνέχεια, στο 2^ο κεφάλαιο, μελετήσαμε τα δομικά στοιχεία της γλώσσας CSS ενώ αναφέραμε και ιδιότητες μορφοποίησης που έχει μόνο η 3^η έκδοση. Για την σωστή συγγραφή κώδικα CSS αναφέραμε διάφορους οδηγούς (Κεφάλαιο 3ο) οι οποίοι βασίζονται σε μετρικές της CSS και τους οποίους πρέπει να ακολουθούν οι προγραμματιστές. Στο 4^ο κεφάλαιο, χρησιμοποιήσαμε διάφορα εργαλεία που σχετίζονται με τις μετρικές που αναλύσαμε στο προηγούμενο κεφάλαιο ενώ στο 5^ο και τελευταίο, παρουσιάσαμε μια δική μας εναλλακτική έκδοση δημοφιλούς αρχικής σελίδας στο Διαδίκτυο, την οποία κατασκευάσαμε με την CSS 3.

Synopsis

In this project, we studied Cascading Style Sheets and we focused to new properties of CSS third release. In first chapter, we made a history report of CSS revolution. In second chapter, we studied the structural components of CSS language and report the differences of CSS third release from previous versions. For the writing of correct code, we referred some helpful guides which are based on CSS's metrics that must be followed by developers. In chapter 4, we used various tools and guides which are related with metrics, which we analyzed in the previous chapter and the fifth and last one chapter we presented an alternative version of our popular home page on the internet, which we built with the CSS 3.

Ευχαριστίες

Περιεχόμενα

Περίληψη.....	4
Ευχαριστίες	6
Περιεχόμενα	8
Κεφάλαιο 1 ^ο	12
1. Εισαγωγή	12
1.1 Ορισμός CSS	13
1.2 Γλώσσα HTML.....	14
1.2.1 Δομή γλώσσας HTML.....	15
1.2.2 Πλεονεκτήματα γλώσσας HTML.....	17
1.2.3 Μειονεκτήματα γλώσσας HTML	17
1.3 Γλώσσα XML	18
1.3.1 Δομή γλώσσας XML	18
1.3.2 Πλεονεκτήματα χρήσης XML.....	20
1.3.3 Μειονεκτήματα χρήσης XML	20
1.4 Γλώσσα XHTML.....	20
1.4.1 Δομή γλώσσας XHTML.....	21
1.4.2 Πλεονεκτήματα χρήσης XHTML.....	23
1.4.3 Μειονεκτήματα χρήσης XHTML.....	23
1.5 Ιστορική αναδρομή των CSS.....	24
1.6 Εκδόσεις CSS	25
1.6.1 Έκδοση επιπέδου CSS 1.....	25
1.6.2 Έκδοση CSS-Positioning (CSS-P)	26
1.6.3 Έκδοση επιπέδου CSS 2.....	26
1.6.4 Έκδοση επιπέδου CSS 2.1	27
1.6.5 Έκδοση επιπέδου CSS 3	27
1.6.6 Έκδοση επιπέδου CSS 4.....	27
1.7 Πλεονεκτήματα χρήσης CSS.....	28
1.8 Μειονεκτήματα χρήσης CSS.....	29
1.9 Σύνοψη	30
Κεφάλαιο 2 ^ο	31
2. Εισαγωγή	31
2.1 Σύνδεση αρχείου HTML με CSS	31
2.2 Κανόνες CSS	34
2.3 Επιλογείς.....	35

2.3.1	Επιλογείς Ετικέτας ή Στοιχείου	35
2.3.2	Επιλογείς Κλάσης.....	36
2.3.3	Επιλογείς ID	37
2.3.4	Καθολικός Επιλογέας	37
2.4	Ενότητες	38
2.5	Ψευδοκλάσεις και Ψευδοστοιχεία	39
2.5.1	Ψευδοκλάσεις Συνδέσμων.....	39
2.5.2	Διαμόρφωση Παραγράφων	39
2.5.3	Ψευδοστοιχεία	40
2.6	Μοντέλο Κουτιού CSS	40
2.7	Ιδιότητες CSS	41
2.7.1	Γενικές Ιδιότητες	41
2.7.2	Ιδιότητες Μορφοποίησης Κειμένου	42
2.7.3	Ιδιότητες Μορφοποίησης Φόντου	43
2.7.4	Ιδιότητες Περιγράμματος	45
2.8	Ιδιότητες CSS 3	46
2.8.1	Μετασχηματισμοί.....	46
2.8.2	Μεταβάσεις.....	47
2.8.3	Κινούμενες εικόνες.....	47
2.9	Σύνοψη	48
Κεφάλαιο 3 ^ο		49
3.	Εισαγωγή	49
3.1	Web Standards	49
3.2	Χαρακτηριστικά Ιστοσελίδων με Web Standards	50
3.2.1	Λογική δόμηση κώδικα	50
3.2.2	Εγκυρότητα κώδικα	50
3.2.3	Προσβασιμότητα κώδικα.....	51
3.2.4	Διαχωρισμός περιεχομένου	51
3.3	Πλεονεκτήματα χρήσης Web Standards.....	52
3.4	Μετρικές.....	52
3.4.1	Απόδοση	53
3.4.1.1	Οδηγός Βελτίωσης της Απόδοσης	54
3.4.2	Προσβασιμότητα	56
3.4.2.1	Οδηγός Προσβασιμότητας Ιστοσελίδων	56
3.4.3	Αποτελεσματικότητα Κώδικα CSS	59
3.4.3.1	Οδηγός Εγγραφής Αποτελεσματικών CSS επιλογέων	59

3.4.4	Συμβατότητα.....	60
3.4.4.1	Οδηγός Επίλυσης Προβλημάτων Συμβατότητας.....	61
3.4.5	Αφαιρετικότητα	63
3.4.5.1	Οδηγός Επίτευξης Αφαιρετικότητας.....	63
3.4.6	Πολυπλοκότητα	66
3.4.6.1	Οδηγός Μέτρησης Πολυπλοκότητας	66
3.5	Σύνοψη	68
Κεφάλαιο 4 ^ο		69
4.1	Εισαγωγή	69
4.2	Εργαλεία	70
4.3	Επικυρωτής CSS W3C	70
4.3.1	Λειτουργία CSS Validator.....	71
4.3.2	Μελέτη CSS Validator	72
4.4	Ελεγκτής Συμβατότητας.....	76
4.4.1	Λειτουργία Ελεγκτή Συμβατότητας	77
4.4.2	Μελέτη Ελεγκτή Συμβατότητας.....	78
4.5	Ελεγκτής Προσβασιμότητας.....	80
4.5.1	Λειτουργία Ελεγκτή Προσβασιμότητας.....	80
4.5.2	Μελέτη Ελεγκτή Προσβασιμότητας.....	81
4.6	Γεννήτρια Παραγωγής Κανόνων CSS.....	85
4.6.1	Λειτουργία Γεννήτριας Κανόνων CSS.....	85
4.7	Οδηγός Συγγραφής Ποιοτικού Κώδικα CSS 3.....	87
4.8	Σύνοψη	88
Κεφάλαιο 5 ^ο		89
5.1	Εισαγωγή	89
5.2	Κατασκευή Σελίδας με CSS 3	89
5.3	Μελέτη Σελίδας και έλεγχος με εργαλεία	91
5.3.1	Έλεγχος Επικυρωτή CSS 3.....	91
5.3.2	Έλεγχος Συμβατότητας	92
5.3.3	Έλεγχος Προσβασιμότητας	93
Βιβλιογραφία.....		Ошибка! Закладка не определена.
ΠΑΡΑΡΤΗΜΑ Α		96
ΠΑΡΑΡΤΗΜΑ Β		100
Κώδικας HTML.....		100
Κώδικας CSS 3.....		105

Κεφάλαιο 1^ο

1. Εισαγωγή

Η ανάγκη των ανθρώπων για αναζήτηση πληροφοριών οδήγησε στην υλοποίηση του διαδικτύου. Ο Παγκόσμιος Ιστός (World Wide Web - WWW) είναι μία αρχιτεκτονική για την προσπέλαση διασυνδεδεμένων εγγράφων τα οποία είναι αποθηκευμένα σε εκατομμύρια διάσπαρτους υπολογιστές σε όλο το διαδίκτυο (Internet) και έχουν σαν στόχο να ικανοποιήσουν αυτή την ανάγκη του ανθρώπου, για πληροφόρηση. Καθώς οι απαιτήσεις για πλοήγηση σε ιστοχώρους, αυξάνεται καθημερινά, οι επιστήμονες των υπολογιστών, ωθήθηκαν στην κατασκευή πιο εκλεπτυσμένων φυλλομετритών (browsers) καθώς και στην ανάπτυξη νέων τεχνολογιών διαδικτύου, επιτρέποντας τους χρήστες να έχουν μια βαθύτερη και πιο αλληλεπιδραστική εμπειρία με τις ιστοσελίδες που επισκέπτονται.

Οι ιστοσελίδες πρέπει να προσφέρουν εύκολη πρόσβαση στους χρήστες, ενώ απαραίτητη θεωρείται η δυνατότητά τους να μεταφέρονται σε διαφορετικά συστήματα. Φυσικά, όλα τα συστήματα δεν μπορούν να έχουν τις ίδιες προδιαγραφές για να εμφανίζουν έγγραφα με πληροφορίες, γι' αυτό τον λόγο αναπτύχθηκαν οι **Γλώσσες Σήμανσης (markup language)**. Οι συγκεκριμένες γλώσσες αποτελούνται από ένα σύνολο εντολών που επιτρέπουν σε διάφορες πηγές δεδομένων να προσθέτουν πληροφορίες οι οποίες μπορεί να είναι κείμενα, εικόνες, αρχεία ήχου. Οι εντολές αυτές ονομάζονται Ετικέτες (tags).

Πολλές από τις σημερινές ιστοσελίδες, ανταγωνίζονται επάξια την λειτουργικότητα που προσφέρουν εφαρμογές εγκαταστημένες στον υπολογιστή. Από διαδικτυακούς κειμενογράφους σε ιστοσελίδες αναπαραγωγής αρχείων μουσικής και βίντεο, τα όρια για τον σχεδιασμό δικτυακών εφαρμογών που υπήρχαν την προηγούμενη δεκαετία, έχουν καταρριφθεί. Αυτό δίνει τεράστια δύναμη στους σχεδιαστές να υλοποιούν εφαρμογές ακριβώς όπως τις οραματίζονταν, ενώ παράλληλα αποτελεί πρόκληση για ανάπτυξη νέων εξελιγμένων τεχνολογιών που θα αυξάνουν την ποιότητα και την αισθητική εικόνα των ιστοσελίδων.



Εικόνα 1.1

Μια τέτοια τεχνολογία είναι τα *Διαδοχικά Φύλλα Στυλ – Cascading Style Sheets (CSS)*, ένα πολύ χρήσιμο εργαλείο για τους σχεδιαστές και κατασκευαστές σελίδων διαδικτύου, μιας και τους δίνει την δυνατότητα να αναπτύσσουν εφαρμογές εύκολες στην διαχείριση, επεκτάσιμες καθώς και άκρως ελκυστικές στους χρήστες. Ο συνδυασμός δε με άλλες τεχνολογίες όπως με την *Γλώσσα Σήμανσης Υπερκειμένου – Hyper Text Markup Language (HTML)*, μπορεί να οδηγήσει στην δημιουργία μοναδικών ιστοσελίδων με περιεχόμενο που μεταβάλλεται δυναμικά ακόμα και από χρήστες που δεν διαθέτουν γνώσεις αυτών των τεχνολογιών.

1.1 Ορισμός CSS

Προσπαθώντας να ορίσουμε τι ακριβώς είναι τα Διαδοχικά Φύλλα Στυλ (CSS) επισκεφτήκαμε την Βικιπαίδεια όπου ορίζονται ως εξής :

“..είναι μια γλώσσα υπολογιστή που ανήκει στην κατηγορία των γλωσσών φύλλων στυλ που χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που έχει γραφτεί με μια γλώσσα σήμανσης. Χρησιμοποιείται δηλαδή για τον έλεγχο της εμφάνισης ενός εγγράφου που γράφτηκε στις γλώσσες HTML και XHTML, δηλαδή για τον έλεγχο της εμφάνισης μιας ιστοσελίδας και γενικότερα ενός ιστοχώρου. Η CSS είναι μια γλώσσα υπολογιστή προορισμένη να αναπτύσσει στυλιστικά μια ιστοσελίδα δηλαδή να διαμορφώνει περισσότερα χαρακτηριστικά, χρώματα, στοίχιση και δίνει περισσότερες δυνατότητες σε σχέση με την HTML.”

Καταλαβαίνουμε λοιπόν πως τα CSS χρησιμοποιούνται καθαρά για την αισθητική παρουσίαση του περιεχομένου μιας ιστοσελίδας που έχει βάση στην γλώσσα HTML. Θεωρείται απαραίτητο, λοιπόν, να αναφέρουμε κάποια βασικά χαρακτηριστικά των γλωσσών HTML, XML και XHTML τα οποία θα μας βοηθήσουν να κατανοήσουμε τόσο την λειτουργία όσο και χρήση των CSS, καλύτερα.

1.2 Γλώσσα HTML

Την δεκαετία του '90, ο Tim Berners -Lee, φυσικός στο Ευρωπαϊκό Κέντρο Πυρηνικών Ερευνών (CERN), ορμώμενος από την ανάγκη επικοινωνίας διαφορετικών ομάδων επιστημόνων, πρότεινε την ιδέα των διασυνδεδεμένων εγγράφων για την μεταξύ τους επικοινωνία, πράγμα που αποτέλεσε τη βάση της επικοινωνίας του Παγκόσμιου Ιστού (WWW). Έτσι σε μικρό χρονικό διάστημα, εκδόθηκε και το πρώτο πρωτότυπο της επικοινωνίας αυτής, που ονομάστηκε **Γλώσσα Σήμανσης Υπερκειμένου – Hyper Text Markup Language (HTML)**. Στην αρχή, η HTML, ήταν μια απλή δομημένη μορφή εγγράφου με ετικέτες σήμανσης (tags) που τοποθετούνταν μεταξύ των σειρών κειμένου και περιείχε στοιχεία περισσότερο λογικά παρά στοιχεία παρουσίας του κειμένου. Για παράδειγμα ήταν δυνατό να οριστεί ότι κάποιο κομμάτι κειμένου ήταν επικεφαλίδα αλλά δεν μπορούσε η HTML να δηλώσει με ποια μορφή αυτή η επικεφαλίδα θα εμφανιζόταν στους χρήστες. Η παρουσίαση του κειμένου, δηλαδή η γραμματοσειρά, το χρώμα και το μέγεθος που χρησιμοποιούταν, προσδιοριζόταν αρχικά από το πρόγραμμα περιήγησης (Web browser).

Γι' αυτό τον λόγο, ο Παγκόσμιος Ιστός και η χρήση της HTML άρχισε να αυξάνεται αλματωδώς, μόνο μετά την κατασκευή του Mosaic, του πρώτου προγράμματος περιήγησης. Ο Mosaic αναπτύχθηκε από την ομάδα του Marc Andreessen στο Πανεπιστήμιο του Ιλινόις και παρουσιάστηκε τον Φεβρουάριο του 1993. Ο ίδιος, ένα χρόνο μετά, μαζί με τον Jim Clark, δημιουργεί την εταιρία Netscape Communications Corp και επικεντρώνονται στην ανάπτυξη εφαρμογών για τον Ιστό, ενώ τον ίδιο χρόνο εκδίδεται νέα έκδοση του πρότυπου, η HTML 2. Το 1995 έρχεται σε κυκλοφορία ο Netscape, ο νέος περιηγητής της εταιρίας. Την ίδια χρονιά οι επιστήμονες της Microsoft άρχισαν την ανάπτυξη του Internet Explorer εκμεταλλευόμενοι της νέες δυνατότητες που είχαν προστεθεί στην 3^η έκδοση της HTML. Η εξέλιξη της γλώσσας συνεχίστηκε με την 4^η έκδοση στα τέλη του 1999. Οι browsers της σημερινής εποχής διαβάζουν πλέον αρχεία HTML που είναι γραμμένα βάσει της τελευταία (δοκιμαστικής) έκδοσης του προτύπου της HTML 5, που ανακοινώθηκε τον Μάιο του 2011, ενώ η W3C¹ αναπτύσσει μια ολοκληρωμένη σουίτα δοκιμών για να επιτευχθεί ευρεία διαλειτουργικότητα για την πλήρη έκδοση, μέχρι το τέλος του 2014.

¹Το 1994 το CERN και το MIT συμφώνησαν στη δημιουργία του οργανισμού World Wide Web Consortium (W3C). Στόχος του W3C είναι η συνεχής ανάπτυξη του Παγκόσμιου Ιστού, η ανάπτυξη προτύπων και η διαλειτουργικότητα μεταξύ των διαφόρων ιστοχώρων.

1.2.1 Δομή γλώσσας HTML

Όπως ήδη αναφέραμε η γλώσσα HTML αποτελείται από ετικέτες (tags) που έχουν την παρακάτω μορφή :

<ετικέτα σήμανσης> περιεχόμενο κειμένου< /ετικέτα σήμανσης >

Έτσι για παράδειγμα, αν θέλουμε να δηλώσουμε ότι ξεκινάμε την εγγραφή ενός αρχείου τύπου html τότε αρκεί να γράψουμε στην αρχή <HTML> και αφού ολοκληρώσουμε το αρχείο μας </HTML>. Άλλες σημαντικές ετικέτες είναι η <HEAD></HEAD> όπου εκεί μπορούμε να βρούμε πληροφορίες όπως ο τίτλος και λέξεις κλειδιά που θέλουμε να εμφανίζονται στο πάνω μέρος του browser. Τέλος, το περιεχόμενο που φαίνεται στην ιστοσελίδα και απεικονίζονται στην οθόνη του υπολογιστή, δηλώνεται μέσα σε ετικέτες <BODY></BODY>. Για την αναγνώριση της HTML 5 από τους browsers, στην αρχή του εγγράφου βάζουμε την εντολή <!DOCTYPE html> μέσω της οποίας ορίζεται ότι ακολουθούνται τα τεχνολογικά πρότυπα (web standards). Άλλες ετικέτες βάσει του είδους τους είναι :

Ετικέτα Επικεφαλίδας	Λειτουργία
<h1> ... </h1> <h2> ... </h2> <h3> ... </h3> <h4> ... </h4> <h5> ... </h5> <h6> ... </h6>	Επικεφαλίδες Κειμένου Όσο μεγαλύτερος είναι ο αριθμός μετά το ήτσο μικρότερη είναι η γραμματοσειρά της επικεφαλίδας
Ετικέτα Διαχωριστικού <p>... </p> <hr>	Αλλαγή γραμμής Αλλαγή παραγράφου Οριζόντια γραμμή που χωρίζει την σελίδα στα δύο
Ετικέτα Μορφοποίησης ... <i>...</i> <u>...</u> ²	Κείμενο σε έντονη γραφή Κείμενο σε πλάγια γραφή Υπογραμμισμένο κείμενο
Ετικέτα Λίστας	Μη αριθμημένη λίστα Αριθμημένη λίστα Στοιχείο λίστας
Ετικέτα Εικόνας 	Εμφάνιση εικόνας
Ετικέτα Εικόνας Κείμενο	Παραπομπή σε άλλη ιστοσελίδα

² Η συγκεκριμένη εντολή δεν αναγνωρίζεται στην τελευταία έκδοση της HTML 5

Η δημιουργία ενός αρχείου τύπου html είναι πολύ εύκολη. Σε έναν οποιοδήποτε κειμενογράφο γράφουμε το κείμενο με τις ετικέτες που αναφέραμε προηγουμένως, αποθηκεύουμε το αρχείο ως **όνομα_αρχείου.html** και στην συνέχεια μπορούμε να το ανοίξουμε με ένα οποιοδήποτε πρόγραμμα περιήγησης όπως τον Chrome, Firefox, Opera, Internet Explorer, Safari.

Κάνοντας χρήση του προγράμματος Notepad++ γράφουμε το αρχείο σε γλώσσα html και το αποθηκεύουμε όπως αναφέραμε προηγουμένως. Είναι πολύ σημαντικό να αναφέρουμε επίσης, πως πρέπει να επιλεγθεί κωδικοποίηση UTF-8 για να μπορεί ο browser να διαβάσει τους ελληνικούς χαρακτήρες.

Ο κώδικας του αρχείου html είναι:

```
<!DOCTYPE html>
<html>
  <head>
    <title>CSS</title>
  </head>
  <body>
    <h1>Εργασία για CSS</h1>
    <p>Αυτή είναι μια παράγραφος που αφορά τα CSS
    <br> Θα αναλύσουμε θέματα όπως:
      <ol>
        <li>Την ιστορία των CSS</li>
        <li>Τον ορισμό των CSS</li>
        <li>Μετρικές των CSS</li>
      </ol>
    <img src = images.jpg></p>
    <p><h4><i>Για περισσότερες επισκευτείτε το <a
href="http://www.w3schools.com/">W3Schools</a><i></h4></p>
  </body>
</html>
```

Στον περιηγητή μας ο παραπάνω κώδικας λαμβάνει την εξής μορφή:



Εικόνα 2

Παρατηρούμε πως τυπώνεται με διαφορετικό μέγεθος η επικεφαλίδα h1 από την h4. Παράλληλα, βλέπουμε τον τρόπο εμφάνισης της λίστας, της εικόνας, ενώ πατώντας την λέξη W3Schools μεταφερόμαστε σε νέα καρτέλα στην αντίστοιχη διεύθυνση.

1.2.2 Πλεονεκτήματα γλώσσας HTML

Πολλά είναι τα πλεονεκτήματα που προσφέρει η γλώσσα HTML και γι' αυτό ακριβώς τον λόγο είναι ευρέως διαδεδομένη και χρησιμοποιείται από εκατομμύρια χρήστες. Αρχικά να αναφέρουμε, πως λόγω της δομής της, δίνει την δυνατότητα τροποποίηση σε οποιονδήποτε χρήστη, ανεξαρτήτως εμπειρίας σε γλώσσες προγραμματισμού. Σε αυτό βοηθάει το γεγονός ότι είναι πολύ εύκολη η εκμάθησή της.

Πληροφορίες χρήσης καθώς και νέες ιδιότητες που προσθέτει η ομάδα ανάπτυξης, ανανεώνονται συνεχώς χωρίς ο χρήστης να αναμένει την δημοσίευση νέας έκδοσης. Επιπλέον, χρήστες από διαφορετικά σημεία του πλανήτη μπορούν να εργαστούν πάνω στο ίδιο έγγραφο. Τέλος, το σημαντικότερο ίσως είναι ότι πλέον, αν όχι όλοι αλλά οι περισσότεροι browsers υποστηρίζουν την HTML.

1.2.3 Μειονεκτήματα γλώσσας HTML

Τα θετικά χαρακτηριστικά και η ελεύθερη δομή της γλώσσας μπορεί να γίνουν κάλλιστα μειονεκτήματα της HTML. Αυτό λόγω του ότι ο οποιοσδήποτε έχει την δυνατότητα να τροποποιήσει το περιεχόμενο ενός αρχείου. Δυστυχώς, οι ιστοσελίδες που είναι γραμμένες σε HTML δεν διαθέτουν δυναμικό περιεχόμενο, γι' αυτό ο χρήστης είναι υποχρεωμένος να κάνει συνεπώς αλλαγές χειρωνακτικά, αν θέλει το περιεχόμενο της σελίδας του να είναι πάντα ενημερωμένο.

Παρόλο που οι περισσότεροι browsers υποστηρίζουν αρχεία HTML, δεν εμφανίζουν το περιεχόμενό τους με τον ίδιο τρόπο. Επιπλέον μειονεκτήματα είναι :

- ❖ Η έλλειψη ευελιξίας σε σχέση με άλλα πρότυπα και τεχνολογίες
- ❖ Η έλλειψη κεντροποιημένης επεξεργασίας, αφού κάθε μία σελίδα πρέπει να επεξεργαστεί ατομικά
- ❖ Η περιορισμένες δυνατότητες για αλλαγή στυλ και τέλος,
- ❖ Η προσθήκη διαφορετικών ετικετών από εταιρείες όπως η Microsoft, οι οποίες δεν υποστηρίζονται από τους περισσότερους browsers

1.3 Γλώσσα XML

Η XML (**Extensible Markup Language**) είναι μία γλώσσα σήμανσης και αφορά περισσότερο την σήμανση εγγράφων που έχουν πληροφορίες με κάποια συγκεκριμένη δομή σύνταξης αλλά δεν περιέχει προκαθορισμένες ετικέτες. Ορίζεται, κυρίως, στην προδιαγραφή XML 1.0, που δημιούργησε ο διεθνής οργανισμός προτύπων W3C, αλλά και σε διάφορες άλλες σχετικές προδιαγραφές ανοιχτών προτύπων. Η XML σχεδιάστηκε δίνοντας έμφαση στην απλότητα, τη γενικότητα και τη χρησιμότητα στο Διαδίκτυο.

Πρόκειται για υποσύνολο της Standard Generalized Markup Language³ (SGML) και κατασκευάστηκε με απώτερο σκοπό να καλύψει ανάγκες που η γλώσσα HTML αδυνατούσε να ικανοποιήσει, όπως η υπερβολική χρήση ετικετών σε ένα τύπο εγγράφου. Τα αποτελέσματα μιας αναζήτησης σε ένα αρχείο τύπου XML είναι πιο ακριβή λόγω της συγκεκριμένης δομής του περιεχομένου του. Επιπλέον δίνει την δυνατότητα στον χρήστη να διαχωρίζει αρχεία ιδίου τύπου βάσει κάποιον χαρακτηριστικών όπως ο τίτλος ή ο συντάκτης του εγγράφου. Τέλος, η βασικότερη ίσως ιδιότητα ενός XML αρχείου είναι η εύκολη μεταφορά στον Ιστό, ανεξαρτήτως της πλατφόρμας και του συστήματος που χρησιμοποιείται.

1.3.1 Δομή γλώσσας XML

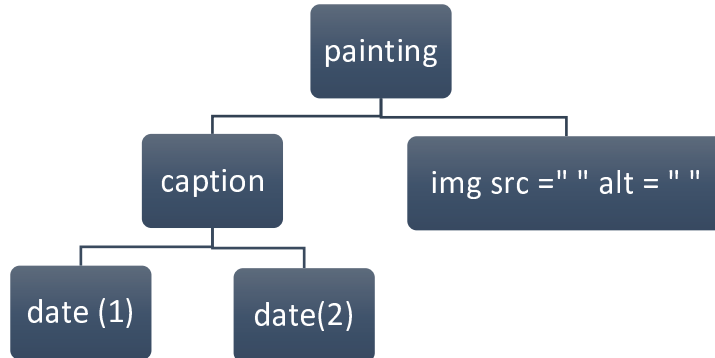
Η δομή ενός αρχείου XML είναι αυστηρά προκαθορισμένη. Για μεγαλύτερη ευκολία μπορεί να αναπαρασταθεί με δεντρική δομή στην οποία η ρίζα (root) του δένδρου απεικονίζει το πρώτο στοιχείο του εγγράφου. Τα υπόλοιπα στοιχεία χωρίζονται σε δύο κατηγορίες σε αυτά που περιέχουν δεδομένα και ονομάζονται γονικές (parent) και σε αυτές που τα στοιχεία τους περιέχονται μέσα σε άλλες και ονομάζονται παιδιά (child elements). Τα στοιχεία που μοιράζονται τον ίδιο γονέα ονομάζονται αδέρφια (siblings). Ένα απλό παράδειγμα XML αρχείου αποτελεί ο παρακάτω κώδικας:

```
<?xml version="1.0" encoding='UTF-8'?>
<painting>

<caption>This is Raphael's "Foligno" Madonna, painted in
<date>1511</date>-<date>1512</date>.</caption>
</painting>
```

³Η SGML, αποτελεί το διεθνές πρότυπο (ISO 8879) γενικευμένης γλώσσας σήμανσης. Διαχειρίζεται μαζικές συλλογές εγγράφων, όπως είναι τα εγχειρίδια συντήρησης των αεροσκαφών F16 ή τα εγχειρίδια συντήρησης των πυρηνικών υποβρυχίων

Το παράδειγμα αυτό αν θέλαμε να απεικονιστεί σε δεντρική μορφή θα ήταν όπως φαίνεται στο διάγραμμα (1). Τα δύο στοιχεία date, είναι παιδιά του στοιχείου caption, το οποίο είναι παιδί του στοιχείου painting που αποτελεί την ρίζα του δένδρου. Το στοιχείο img έχει δύο χαρακτηριστικά, τα src και alt.



Διάγραμμα 1

Τα βασικά συστατικά ενός αρχείου XML κατηγοριοποιούνται σε :

- **Ετικέτες** συμβολίζονται με "< ... >", ενώ διακρίνονται σε *ετικέτες-αρχής*, *ετικέτες-τέλους* και *ετικέτες-χωρίς-περιεχόμενο*
- **Στοιχεία** που αποτελούνται από κάποιο κομμάτι κειμένου που είτε ξεκινά με μία ετικέτα-αρχής και καταλήγει σε μία ετικέτα-τέλους, είτε αποτελείται μόνο από μία ετικέτα-χωρίς-περιεχόμενο
- **Χαρακτηριστικά** που ουσιαστικά είναι markup δομές και αφορούν ένα στοιχείο σήμανσης που αποτελείται από ένα ζευγάρι *όνομα/τιμή*, το οποίο υπάρχει μέσα σε μία ετικέτα-αρχής ή σε μία ετικέτα-χωρίς-περιεχόμενο
- **Σήμανση** που σχετίζεται με τα αλφαριθμητικά ενός κειμένου XML τα οποία είτε ξεκινούν με το χαρακτήρα "<" και καταλήγουν στο χαρακτήρα ">", είτε ξεκινούν με το χαρακτήρα "&" και καταλήγουν στο χαρακτήρα ";"
- **Περιεχόμενο** που σχετίζεται με τα αλφαριθμητικά ενός κειμένου XML δεν ξεκινούν με του χαρακτήρες που αναφέρθηκαν στην σήμανση

Έχουν αναπτυχθεί εκατοντάδες γλώσσες και τεχνολογίες που βασίζονται στην XML όπως το RSS και SOAP καθώς και η XHTML. Οι περισσότερες σουίτες εφαρμογών γραφείου περιέχουν κωδικοποιήσεις που βασίζονται στην XML. Έτσι για παράδειγμα η εταιρεία Microsoft Office έχει το πρόγραμμα Office Open XML, ο οργανισμός OpenOffice.org έχει το OpenDocument ενώ η Apple το πρόγραμμα iWork.

1.3.2 Πλεονεκτήματα χρήσης XML

Η χρήση XML για την ανταλλαγή πληροφοριών προσφέρει πολλά οφέλη. Οι βασικοί παράγοντες επιτυχίας της είναι οι εξής:

- Η XML είναι απλό κείμενο, το οποίο μπορεί με ευκολία να διαβαστεί από ανθρώπους, ενώ μπορεί να γίνει κατανοητό, ακόμα και από αρχάριους
- Η XML επικεντρώνεται μόνο στην σημασία των δεδομένων και όχι με την επεξεργασία τους
- Η XML είναι πλήρως συμβατή με την γλώσσα Java και μπορεί να μεταφερθεί μεταξύ διαφορετικών συστημάτων. Κάθε εφαρμογή που επεξεργάζεται την XML μπορεί να εξάγει πληροφορίες ανεξάρτητα από την πλατφόρμα που χρησιμοποιεί
- Η XML είναι επεκτάσιμη και δίνει την ικανότητα στον χρήστη να δημιουργεί τις δικές του ετικέτες, ή να χρησιμοποιήσει τις ετικέτες που δημιουργούνται από άλλους χρησιμοποιώντας φυσική γλώσσα

1.3.3 Μειονεκτήματα χρήσης XML

Παρά τα πλεονεκτήματα της XML, κάποιες φορές η χρήση της είναι απαγορευτική κυρίως όταν η κωδικοποίηση σε XML οδηγεί σε μεγάλα αρχεία, αφού όλα τα δεδομένα βρίσκονται μέσα σε ετικέτες. Δεν είναι σπάνιο το φαινόμενο παραπάνω των 2/3 ενός αρχείου XML να καταλαμβάνεται από ετικέτες. Επιπλέον η κωδικοποίηση των δεδομένων σε απλό κείμενο είναι χρονοβόρα, και πολλές φορές δεν ενδείκνυται, ιδίως για επιστημονικές εφαρμογές ή για εφαρμογές όπου η απόδοση έχει μεγάλη σημασία.

1.4 Γλώσσα XHTML

Η εξέλιξη της HTML έχει ουσιαστικά σταματήσει. Για αυτόν τον λόγο η HTML τείνει να αντικατασταθεί από μια νέα γλώσσα, που ονομάζεται XHTML. Στόχος ήταν η τροποποίηση της HTML σε ένα πρότυπο που βασίζεται στην XML, η πρώτη θα γινόταν συμβατή με όλα τα εργαλεία που υπάρχουν για τη δεύτερη. Ενώ η HTML αποτελεί υποσύνολο της SGML (Standard Generalized Markup Language), η XHTML είναι μια εφαρμογή της XML η οποία είναι μια πιο περιορισμένη εφαρμογή της SGML. Η συγγραφή ενός XHTML έγγραφου προϋποθέτει την τήρηση ενός συνόλου κανονισμών. Η ανάγνωση των εγγράφων αυτών, γίνεται χρησιμοποιώντας

αυστηρώς τυποποιημένα λογισμικά που είναι γνωστά με την ονομασία XML parsers⁴. Αντίθετα, για την ανάγνωση των HTML εγγράφων, απαιτείται ένας parser ο οποίος να είναι πιο ανεκτός στα λάθη. Τον Ιανουάριο του 2000 η W3C εκδίδει την 1^η έκδοση ενώ τον Μάιο του επόμενου χρόνου η 2^η έκδοση ήταν πραγματικότητα. Τέλος, η XHTML 5 είναι υπό κατασκευή από το Σεπτέμβριο του 2009 και αποτελεί μέρος της HTML 5.

Υπάρχουν τρεις τύποι εγγράφων για την HTML 1.0 :

- Του **XHTML 1.0 Strict** που ουσιαστικά περιέχει την γλώσσα XML που ισοδυναμεί στο αυστηρό πρότυπο της HTML 4.01, και περιλαμβάνει τα στοιχεία και τα χαρακτηριστικά που δεν έχουν σημαθεί ότι προκαλούν κάποιο πρόβλημα στο πρότυπο της HTML 4.01. Από την 25^η Μαΐου 2011, η XHTML 1.0 Strict χρησιμοποιείται για την αρχική σελίδα του δικτυακού τόπου της W3C
- Του **XHTML 1.0 Transitional** που περιέχει την ισοδύναμη έκδοση της XML για την HTML 4.01 Transitional και περιλαμβάνει κυρίως στοιχεία παρουσίασης (όπως center, font και strike) τα οποία εξαιρούνταν από το αυστηρό πρότυπο της HTML 4.01 και
- Του **XHTML 1.0 Frameset** είναι η ισοδύναμη έκδοση της XML για την HTML 4.01 Frameset και επιτρέπει τον ορισμό των πλαισίων εγγράφων

Άξιο αναφοράς αποτελεί το γεγονός ότι, η W3C εξέδωσε και Mobile εκδόσεις της XHTML δίνοντας την δυνατότητα ακόμα και σε συστήματα με πολύ περιορισμένους πόρους να κάνουν χρήση πληροφοριών και διάφορων τύπων δεδομένων.

1.4.1 Δομή γλώσσας XHTML

Η δομή ενός αρχείου XHTML είναι απλή και μοιάζει με αυτή της HTML. Ομοίως βασικά της στοιχεία είναι ο τίτλος, οι επικεφαλίδες, οι παράγραφοι. Διαθέτει επιπλέον:

- Διαχωριστικά (divs), που χρησιμοποιούνται στον διαχωρισμό του περιεχομένου σε ενότητες/ μπλόκς
- Διαστήλια (spans), χρησιμοποιούνται για τον χωρισμό κειμένων σε μικρότερα κομμάτια

⁴Είναι το λογισμικό που επεξεργάζεται ένα κείμενο XML. Ο επεξεργαστής, αναφέρεται συχνά, με τον αγγλικό όρο XML parser.

Ένα απλό αρχείο HTML είναι το παρακάτω, ενώ για την δημιουργία του μπορεί να χρησιμοποιηθεί οποιοσδήποτε κειμενογράφος και αποθηκεύοντας το αρχείο με την κατάληξη .xhtml

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

Ανοίγοντας το αρχείο μας με ένα πρόγραμμα περιήγησης το αποτέλεσμα που εμφανίζεται είναι το εξής :



Εικόνα 3

1.4.2 Πλεονεκτήματα χρήσης XHTML

Μία από τις βασικές προκλήσεις που αντιμετωπίζουν οι σχεδιαστές ιστοσελίδων είναι η χρήση καλά δομημένης γλώσσας σήμανσης που να πληροί τους κανόνες που έχει θέσει το πρότυπο HTML 4.01. Η XHTML προσφέρει πολλά πλεονεκτήματα σε σχέση με την HTML τα οποία σχετίζονται με :

- **Την Βιωσιμότητα** – οι νέες εφαρμογές τείνουν να χρησιμοποιούν την XML όλο και περισσότερο. Χρησιμοποιώντας την XHTML σε μια ιστοσελίδα αντί της HTML κάνει ευκολότερη οποιαδήποτε μελλοντική μετατροπή της ιστοσελίδας
- **Την Επεκτασιμότητα** – αρχεία που είναι γραμμένα σε XML και XHTML λόγω της χρήσης namespaces, μπορούν να επεκτείνουν τις λειτουργίες τους και με άλλες μορφές σήμανση όπως την MathML (Math Markup Language) και την SVG (Scalable Vector Graphics)
- **Την Συμβατότητα** - αρχεία γραμμένα σε XHTML συμμορφώνονται με τους κανόνες της XML και δίνουν την δυνατότητα σε προγράμματα επεξεργασίας να μετατρέπουν ένα αρχείο XHTML σε άλλη μορφή (π.χ. PDF, RSS ή RTF) με μεγάλη ευκολία
- **Την Αποτελεσματικότητα της επεξεργασίας αιτήσεων** – μελλοντικά οι browsers θα μπορούν να υποστηρίξουν XHTML έγγραφα και τους αυστηρούς κανόνες της XML γρηγορότερα χάρη στις μικρότερη επεξεργασία σφαλμάτων. Προς το παρόν όμως, μεγάλο μέρος της επεξεργαστικής ισχύος ενός προγράμματος περιήγησης εξακολουθεί να δαπανάται για την επεξεργασία σφαλμάτων που εμφανίζουν τα έγγραφα με λανθασμένη σήμανση HTML. Αυτό είναι το τίμημα της γρήγορης αλλά χαοτικής επέκτασης του Διαδικτύου.

1.4.3 Μειονεκτήματα χρήσης XHTML

Τα μειονεκτήματα της XHTML δεν είναι πολλά. Αυτό ίσως σχετίζεται στο γεγονός, ότι η χρήση της είναι περιορισμένη και έτσι δεν μπορούμε να έχουμε ένα συνεχές feedback. Προς το παρόν, είναι ελάχιστα τα προγράμματα περιήγησης που υποστηρίζουν την λήψη και την επεξεργασία εγγράφων XHTML. Για παράδειγμα ο Internet Explorer, browser της εταιρείας Microsoft, δεν υποστηρίζει ακόμα την XHTML και όλες τις λειτουργίες.

1.5 Ιστορική αναδρομή των CSS

Η μορφοποίηση που προσφέρουν όλες οι προαναφερθείσες γλώσσες είναι απλή και δίνει περιορισμένες επιλογές στον χρήστη. Το κενό αυτό, έρχονται να καλύψουν τα *Διαδοχικά Φύλλα Στυλ – CSS*. Οι αρχές, πάνω στις οποίες στηρίχτηκε η υλοποίηση των Διαδοχικών Φύλλων Στυλ, έχουν τις ρίζες του από την δεκαετία του '80 και συγκεκριμένα στην γλώσσα SGML. Βασικός στόχος, ήταν η δημιουργία και η διατήρηση ενός σταθερού ίδιου στυλ σε πολλές σελίδες του διαδικτύου, καθώς και ο διαχωρισμός της δομής του εγγράφου από την παρουσίαση, ώστε να μπορούν να χρησιμοποιηθούν διαφορετικά στυλ ανάλογα με την χρήση μιας σελίδας.

Ο Robert Cailliau ήταν ο άνθρωπος ο οποίος σκέφτηκε πως αν υπήρχε διαχωρισμός μεταξύ της εμφάνισης και της δομής ενός εγγράφου τότε τα πράγματα θα ήταν πιο απλά. Ο ιδανικός τρόπος για να γίνει αυτό ήταν να δώσει στον χρήστη διαφορετικές επιλογές εμφάνισης ενός εγγράφου και έτσι δημιουργήθηκαν τρία είδη από Style Sheets: ένα για εκτύπωση εγγράφων, ένα για απεικόνιση στην οθόνη και ένα για την επεξεργασία του εγγράφου

Η γέννηση της CSS, ξεκίνησε το 1994 από τον Håkon Wium Lie (Lie, 2005) όταν θέλησε να στείλει ένα αρχείο που να έχει το στυλ μιας εφημερίδας και τελικά συνειδητοποίησε πως δεν μπορούσε με τις τότε υπάρχουσες τεχνολογίες. Λίγο καιρό αργότερα, εκδίδει το πρώτο του προσχέδιο για την CSS. Το προσχέδιο αυτό ήταν ουσιαστικά μια πρόταση για τον τρόπο με τον οποίο αρχεία HTML μπορούσαν να αλλάξουν την μορφοποίησή τους με απλές δηλώσεις. Το πρόγραμμα περιήγησης είχε την δυνατότητα να εμφανίσει ιστοσελίδες με διαφορετικά στυλ τα οποία βρισκόταν σε μια διατεταγμένη λίστα. Το στυλ που επέλεγε δηλωνόταν στην αρχή του εγγράφου και είχε για παράδειγμα την παρακάτω μορφή:

```
<LINK REL="style" HREF="http://NYT.com/style">
```

όπου το στοιχείο LINK χρησιμοποιούταν για να δείξει την διεύθυνση URL στην οποία βρισκόταν το φύλλο στυλ. Το ίδιο αρχείο μπορούσε να έχει παραπάνω από ένα φύλλα στυλ ακόμα και εμφωλευμένα. Ο πρώτος που απάντησε σε αυτή την πρόταση ήταν ο Bert Bos, που εκείνο τον καιρό εργαζόταν σε μία αντίστοιχη πρόταση για τα φύλλα στυλ. Οι δύο τους συνεργάστηκαν και έτσι γεννήθηκε η CSS η οποία πήρε τα πρωτεία από την γλώσσα HTML, μιας και κατά γενική ομολογία ήταν καλύτερη επειδή μπορούσε να εφαρμοστεί σε πολλά περισσότερα είδη αρχείων .

Αρχικά, έγιναν πολλές διαμάχες για την χρήση της CSS γιατί κάποιοι θεωρούσαν αδιανόητο το γεγονός ότι ο χρήστης είχε την δυνατότητα να ελέγχει και να επιλέγει την παρουσίαση ενός εγγράφου στο Διαδίκτυο. Τελικά όμως, η κοινωνία του Ιστού αποδέχτηκε την CSS. Θερμός υποστηρικτής της CSS υπήρξε η εταιρεία Microsoft, ο οποία δεσμεύθηκε να χρησιμοποιήσει την νέα γλώσσα φύλλων στυλ στον Internet Explorer, το πρόγραμμα περιήγησης που είχε αναπτύξει. Αντιθέτως, οι

κατασκευαστές της εταιρία Netscape Communication Corporation παρέμειναν επιφυλακτικοί απέναντι στην CSS και επέλεξαν να παρουσιάσουν μια εναλλακτική γλώσσα που ονομαζόταν Java Script StyleSheets (JSSS). Η γλώσσα δεν ήταν επιτυχής και αυτό είχε σαν αποτέλεσμα ο Internet Explorer να γίνει ο πιο δημοφιλής και διαδεδομένος browser της εποχής.

Η ανάπτυξη της HTML και της CSS έγινε από μια ομάδα εργασίας που ονομάστηκε ERB (HTML Editorial Review Board). Στη συνέχεια η ομάδα αυτή χωρίστηκε σε τρεις άλλες ομάδες εργασίας:

- την HTML ομάδα εργασίας υπό την ηγεσία του Dan Connolly που προέρχονταν από τον οργανισμό W3C
- την DOM ομάδα εργασίας υπό την ηγεσία του Lauren Wood της εταιρείας Soft Quad και
- την CSS ομάδα εργασίας υπό την ηγεσία του Chris Lilley επίσης προερχόμενου από τον οργανισμό W3C

1.6 Εκδόσεις CSS

Η γλώσσα CSS εξελίχθηκε τα τελευταία (Μητσαράκης, 2009) χρόνια υπό την καθοδήγηση της W3C, μέχρι να φτάσει στην σημερινή της μορφή. Τα περισσότερα προγράμματα περιήγησης της σημερινής εποχής υποστηρίζουν την 2^η έκδοση της CSS, η οποία περιλαμβάνει την τόσο την 1^η έκδοση όσο και την γλώσσα CSS-Positioning. Μετά από πολλά χρόνια ανάπτυξης, η νέα έκδοση CSS 3 είναι πλέον γεγονός.

1.6.1 Έκδοση επιπέδου CSS 1

Ο οργανισμός W3C δημοσίευσε τις προτάσεις της για την 1^η έκδοση το 1996 και τις αναθεώρησε το 1999. Αυτή η πρώτη έκδοση είχε ικανότητες όπως :

- a. Μορφοποίηση γραμματοσειρών (πλάγια, έντονη, υπογραμμισμένη γραφή)
- b. Μορφοποίηση κειμένου παραγράφου (εισαγωγή εσοχής μεταξύ παραγράφων, κενά μεταξύ των γραμμάτων, λέξεων και σειρών του κειμένου)
- c. Αλλαγή χρώματος γραμματοσειράς, εισαγωγή χρωματιστού φόντου στο κείμενο
- d. Επιλογές διαφορετικής στοίχισης σε παραγράφους, εικόνες και άλλα στοιχεία
- e. Κατασκευή στηλών

- f. Δημιουργία μοναδικών ή συνόλου στοιχείων με συγκεκριμένη ταυτότητα

Ο περιηγητής Netscape Navigator 4 καθώς και οι 3^η και 4^η έκδοση του Internet Explorer υποστηρίζουν το επίπεδο CSS 1. Ο W3C δεν υποστηρίζει πια το πρότυπο CSS 1.

1.6.2 Έκδοση CSS-Positioning (CSS-P)

Οι σχεδιαστές Web εφαρμογών έψαχναν έναν τρόπο που να τους επιτρέπει να στοιχίζουν τα στοιχεία μιας ιστοσελίδας, ακριβώς στο σημείο που τα οραματιζόταν. Η πρώτη έκδοση της CSS είχε ήδη δημοσιευτεί και η υποστήριξη των περιηγητών για την 2^η έκδοση ήταν πολύ μακριά, γι' αυτό η W3C δημοσίευσε ως προσωρινή λύση την CSS-P. Η έκδοση αυτή είχε σαν επιπρόσθετη λειτουργία (σε σχέση με την CSS1) την δυνατότητα εμφάνισης ενός στοιχείου πίσω από ένα δεύτερο. Τόσο ο Netscape όσο και ο Internet Explorer υποστήριζαν την νέα αυτή έκδοση.

1.6.3 Έκδοση επιπέδου CSS 2

Αυτή η έκδοση του CSS παρουσιάστηκε το 1998 και υιοθετήθηκε από τους περισσότερους κατασκευαστές προγραμμάτων περιήγησης. Στο CSS 2 επίπεδο, περιλαμβάνονται όλα τα χαρακτηριστικά των δύο προηγούμενων εκδόσεων και δίνεται ιδιαίτερη στα διαφορετικά μέσα που γινόταν η αναπαραγωγή των στοιχείων CSS, γιατί με διαφορετικό τρόπο παρουσιαζόταν ένα αρχείο σε μια οθόνη απ' ότι σε ένα κομμάτι χαρτί ή σε ένα πρόγραμμα σύνθεσης ομιλίας ή σε μια συσκευή γραφής Braille. Το 2^ο επίπεδο της γλώσσας αποτελεί ένα υπερσύνολο της CSS επιπέδου 1 και περιέχει ένα σύνολο νέων χαρακτηριστικών όπως:

- a. Ρύθμιση απόλυτης, σχετικής και προκαθορισμένης θέσης στοιχείων
- b. Δυνατότητα δημιουργίας επιπέδων (layers) στην απεικόνιση με την βοήθεια ενός νέου χαρακτηριστικού, του z-index
- c. Υποστήριξη για aural stylesheet που δίνει επιπλέον ιδιότητες μορφοποίησης μιας γραμματοσειράς όπως για παράδειγμα το εφέ της σκιάς

Ο W3C δεν υποστηρίζει πλέον το πρότυπο CSS 2

1.6.4 Έκδοση επιπέδου CSS 2.1

Η 2^η έκδοση της CSS εμφάνισε κάποια προβλήματα τα οποία διορθώθηκαν με την 2.1 έκδοση. Τα προβλήματα αυτά αφορούσαν την διακοπή κάποιων χαρακτηριστικών που δεν υποστηρίζονται πλέον από τους περιηγητές. Η νέα έκδοση, προσθέτει νέα χαρακτηριστικά για την γλώσσα τα οποία πιο πριν υποστηρίζονταν από επεκτάσεις των περιηγητών. Η συγκεκριμένη έκδοση αποτέλεσε project εργασίας της W3C για πολλά χρόνια. Η επιτροπή του W3C εξέδωσε το τελικό πρότυπο στις 7 Ιουνίου του 2011.

1.6.5 Έκδοση επιπέδου CSS 3

Η CSS επιπέδου 3 βρίσκεται υπό ανάπτυξη από τις 15 Δεκεμβρίου του 2005. Η CSS επιπέδου 3 αποτελείται από διαφορετικές ενότητες που ονομάζονται modules. Κάθε ενότητα προσθέτει νέες δυνατότητες ή επεκτείνει τα χαρακτηριστικά που ορίζονται από την CSS 2⁰⁰ επιπέδου, διατηρώντας παράλληλα συμβατότητα με παλαιότερες εκδόσεις. Τον Ιούνιο του 2012 καταμετρήθηκαν πάνω από πενήντα ενότητες CSS, ενώ η Ομάδα Εργασίας CSS δημοσίευσε τέσσερις από αυτές επίσημα και αφορούσαν :

- a. Τα Media Queries, ενότητα που επιτρέπει το περιεχόμενο της σελίδας να προσαρμοστεί σε συνθήκες όπως η ανάλυση της οθόνης σε διαφορετικού τύπου οθόνης (πχ. smartphones)
- b. Τα Namespaces, ενότητα που επιτρέπει το ίδιο αρχείο να χρησιμοποιείται από πολλούς χρήστες χωρίς να υπάρχει σύγκρουση μεταξύ τους
- c. Τα Selectors, ενότητα που περιγράφει στοιχεία επιλογής που χρησιμοποιεί η CSS για να επισυνάψει ιδιότητες στυλ. Τα στοιχεία αυτά μπορεί να επιλεγούν βάση του ονόματός τους, των γνωρισμάτων τους, του περιεχομένου τους και άλλων ιδιοτήτων.
- d. Τα Colors, ενότητα που προσδιορίζει όλες τις πτυχές που σχετίζονται με το χρώμα των CSS, συμπεριλαμβανομένης της ιδιότητας της διαφάνειας

1.6.6 Έκδοση επιπέδου CSS 4

Δεν υπάρχει ένα ενιαίο ολοκληρωμένο πρότυπο δεδομένου ότι η CSS 4 χωρίζεται σε ενότητες και επίσης ακόμα δεν έχουν ολοκληρωθεί οι εργασίες για το 3^ο επίπεδο. Ωστόσο υπάρχουν αρκετές ενότητες επιπέδου 4. Από την στιγμή που η CSS επιπέδου χωρίστηκε σε πολλές ενότητες, αυτές έχουν την δυνατότητα να αναπτυχθούν ανεξάρτητα. Υπάρχουν λίγες ενότητες 4⁰⁰ επιπέδου οι οποίες

βασίζονται σε λειτουργικότητες του προηγούμενου επιπέδου. Έτσι, ενώ δεν υπάρχει ακόμα επίσημα η CSS4 έκδοση πάνω στην οποία θα εργαστεί ο W3C, ωστόσο υπάρχουν ενότητες που μπορούν να αναφέρονται συλλογικά ως CSS4.

1.7 Πλεονεκτήματα χρήσης CSS

Έχουμε ήδη αναφέρει έμμεσα ή άμεσα αρκετά από τα πλεονεκτήματα που προσφέρει η χρήση των CSS στους προγραμματιστές ιστοσελίδων και όχι μόνο. Συνοπτικά αναφέρουμε ως επιπρόσθετα πλεονεκτήματα την μεγαλύτερη ευελιξία που προσφέρει μιας και η CSS κατέστησε εφικτές μορφοποιήσεις οι οποίες ήταν αδύνατες ή πολύ δύσκολες να πραγματοποιηθούν με την κλασσική HTML. Επίσης σημαντικό είναι να αναφέρουμε το πόσο ευκολότερη γίνεται η δουλειά των συντηρητών των ιστοσελίδων μιας και η εμφάνιση ενός ολόκληρου site μπορεί να ελέγχεται από ένα μόνο εξωτερικό αρχείο CSS. Έτσι, κάθε αλλαγή στο στυλ της ιστοσελίδας μπορεί να γίνεται με μια μοναδική αλλαγή σε αυτό το αρχείο, αντί για την επεξεργασία πολλών σημείων σε κάθε σελίδα που υπάρχει στο site.

Φυσικά πολύ σημαντικό είναι το γεγονός πως η χρήση CSS έχει ελαχιστοποιήσει το μέγεθος του αρχείου μορφοποίησης, δεδομένου ότι ο κάθε κανόνας που χρησιμοποιείται γράφεται μόνο μια φορά και όχι σε κάθε σημείο που εφαρμόζεται. Επιπλέον, κατασκευάζουμε σελίδες που είναι πολύ πιο γρήγορες αφού όταν χρησιμοποιούμε εξωτερικό αρχείο CSS, ο browser την πρώτη φορά που θα φορτώσει κάποια σελίδα του site μας το αποθηκεύει στην cache, οπότε δεν χρειάζεται να το κατεβάσει ξανά κάθε φορά που κατεβάζει ο χρήστης του κάποια άλλη σελίδα του site μας.

Τέλος, η χρήση των CSS μας δίνει καλύτερα αποτελέσματα όταν κάνουν αναζήτηση της σελίδας σε μια μηχανή αναζήτησης (Search engine optimization - SEO). Οι μηχανές αναζήτησης δεν «μπερδεύονται» ανάμεσα σε περιεχόμενο και τη μορφοποίηση του, αλλά έχουν πρόσβαση στο περιεχόμενο σκέτο, οπότε είναι πολύ ευκολότερο να το καταγράψουν και να το αρχειοθετήσουν (indexing). Καλύτερα SEO δίνουν μεγαλύτερες πιθανότητες επισκεψιμότητας μιας σελίδας πράγμα που είναι συνδεδεμένος με την επιτυχία της ιστοσελίδας

1.8 Μειονεκτήματα χρήσης CSS

Πέρα από τα πλεονεκτήματα της CSS, μπορούμε να παρατηρήσουμε και τα εξής μειονεκτήματα τα οποία θα πρέπει να αντιμετωπίζουν με σοβαρότητα και προσοχή οι σχεδιαστές ιστοσελίδων.

- **Απαιτείται πρόσβαση σε εξωτερικά αρχεία**, γεγονός που μπορεί να προκαλέσει μεγάλο πρόβλημα αν αποθηκευτεί ένα έγγραφο HTML στο σκληρό δίσκο δίχως το αρχείο CSS που ορίζει τη μορφοποίησή. Έτσι το έγγραφο δεν θα εμφανίζεται σωστά όταν ο χρήστης το βλέπει ενώ είναι αποσυνδεδεμένος από το δίκτυο. Με τον ίδιο τρόπο, όλες οι σελίδες των οποίων η μορφοποίηση εξαρτάται από ένα αρχείο CSS, όταν χάσουν τη σύνδεση με το αρχείο αυτό, δεν θα εμφανίζονται σωστά. Επίσης υλικό όπως φωτογραφίες δεν θα εμφανίζονται γιατί η δήλωσή τους θα βρίσκεται στο αρχείο της μορφοποίησης
- **Μείωση ταχύτητας στην φόρτωση της ιστοσελίδας** καθώς ο browser πρέπει να κατεβάσει δύο αρχεία ταυτόχρονα ένα HTML και ένα CSS αρχείο, πράγμα που είναι πιο χρονοβόρο από το να κατεβάσει ένα μόνο αρχείο HTML. Η ταχύτητα, βέβαια είναι το τελευταίο που σκεφτόμαστε αν ο χρήστης θέλει να περιηγηθεί σε ένα σύνολο σελίδων οι οποίες χρησιμοποιούν την ίδια μορφοποίηση (ίδιο αρχείο CSS)
- **Αυξημένη πολυπλοκότητα σε συστήματα διαχείρισης περιεχομένου** (CMS – Content Management Systems) όπως το Joomla, Drupal, Wordpress κτλ. Η πολυπλοκότητα αυτή οφείλεται στο ότι πρέπει να διατηρούνται σωστά ένα σύνολο από CSS links σε ένα μεγάλο εύρος φακέλων και αρχείων τα οποία αλλάζουν και ανανεώνονται συνεχώς. Τα περισσότερα από τα συστήματα τέτοιου είδους χρησιμοποιούν CSS αρχεία για τη διαμόρφωση των σελίδων αλλά είναι πολύπλοκα, και μερικές φορές κάποια αλλαγή σε αυτά μπορεί να προκαλέσει δυσάρεστες συνέπειες
- **Έλλειψη ασφάλειας στην τροποποίηση του αρχείου** γεγονός που μπορεί να είναι πολύ καταστροφικό αφού οποιοσδήποτε που έχει δικαίωμα να διαβάσει ή να εγγράψει στο αρχείο μπορεί να διακόψει κάποιες λειτουργίες ή να αλλάξει υπερσυνδέσμους στις ιστοσελίδες
- **Διαφορετική σύνταξη από την HTML**, το οποίο είναι λογικό αφού η CSS αναπτύχθηκε εντελώς ανεξάρτητα από την HTML και χρησιμοποιεί εντελώς διαφορετική σύνταξη. Αυτό βέβαια επιβαρύνει τον προγραμματιστή που αφού θα πρέπει να γνωρίζει δύο διαφορετικούς τρόπους μορφοποίησης για να αναπτύσσει εφαρμογές διαδικτύου

1.9 Σύνοψη

Η αλματώδης εξέλιξη του Διαδικτύου προκάλεσε την διαδοχική ανάπτυξη νέων τεχνολογιών προκειμένου να καλυφθούν οι ανάγκες των χρηστών. Η πρώτη γλώσσα σήμανσης, η HTML, αποτέλεσε θεμέλιο λίθο για όλες τις νέες και εξελιγμένες γλώσσες που δημιουργήθηκαν στην συνέχεια όπως η XML και XHTML. Βέβαια, καμία από τις προηγούμενες δεν μπόρεσε να καλύψει ένα μεγάλο κενό, αυτό της μορφοποίησης. Η CSS έδωσε την λύση που όλοι έψαχναν. Οι προγραμματιστές ιστοσελίδων έχουν πλέον, άπειρες σχεδιαστικές δυνατότητες, ενώ η τελευταία έκδοση CSS 3 μπορεί να υλοποιήσει ακόμα και τις πιο τρελές σχεδιαστικές απαιτήσεις. Στο επόμενο κεφάλαιο, θα εισαχθούμε αναλυτικά, σε όλες αυτές τις δυνατότητες που μας παρέχουν τα Διαδοχικά Φύλλα Στυλ.



Εικόνα 4

Κεφάλαιο 2^ο

2. Εισαγωγή

Όπως ήδη αναφέραμε στο 1^ο κεφάλαιο, τα CSS είναι ένα πολύ δυνατό εργαλείο στα χέρια των κατασκευαστών ιστοσελίδων καθώς τους δίνει την δυνατότητα να παρουσιάσουν ένα ή περισσότερα αρχεία με όποιον τρόπο επιθυμούν. Στην ουσία η χρήση απλά ενός αρχείου CSS δεν μπορεί να μας εμφανίσει στο παράθυρο μια σελίδα, μιας και δεν θα έχει κανένα απολύτως περιεχόμενο να παρουσιάσει, γι' αυτό και η συνοδεία ενός αρχείου, συνήθως HTML, κρίνεται απαραίτητη.

2.1 Σύνδεση αρχείου HTML με CSS

Ένα αρχείο HTML φέρει το περιεχόμενο που θέλουμε να εμφανίσουμε. Για παράδειγμα στην ενότητα [1.2.1](#) κατασκευάσαμε ένα παράδειγμα όπου εμφανίζαμε διαφορετικές επικεφαλίδες, εικόνες και παραγράφους. Αν επιθυμούσαμε να αλλάξουμε γραμματοσειρά ή το φόντο της σελίδας μας αυτό ήταν αρκετά περίπλοκο και χρονοβόρο. Γι' αυτό τον λόγο κατασκευάζουμε ένα αρχείο CSS που αποκλείστηκα περιέχει την διαφορετική μορφοποίηση που θέλουμε να έχει κάθε κομμάτι της σελίδας μας. Αυτό γίνεται σε δύο βήματα.

- Βήμα 1^ο : Δημιουργούμε ένα νέο αρχείο σε έναν κειμενογράφο (εδώ χρησιμοποιούμε τον Notepad++) με κατάληξη .css μέσα στο οποίο θα γράψουμε τον κώδικα της μορφοποίησης
- Βήμα 2^ο : Μέσα στο αρχείο HTML που κατασκευάσαμε προηγουμένως προσθέτουμε μια ετικέτα που θα συνδέσει τα δύο αρχεία και η οποία έχει την μορφή

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

όπου style.css το όνομα αρχείου που κατασκευάσαμε στο 1^ο βήμα. Να επισημάνουμε πως τα δύο αυτά αρχεία πρέπει να βρίσκονται στον ίδιο φάκελο.

Εναλλακτικά η μορφοποίηση μπορεί να εισαχθεί στο αρχείο HTML με δύο τρόπους :

- **Εσωτερικά στο HTML αρχείο:** Αν επιθυμούμε να μορφοποιήσουμε ένα στοιχείο μόνο, και δεν πρόκειται να χρειαστούμε αυτό το είδος μορφοποίησης σε άλλο σημείο μπορούμε να το δηλώσουμε μέσα σε ένα **attribute style**. Για παράδειγμα, για να κάνουμε μια συγκεκριμένη παράγραφο να έχει μπλε φόντο, προσθέτουμε `<p style = "color: #990066;">`⁵. Αξίζει να σημειωθεί, ότι αν κάποιες από τις ιδιότητες που θα γράψουμε εντός του attribute style ενός στοιχείου «συγκρούονται» με κανόνες CSS που έχουν οριστεί σε ένα εξωτερικό αρχείο .css, ο περιηγητής θα επιλέξει να εφαρμόσει την μορφοποίηση εντός του attribute style για το στοιχείο αυτό.
- **CSS για μια συγκεκριμένη σελίδα:** Πολλές φορές, μπορεί να θέλουμε να εφαρμόσουμε κάποιους κανόνες CSS μόνο σε συγκεκριμένη σελίδα και όχι στο σύνολο ενός διαδικτυακού χώρου. Ένας τρόπος να το κάνουμε αυτό είναι να εισάγουμε εντός των ετικετών `<head>...</head>` της σελίδας τους κανόνες CSS μας μέσα σε `<style>` ετικέτες. Παραδείγματος χάριν, για να κάνουμε το χρώμα γραμμάτων μιας συγκεκριμένης σελίδας κόκκινο, θα γράφαμε στο head της: `<style type = "text/css"> body { color : red; } </style>`

Οι δύο αυτοί, τρόποι χρησιμοποιούνται σπάνια (κυρίως σε ιστοχώρους με μικρό αριθμό σελίδων), μιας και αυξάνουν την περιπλοκότητα του κώδικα.

Με τον πρώτο τρόπο, το HTML αρχείο που κατασκευάσαμε μετά την προσθήκη της διεύθυνσης για την μορφοποίηση θα είναι:

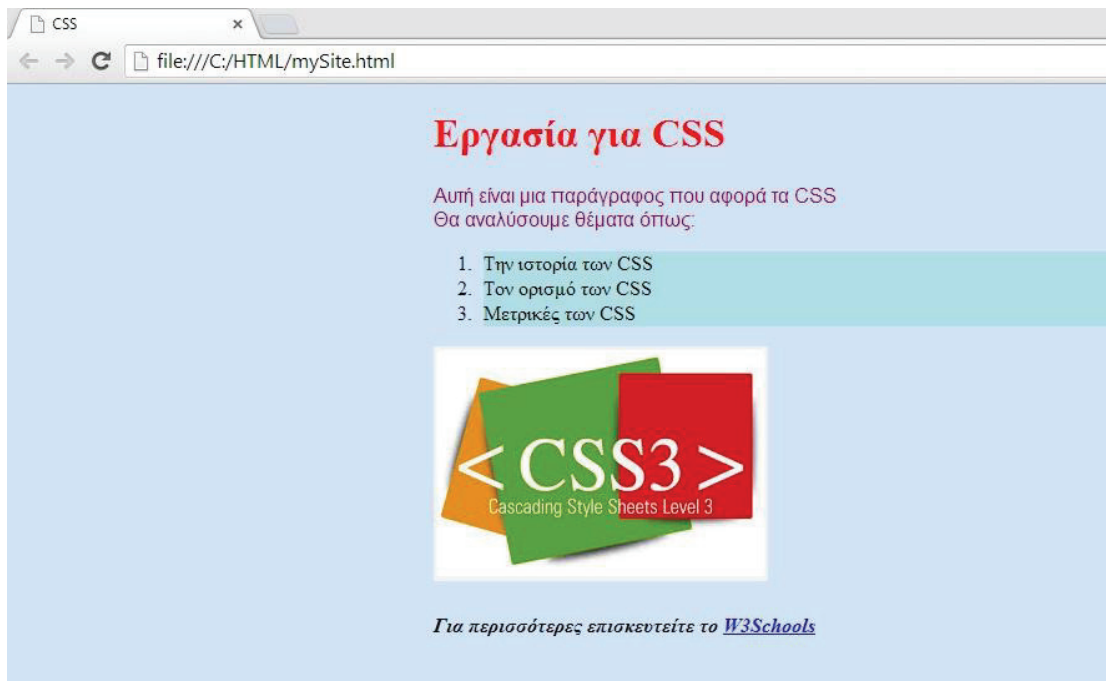
```
<!DOCTYPE html>
<html>
  <head>
    <title>CSS</title>
    <link rel="stylesheet" type="text/css" href="style.css" >
  </head>
  <body>
    <h1>Εργασία για CSS</h1>
    <p>Αυτή είναι μια παράγραφος που αφορά τα CSS
    <br> Θα αναλύσουμε θέματα όπως:
      <ol>
        <li>Την ιστορία των CSS</li>
        <li>Τον ορισμό των CSS</li>
        <li>Μετρικές των CSS</li>
      </ol>
    <img src = images.jpg></p>
    <p><h4><i>Για περισσότερες επισκευτείτε το <a
href="http://www.w3schools.com/">W3Schools</a> <i></h4></p>
  </body>
</html>
```

⁵ Τα χρώματα μπορούν να δηλωθούν είτε με νούμερα είτε με το όνομά τους πχ. Red,yellow, grey κ.α

Ενώ το αρχείο style.css περιέχει την εξής μορφοποίηση που επηρεάζει το περιεχόμενο της σελίδας HTML

```
body
{
  background-color:#d0e4fe;
  margin:auto;
  width:50%;
}
h1 {
  color: red;
}
p {
  color: #990066;
  font: medium Helvetica;
}
```

Σε επόμενες ενότητες θα γίνει λεπτομερής ανάλυση στα στοιχεία της CSS αλλά για να κατανοήσουμε το παράδειγμα αναφέρουμε ότι μπορούμε να αλλάξουμε το χρώμα του φόντου της σελίδας (**background-color:#d0e4fe**), την θέση κάποιου στοιχείου μέσα στην σελίδα (**margin:auto** και **width:50%**), το χρώμα της γραμματοσειράς (**color: red**) αλλά και το είδος της γραφής της γραμματοσειράς (**font: medium Helvetica**).



Εικόνα 5

2.3 Επιλογείς

Οι επιλογείς CSS δίνουν την δυνατότητα να επιλέξουμε τα στοιχεία που θέλουμε να αλλάξουν μορφή. Δεν είναι τίποτα άλλο από απλούς κανονισμούς για να μπορέσει το πρόγραμμα περιήγησης να καταλάβει τι ακριβώς θέλουμε να εμφανίσουμε.

2.3.1 Επιλογείς Ετικέτας ή Στοιχείου

Οι επιλογείς ετικέτας ή στοιχείου (Κομβούτη Βέρου, n.d.), όπως αλλιώς μπορούν να ονομαστούν, είναι ένα πολύ αποτελεσματικό εργαλείο αλλαγής στυλ, δεδομένου ότι επηρεάζουν κάθε εμφάνιση μιας ετικέτας HTML σε μια ιστοσελίδα. Για παράδειγμα, αν θέλουμε να διαμορφώσουμε κάθε παράγραφο κειμένου σε μια σελίδα, χρησιμοποιώντας την ίδια γραμματοσειρά, χρώμα και μέγεθος, αυτό μπορεί να γίνει απλώς δημιουργώντας ένα στυλ χρησιμοποιώντας το γράμμα p στον επιλογέα που σχετίζεται με την ετικέτα `<p>...</p>` που βρίσκεται μέσα στο HTML αρχείο. Στην ουσία ένας επιλογέας ετικέτας, επαναπροσδιορίζει το πώς ένα πρόγραμμα περιήγησης εμφανίζει μια συγκεκριμένη ετικέτα. Πριν την χρήση των CSS, προκειμένου να αλλάξουμε το στυλ στο κείμενο μιας παραγράφου, θα έπρεπε να περικλύσουμε το κάθε κείμενο σε μια ετικέτα `` η οποία όριζε την ίδια εμφάνιση σε κάθε παράγραφο σε μια σελίδα. Η υπερβολική χρήση αυτών των ετικετών μέσα στην HTML, όμως, έκανε την λήψη και την εμφάνιση των σελίδων πολύ αργή ενώ η διαδικασία ενημέρωσης ήταν ακόμη πιο χρονοβόρα. Όλα αυτά τα προβλήματα λύνονται με την δημιουργία ενός κανόνα CSS. Οι επιλογείς ετικέτας είναι εύκολο να εντοπιστούν σε έναν κανόνα CSS καθώς έχουν το ίδιο όνομα με το στοιχείο που επηρεάζουν. Γι' αυτό το λόγο στο παράδειγμα 2.1 στο αρχείο CSS δηλώσαμε :

```
h1 {
  color: red;
}
```

Προκειμένω όλες οι επικεφαλίδες h1 τις σελίδας μας να έχουν κόκκινη γραμματοσειρά. Η χρήση επιλογέων ετικέτας έχει όμως σημαντικά μειονεκτήματα. Παραδείγματος χάριν αν θέλαμε μόνο μια παράγραφος της σελίδας μας να έχει διαφορετική μορφή αυτό δεν θα ήταν εφικτό αφού ο περιηγητής δεν θα μπορούσε να καταλάβει την διαφορά και θα εφάρμοζε ίδια μορφοποίηση. Την λύση σε αυτό το πρόβλημα έρχονται να δώσουν οι επιλογείς κλάσης.

2.3.2 Επιλογείς Κλάσης

Όταν θέλουμε να δώσουμε σε κάποια στοιχεία διαφορετική μορφοποίηση από αυτήν που ορίζεται από τους επιλογείς ετικέτας μπορούμε να χρησιμοποιήσουμε έναν επιλογέα κλάσης. Έτσι, ομαδοποιούμε στοιχεία, ώστε να φέρουν ίδια χαρακτηριστικά. Κατασκευάζουμε έναν επιλογέα κλάσης ως **.όνομα_κλάσης** και έτσι δίνουμε την εντολή στο πρόγραμμα περιήγησης να ψάξει όσα στοιχεία στη σελίδα περιλαμβάνουν την ιδιότητα της κλάσης και στην συνέχεια εφαρμόζει τις ιδιότητες που θα γράψουμε στον κανόνα CSS σε οποιοδήποτε στοιχείο περιλαμβάνει την κλάση. Έτσι, αν θέλαμε μια συγκεκριμένη παράγραφος να έχει πράσινα γράμματα, ενώ οι υπόλοιπες να είχαν μαύρα, θα δημιουργούσαμε μια κλάση έστω `.mygreen` μέσα στο αρχείο CSS ως εξής:

```
.mygreen {
    color: green;
}
```

Ενώ μέσα στον κώδικα της HTML προσθέτουμε της εντολή `<p class = "mygreen"> ...</p>` ώστε ο περιηγητής να καταλάβει πως πρέπει να ακολουθήσει διαφορετική μορφοποίηση από την γενική που είχε δηλωθεί για όλες τις παραγράφους p.



Εικόνα 7

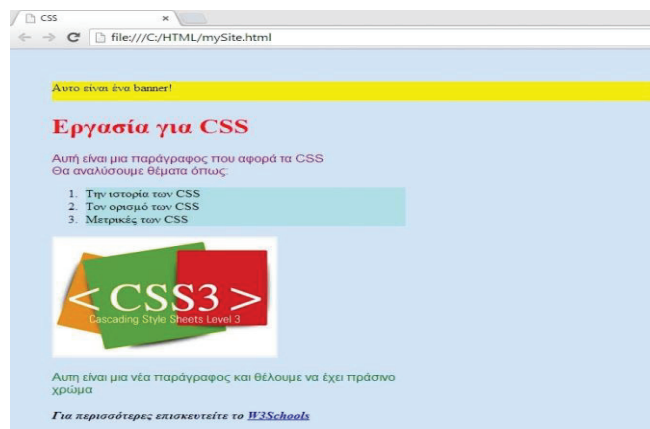
Οι κλάσεις γενικά χρησιμοποιούνται όταν θέλουμε να ομαδοποιήσουμε κάποια στοιχεία HTML για τα οποία δεν μπορούμε να βρούμε κάποιον άλλο επιλογέα που να αφορά όλα αυτά και μόνον αυτά, οπότε τους προσδίδουμε μια συγκεκριμένη κλάση, ώστε να μπορούμε στο CSS μας να αναφερθούμε μόνο σε αυτά και να τα μορφοποιήσουμε.

2.3.3 Επιλογείς ID

Με την βοήθεια ID επιλογέων μπορούμε να απομονώσουμε και να τροποποιήσουμε ένα μοναδικό μέρος μιας ιστοσελίδας. Χρησιμοποιούνται κυρίως για να εμφανίζουμε banners, ή μπάρες πλοήγησης μέσα στον ιστοχώρο. Κατασκευάζονται όπως οι επιλογείς κλάσης δίνοντας ένα όνομα ID το οποίο το χρησιμοποιούμε στην συνέχεια μέσα στην σελίδα HTML. Διαφοροποιούνται από τους επιλογείς κλάσης μιας και διαθέτουν συγκεκριμένες χρήσεις ιδανικές για σελίδες που έχουν κατασκευαστεί με JavaScript ή έχουν πολύ μεγάλο μέγεθος. Μέσα στο αρχείο CSS προσθέτουμε :

```
#banner {
    background: yellow;
    height: 30px;
    width: 720px;
}
```

Καθώς και την γραμμή `<p id="banner">...</p>` μέσα στο HTML. Αυτό μας δίνει το παρακάτω αποτέλεσμα:



Εικόνα 8

2.3.4 Καθολικός Επιλογέας

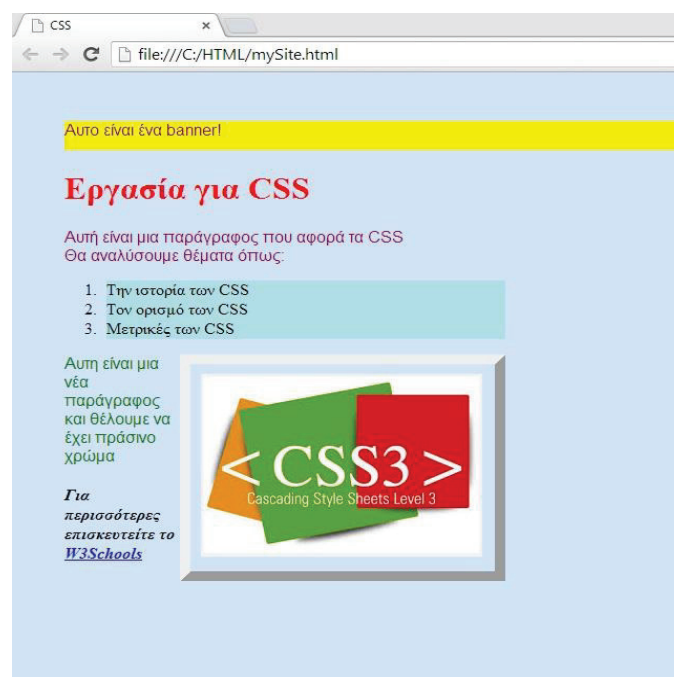
Αν θέλουμε να έχουμε ένα σύνολο επιλογέων και να εφαρμόσουμε τις ίδιες ιδιότητες στυλ σε πολλά διαφορετικά στοιχεία που είναι διάσπαρτα σε σελίδες η CSS μας δίνει την δυνατότητα να το κάνουμε με τον καθολικό επιλογέα που συμβολίζεται με αστερίσκο (*). Για παράδειγμα αν θέλουμε όλες ετικέτες στην ιστοσελίδα μας (παράγραφος, επικεφαλίδα κ.α) να εμφανίζονται με έντονη γραμματοσειρά τότε προσθέτουμε στον CSS κώδικα `* { font-weight: bold; }` και η γραφή αλλάζει.

2.4 Ενότητες

Για την καλύτερη επεξεργασία της ιστοσελίδας πολύ συχνά χρησιμοποιούμε ενότητες που περικλείουν κώδικα HTML ο οποίος εκτελεί κάποια συγκεκριμένη λειτουργία. Για παράδειγμα στην σελίδα του παραδείγματος που κατασκευάσαμε στις προηγούμενες ενότητες αν θέλαμε να προσθέσουμε ένα πλαίσιο γύρω από την εικόνα και επίσης να αλλάξουμε την θέση που έχει αυτή μέσα στην ιστοσελίδα θα δημιουργούσαμε μια ενότητα `<div>..</div>` και θα περικλείαμε τον κώδικα HTML που είναι υπεύθυνος για την εμφάνιση της εικόνας. Στο αρχείο CSS προσθέτουμε τον κώδικα

```
div.my_image {
    border-style: outset;
    border-width: 10px;
    float : right;
    width: 270px;
    padding: 15px;
    margin: 0px;
}
```

Και μέσα στο HTML αρχείο `<div class = "my_image">...</div>` . Η ενότητα `my_image` έχει διαφορετική μορφοποίηση από το κυρίως σώμα της ιστοσελίδας καθώς και ακολουθεί διαφορετική διάταξη μέσα στην ιστοσελίδα αφού τοποθετείται δίπλα από την παράγραφο και όχι στην σειρά όπως προηγουμένως. Σε επόμενη ενότητα θα αναφέρουμε τις διαφορετικές ιδιότητες μορφοποίησης.



Εικόνα 9

2.5 Ψευδοκλάσεις και Ψευδοστοιχεία

Ορισμένες φορές χρησιμοποιούμε τις λεγόμενες ψευδοκλάσεις (pseudo-classes) ή ψευδό-στοιχεία (pseudo-elements), τα οποία μας επιτρέπουν να επιλέγουμε στοιχεία τα οποία δεν αποτελούν στοιχεία HTML, αλλά κομμάτια τους ή συγκεκριμένες καταστάσεις τους. Ουσιαστικά αποτελούν κάποιες λέξεις - κλειδιά που ξεκινούν με : και γράφουμε μετά από έναν επιλογή τύπου στοιχείο/ετικέτας.

2.5.1 Ψευδοκλάσεις Συνδέσμων

Υπάρχουν πέντε ψευδοκλάσεις που επιτρέπουν την αλληλεπίδραση του χρήστη με άλλες διαδικτυακές ιστοσελίδες. Καταδεικνύουν μια από τις παρακάτω καταστάσεις:

- ❖ **a: link** → αφορά τους συνδέσμους που ο χρήστης δεν έχει ακόμη επισκεφθεί
- ❖ **a:visited** → αφορά τους συνδέσμους που ο χρήστης έχει ακόμη επισκεφθεί βάσει του ιστορικού του περιηγητή
- ❖ **a:hover** → αφορά τα στοιχεία που αλλάζουν μορφοποίηση τη στιγμή που ο χρήστης έχει το δείκτη του ποντικιού πάνω σε αυτά χωρίς να έχει πατήσει κλικ
- ❖ **a:active** → αφορά τα στοιχεία που αλλάζουν μορφοποίηση τη στιγμή που ο χρήστης πατήσει κλικ στο δείκτη του ποντικιού, κλάσματα δευτερολέπτου πριν μεταβεί στην νέα σύνδεση
- ❖ **:focus** → λειτουργεί παρόμοια με την hover αλλά κυρίως για στοιχεία ηλεκτρονικών φορμών που βρίσκονται σε μία ιστοσελίδα

2.5.2 Διαμόρφωση Παραγράφων

Η CSS διαθέτει δύο ψευδοστοιχεία που αυξάνουν την αισθητική εικόνα μιας σελίδας. Το στοιχείο **:first – letter** που αφορά στην διαμόρφωση του πρώτου γράμματος ενός στοιχείου (π.χ. μιας παραγράφου). Το αρχικό αυτό γράμμα, μπορεί να εξέχει από το υπόλοιπο κείμενο, να έχει μεγαλύτερο μέγεθος ή και είδος γραμματοσειράς. Το δεύτερο ψευδοστοιχείο **:first - line** αφορά αποκλειστικά την πρώτη γραμμή ενός στοιχείου η οποία μπορεί να έχει διαφορετική μορφή, όπως χρωματιστό φόντο, προκειμένου να κεντρίσει το ενδιαφέρον του χρήστη.

2.5.3 Ψευδοστοιχεία

Εκτός από τα ψευδοστοιχεία διαμόρφωσης παραγράφου η CSS περιέχει επιπλέον ψευδοστοιχεία που διευκολύνουν ακόμα πιο πολύ την κατασκευή μιας ιστοσελίδας όπως:

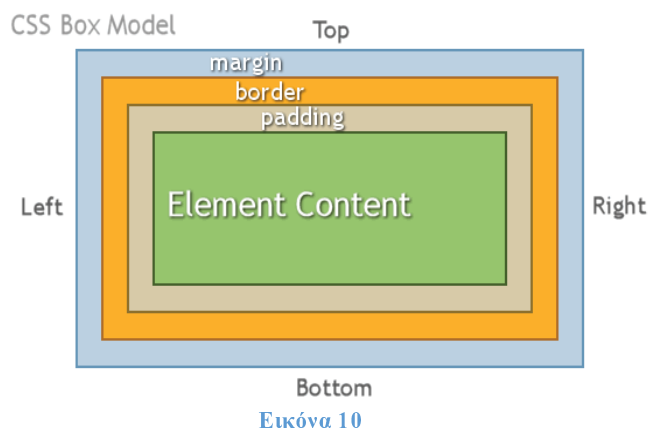
- **Before** : διευκολύνει την προσθήκη μικρών κομματιών κώδικα. Για παράδειγμα αν είχαμε μια ιστοσελίδα ανακοινώσεων μιας γραμματείας θα ήταν πολύ χρήσιμο οι νέες ανακοινώσεις να είχαν μπροστά μια μικρή επισήμανση που θα το όριζε. Για την κατασκευή της θα δημιουργούσαμε ένα στοιχείο με το όνομα `tip` και θα το ορίζαμε στο CSS αρχείο ως `p.tip: before { content: "Νέο! "}` . Με αυτό τον τρόπο μειώνεται η πολυπλοκότητα του κώδικα της HTML σελίδας.
- **After** : όμοια λειτουργία με το before μόνο που αντί να προσθέτει την επισήμανση πριν το στοιχείο (π.χ παράγραφο) το προσθέτει μετά από αυτό. Χρησιμοποιείτε για να προσθέσουμε εισαγωγικά σε μία παράθεση

Τέλος, θα ήταν παράληψη να μην αναφέρουμε ένα επιλογή CSS 3 με την ονομασία **::SELECTION** που χρησιμοποιείται κυρίως για να αλλάξει η μορφοποίηση ενός κειμένου όταν το επιλέγουμε. Ένας επιλογέας με τον οποίο ο χρήστης μπορεί να δει ξεκάθαρα πιο κομμάτι ιστοσελίδας έχει επιλέξει όταν για παράδειγμα θέλει να το αντιγράψει κάπου διαφορετικά. Οι περισσότεροι περιηγητές εξ' ορισμού το πράττουν αυτό, κάνοντας το φόντο του επιλεγμένου κειμένου σκούρο μπλε, αλλά αν θέλουμε το φόντο να έχει διαφορετικά χρώμα τότε αρκεί να προσθέσουμε την εντολή `::selection {background-color: #993366;}`

2.6 Μοντέλο Κουτιού CSS

Τα στοιχεία μιας σελίδας HTML μπορεί να θεωρηθούν ως κουτιά με διαφορετικές διατάξεις (το ένα εμφωλευμένο στο άλλο, το ένα δίπλα/πάνω/κάτω από κάποιο άλλο). Στην CSS ο όρος “μοντέλο κουτιού” χρησιμοποιείται όταν μιλάμε για το σχεδιασμό και τη διάταξη των στοιχείων. Το μοντέλο CSS κουτί είναι ουσιαστικά ένα κουτί που τυλίγεται γύρω από τα στοιχεία HTML και αποτελείται από περιθώρια, περιγράμματα, το περιεχόμενο της ιστοσελίδας (εικόνα 10).

Εξωτερικά υπάρχει το κουτί `margin` που μέσα του περικλείει το κουτί `border` που με την σειρά του περικλείει το κουτί `padding`.



Εσωτερικά όλων υπάρχει το περιεχόμενο που είναι γραμμένο σε HTML. Αναλυτική αναφορά για τις προηγούμενες ιδιότητες γίνεται στην επόμενη ενότητα

2.7 Ιδιότητες CSS

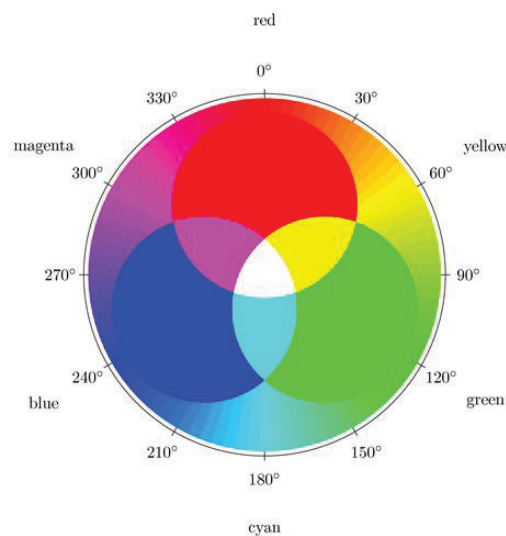
Ήδη σε προηγούμενες ενότητες αναφέραμε κάποιες από τις ιδιότητες μορφοποίησης όπως η διαφορετική γραμματοσειρά, το χρώμα ή το φόντο που έχει αυτή καθώς και σε ιδιότητες που συναντάμε στο μοντέλο κουτιού. Στην συνέχεια θα αναφερθούμε με μεγαλύτερη λεπτομέρεια στην κάθε μία από αυτές.

2.7.1 Γενικές Ιδιότητες

- ◆ **Margin** : Ορίζει τον χώρο μεταξύ των ορίων ενός στοιχείου και όσων το περιβάλλουν. Είναι ιδιαίτερα σημαντικό να ορίζουμε margin σε εικόνες, ώστε να έχουν απόσταση από το κυρίως περιεχόμενο. Για να ορίσουμε διαφορετικά margin για κάθε πλευρά χρησιμοποιούμε τις margin-top, margin-right, margin-bottom και margin-left. Το margin μπορεί να λάβει τέσσερις τιμές τις auto (ο περιηγητής υπολογίζει το margin), length (ο σχεδιαστής ορίζει το margin σε μια μονάδα px, pt, cm) και % (το margin υπολογίζεται βάσει του πλάτους του περιεχομένου)
- ◆ **Padding** : Ορίζει το κενό που θα υπάρχει μεταξύ των ορίων ενός στοιχείου και των περιεχομένων του. Είναι πολύ σημαντικό να ορίζουμε padding σε στοιχεία στα οποία έχουμε ορίσει κάποιο περίγραμμα, ώστε να μην επικαλύπτονται τα περιεχόμενα τους με το περίγραμμα. Ομοίως με το margin και εδώ για ορίσουμε διαφορετικό padding ανά πλευρά, χρησιμοποιούμε τις ιδιότητες padding-top, padding-right, padding-bottom και padding-left. Το padding μπορεί να λάβει δύο τιμές τις length (για καθορισμένο padding σε μια μονάδα px, pt, cm) και % (το padding υπολογίζεται βάσει του περιεχομένου)
- ◆ **Width** : Ορίζει το πλάτος ενός στοιχείου είτε σε κάποια μονάδα μήκους είναι σε ποσοστό
- ◆ **Height** : Ορίζει το ύψος ενός στοιχείου είτε σε κάποια μονάδα μήκους είναι σε ποσοστό
- ◆ **Float** : Ορίζει την διάταξη των στοιχείων σε μια σελίδα. Δέχεται τις τιμές right, left και none και επιτρέπει τα στοιχεία που περιβάλλουν το στοιχείο εκείνο στο οποίο εφαρμόζουμε την ιδιότητα αυτή να “ρέουν” τριγύρω του

2.7.2 Ιδιότητες Μορφοποίησης Κειμένου

- **COLOR** : Αναφέρεται στο χρώμα της γραμματοσειράς του κειμένου. Τα χρώματα στην CSS ακολουθούν το πρότυπο RGB (Red,Green,Blue) δηλαδή όλους τους πιθανούς συνδυασμούς που προκύπτουν από τα χρώματα κόκκινο, πράσινο και μπλε. Η αναφορά σε ένα συγκεκριμένο χρώμα μπορεί να γίνει βάσει του δεκαεξαδικού ή του δεκαδικού συστήματος με τιμές από hex 00 – hex FF ή από 0 -255 αντίστοιχα.



Εικόνα 11

Τέλος, στα χρώματα μπορούμε να αναφερθούμε απευθείας και με την ονομασία τους όπως magenta, red, yellow, green κ.α. Ο αναγνώστης μπορεί να επισκεφθεί την [ιστοσελίδα της W3C](#) όπου μπορεί να βρει όλα τα χρώματα με τις αντίστοιχες τιμές τόσο στην δεκαδική όσο και στην δεκαεξαδική μορφή.

- **FONT**: Σχετίζεται με το μέγεθος, το στυλ, τον τύπο γραφής και σε ποια οικογένεια φόντου της γραμματοσειράς ανήκει.
 - **Font-size** → Αφορά το μέγεθος των γραμμάτων. Οι τιμές που δέχεται μπορούν να εκφραστούν σε pixels ή σε μονάδες em⁶
 - **Font-style** → Αφορά τον τρόπο γραφής ο οποίος μπορεί να αλλάξει σε πλάγιο (italic και oblique) ή να παραμείνει απλό (normal)
 - **Font-family** → Αφορά τον τύπο της γραμματοσειράς και ο δημιουργός της ιστοσελίδας μπορεί να διαλέξει μέσα από τρεις οικογένειες γραμματοσειρών Serif, Sans-serif και Monospace

⁶ Μια μονάδα em ισούται με 16 pixels

- **ALIGNMENT**: Σχετίζεται με την στοίχιση του κειμένου μέσα στην σελίδα. Μπορεί να είναι κεντρική (center), δεξιά / αριστερή ή πλήρης (justify) στην οποία όλες οι γραμμές του κειμένου έχουν ίδια απόσταση από το δεξί και αριστερό άκρο της σελίδας.
 - **DECORATION** : Προκειμένου να τονίζουμε κάποια επικεφαλίδα ή κάποιο σημαντικό κομμάτι του κειμένου χρησιμοποιούμε υπογράμμιση. Μπορεί να λάβει τρεις τιμές, υπογράμμισης πάνω από το κείμενο (overline), διαπέρασης του κειμένου (line-through) και υπογράμμισης κάτω από το κείμενο (underline).
 - **TRANSFORMATION** : Σχετίζεται με τον τρόπο γραφής μιας γραμματοσειράς σε κεφαλαία (uppercase) ή πεζά (lowcase). Μια επιπλέον λειτουργία είναι και η capitalize κατά την οποία το πρώτο γράμμα από κάθε λέξη τυπώνεται με κεφαλαία.
1. **EFFECTS** : Η 3^η έκδοση της CSS έχει εισάγει δύο εφέ το πρώτο είναι η δημιουργία σκιάς σε ένα κείμενο (text-shadow) καθώς και το ταίριασμα λέξεων σε πλαίσιο χωρίς να βγαίνουν εκτός ορίων (word-wrap) και το χωρισμό αυτών προκειμένου να χωρέσουν.

2.7.3 Ιδιότητες Μορφοποίησης Φόντου

Το φόντο είναι ίσως ένα από τα πιο σημαντικά στοιχεία μιας ιστοσελίδας. Θα πρέπει να έχει επιλεγεί με τέτοιο τρόπο ώστε να αναδεικνύει καλύτερα την γραμματοσειρά του κειμένου και να μην κουράζει το μάτι του χρήστη. Γι αυτό τον λόγο, χρησιμοποιούνται λιγότερο έντονα χρώματα (James, 2007). Βέβαια αυτό έγκειται στην αισθητική του σχεδιαστή, αλλά είναι σημαντικό να γίνει σωστή επιλογή χρωμάτων αν επιθυμεί να προσελκύσει χρήστες που θα επισκέπτονται την ιστοσελίδα του συχνά. Ιδιότητες που μεταβάλλουν το φόντο είναι:

- a. **background-color** : Χρησιμοποιείται για να ορίσει χρώμα φόντου στα στοιχεία που αφορά ο επιλογέας. Το χρώμα μπορεί να γραφεί σε οποιαδήποτε από τις μορφές που περιγράφηκαν για την ιδιότητα COLOR της 2.6.1 ενότητας
- b. **background-image** : Χρησιμοποιείται για να ορίσει μια εικόνα φόντου. Η τιμή που δέχεται είναι η διεύθυνση στην οποία βρίσκεται η εικόνα που θέλουμε να χρησιμοποιήσουμε π.χ. `background-image: url(img_flwr.gif)`. Η 3^η έκδοση της CSS μας βοηθάει να έχουμε για φόντο πολλές εικόνες απλά εισάγοντας τις διευθύνσεις των εικόνων την μία πίσω από την άλλη

- c. **background-repeat** : Σχετίζεται με τον τρόπο που θα επαναλαμβάνεται η εικόνα που ορίσαμε στην προηγούμενη ιδιότητα. Οι τιμές που δέχεται είναι: no-repeat (καμία επανάληψη) repeat-x (επαναλαμβάνεται γύρω από τον οριζόντιο άξονα), repeat-y (επαναλαμβάνεται γύρω από τον κάθετο άξονα) και repeat (επαναλαμβάνεται γύρω και από τους δύο άξονες)
- d. **background-position** : Η ιδιότητα αυτή επιτρέπει να ορίσουμε τη θέση που θα τοποθετηθεί η εικόνα φόντου βάσει της ιδιότητας *background-image* σε σχέση με τους δύο άξονες. Συγκεκριμένα χρησιμοποιούμε τις λέξεις left, right, και center για να ορίσουμε την θέση με άξονα αναφοράς τον οριζόντιο και τις λέξεις top, bottom και center βάσει του κάθετου άξονα.
- e. **background-size** : Πρόκειται για νέα ιδιότητα της 3^{ης} έκδοσης που καθορίζει το μέγεθος της εικόνας φόντου. Πριν την CSS 3, το μέγεθος της εικόνας φόντου προσδιοριζόταν από το πραγματικό μέγεθος της εικόνας. Με την εντολή αυτή, κάθε εικόνα μπορεί να μεταβάλει το μέγεθός της το οποίο καθορίζεται σε pixels ή σε ποσοστά.
- f. **background-origin** : Προσδιορίζει το χώρο τοποθέτησης των εικόνων φόντου. Η εικόνα φόντου μπορεί να τοποθετηθεί στο content-box, στο padding-box, ή περιοχή border-box (εικόνα 11)



Εικόνα 12

- g. **background** : Η ιδιότητα αυτή αποτελεί ουσιαστικά συντόμευση για να ορίσουμε με μία μόνο ιδιότητα όλες τις παραπάνω. Η CSS 3 έχει εισάγει ένα καινούριο εφέ διαβάθμισης του χρώματος του φόντου που πριν γινόταν με την χρήση εικόνων. Αυτό έχει μειώσει το χρόνο φόρτωσης μιας ιστοσελίδας καθώς και την χρήση bandwidth. Μπορεί να λάβει δύο τύπους διαβάθμισης χρώματος linear-gradient και radial-gradient

2.7.4 Ιδιότητες Περιγράμματος

Η CSS (Zihou, 2009) δίνει την ιδιότητα εισαγωγής πλαισίων που ομορφαίνουν ακόμα πιο πολύ μια ιστοσελίδα. Πέρα από την στιλιστική τους ιδιότητα είναι πολύ χρήσιμο εργαλείο για να καταδεικνύουμε σημαντικά κομμάτια στην ιστοσελίδα μας. Ιδιότητες διαχείρισης πλαισίων είναι :

- i. **border-color** : Με την ιδιότητα αυτή ρυθμίζεται το χρώμα περιγράμματος. Το χρώμα μπορεί να γραφεί σε οποιαδήποτε από τις μορφές που περιγράφηκαν για την ιδιότητα COLOR της 2.6.1 ενότητας
- ii. **border-width** : Με την ιδιότητα αυτή ρυθμίζεται το πάχος του περιγράμματος σε κάποια από τις μονάδες μέτρησης, pixels ή em
- iii. **border-style** : Με την ιδιότητα αυτή ρυθμίζεται το στυλ του περιγράμματος. Οι τιμές που χρησιμοποιούνται συνήθως για την ιδιότητα αυτή είναι οι εξής:
 - *solid* : Περίγραμμα μιας ενιαίας γραμμής
 - *dashed* : Περίγραμμα που αποτελείται από παύλες
 - *dotted* : Περίγραμμα που αποτελείται από τελείες
 - *double* : Δημιουργεί δύο περιγράμματα
- iv. **border-radius** : Με την ιδιότητα αυτή κατασκευάζονται περιγράμματα με στρογγυλεμένες άκρες
- v. **box-shadow** : Με την ιδιότητα αυτή κατασκευάζονται πλαίσια σε σχήμα τετράγωνου με την προσθήκη του εφέ σκίασης
- vi. **border-image** : Με την ιδιότητα αυτή μπορούμε να χρησιμοποιήσουμε μια εικόνα για να δημιουργήσουμε ένα πλαίσιο
- vii. **border** : Η ιδιότητα αυτή αποτελεί ουσιαστικά συντόμευση για να ορίσουμε με μία μόνο ιδιότητα όλες τις παραπάνω.

2.8 Ιδιότητες CSS 3

Η 3^η έκδοση της CSS προσφέρει νέες και μοναδικές ιδιότητες στους σχεδιαστές ιστοσελίδων που οι προηγούμενες εκδόσεις δεν διέθεταν. Ήδη έχουμε αναφέρει μερικές στον σχεδιασμό περιγραμμάτων και κειμένων, υπάρχουν όμως και άλλες που μπορούν προσδίδουν ακόμα και το εφέ κίνησης στο περιεχόμενο της σελίδας.

2.8.1 Μετασχηματισμοί

Με τους μετασχηματισμούς (transforms) της CSS 3 μπορούμε να μετακινήσουμε, να αλλάξουμε την κλίμακα, να περιστρέψουμε και να εκτείνουμε στοιχεία. Ο μετασχηματισμός είναι ένας μηχανισμός που επιτρέπει να αλλάξουν σχήμα μέγεθος και θέση μέσα στην ιστοσελίδα. Μπορούμε να μεταβάλλουμε τα στοιχεία χρησιμοποιώντας είτε δισδιάστατο (2D) είτε τρισδιάστατο (3D) μετασχηματισμό.

2D Transforms : Ο δισδιάστατος μετασχηματισμός διαθέτει 5 μεθόδους οι οποίες μπορεί να επιδράσουν και να αλλάξουν ένα στοιχείο

1. Translate() → ένα στοιχείο μπορεί να κινηθεί από την τρέχουσα θέση του προς μια άλλη (άξονες x,y) η οποία δίνεται ως παράμετρος
2. Rotate() → ένα στοιχείο μπορεί να περιστραφεί δεξιόστροφα ή αριστερόστροφα
3. Scale() → ένα στοιχείο μπορεί να αυξομειώσει το μέγεθός του βάσει παραμέτρων που δίνονται
4. Skew() → ένα στοιχείο μπορεί να περιστραφεί κατά μια γωνία που δίνεται ως παράμετρος
5. Matrix() → συνδυάζει όλες τις προηγούμενες ιδιότητες σε μια

3D Transforms : Ο τρισδιάστατος μετασχηματισμός διαθέτει 2 μεθόδους οι οποίες μπορεί να επιδράσουν και να αλλάξουν ένα στοιχείο

1. rotateX() → με την μέθοδο αυτή ένα αντικείμενο μπορεί να περιστραφεί γύρω από τον άξονα X βάσει μιας δοσμένης γωνίας
2. rotateY() → όμοια με την rotate μόνο που εδώ περιστρέφεται γύρω από τον άξονα Y

2.8.2 Μεταβάσεις

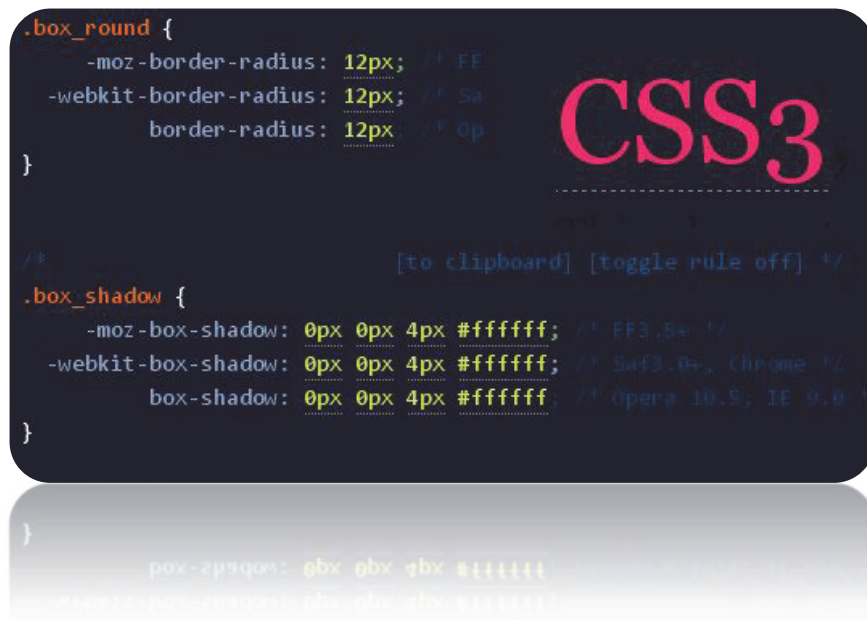
Με την CSS 3 μπορούμε να εισάγουμε ένα εφέ όταν μεταβαίνουμε από ένα στυλ μορφοποίησης σε ένα άλλο χωρίς να κάνουμε χρήση Flash ή Javascript. Η μετάβαση (Transition) έχει τέσσερις ιδιότητες την `transition - property` η οποία συγκεκριμενοποιεί το όνομα της CSS ιδιότητας πάνω στην οποία εφαρμόζεται η `transition`, την `transition - duration` που ορίζει την χρονική διάρκεια της μετάβασης, την `transition- timing-function` που δηλώνει πως υπολογίζεται η ταχύτητα μετάβασης και την `transition-delay` που ορίζει την χρονική στιγμή που αρχίζει η μετάβαση.

2.8.3 Κινούμενες εικόνες

Με CSS3, μπορούμε να δημιουργήσουμε κινούμενες εικόνες (animations), τα οποία μπορούν να αντικαταστήσουν animations που έχουν γίνει με Flash ή JavaScripts σε υπάρχουσες ιστοσελίδες. Η δημιουργία animations στην CSS3 γίνεται βάσει ενός κανόνα `@ keyframes` ο οποίος καθορίζει ένα στυλ CSS στο εσωτερικό του κανόνα `@ keyframes` καθώς και το `animation` που αλλάζει σταδιακά από το τρέχον στυλ στο νέο στυλ. Όταν ένα animation δημιουργείται από τον κανόνα `@ keyframe`, θα πρέπει να συνδεθεί με έναν επιλογέα, αλλιώς η κίνηση θα έχει καμία επίδραση στο αντικείμενο.

2.9 Σύνοψη

Στο κεφάλαιο αυτό έγινε ανάλυση των σχεδιαστικών δυνατοτήτων που προσφέρουν τα Διαδοχικά Φύλλα Στυλ καθώς και τις νέες ιδιότητες που έχει η 3^η έκδοση αυτών. Μια ιστοσελίδα για να παρουσιαστεί στον χρήστη θα πρέπει να υποστηρίζεται και από αντίστοιχα προγράμματα περιήγησης. Πολλοί φυλλομετρητές δεν μπορούν να αναγνωρίσουν τις ιδιότητες που αναφέραμε προηγουμένως και έτσι η ιστοσελίδα δεν μπορεί να γίνει διαθέσιμη στον χρήστη. Για τον λόγο αυτό οι μηχανικοί λογισμικού αναπτύσσουν και ανανεώνουν συνεχώς τα προγράμματα περιήγησης. Στο [ΠΑΡΑΡΤΗΜΑ Α](#) αναφέρονται όλες οι ιδιότητες της CSS καθώς και οι εκδόσεις των πιο σημαντικών browsers που μπορούν να τα υποστηρίξουν.



Εικόνα 13

Κεφάλαιο 3^ο

3. Εισαγωγή

Τα τεχνικά πρότυπα που χρησιμοποιούνται ευρέως σε διάφορους τομείς της ζωής, έτσι για παράδειγμα χρησιμοποιούμε άλλο μέγεθος φακέλου για να τοποθετήσουμε ένα χαρτί μεγέθους A4 και άλλο για μέγεθος A2. Τα πρότυπα Web (Web Standards), λειτουργούν παρόμοια με τα υπόλοιπα πρότυπα, μόνο που έχουν σαν σκοπό καθορίζουν το σύνολο των απαιτήσεων που πρέπει να ικανοποιεί μια ιστοσελίδα.

Τα πρότυπα δεν είναι απαραίτητα μόνο από τεχνικής άποψης αλλά γιατί ουσιαστικά ικανοποιούν τις ανάγκες των χρηστών. Ωστόσο, τα Web Standards συχνά αγνοούνται. Το διαδίκτυο αποτελείται από δισεκατομμύρια έγγραφα που δεν πληρούν περιορισμούς ή κανονισμούς, προκαλώντας σοβαρά προβλήματα. Αυτό οφείλεται στο γεγονός ότι το Διαδίκτυο χαρακτηρίζεται από ελευθερία, όπου ο καθένας έχει την δυνατότητα να αναρτήσει σελίδες ποικίλου περιεχομένου, ανεξαρτήτως αν διαθέτει ή όχι το τεχνικό υπόβαθρο. Δυστυχώς αυτό έχει ένα μεγάλο τίμημα στην περιήγηση των χρηστών μια και συναντούν συχνά ιστοσελίδες οι οποίες καθυστερούν αρκετά να εμφανίσουν το περιεχόμενό τους, έχουν ανεπαρκή εμφάνιση, ή έχουν κακή λειτουργικότητα.

3.1 Web Standards

“Τα Web standards σκοπεύουν να είναι μια κοινή βάση... μία δομή για ολόκληρο το web ώστε οι browsers κι άλλο λογισμικό να μπορούν να καταλαβαίνουν το ίδιο λεξιλόγιο”

Eric Meyer

Η W3C (World Wide Web Consortium) κι άλλες οργανώσεις έχουν αναπτύξει τεχνολογίες για τη δημιουργία και τη μετάφραση υλικού για το Διαδίκτυο. Τα Web Standards σχεδιάστηκαν ώστε να:

- προσφέρουν τα μεγαλύτερα πλεονεκτήματα σε όσο το δυνατό περισσότερους χρήστες
- προσφέρουν τη μεγαλύτερη δυνατή βιωσιμότητα αρχείων για το web απλοποιούν τον κώδικα και να μειώνουν το κόστος παραγωγής

- προσφέρουν δικτυακούς τόπους προσβάσιμους σε περισσότερους ανθρώπους και σε περισσότερες συσκευές σχετιζόμενες με το Internet
- συνεχίζουν να δουλεύουν σωστά καθώς οι φυλλομετρητές εξελίσσονται και νέες συσκευές δημιουργούνται για τις ανάγκες της αγοράς

Για τους web designers και τους developers, τα Web Standards σχετίζονται με τη χρήση standards (για τη δόμηση, την παρουσίαση κλπ υλικού) και με τις καλύτερες πρακτικές ώστε να ωφεληθούν οι χρήστες, οι πελάτες και οι ίδιοι.

3.2 Χαρακτηριστικά Ιστοσελίδων με Web Standards

Τα Web Standards αντιλαμβάνονται το web ως ένα γενικευμένο εργαλείο επικοινωνίας, το οποίο μπορεί να είναι προσβάσιμο (Μπαλής, n.d.) σε μεγάλες ομάδες ανθρώπων και συσκευών. Τα κοινά χαρακτηριστικά περιλαμβάνουν:

3.2.1 Λογική δόμηση κώδικα

Ο κώδικας που γράφει ένας προγραμματιστής θα πρέπει να διακρίνεται από λογική. Έτσι θα πρέπει να χρησιμοποιούνται html στοιχείων ανάλογα την μορφοποίηση που θέλουμε να έχουμε στο περιεχόμενο. Για παράδειγμα χρήση των heading tags για τις επικεφαλίδες, χρήση του p tag για την κατασκευή παραγράφων, εισαγωγή του li tag για κατασκευή λιστών κτλ.

3.2.2 Εγκυρότητα κώδικα

Ο κώδικας μιας ιστοσελίδας απαραίτητα πρέπει να είναι έγκυρος. Η διαδικασία κατά την οποία ένα κομμάτι κώδικα αξιολογείται για την εγκυρότητά του ονομάζεται Validation. Στην περίπτωση μιας ιστοσελίδας, το αρχείο μας, ελέγχεται κατά πόσο πληροί τα W3C Standards που έχουν καθοριστεί.

Η διαδικασία του Validation είναι πολύ σημαντική γιατί ένας έγκυρος κώδικας θα "διαβαστεί" γρηγορότερα και πιο σωστά από το μη έγκυρο, ενώ οι browsers γίνονται με το χρόνο συμβατοί με τα standards και αυτό έχει ως αποτέλεσμα να επιβάλλεται το γράψιμο έγκυρου κώδικα

3.2.3 Προσβασιμότητα κώδικα

Η προσβασιμότητα είναι άλλο ένα πολύ χρήσιμο χαρακτηριστικό που πρέπει να έχουν οι ιστοσελίδες. Ο προγραμματιστής θα πρέπει να λαμβάνει υπόψη ότι η ιστοσελίδα του θα πρέπει να είναι διαθέσιμη στους περισσότερους ανθρώπους, ακόμα και σε αυτού που έχουν με προβλήματα όρασης ή άλλες ειδικές ανάγκες. Επιπλέον, θα πρέπει να είναι διαθέσιμο το περιεχόμενό του όχι μόνο από τις οθόνες των υπολογιστών αλλά και από άλλες συσκευές όπως screen readers, text browsers, search engines. Τέλος, η προσβασιμότητα είναι βασική προϋπόθεση μιας ιστοσελίδας που κατασκευάζεται για κρατικούς σκοπούς, όπως για παράδειγμα η ιστοσελίδα της εφημερίδας της κυβέρνησης, της βουλής κ.α.

Ένας προγραμματιστής που θέλει να γράψει προσβάσιμο κώδικα θα πρέπει να :

- Χρησιμοποιεί προσβάσιμους πίνακες κειμένου
- Χρησιμοποιεί προσβάσιμες φόρμες (label for, id, legend)
- Χρησιμοποιεί κώδικα αντί εικόνων για να μεταφέρει την πληροφορία
- Παρέχει επιλογές αποφυγής των μενού
- Παρέχει πλήκτρα πρόσβασης
- Χρησιμοποιεί style sheets με τρόπο ώστε να ελέγχει το πως εμφανίζεται μια σελίδα
- Επιτρέπει το διάβασμα μιας σελίδας ακόμα και χωρίς style sheets
- Παρέχει metadata για να πληροφορήσει τους χρήστες του

3.2.4 Διαχωρισμός περιεχομένου

Οι web developers στοχεύουν στο να απομακρύνουν όλα εκείνα τα κομμάτια κώδικα που καθορίζουν πως θα εμφανίζεται με σελίδα από το περιεχόμενο αυτής αφήνοντας τον HTML κώδικα όσο πιο "καθαρό" γίνεται. Αυτό προσφέρει:

- ◆ Σελίδες προσβάσιμες σε περισσότερες συσκευές
- ◆ Ευκολότερη αλλαγή του τύπου στο μέλλον, καθώς η εμφάνιση της σελίδας καθορίζεται από ένα μόνο αρχείο
- ◆ Μικρότερα αρχεία/γρηγορότερο "κατέβασμα" των σελίδων - λιγότερος κώδικας σε μια σελίδα
- ◆ Λιγότερο κώδικα σε μια σελίδα - ευκολότερο γράψιμο νέου κώδικα
- ◆ Οι χρήστες μπορούν να χρησιμοποιήσουν διαφορετικά στυλ ανάλογα με τις ανάγκες τους
- ◆ Μεγαλύτερο και καλύτερο έλεγχο του κώδικα

3.3 Πλεονεκτήματα χρήσης Web Standards

Τα πλεονεκτήματα της χρήσης Web Standards είναι πάρα πολλά και ωφελούν και τις τρεις ενδιαφερόμενες ομάδες, του επισκέπτη, του επενδυτή και του προγραμματιστή. Έτσι :

1. **Ο Χρήστης** : μπορεί να “κατεβάσει ” αρχεία πιο γρήγορα, να διαβάσει το περιεχόμενο ακόμα και αν είναι χρήστης με προβλήματα όρασης γενικώς ή χρήστης με δυσλεξία, να δει το περιεχόμενο και απο άλλες συσκευές, να προσαρμόσει την εμφάνιση του τόπου ανάλογα με τις ανάγκες του και τέλος να εκτυπώσει την ιστοσελίδα ένα το επιθυμεί.
2. **Ο πελάτης** : μπορεί πιο εύκολα να ανανεώσει τον κώδικα, να έχει μικρότερα έξοδα φιλοξενίας (hosting), λόγω του λιγότερου κώδικα, να έχει καλύτερη σειρά στις μηχανές αναζήτησης, να διαθέτει περιεχόμενο που μπορεί να παρουσιαστεί διαφορετικά χωρίς να αλλαχθεί ο κώδικας στα HTML αρχεία, να δίνει την επιλογή στους χρήστες να μπορούν να εξατομικεύσουν την εμφάνιση καθώς και να εκτυπώσουν γρήγορα και φθηνά περιεχόμενα του τόπου
3. **Ο προγραμματιστής** : μπορεί να γράψει πιο εύκολο και πιο μικρό κώδικα, ο οποίος να εξαρτάται λιγότερο από τις συσκευές ανάγνωσής του, γεγονός που θα αυξήσει την ανταγωνιστικότητά, καθώς ο κόσμος του web κινείται προς τη γενικότερη χρήση Web Standards.

3.4 Μετρικές

Στην επόμενη ενότητα θα αναφέρουμε σημαντικούς δείκτες που μπορούν να βελτιώσουν την ποιότητα του κώδικα CSS και κατ’ επέκταση την ποιότητα μιας ιστοσελίδας. Παράλληλα παρουσιάζονται οδηγοί που πρέπει να ακολουθήσει ένας προγραμματιστής εάν επιθυμεί να κατασκευάσει ιστοσελίδες που θα μεγιστοποιούν την εμπειρία του χρήστη.

3.4.1 Απόδοση

Η απόδοση (Performance) μιας ιστοσελίδας ήταν πάντα ένας παράγοντας ζωτικής (Tammy, 2013) σημασίας για την επιτυχία μιας ιστοσελίδας. Οι ερευνητές έχουν καταλήξει πως ακόμα και η πιο μικρή βελτίωση στον χρόνο φόρτωσης μιας ιστοσελίδας, μπορεί να οδηγήσει σε αύξηση πωλήσεων των προϊόντων, σε περισσότερα έσοδα από τις διαφημίσεις που περιέχονται σε αυτή και γενικότερα να προσφέρουν μεγαλύτερη ικανοποίηση στους πελάτες.

Τα προηγούμενα χρόνια οι προγραμματιστές ιστοσελίδων κατέφευγαν στην βελτίωση του υλικού τους συστήματος στο οποίο έτρεχε η ιστοσελίδα, αλλά και στην αύξηση του εύρους ζώνης (bandwidth) προκειμένου να προσφέρουν την βέλτιστη εμπειρία στον χρήστη. Τα τελευταία χρόνια όμως, ιδίως λόγω της εκτεταμένης περιήγησης στο διαδίκτυο μέσω κινητών συσκευών, αυτός ο τρόπος δεν ακολουθείτε. Το χαμηλό εύρος ζώνης καθώς και η (Hall, 2009) μικρή επεξεργαστική ισχύς των κινητών συσκευών, κάνουν επιτακτική την ανάγκη για βελτιστοποίηση μιας ιστοσελίδας στο front-end επίπεδο, προκειμένου να ικανοποιήσουν τις προσδοκίες των χρηστών.

Δεν έχει σημασία πόσο ενδιαφέρουσα, όμορφη ή διαδραστική είναι μια ιστοσελίδα. Εάν ο χρόνος φόρτωσής της, από έναν επιτραπέζιο υπολογιστή ή από μια κινητή συσκευή, διαρκέσει περισσότερο από δύο με τρία δευτερόλεπτα, ο χρήστης γίνεται ανυπόμονος και είναι πολύ πιθανόν να την εγκαταλείψει γρήγορα (Oehler Matthias, 2008). Έχει υπολογισθεί πως ο μέσος χρήστης μπορεί να μεταβεί σε νέα σελίδα, ή να πατήσει το κουμπί back για να μεταβεί σε προηγούμενη σελίδα, ακόμα και να κλείσει το πρόγραμμα περιήγησης πριν καν φορτώσει η ιστοσελίδα που είχε ζητήσει. Ακόμη και καθυστερήσεις που διαρκούν λιγότερο από ένα δευτερόλεπτο μπορούν να επηρεάσουν σημαντικά τα έσοδα.

Το 2006 ο Marissa Mayer (Tammy, 2013), πειραματίστηκε με τον δείκτη απόδοση μιας ιστοσελίδας και συγκεκριμένα με την σελίδα επιστροφής αποτελεσμάτων στην μηχανή Google. Οι χρήστες είχαν αναφέρει σε σχόλιά τους ότι θα επιθυμούσαν να παίρνουν πάνω από 10 αποτελέσματα ανά σελίδα κατά την αναζήτησή τους. Έτσι η Google αύξησε αυτό το όριο σε 30 αποτελέσματα ανά σελίδα. Προς μεγάλη τους έκπληξη είδαν πως τα έσοδα της εταιρίας μειώθηκαν κατά 20% και αυτό γιατί η σελίδα απαιτούσε μισό δευτερόλεπτο για να φορτώσει τα επιπλέον αποτελέσματα.

Οι προσδοκίες των χρηστών έχουν κλιμακωθεί από τότε. Μια μελέτη του 2009 από τη Forrester Research για λογαριασμό της τεχνολογίας Akamai, εντόπισε ως κατώτατο όριο αποδεκτού χρόνου απόκρισης μιας ιστοσελίδας στο ιντερνέτ, τα δύο δευτερόλεπτα και διαπίστωσε ότι το 40% των χρηστών εγκαταλείπουν μια σελίδα που η διάρκεια φόρτωσης του περιεχομένου της είναι περισσότερο από τρία

δευτερόλεπτα. Μόλις ένα χρόνο αργότερα, στην ίδια έρευνα, το ποσοστό αυτό αυξήθηκε στο 57% .

3.4.1.1 Οδηγός Βελτίωσης της Απόδοσης

Για να έχουμε την βέλτιστη εμπειρία χρήστη θα πρέπει να ακολουθηθούν διάφοροι κανόνες τόσο από την πλευρά του πελάτη (client) όσο και από την πλευρά του εξυπηρετητή της ιστοσελίδας (server). Στην συνέχεια θα αναφέρουμε μερικούς χωρισμένους σε τρεις κατηγορίες.

Ελαχιστοποίηση HTTP Αιτήσεων

Οι πολλές αιτήσεις που μπορεί να γίνουν από μια ιστοσελίδα αυξάνουν τον χρόνο RTT (round-trip-time) που χρειάζεται ένα πακέτο δεδομένων για να μεταφερθεί στον τελικό του προορισμό. Αυτό έχει σαν αποτέλεσμα να επιβραδύνει την απόδοση αλλά και την εκτέλεση μιας ιστοσελίδας . Συνήθως, μια ιστοσελίδα μπορεί να παράγει πρόσθετα αιτήματα HTTP για εικόνες, φύλλα στυλ (π.χ. CSS), αρχεία script (π.χ. Javascript). Για να μειώσουμε τον αριθμό αυτό πρέπει :

- Όταν έχουμε πολλαπλά αρχεία style sheets τα οποία χρειάζονται για να διαμορφώσουμε μια ιστοσελίδα θα πρέπει να τα συγχωνεύουμε σε ένα αρχείο πριν το στείλουμε στον χρήστη
- Όταν υπάρχουν πολλές φωτογραφίες οι οποίες χρησιμοποιούνται σαν φόντο στην ιστοσελίδα τότε θα πρέπει να τις συνδυάζουμε και να δημιουργούμε μία φωτογραφία που θα είναι μια οντότητα CSS Sprite. Στην συνέχεια θα πρέπει να τροποποιήσουμε και τον κώδικα HTML προκειμένου να καλεί τις εικόνες από το CSS Sprite
- Εάν υπάρχουν ενσωματωμένες εικόνες, θα πρέπει να ενσωματώνεται το περιεχόμενο της απάντησης HTTP στη θέση ενός URL
- Να τροποποιήσουμε την ιστοσελίδα έτσι ώστε όλα τα ενσωματωμένα αντικείμενα έχουν σχετικό Expiration Header το οποίο ουσιαστικά δηλώνει πότε χρησιμοποιούνται τα αντικείμενα. Αυτό θα αποτρέψει το πρόγραμμα περιήγησης του πελάτη να ξαναζητήσει αντικείμενα που χρησιμοποιήθηκαν πρόσφατα
- Να εξετάσουμε το HTML αρχείο για να δούμε αν υπάρχουν ανακατευθύνσεις σε όλες τις συνδέσεις HTTP. Εάν υπάρχουν τότε θα πρέπει να τις αντικαταστήσουμε με την αρχική πλήρη διεύθυνση URL

Ελαχιστοποίηση του μεγέθους των Web αντικειμένων

Από πλευρά κατασκευής του κώδικα μιας ιστοσελίδας, θα πρέπει επίσης να εφαρμόσουμε ένα σύνολο κανόνων που με την σειρά τους, θα αυξήσουν την απόδοση της ιστοσελίδας. Είναι απαραίτητο να :

- Εφαρμόσουμε συμπίεση (π.χ. gzip) σε όλα τα αρχεία HTML, τα αρχεία script και στυλ μορφοποίησης που θα αποσταλούν στον χρήστη και να αναφέρουμε ρητά ότι έχουμε στείλει συμπιεσμένα αρχεία στην κωδικοποίηση του περιεχομένου της απόκρισης HTTP
- Εκτελέσουμε μια διαδικασία γνωστή ως “σμίκρυνση”. Για παράδειγμα, θα πρέπει να εξετάσουμε τα αρχεία με JS και CSS κώδικα και να εντοπίσουμε περιττούς χαρακτήρες από τον κώδικα (π.χ. σχόλια, διάστημα, κ.λπ.) και στην συνέχεια να τους αφαιρέσουμε
- Εκτελέσουμε συσκότιση δηλαδή να μειώσουμε το μήκος των ονομάτων που έχουμε δώσει στα σύμβολά μας και τέλος
- Βελτιστοποιήσουμε τα γραφικά της ιστοσελίδας μας και συγκεκριμένα να βελτιστοποιήσουμε κινούμενες εικόνες GIF, προοδευτικές εικόνες τύπου JPEG κ.τ.λ.

Αναδιοργάνωση σύνθεσης της ιστοσελίδας

Η θέση των διαφόρων στοιχείων μέσα στο αρχείο της ιστοσελίδας μπορεί να επηρεάσει σημαντικά την ταχύτητα με την οποία εκτελείται η σελίδα. Για να αναδιοργανώσουμε με σωστό τρόπο την ιστοσελίδα θα πρέπει να:

- Τροποποιήσουμε την ιστοσελίδα ώστε το αρχείο στυλ CSS να βρίσκεται στην κορυφή του αρχείου HTML επιτρέποντας βαθμιαία απόδοση
- Τροποποιήσουμε την ιστοσελίδα ώστε το κομμάτι κώδικα script να βρίσκονται στο κάτω μέρος του αρχείου HTML, επιτρέποντας έτσι παράλληλες λήψεις και τέλος,
- Ελέγξουμε με διάφορα σενάρια, είτε τοποθετώντας τα script κομμάτια και τα αρχεία style sheet ενσωματωμένα στο HTML αρχείου ή εξωτερικά σε άλλο αρχείο, προκειμένου να δούμε ποιος τρόπος αυξάνει την απόδοση της ιστοσελίδα περισσότερο

3.4.2 Προσβασιμότητα

Η δυνατότητα πρόσβασης στο διαδίκτυο σημαίνει ότι οι ανάπηροι άνθρωποι μπορούν να χρησιμοποιήσουν το διαδίκτυο. Πιο συγκεκριμένα, η δυνατότητα πρόσβασης στο διαδίκτυο σημαίνει ότι οι ανάπηροι άνθρωποι μπορούν να αντιληφθούν, να καταλάβουν, να πλοηγηθούν, και να αλληλεπιδράσουν στο Διαδίκτυο και μπορούν να συμμετέχουν σ' αυτό (Termens Miquel, 2009) (Cornelia, 2002). Η δυνατότητα πρόσβασης στο Διαδίκτυο ωφελεί επίσης και άλλους, συμπεριλαμβανομένων και των ηλικιωμένων ανθρώπων που λόγω της γήρανσης οι δυνατότητές τους είναι διαφορετικές.

Η δυνατότητα πρόσβασης στο Διαδίκτυο καλύπτει όλες τις ανικανότητες που επηρεάζουν την πρόσβαση στο Διαδίκτυο και περιλαμβάνει: οπτικές, ακουστικές, σωματικές, δυσκολίες ομιλίας, γνωστικές, και νευρολογικές ανικανότητες. Εκατομμύρια ανθρώπων με ανικανότητες έχουν επιπτώσεις στη χρήση του Διαδικτύου (Lourdes Moreno, 2008). Αυτήν την περίοδο, οι περισσότερες ιστοσελίδες και λογισμικά παρουσιάζουν πολλές δυσκολίες στη δυνατότητα πρόσβασης και η χρήση τους γίνεται πιο δύσκολη ή αδύνατη για πολλούς ανάπηρους ανθρώπους. Δεδομένου ότι οι ιστοσελίδες και τα λογισμικά θα διατίθενται πιο εύκολα, οι ανάπηροι άνθρωποι θα είναι σε θέση να χρησιμοποιήσουν και να συμμετέχουν στο Διαδίκτυο αποτελεσματικότερα.

Η δυνατότητα πρόσβασης στο Διαδίκτυο ωφελεί επίσης και τους ανθρώπους χωρίς ανικανότητες. Παραδείγματος χάριν, μια βασική αρχή της δυνατότητας πρόσβασης Διαδικτύου είναι να σχεδιάζει ιστοσελίδες και λογισμικά που είναι ευέλικτα για να ικανοποιήσουν τις διαφορετικές ανάγκες, τις προτιμήσεις, και τις καταστάσεις των χρηστών. Αυτή η ευελιξία ωφελεί επίσης τους ανθρώπους χωρίς ανικανότητες σε ορισμένες καταστάσεις, όπως οι άνθρωποι που χρησιμοποιούν μια αργή σύνδεση στο Διαδίκτυο, οι άνθρωποι με τις «προσωρινές ειδικές ανάγκες» όπως ένας σπασμένος βραχίονας, και οι άνθρωποι με τις διαφορετικές δυνατότητες λόγω της γήρανσης.

3.4.2.1 Οδηγός Προσβασιμότητας Ιστοσελίδων

Η προσβασιμότητα στον Παγκόσμιο Ιστό καλύπτει τις πρακτικές ανάπτυξης ιστοσελίδων που παρέχουν περιεχόμενο που μπορεί να χρησιμοποιηθεί από όλους, συμπεριλαμβανομένων των ατόμων με ειδικές ανάγκες (Bailey John, 2006). Ο διευθυντής του W3C, Tim Berners-Lee, ανακοίνωσε την έναρξη του Διεθνούς Προγράμματος (IPO) με στόχο την προσβασιμότητα - Web Accessibility Initiative (WAI) (Leitner Michael, 2005) το 1997 και όρισε την προσβασιμότητα ως εξής: «Η

δύναμη του Παγκόσμιου Ιστού έγκειται στην καθολικότητά του. Η πρόσβαση από όλους, ανεξάρτητα εάν κάποιος αντιμετωπίζει κάποια αναπηρία, είναι πολύ σημαντικός παράγοντας». Ο οργανισμός W3C εκδίδει για πρώτη φορά το 1999 έναν οδηγό προσβασιμότητας ιστοσελίδων με την ονομασία Web Content Accessibility Guidelines, (WCAG 1.0) και διαφημίζει την σημαντικότητά (Guarino Reid Loretta, 2008) του με το σλόγκαν «Πώς να κάνετε προσβάσιμο το περιεχόμενο του Διαδικτύου, στα άτομα με αναπηρία».

Ο οδηγός WCAG 1.0 αποτελείται από 14 οδηγούς (Sikos, 2011) (pg. 369):

1. Όταν έχουμε οπτικοακουστικό περιεχόμενο στις ιστοσελίδες μας πρέπει να παρέχουμε ισοδύναμες εναλλακτικές λύσεις για τα άτομα με αναπηρίες
2. Οι πληροφορίες που τονίζονται με συγκεκριμένα σε χρώματα (π.χ κόκκινο χρώμα για μια νέα δημοσίευση) πρέπει να είναι διαθέσιμες και αντιληπτές και σε γραμματοσειρές χωρίς χρώματα
3. Η σήμανση (markup) και φύλλα μορφοποίησης στυλ (CSS) θα πρέπει να εφαρμόζονται σωστά
4. Η φυσικές γλώσσες των εγγράφων του Ιστού πρέπει να δηλώνονται.
5. Οι πίνακες πρέπει να κατασκευάζονται κατά τρόπο που να μπορούν να μετασχηματίζονται ομαλά
6. Οι σελίδες που εφαρμόζουν νέες τεχνολογίες (HTML, JavaScript, Ajax, CSS) πρέπει επίσης να μετασχηματίζονται με ομαλό τρόπο
7. Πρέπει να παρέχεται η δυνατότητα ελέγχου από τον χρήστη όταν έχουμε ιστοσελίδες που αλλάζουν δυναμικά το περιεχόμενό τους μέσα στον χρόνο
8. Πρέπει να εξασφαλίζεται άμεση προσβασιμότητα όλων των ενσωματωμένων διεπαφών
9. Ο σχεδιασμός των ιστοσελίδων θα πρέπει να είναι ανεξάρτητος από τη συσκευή που εκτελείται
10. Πρέπει να χρησιμοποιούνται ενδιάμεσες λύσεις.
11. Τεχνολογίες και οδηγίες που έχουν εκδοθεί από τον W3C πρέπει να ακολουθούνται
12. Πρέπει να παρέχονται πληροφορίες σχετικά με το πλαίσιο και τον προσανατολισμό
13. Η πλοήγηση μέσα πρέπει να είναι εύκολη στην κατανόηση
14. Τα έγγραφα πρέπει να είναι σαφή και απλά

Κάθε οδηγός υποδιαιρείται σε σημεία ελέγχου που χρησιμεύουν στον έλεγχο του κώδικα για το αν συμμορφώνεται στον οδηγό WCAG. Υπάρχουν 65 συνολικά σημεία ελέγχου. Καθένα από αυτά έχει μια προτεραιότητα με τιμή από 1 έως 3 και λειτουργούν παρόμοια με τα επίπεδα συμμόρφωσης εισήχθησαν στον δεύτερο οδηγό προσβασιμότητας WCAG 2.0. Τα τρία επίπεδα προτεραιότητας είναι οι εξής:

- **Προτεραιότητα 1:** Οι προγραμματιστές πρέπει να ικανοποιούν συγκεκριμένες απαιτήσεις αλλιώς μία ή περισσότερες ομάδες δεν μπορούν να έχουν πρόσβαση στο περιεχόμενο της σελίδας
- **Προτεραιότητα 2:** Οι προγραμματιστές θα ήταν καλό να ικανοποιούν αυτές τις απαιτήσεις διαφορετικά, το περιεχόμενο θα είναι δύσκολα προσβάσιμο για ορισμένες ομάδες
- **Προτεραιότητα 3:** Οι προγραμματιστές μπορούν να αντιμετωπίσουν αυτά τα σημεία ελέγχου, προκειμένου να μεγιστοποιηθεί η προσβασιμότητα των χρηστών

Η δεύτερη έκδοση των οδηγιών για την προσβασιμότητα του περιεχομένου του ιστού με την ονομασία WCAG 2.0, προτάθηκε από το W3C το 2008. Η έκδοση WCAG 2.0 αποτελείται από δώδεκα οδηγούς που χωρίζονται στις ακόλουθες τέσσερις αρχές (Sikos, 2011) (pg. 370) :

Αρχή 1: Τα στοιχεία της διεπαφής του χρήστη καθώς και δημοσιευμένες πληροφορίες πρέπει να γίνονται αντιληπτές από οποιονδήποτε.

1. Πρέπει να παρέχονται εναλλακτικές επιλογές για περιεχόμενο που δεν περιλαμβάνει κείμενο (π.χ. βίντεο, εικόνα) και να μπορεί να μετασχηματιστεί σε αυτές τις εναλλακτικές
2. Τα media στοιχεία που βασίζονται στον χρόνο πρέπει να έχουν εναλλακτικές επιλογές για να εμφανίζονται
3. Τα περιεχόμενα του ιστού πρέπει να είναι διαθέσιμα μέσω διαφόρων παρουσιάσεων, χωρίς να υπάρχει απώλεια πληροφοριών ή δομής της σελίδας
4. Το οπτικοακουστικό περιεχόμενο πρέπει να διακρίνεται εύκολα

Αρχή 2: Η διεπαφή του χρήστη πρέπει να είναι λειτουργική και εύχρηστη στην πλοήγηση

5. Όλες οι λειτουργίες θα πρέπει να είναι διαθέσιμες από το πληκτρολόγιο
6. Οι χρήστες δεν μπορούν να υποχρεωθούν να εκτελούν ενέργειες εντός κάποιων χρονικών προθεσμιών
7. Σχέδια που μπορεί να προκαλέσουν επιληπτικές κρίσεις πρέπει να αποφεύγονται
8. Πρέπει να παρέχεται καθοδήγηση και βοήθεια στους χρήστες προκειμένου να μπορούν να περιηγηθούν μέσα στην σελίδα (Ahmad Rashid, 2006)

Αρχή 3: Η διεπαφή του χρήστη πρέπει να έχει κατανοητό περιεχόμενο

9. Το κείμενο το περιεχομένου πρέπει να διαβάζεται και να κατανοείται, εύκολα
10. Η εμφάνιση και λειτουργία του περιεχομένου πρέπει να είναι προβλέψιμη
11. Πρέπει να παρέχεται βοήθεια για τους χρήστες προκειμένου να αποφύγουν, να βρουν αλλά και να διορθώσουν τα λάθη

Αρχή 4: Το ανθεκτικό περιεχόμενο μιας σελίδας που χαρακτηρίζεται από υψηλή διαλειτουργικότητα μπορεί να χρησιμοποιηθεί από οποιοδήποτε είδος πράκτορα χρήστη, συμπεριλαμβανομένων των υποστηρικτικών τεχνολογιών

12. Η συμβατότητα πρέπει να μεγιστοποιηθεί με τρέχοντες και μελλοντικούς πράκτορες χρήστη και υποστηρικτικές τεχνολογίες, συμπεριλαμβανομένων εκείνων που λειτουργούν με περιορισμένους πόρους

Αυτοί οι οδηγοί περιλαμβάνουν 61 κριτήρια επιτυχίας ενώ μπορούν να υλοποιηθούν με την χρήση μιας σειράς τεχνικών. Μερικές από αυτές πρέπει να πληρούν τα κριτήρια επιτυχίας (επαρκείς τεχνικές), ενώ άλλες είναι προαιρετικές (συμβουλευτικές τεχνικές) (Kepler, 2005) .

3.4.3 Αποτελεσματικότητα Κώδικα CSS

Τα αρχεία CSS τα οποία έχουν την ίδια επίδραση στο περιεχόμενο ενός αρχείου HTML, μπορεί να διαφέρουν σημαντικά ως προς τον τρόπο με τον οποίο εκχωρούν τιμές στις ιδιότητες. Το μήκος του κώδικα είναι ένα κριτήριο εκχώρησης τιμής, αλλά το πιο σημαντικό είναι οι εκφράσεις που χρησιμοποιούνται για να αναφερθούν στα στοιχεία του αρχείου HTML για την ανάθεση ιδιοτήτων. Μεγιστοποίηση της αποδοτικότητας του κώδικα CSS σημαίνει εφαρμογή εκχώρησης ιδιοτήτων κατά τρόπο που να ελαχιστοποιείται η προσπάθεια σύνταξης κώδικα, καθώς και να επιτυγχάνεται η διατήρηση και επαναχρησιμοποίησή του. Στις επιλογές CSS που αναφέραμε στο 2^ο κεφάλαιο διαδραματίζουν βασικό ρόλο στην πολυπλοκότητα του κώδικα καθώς και στην ποιότητα του εγγράφου HTML που διαθέτει το περιεχόμενο.

3.4.3.1 Οδηγός Εγγραφής Αποτελεσματικών CSS επιλογών

Ο David Hyatt, αρχιτέκτονας του Safari και του WebKit, (Steve, 2009) στο άρθρο του “Writing Efficient CSS for use in the Mozilla UI” δίνει ένας εύχρηστο οδηγό για την συγγραφή CSS, στον οποίο θα πρέπει να καταφεύγουν όλοι οι προγραμματιστές εάν επιθυμούν να γράψουν σωστό και αποδοτικό κώδικα. Θα

αναφέρουμε κάποιους κανόνες ειδικά για την κατασκευή επιλογών που σχετίζονται άμεσα με την απόδοση της ιστοσελίδας. Έτσι όταν γράφουμε CSS κώδικα πρέπει :

1. **Να αποφεύγεται η χρήση καθολικών κανόνων:** Εκτός τους καθολικούς επιλογείς ο Hyatt βάζει στην ίδια κατηγορία προς αποφυγήν, τους επιλογείς παιδιά, αδέρφια, απογόνους και χαρακτηριστικών. Αντί αυτών, συνιστά την χρήση επιλογών ID, class και tag
2. **Να μην καθίσταται αρμόδιος ένας ID επιλογέας:** Επειδή υπάρχει μόνο ένα στοιχείο στη σελίδα με ένα συγκεκριμένο ID, δεν υπάρχει ανάγκη να προστεθούν επιπλέον αρμοδιότητες. Για παράδειγμα, το DIV #toc δεν είναι απαραίτητο να υπάρχει και θα πρέπει να απλοποιηθεί σε # toc
3. **Να μην καθίσταται αρμόδιος ένας επιλογέας τάξης:** Αντί για επιλογείς κλάσης οι οποίοι πληρούν τις προϋποθέσεις για ειδικές ετικέτες, να επεκτείνεται το όνομα της κλάσης ώστε να καταδεικνύεται η χρήση του. Για παράδειγμα εάν έχουμε τον επιλογέα κλάσης LI.chapter θα πρέπει να αλλάξει σε .list-chapter για να γίνεται κατανοητή η χρήση του.
4. **Να κατασκευάζονται όσο το δυνατό πιο συγκεκριμένοι κανόνες:** Θα πρέπει να αποφεύγεται η κατασκευή μακροσκελών επιλογών. Συνίσταται να κατασκευάζεται μια κλάση και να προσθέτουμε στην συνέχεια τα κατάλληλα στοιχεία που επιθυμούμε
5. **Να αποφεύγονται οι επιλογείς τύπου «απόγονοι»:** Το κόστος επεξεργασίας των απογόνων είναι μεγάλο. Οι επιλογείς «παιδιά» πρέπει να προτιμούνται σε σχέση με τους «απογόνους» μιας και είναι πιο αποτελεσματικοί. Η καλύτερη επιλογή βέβαια, είναι να μην χρησιμοποιούνται ούτε επιλογείς «παιδιά»
6. **Να αποφεύγονται επιλογείς «παιδιά» τύπου tag:** Εάν έχουμε έναν επιλογέα παιδί τύπου ετικέτας, για παράδειγμα τον #toc> LI> A, θα ήταν καλύτερο να χρησιμοποιηθεί μια τάξη που σχετίζεται με καθένα από τα στοιχεία ετικέτας, όπως .toc-anchor
7. **Να εντοπίζονται όλες οι χρήσεις των επιλογών «παιδί»:** Θα πρέπει να επανεξετάσουμε την χρήση των επιλογών τύπου «παιδί» και να αντικαταστήσουμε την χρήση τους με κλάσεις
8. **Να βασιστούμε στην κληρονομικότητα:** θα πρέπει να καθορίσουμε ποιες ιδιότητες κληρονομούνται και να αποφύγουμε κανόνες στυλ που σχετίζονται με αυτές.

3.4.4 Συμβατότητα

Ήδη στο προηγούμενο κεφάλαιο, κάναμε αναφορά για την σημασία της συμβατότητας (Compatibility) ενός προγράμματος περιήγησης και των κανόνων

CSS, ενώ στο Παράρτημα Α ορίσαμε ποιες λειτουργίες και από ποιες εκδόσεις περιηγητών, υποστηρίζονται. Η αιτία των προβλημάτων συμβατότητας είναι ότι τα προγράμματα περιήγησης αναλύουν τα CSS αρχεία με διαφορετικό τρόπο. Ακόμη και προγράμματα περιήγησης των ίδιων κατασκευαστών, τα οποία όμως ανήκουν σε άλλη έκδοση όπως για παράδειγμα ο IE6.0 και IE7 (Hao Zheng, 2012) μπορεί να εμφανίσουν προβλήματα συμβατότητας. Ένας άλλος λόγος, είναι ότι η έκδοση της CSS και οι εκδόσεις των προγραμμάτων περιήγησης είναι σε κατάσταση συνεχής ενημέρωσης και όσο συμβαίνει αυτό οι δύο πλευρές αλλάζουν δυναμικά αυτό πλήττει την συμβατότητα.

3.4.4.1 Οδηγός Επίλυσης Προβλημάτων Συμβατότητας

Στην επόμενη ενότητα θα προσεγγίσουμε τα προβλήματα συμβατότητας και θα προτείνουμε τρόπους επίλυσης αυτών.

Πρόβλημα 1^ο : Η συνομιλία μεταξύ την τρέχουσας έκδοσης του περιηγητή και της επόμενης δημιουργεί πρόβλημα συμβατότητας. Για να λύσουμε το πρόβλημα αυτό προσθέτουμε στην κεφαλή HEAD του HTML αρχείου την εντολή :

```
< meta HTTPequiv = "x-ua-compatible" content = "version_of_browser" / >
```

Πρόβλημα 2^ο : Η θέση ενός αντικειμένου εμφανίζεται διαφορετικά όταν αυτό εκτελείται στον Internet Explorer και διαφορετικά στον Firefox. Αυτό συμβαίνει γιατί το προεπιλεγμένο κομμάτι **body** στον IE είναι ολόκληρο το πλαίσιο ενώ στον Firefox είναι μόνο η περιοχή όπου υπάρχει περιεχόμενο. Στον Firefox το προεπιλεγμένο body έχει ένα κενό διάστημα στο πάνω μέρος της φόρμας. Για να λύσουμε το πρόβλημα αυτό προσθέτουμε την δήλωση:

```
* { padding : 0; margin : 0; }
```

η οποία δουλεύει μόνο στον Firefox και καθιστά τα στοιχεία των σελίδων που τρέχουν στον Firefox να ξεκινάνε την διάταξη τους από την κορυφή της φόρμας

Πρόβλημα 3^ο : Πριν από την υλοποίηση της ιστοσελίδας θα πρέπει να υπάρχει το στάδιο σχεδιασμού όπου το πλάτος της σελίδας είναι από τα πρώτα χαρακτηριστικά που πρέπει να δηλωθούν. Για να λύσουμε το πρόβλημα αυτό προσθέτουμε τις εντολές:

```
body {
margin: 0 auto;
text-align: left;
width: [the width of the contents in main page];
position: relative;}
```

Η εντολή `text-align` αναφέρεται σε ένα από τα χαρακτηριστικά της ιστοσελίδας και χρησιμοποιείται για να λύσει το πρόβλημα της κεντρικής στοίχισης του περιεχομένου, στον ΙΕ. Οι εντολές `margin: 0 auto` και `width: [the width of the contents in main page]` χρησιμοποιούνται για να λύσουν το ίδιο πρόβλημα στον Firefox. Όμως, αυτό το είδος της λύσης, που λύνει το πρόβλημα προσαρμοστικότητας του κέντρου της σελίδας γεννά δύο νέα ερωτήματα i. Πως θα εμφανιστεί η γραμματοσειρά στο κέντρο και ii. Πως θα εξαλείψουμε τα λάθη που εμφανίζονται κατά την χρήση χαρακτηριστικών τύπου `float`. Το κέντρο για τη γραμματοσειρά μπορεί να αλλάξει προσθέτοντας τον κώδικα : **`text-align: left/right`** ενώ η λύση για το πρόβλημα των `float` χαρακτηριστικών παρουσιάζεται παρακάτω.

Πρόβλημα 4^ο : Όπως ήδη αναφέραμε, η χρήση `float` attributes δημιουργεί πολλά προβλήματα. Όταν μια σελίδα χρησιμοποιεί την εντολή **`text-align: center/left/right`** για να καθορίσει την θέση των στοιχείων, τα τρέχοντα στοιχεία θα πρέπει να αναφέρονται στο `body` για να ορίσουν την θέση τους, με αποτέλεσμα να καταστρέφεται η προσαρμοστικότητα του κέντρου. Λύση σε αυτό το πρόβλημα είναι ο ορισμός ενός ορίου, με τα εσωτερικά στοιχεία να έχουν ως σημείο αναφοράς το όριο αυτό. Το όριο αυτό θέτεται στον HTML κώδικα ως εξής:

```
<div id = "first" >
    <div id = "first01" ></div >
    <div id = "first02" ></div >
    <div id = "first03" ></div >
</div >
```

Όπου χρησιμοποιούμε τα `id = "first01", "first02", "first03"` για να υπολογίσουμε το ύψος και το πλάτος της διάταξης των στοιχείων χωρίς να χρησιμοποιούμε `float` ιδιότητες.

Πρόβλημα 5^ο : Όταν χρησιμοποιούμε `border` για να δούμε την διάταξη αν δεν επεξεργαστούμε τους κώδικες με συγκεκριμένο τρόπο τότε εμφανίζεται σφάλμα σε μερικά pixels στον ΙΕ. Αυτό συμβαίνει λόγω της συμβατότητας του μοντέλου «CSS κουτί». Για να λύσουμε το πρόβλημα αυτό, προσθέτουμε στον κώδικα την κατάσταση:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml/DTD/xhtmltransitional.dtd">
```

Με την προσθήκη της αναγνώρισης DTD, η σελίδα θα ρυθμίζεται βάση του πρωτότυπου μοντέλου.

Πρόβλημα 6^ο : Όταν το αντικείμενο στο `box model` έχει την κατάσταση `float`, η απόσταση από το αριστερό ή το δεξί περιθώριο `box` αντικείμενο διπλασιάζεται. Αυτό αποτελεί πρόβλημα στην έκδοση ΙΕ6.0. Προσθέτοντας την εντολή :

`display: inline;`

το πρόβλημα λύνεται.

Πρόβλημα 7^ο : Επειδή το μοντέλο BOX ορίζεται διαφορετικά στον Firefox και τον IE τα αντικείμενα εμφανίζονται με μια διαφορά 2px. Για να λύσουμε το πρόβλημα αυτό προσθέτουμε :

```
div {
    margin: 30px! Important;
    margin: 28px;
}
```

3.4.5 Αφαιρετικότητα

Οι παρουσιάσεις προδιαγραφών με υψηλό επίπεδο αφαιρετικότητας βασίζονται σε δομικά στοιχεία καθορισμένης σημασιολογίας, πέρα από το πλαίσιο του εγγράφου και δεν απαιτούν πρόσθετη δομή στο περιεχόμενο του εγγράφου. Έτσι, μια πιο αφηρημένη παρουσίαση προδιαγραφών κάνει το περιεχόμενο του εγγράφου εύκολα επαναχρησιμοποιήσιμο. Όσον αφορά την αφαιρετικότητα σε επίπεδο γλώσσας CSS, αυτή εκφράζει το πόσο έντονα ανατίθενται ιδιότητες παρουσίασης σε στοιχεία HTML (Keller Matthias, 2009).

Ένα υψηλό επίπεδο αφαίρεσης σημαίνει πως έχουμε έναν κώδικα που περιέχει πιο γενικές δηλώσεις και κατ' επέκταση εμφανίζει μεγαλύτερη ανοχή στις αλλαγές που γίνονται στο περιεχόμενο του εγγράφου. Στην συνέχεια παρουσιάζεται ένας οδηγός για την κατασκευή κώδικα CSS με υψηλό επίπεδο αφαιρετικότητας.

3.4.5.1 Οδηγός Επίτευξης Αφαιρετικότητας

Βήμα 1^ο : Κατασκευή Επιλογέων CSS

Στην γλώσσα CSS (Keller Matthias, 2010) όπως έχουμε ήδη αναφέρει, χρησιμοποιούνται σε μεγάλο βαθμό απλοί επιλογείς, επιλογείς στοιχείων και τάξη καθώς και επιλογείς τύπου id. Ένας επιλογέας στοιχείο επιλέγει στοιχεία HTML με συγκεκριμένο όνομα, ενώ ένας επιλογέας κλάσης ή id επιλέγει στοιχεία HTML βάσει της κλάσης και του id χαρακτηριστικού, αντίστοιχα. Πιο περίπλοκοι επιλογείς μπορούν να προκύψουν συνδυάζοντας πολλούς απλούς επιλογείς. Όλοι οι πιθανοί συνδυασμοί επιλογέων μπορούν να αναπαρασταθούν σε ένα δένδρο (document tree). Κατασκευάζουμε το δένδρο ως εξής:

1. Το πρώτο τμήμα του απλού επιλογέα περιορίζει το πεδίο εφαρμογής σε όλους τους απογόνους που περιέχουν την κλάση την οποία εξετάζουμε. Το δεύτερο τμήμα του επιλογέα επιλέγει τους απογόνους

- που οριστήκαν προηγουμένως. Οι επιλογείς - απόγονοι κατασκευάζονται όταν υπάρχουν πάνω από δύο κοινοί απλοί επιλογείς
2. Οι δεύτεροι απλοί επιλογείς περιορίζουν το πεδίο εφαρμογής, σε όλα τα στοιχεία απόγονος των κόμβων εντός του πεδίου εφαρμογής που έχει οριστεί από το πεδίο επιλογής των πρώτων απλών επιλογέων
 3. Τα βήματα 1 και 2 επαναλαμβάνονται μέχρι φτάσουμε στον τελευταίο απλό επιλογέα ο οποίος θα ορίζει ποιο από τα στοιχεία πρέπει να μορφοποιηθεί ενώ ο προηγούμενος απλός επιλογέας, ορίζει την θέση των υπολοίπων μέσα στο δένδρο.

Κατασκευάζοντας το δένδρο του αρχείου CSS με τον τρόπο που περιγράψαμε παίρνουμε μετρικές αφαιρετικότητας, οι οποίες περιλαμβάνουν το μέγεθος του πεδίου εφαρμογής ενώ ορίζουν την καθολικότητα του τελευταίου στοιχείου σε έναν πιθανό συνδυασμό επιλογέων.

Βήμα 2^ο : Ορισμός Καθολικότητας U

Όταν έχουμε έναν συνδυασμό επιλογέων, ο τελευταίος απλός επιλογέας ορίζει ποια στοιχεία έχουν επιλεγεί. Εάν επισκεπτόμαστε έναν – έναν τους επιλογείς έχουμε μέγιστο επίπεδο αφαιρετικότητας μόνο όταν επιλέγονται στοιχεία από την δομή του περιεχομένου των οποίων η σημασιολογία ορίζεται πέρα του περιεχομένου του εγγράφου. Με βάση αυτές τις εκτιμήσεις μια μετρική για την αφαιρετικότητα των απομονωμένων απλών επιλογέων καθώς και των απλών επιλογέων που είναι το τελευταίο μέρος ενός συνδυασμού επιλογέων μπορεί να οριστεί βάση του τύπου:

$$\text{Universality } U = \frac{\# \text{ element selectors in SSel}}{\# \text{ selectors}}$$

Όπου SSel είναι το σύνολο των απομονωμένων απλών επιλογέων καθώς και ο τελευταίος απλός επιλογέας από κάθε συνδυασμό επιλογέων.

Βήμα 3^ο : Ορισμός Μέσου Πεδίου Επιλογέα S

Το μέσο πεδίο επιλογέα S αντιπροσωπεύει όλους τους περιορισμούς των συνδυασμών επιλογέων, στο δένδρο που κατασκευάσαμε στο βήμα 1. Με το μέγεθος του πεδίου παρατηρούμε πως ακόμα και αν υπάρχει μόνο ένας επιλογέας - στοιχείο που χρησιμοποιείται σε συνδυασμό με άλλους επιλογείς, η έκφραση μπορεί να αντιπροσωπεύει εμφωλευμένες δομές που είναι δύσκολο να ερμηνευθούν έξω από

αυτό το πλαίσιο. Για να μετρήσουμε το μέσο μέγεθος πεδίου χρησιμοποιούμε τον τύπο:

$$\text{Average Scope} = \frac{\sum_{i=1}^m \# \text{ of elements in scope of selector } i}{m * n}$$

Όπου m ο αριθμός των επιλογών και n ο αριθμός των στοιχείων στο δένδρο του εγγράφου που κατασκευάσαμε. Όταν έχουμε απλούς επιλογείς που δεν αποτελούν μέρος ενός συνδυασμού επιλογών και είναι στοιχεία ή επιλογείς κλάσης τότε ως πεδίο εφαρμογής θεωρείται όλο το δέντρο που κατασκευάσαμε. Για επιλογείς id μόνο ένα στοιχείο μετράει ως πεδίο εφαρμογής, το οποίο είναι το ίδιο το στοιχείο id .

Βήμα 4^ο : Υπολογισμός Παράγοντα Αφαιρετικότητας

Για να επιτύχουμε ένα υψηλό επίπεδο αφαιρετικότητα είναι απαραίτητο να έχουμε ταυτόχρονα ένα μεγάλο πεδίο S καθώς και έναν υψηλό συντελεστή καθολικότητα U . Εάν έχουμε χαμηλό συντελεστή καθολικότητας U σε συνδυασμό με μέγιστο πεδίο S τότε έχουμε σαν αποτελέσματα με την διαδικασία κωδικοποίησης CSS να εισάγονται νέες κλάσεις χαρακτηριστικών προκειμένου να επιτευχθεί η σύνδεση των πληροφοριών παρουσίασης και έτσι δεν έχουμε την κατασκευή συνδυασμού επιλογών.

Από την άλλη, εάν έχουμε ένα υψηλό παράγοντα καθολικότητα U σε συνδυασμό με ένα μικρό πεδίο S τότε καταλαβαίνουμε πως χρησιμοποιούμε έναν πολύ σύνθετο συνδυασμό επιλογών που αντανακλά την βαθιά εμφωλευμένη δομή του περιεχομένου του εγγράφου μας. Οι συνδυασμοί που αναφέραμε προηγουμένως έχουν χαμηλό δείκτη αφαιρετικότητας. Για να επιτύχουμε το μέγιστο επίπεδο αφαιρετικότητας υπολογίζουμε :

$$\text{Abstractness factor} = \min (S , U)$$

Ο παραπάνω δείκτης είναι ένα κριτήριο για να προσδιορίσουμε τη σχέση μεταξύ των προδιαγραφών παρουσίαση και του περιεχομένου του εγγράφου, καθορίζει λοιπόν κατά πόσο το περιεχόμενο του εγγράφου διαχωρίζεται από την παρουσίασή του.

3.4.6 Πολυπλοκότητα

Η πολυπλοκότητα (Complexity) της γλώσσας CSS (Adewumi Adewole, 2012) αναφέρεται στο πόσο μπορεί να γίνει κατανοητή και εύκολα διαχειρίσιμη. Όλοι οι παράγοντες που κάνουν την χρήση της CSS δύσκολη είναι υπεύθυνοι για την αύξηση της πολυπλοκότητας του κώδικα. Τέτοιοι παράγοντες είναι το μέγεθος, η ποικιλία στην χρήση κανόνων, η πολλαπλή χρήση κανόνων κ.α. Στην συνέχεια θα αναφέρουμε έναν οδηγό, βάση του οποίου μπορεί να γίνει μέτρηση της πολυπλοκότητας.

3.4.6.1 Οδηγός Μέτρησης Πολυπλοκότητας

- 1) **Μέγεθος Κανόνα (Rule Length - RL)**: Δεδομένου ότι το μέγεθος είναι ένα σημαντικό μέτρο προτείνουμε έναν κανόνα μέτρησης μήκους, που είναι παρόμοιος με τον αριθμό των modules στον δομημένο προγραμματισμό. Ένα αρχείο CSS αποτελείται από μια λίστα κανόνων. Η μετρική RL υπολογίζει τον αριθμό των γραμμών των κανόνων της CSS. Αυτή η μετρική αγνοεί τα κενά και τις γραμμές που περιέχουν σχόλια και αυτό γιατί η ίδια η CSS δεν τα εκτελεί. Η RL υπολογίζεται βάσει του τύπου :

$$RL = \sum \text{rule statements in a CSS file}$$

Όπου rule statement είναι ένα από τα ακόλουθα:

- Επιλογείς μαζί με άνοιγμα αγκύλης ({), για παράδειγμα
body {
- Ιδιότητες ενός επιλογέα που να τελειώνει με ελληνικό ερωτηματικό (;), για παράδειγμα color: #FFFFFF;
- Αγκύλη που κλίνει έναν μπλοκ κανόνα (}

- 2) **Αριθμός Μπλοκ των Κανόνων (Number of Rule Blocks - NORB)**: Ένα μπλοκ με κανόνα αναφέρεται σε έναν επιλογέα και στα χαρακτηριστικά αυτού και ορίζεται με την παρακάτω σύνταξη

```
/* Syntax of a rule block */
selector [, selector2, ...] [:pseudo-RLass] {
property: value;
[property2: value2;
...]}
```

Ένα αρχείο CSS θα πρέπει να περιέχει τουλάχιστον ένα μπλοκ με έναν κανόνα.

- 3) **Μέτρηση της εντροπίας (Entropy Metric - E)**: Η λέξη εντροπία υιοθετήθηκε από την θεωρία πληροφορίας και ορίστηκε ως δείκτης μέτρησης της ποικιλομορφίας. Στην κατασκευή λογισμικού, χρησιμοποιείται ως μέτρο εκτίμησης της πολυπλοκότητας των κανόνων. Τα τελευταία χρόνια έχει εφαρμοστεί ως μέτρο αξιολόγησης της δομικής πολυπλοκότητας των εγγράφων XML, τα οποία συντάσσονται βάσει της γλώσσας Document Type Definition (DTD) που έχει ορίσει το W3C. Η εντροπία υπολογίζεται από τον τύπο

$$E = \langle S, F, P \rangle$$

Όπου E είναι ένα πείραμα με S το σύνολο των στοιχειωδών γεγονότων, F είναι ένα πεδίο Borel πάνω στο S και P είναι μια πιθανοτική συνάρτηση εκτίμησης των πραγματικών τιμών για τα γεγονότα του F. Στην συνέχεια, και για έναν περιορισμένο αριθμό γεγονότων C_1, C_2, \dots, C_n η εντροπία του δοθέντος πειράματος E δίνεται από τον τύπο:

$$H = - \sum P(C_t)P(C_t) \text{ όπου } t = 1 \dots n$$

Με βάση τον παραπάνω ορισμό, η εντροπία ενός δεδομένου έγγραφο CSS με n διακριτές κλάσεις στοιχείων μπορεί να υπολογιστεί με τις σχετικές συχνότητες που προκύπτουν από εκτιμήσεις των πιθανοτήτων $P(C_i)$, $i = 1, 2, \dots, n$. Με τον όρο διακριτές κλάσεις στοιχείων αναφερόμαστε στα στοιχεία που έχουν τις ίδιες δομικές πολυπλοκότητες και ομαδοποιούνται στην ίδια κλάση που ονομάζεται *ισοδύναμη κλάση* (C).

- 4) **Αριθμός Εκτεταμένων Μπλόκ Κανόνων (Number of Extended Rule Blocks - NERB)**: Αυτή η μετρική υπολογίζει τον αριθμό των μπλοκ που περιέχουν εκτεταμένους κανόνες μέσα σε ένα CSS αρχείο. Υπολογίζεται από τον τύπο:

$$NERB = \sum \text{extended rule block}(i), \text{ όπου } i = 1 \dots n$$

- 5) **Αριθμός Χαρακτηριστικών ανά Μπλοκ Κανόνα (Number of Attributes Defined per Rule Block - NADRB)**: Αυτή η μετρική προσδιορίζει το μέσο αριθμό των χαρακτηριστικών που ορίζονται στο μπλοκ κανόνα ενός αρχείου CSS. Μπορεί να υπολογιστεί ως εξής:

$$NADRB = \frac{\text{Total no. of attributes in all rule blocks}}{\text{Total no. of rule blocks}}$$

- 6) **Αριθμός Συνεκτικών Μπλόκ Κανόνων (Number of Cohesive Rule Blocks - NCRB):** Η συνοχή μπορεί να περιγραφεί ως το «ενιαίο πνεύμα» ενός στοιχείου. Στην περίπτωση της γλώσσας CSS, αναφέρεται στον αριθμό των μπλόκ κανόνων που διαθέτουν ένα μοναδικό χαρακτηριστικό. Η μετρική αυτή καταγράφει όλα τα μπλόκ κανόνων τα οποία έχουν μόνο ένα χαρακτηριστικό και υπολογίζεται από τον τύπο :

$$NCRB = \sum rule\ block\ (i)\ possessing\ only\ one\ attribute, \acute{o}που\ i = 1...n$$

3.5 Σύνοψη

Στο κεφάλαιο αυτό, κατανοήσαμε πόσο σημαντικό είναι οι ιστοσελίδες να πληρούν προϋποθέσεις που έχουν τεθεί από τον οργανισμό W3C με αποτέλεσμα να εμφανίζουν πολλά πλεονεκτήματα όπως να επιτυγχάνουν γρηγορότερο "κατέβασμα" στην συσκευή που επισκέπτεται την ιστοσελίδα, να διαθέτουν κώδικα προσβάσιμο σε περισσότερους ανθρώπους και συσκευές ανάγνωσης, να δίνουν την δυνατότητα παραμετροποίηση μιας σελίδας από τους χρήστες αλλά και να γίνονται πιο ανταγωνιστικές. Στην συνέχεια, αναλύθηκε ένα σύνολο μετρικών χρήσιμων για να μεγιστοποιηθεί η εμπειρία του χρήστη. Τέλος, για κάθε μία από τις μετρικές διατυπώθηκαν οδηγίες που θα έπρεπε να ακολουθήσει κάθε προγραμματιστής όταν κατασκευάζει μια ιστοσελίδα, προκειμένου να εξασφαλίσει όσο το δυνατόν πιο ποιοτικά Web Sites.

Κεφάλαιο 4^ο

4.1 Εισαγωγή

Όπως ήδη έχουμε αναφέρει στα προηγούμενα κεφάλαια η CSS είναι μια γλώσσα για την παρουσίαση των στοιχείων HTML συμπεριλαμβανομένης της θέσης, της διάταξης, του χρώματος και της γραμματοσειράς των στοιχείων. Η κύρια κινητήριος δύναμη πίσω από την υιοθέτηση της CSS, υπήρξε ο διαχωρισμός της δομής από την παρουσίαση του περιεχομένου. Παρά το γεγονός ότι αυτός ο διαχωρισμός κατάφερε να βοηθήσει την εξέλιξη των εφαρμογών του ιστού όσον αφορά τη δομή και το περιεχόμενο, ο ίδιος ο κώδικας CSS δεν είναι εύκολα διαχειρίσιμος.

Το να προγραμματίζει κάποιος σε γλώσσα CSS δεν είναι εύκολο. Απαιτεί έναν συνδυασμό αλληλεπίδρασης του ανθρώπου με τον υπολογιστή, σχεδιασμό γραφικών καθώς και εξειδικευμένες γνώσεις προγραμματισμού. Επιπροσθέτως, η γλώσσα έχει ένα σύνολο χαρακτηριστικών όπως η κληρονομικότητα, η δημιουργία ειδικών επιλογέων κ.α που καθιστούν τον τρόπο που αυτά εφαρμόζονται σε στοιχεία Document Object Model (DOM) ακόμα πιο δύσκολο. Ενώ ο προγραμματιστής συνεχώς αντιμετωπίζει ερωτήματα όπως «Χρησιμοποιεί η εφαρμογή μου τους σωστούς κανόνες CSS;», «Μήπως η έκδοση CSS που έχω είναι ξεπερασμένη;», «Τι θα συμβεί αν αλλάξω έναν κανόνα CS, θα προκαλέσει κατάρρευση της ιστοσελίδας μου;», «Είναι κατάλληλος ο επιλογέας που έχω ή θα παρακαμφθεί κατά την εκτέλεση του αρχείου;» και πολλά άλλα. Λύση σε αυτά τα προβλήματα προσπάθησαν να δώσουν οι προγραμματιστές, με την ανάπτυξη κατάλληλων εργαλείων τα οποία καταφέρνουν να κάνουν την δουλειά τους πιο εύκολη, αποτελεσματική ενώ παράλληλα ελαχιστοποιούν τα ανθρώπινα λάθη που προκύπτουν κατά την συγγραφή του κώδικα.

Στην επόμενη ενότητα θα αναλύσουμε χαρακτηριστικά παραδείγματα εργαλείων ενώ παράλληλα θα γίνει σύνδεση αυτών με κάποιες από τις μετρικές που αναφέραμε στο προηγούμενο κεφάλαιο και μας βοηθάνε να έχουμε ποιοτικότερες ιστοσελίδες και εφαρμογές στον ιστό. Τέλος, θα χρησιμοποιήσουμε τον δείκτη Alexa ο οποίος καταδεικνύει την επισκεψιμότητα ενός ιστοχώρου και κατ' επέκταση πόσο δημοφιλής είναι αυτός και στην συνέχεια θα εξετάσουμε πως ανταποκρίνονται αυτοί στα εργαλεία που αναλύουμε. Θα επιλέξουμε ιστοσελίδες που έχουν κατασκευαστεί αποκλειστικά στην Ελλάδα και δεν αποτελούν επίσημες μεταφράσεις στα ελληνικά, όπως Google.gr, Facebook.com/gr κ.ο.κ

4.2 Εργαλεία

Οι προγραμματιστές χρησιμοποιούν βασικά δύο είδη εργαλείων για να εντοπίσουν λάθη σε αρχεία CSS τους validators (επικυρωτές) και τους debuggers (διορθωτές σφαλμάτων).

- **Validators:** εντοπίζουν ζητήματα συντακτικής φύσης. Ελέγχουν με στατικό τρόπο, αν ένα αρχείο CSS ακολουθεί αυστηρά την CSS γραμματική. Ένα χαρακτηριστικό παράδειγμα τέτοιων εργαλείων είναι ο W3C CSS validator. Τα εργαλεία αυτά είναι πολύ χρήσιμα, αλλά δεν έχουν την δυνατότητα να εντοπίζουν τους κανόνες που δεν συμπεριφέρονται όπως αναμένεται.
- **Debuggers:** είναι δυναμικά και όχι στατικά εργαλεία όπως οι validators και δρουν σε συνδυασμό με μια μηχανή μορφοποίησης που εκτελεί αρχεία CSS και τα εφαρμόζει στο περιεχόμενο διάφορων ιστοσελίδων εμφανίζοντας το αποτέλεσμα που προκαλούν. Επιτρέπουν στον χρήστη, με λίγα λόγια, να δει πως θα φαίνεται η διαμόρφωση που έχει κατασκευάσει σε ένα σύνολο αρχείων που χρησιμοποιούνται για τεστ. Όλοι οι μοντέρνοι περιηγητές έχουν debuggers όπως τον Firebug (Firefox), τον Developer Toolbar (Internet Explorer), τον Dragonfly (Opera) και τον Web Inspector (Safari).

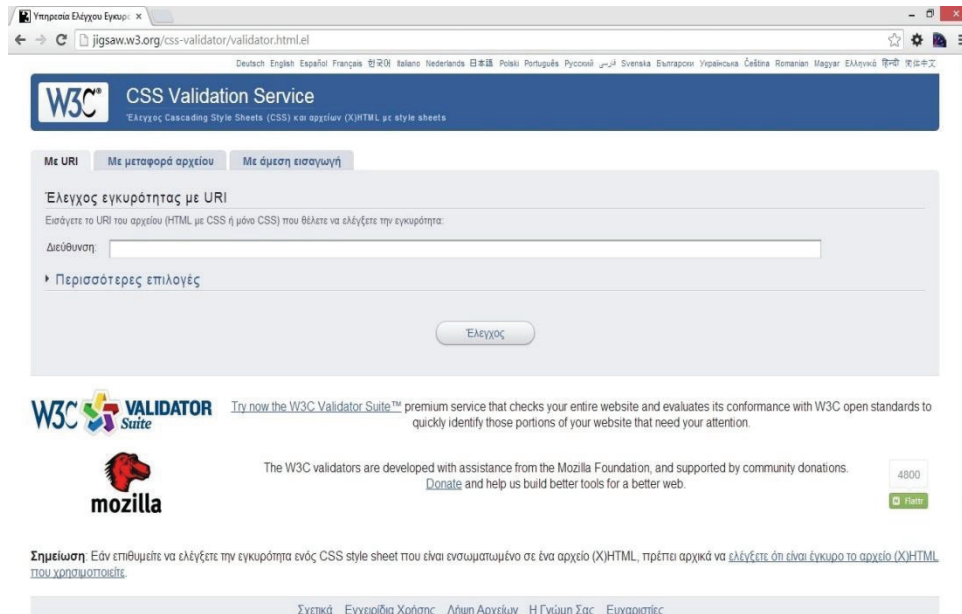
4.3 Επικυρωτής CSS W3C

Ο επικυρωτής CSS θα πρέπει να εφαρμόζεται ανάλογα με την έκδοση CSS που χρησιμοποιείται. Όταν έχουμε ένα αρχείο σε επίπεδο CSS 1 τότε συνήθως είναι συμβατό και στην έκδοση CSS 2 αλλά ωστόσο υπάρχουν κάποιοι κανόνες που γράφονται με ελαφρώς διαφορετικότερο τρόπο. Οι περισσότερες νέες ιδιότητες που έχουν προστεθεί στην 3^η έκδοση δεν είναι συμβατές στις προηγούμενες εκδόσεις.

Τα έγκυρα CSS φύλλα στυλ πρέπει να συμμορφώνονται με τους κανόνες γραμματικής της αντίστοιχης έκδοσης και πρέπει να περιέχουν μόνο κανόνες και ιδιότητες που ορίζονται βάσει των εν λόγω προδιαγραφών. Ο οργανισμός W3C παρέχει τον επικυρωτή CSS validator ο οποίος παίρνει σαν όρισμα ένα αρχείο CSS, το αναλύει και επιστρέφει αποτελέσματα για το πόσο ο κώδικας ακολουθεί τους κανονισμούς που έχουν τεθεί. Εκτός όμως από τον επικυρωτή του W3C υπάρχουν πολλοί ακόμα επικυρωτές που προσφέρουν ακριβώς την ίδια λειτουργία. Μια ιστοσελίδα που έχει επικυρωθεί και δεν περιέχει σφάλματα, **αυξάνει σε μεγάλο βαθμό την απόδοση** της ιστοσελίδας και άρα μπορεί να θεωρηθεί θεμελιώδης εργαλείο για την κατασκευή ποιοτικών σελίδων.

4.3.1 Λειτουργία CSS Validator

Οποιοσδήποτε χρήστης μπορεί να έχει δωρεάν πρόσβαση στον επικυρωτή CSS αρκεί να επισκεφθεί την διεύθυνση <http://jigsaw.w3.org/css-validator/> ενώ μπορεί να επιλέξει την γλώσσα που επιθυμεί και να εμφανίζεται αντίστοιχα η ιστοσελίδα. Εμείς διαλέγουμε τα ελληνικά. Ο επικυρωτής έχει την εξής μορφή:

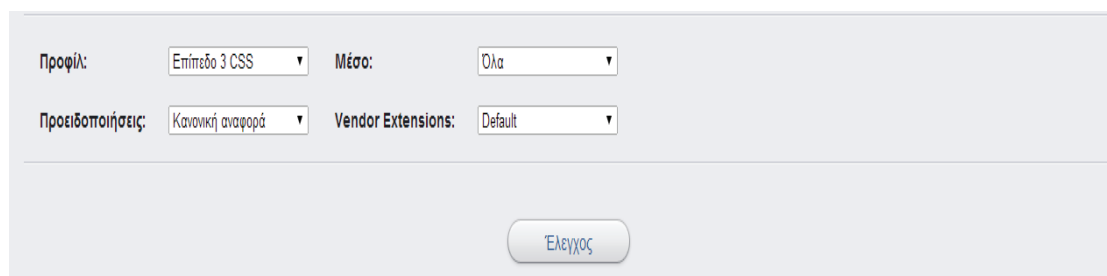


Εικόνα 1

Ενώ μας δίνει τρεις εναλλακτικές επιλογές για να ελέγξουμε τον κώδικα CSS

- Με την εισαγωγή URL που περιέχει τον κώδικα CSS
- Με την εισαγωγή αρχείου κάνοντας το upload από τον υπολογιστή
- Με άμεση εισαγωγή κάνοντας αντιγραφή και επικόλληση τον κώδικα CSS

Είναι πολύ σημαντικό να αναφέρουμε πως η επικύρωση μπορεί να γίνει σε μια από τις τρεις εκδόσεις της CSS.



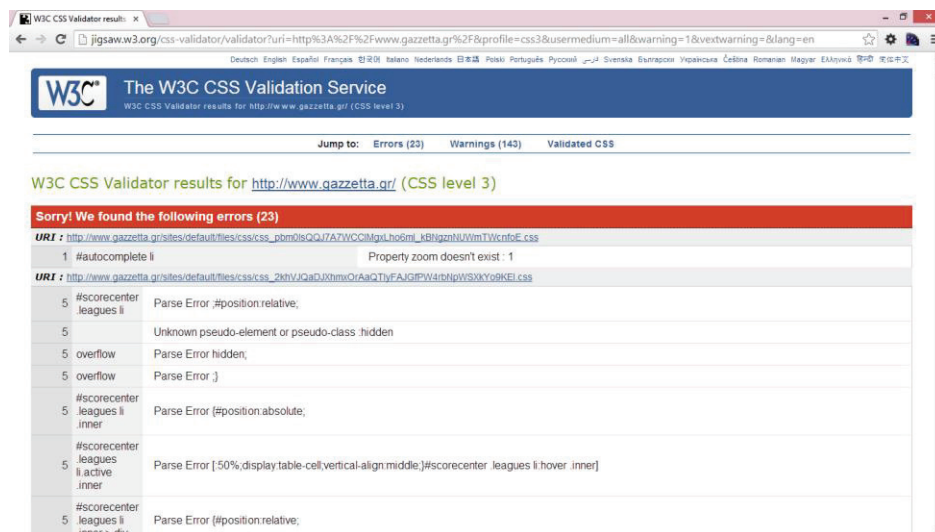
Εικόνα 2

4.3.2 Μελέτη CSS Validator

Επιλέγουμε ένα σύνολο ιστοσελίδων που είναι αρκετά δημοφιλή στην Ελλάδα και θα εξετάσουμε πως αντιδρούν στο επικυρωτή.

1. Gazzetta.gr

Η ιστοσελίδα Gazzeta.gr περιλαμβάνει ένα αθλητικό magazine όπου φίλαθλοι διάφορων σπορ μπορούν να ενημερωθούν για την πορεία της ομάδας τους, για κοινωνικά πολιτιστικά θέματα αλλά και για άλλα, ποικίλου περιεχομένου. Εισάγοντας την διεύθυνση <http://www.gazzetta.gr/> μεταβαίνουμε στην ιστοσελίδα, ενώ αν την εξετάσουμε με τον επικυρωτή επιστρέφει τα εξής αποτελέσματα:

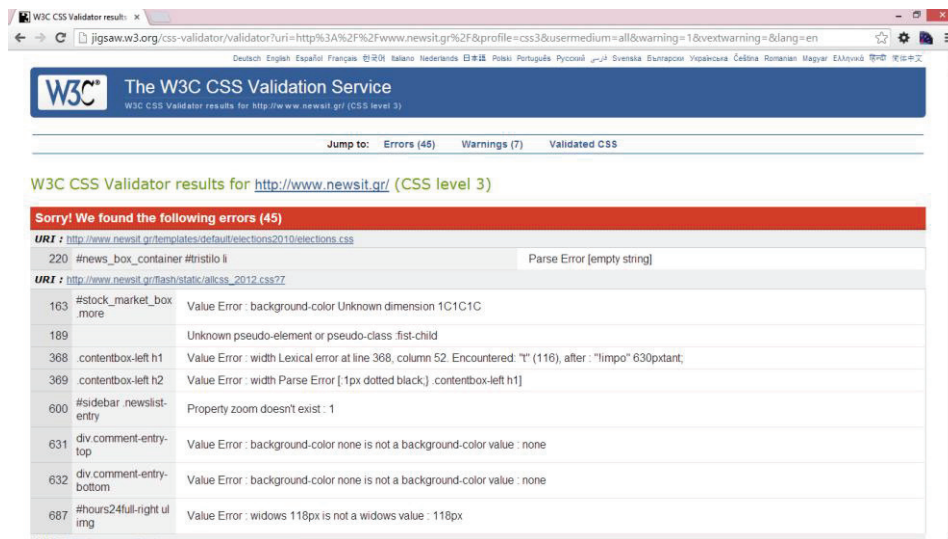


Εικόνα 3

Δηλαδή εμφανίζονται 23 σφάλματα και δεν περνάει την επικύρωση για το 3^ο επίπεδο CSS. Βέβαια αυτό δεν καθιστά προβληματικό το site, απλά μειώνει την απόδοσή του καθώς και τον χρόνο που κάνει να φορτώσει το περιεχόμενο, που είναι όμως σε επιτρεπτά χρονικά όρια.

2. Newsit.gr

Η ιστοσελίδα Newsit.gr είναι αναμενόμενο να είναι πολύ δημοφιλής μιας και περιέχει ειδήσεις και νέα από την Ελλάδα και τον κόσμο. Ο χρήστης μπορεί να μεταβεί σε αναρτήσεις βάσει των προσωπικών προτιμήσεων, ενώ μπορεί να επιλέξει από διάφορες κατηγορίες όπως πολιτική, οικονομία, αθλητικά, τεχνολογία, υγεία, lifestyle κ.ο.κ. Ο χρήστης μπορεί να μεταβεί πληκτρολογώντας την διεύθυνση <http://www.newsit.gr/> ενώ τα αποτελέσματα που επιστρέφονται μετά την επικύρωση της ιστοσελίδας έχουν ως εξής :

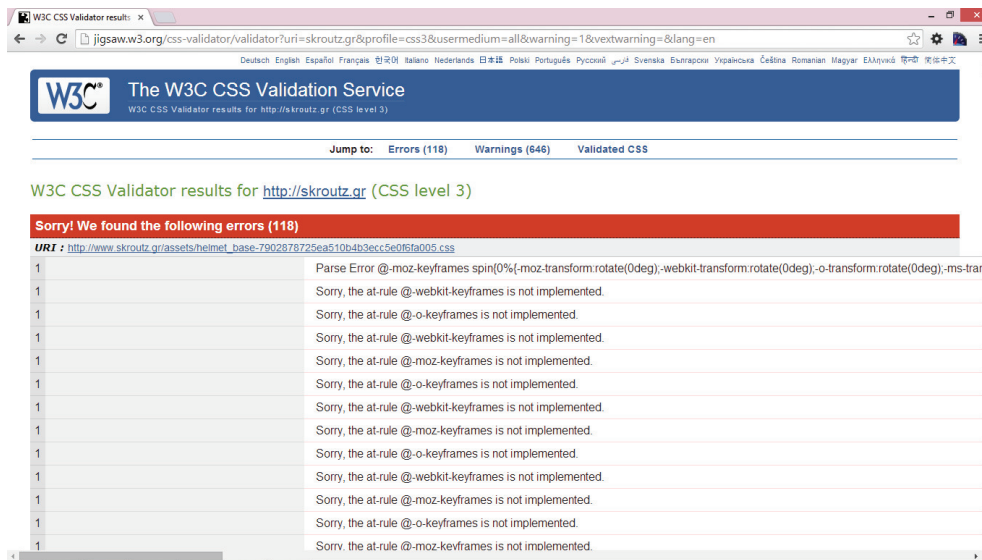


Εικόνα 4

Τα σφάλματα που παρουσιάζονται είναι πολύ περισσότερα από την προηγούμενη σελίδα και φτάνουν στα 45. Παρατηρούμε πως η σελίδα φορτώνει πιο αργά το περιεχόμενό της από την προηγούμενη και αυτό αφαιρεί της αποτελεσματικότητας.

3. Skrouz.gr

Η συγκεκριμένη ιστοσελίδα αποτελεί έναν οδηγό τιμών για διάφορες κατηγορίες προϊόντων. Ο επισκέπτης μπορεί να συγκρίνει τιμές και στην συνέχεια να μεταβεί στο ηλεκτρονικό κατάστημα που επιθυμεί ώστε να παραγγείλει προϊόντα. Η διεύθυνση είναι η <http://www.skrouz.gr/> και βλέπουμε πως εμφανίζει αρκετά λάθη με πιο συχνό το **Sorry, the at-rule @-“rule_name” is not implemented.**

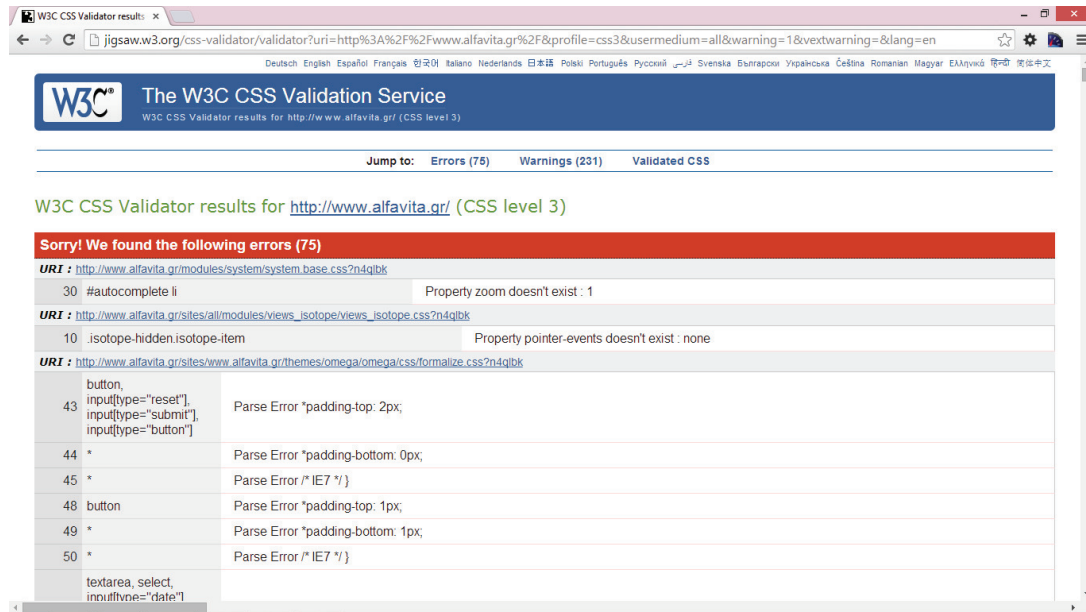


Εικόνα 5

Αξίζει σε αυτό το σημείο να αναφέρουμε πως οι developers και οι σχεδιαστές ιστοσελίδων προσπαθούν συνεχώς να βρουν νέους τρόπους για να γράφουν κώδικα CSS ο οποίος θα κάνει πιο γρήγορα download στο μηχάνημα από το οποίο επισκέπτονται την σελίδα, να αυξήσουν την προσβασιμότητα καθώς και να έχουν πιο εύκολα διαχειρίσιμες ιστοσελίδες. Έτσι αναπτύσσουν CSS hacks κομμάτια κώδικα τα οποία όμως δεν αναγνωρίζονται από όλα τα προγράμματα περιήγησης γι' αυτό και το αποτέλεσμα από το validation επιστρέφει 118 λάθη. Ο αναγνώστης με μια αναζήτηση στον ιστό μπορεί να βρει αρκετές τέτοιες μικρές παρεμβάσεις των προγραμματιστών και να τις εντάξει στην σελίδα του, γνωρίζοντας βέβαια ότι δεν θα επικυρωθεί.

4. Alfavita.gr

Η ιστοσελίδα <http://www.alfavita.gr/> είναι μια πύλη και περιέχει πολλές πληροφορίες που σχετίζονται με την εκπαίδευση από την πρώτη μέχρι και την τελευταία βαθμίδα του συστήματος. Επιπλέον συγκεντρώνει πληροφορίες για αγγελίες εργασίας, για την νομοθεσία, την κοινωνία κ.α.



W3C CSS Validator results for <http://www.alfavita.gr/> (CSS level 3)

Jump to: [Errors \(75\)](#) [Warnings \(231\)](#) [Validated CSS](#)

Sorry! We found the following errors (75)

URI	Line	Column	Error
http://www.alfavita.gr/modules/system/system.base.css?n4qibk	30		#autocomplete li Property zoom doesn't exist : 1
http://www.alfavita.gr/sites/all/modules/views_isotope/views_isotope.css?n4qibk	10		isotope-hidden.isotope-item Property pointer-events doesn't exist : none
http://www.alfavita.gr/sites/www.alfavita.gr/themes/omega/omega/css/formalize.css?n4qibk	43		button, input[type="reset"], input[type="submit"], input[type="button"] Parse Error "padding-top: 2px;
	44	*	Parse Error "padding-bottom: 0px;
	45	*	Parse Error "IE7 ?" }
	48		button Parse Error "padding-top: 1px;
	49	*	Parse Error "padding-bottom: 1px;
	50	*	Parse Error "IE7 ?" }
			textarea, select, input[type="date"]

Εικόνα 6

Και εδώ εμφανίζονται αρκετά λάθη, στο σύνολο 75, που αφορούν κυρίως παραλήψεις σε κανόνες, και κακή σύνταξη αυτών. Παρ' όλο όμως που δεν περνάει την επικύρωση, το συγκεκριμένο site παρουσιάζει μια ανοδική πορεία στον αριθμό των ατόμων που το επισκέπτονται καθημερινά.

5. Gsis.gr

Η Γενική Γραμματεία Πληροφοριακών Συστημάτων έχει κατασκευαστεί για λογαριασμό του υπουργείου οικονομικών και διευκολύνει τους πολίτες να εκτελέσουν σημαντικές υπηρεσίες, να ενημερωθούν για τα φορολογικά δεδομένα τους, για διαγωνισμούς του δημοσίου κ.ο.κ.

The W3C CSS Validation Service

W3C CSS Validator results for http://www.qsis.gr/gsis_site/ (CSS level 3)

Jump to: [Errors \(1\)](#) [Warnings \(7\)](#) [Validated CSS](#)

W3C CSS Validator results for http://www.qsis.gr/gsis_site/ (CSS level 3)

Sorry! We found the following errors (1)

URI : http://www.qsis.gr/gsis_site/

78	Property valign doesn't exist : top
----	-------------------------------------

↑ TOP

W3C VALIDATOR Suite Try now the **W3C Validator Suite™** premium service that checks your entire website and evaluates its conformance with W3C open standards to quickly identify those portions of your website that need your attention.

The W3C validators are developed with assistance from the Mozilla Foundation, and supported by community donations. [Donate](#) and help us build better tools for a better web.

4834 [Flatlr](#)

Warnings (7)

URI : <http://www.qsis.gr/gsis/export/system/modules/gr.qsis.site/resources/css/style3.css>

312	box_schema2 h4	Same color for background-color and color
1158	hr	Same color for background-color and color
1159	hr.space	Same color for background-color and color

Εικόνα 7

Οι επίσημες κυβερνητικές ιστοσελίδες είναι υποχρεωμένες να κατασκευάζονται ακολουθώντας αυστηρά τους κανόνες που έχουν θεσπιστεί από την W3C και αυτό γιατί πρέπει να τηρούν όλες τις προϋποθέσεις ώστε να όλοι οι πολίτες να έχουν την δυνατότητα να τις επισκέπτονται. Αυτό φαίνεται άλλωστε και από τα αποτελέσματα του validation αφού επιστρέφεται μόνο ένα μοναδικό λάθος .

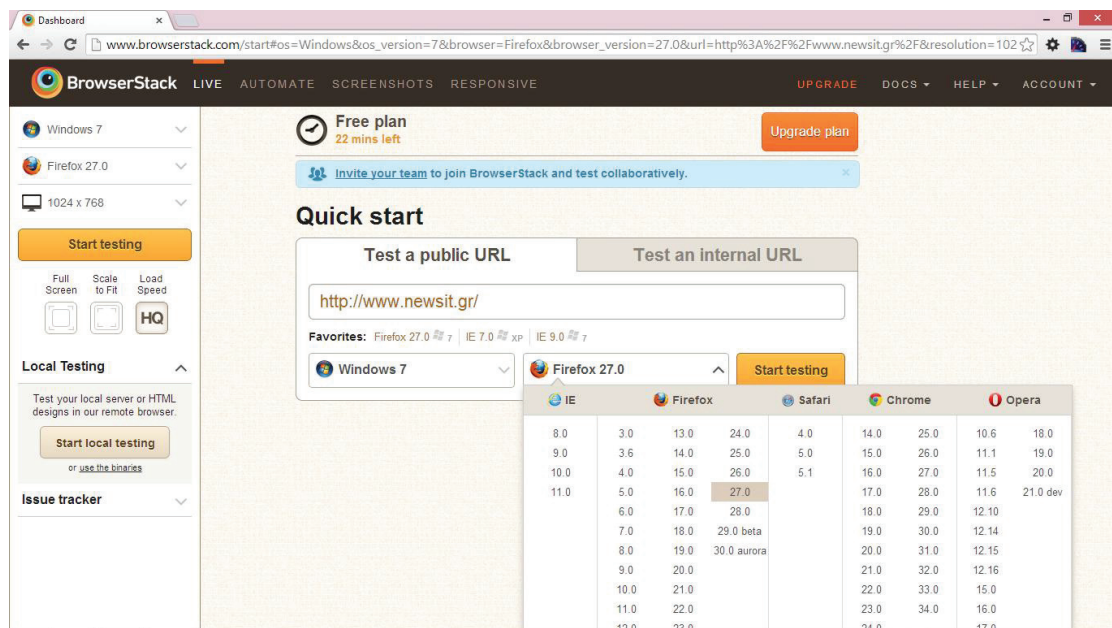
4.4 Ελεγκτής Συμβατότητας

Όπως ήδη αναφέραμε, μια ιστοσελίδα πρέπει να μπορεί να είναι προσβάσιμη όχι μόνο από υπολογιστές αλλά και από κινητές συσκευές όπως τηλέφωνα και tablets. Γι' αυτό είναι απαραίτητο να εκτελείται σωστά. Επιπλέον, θα πρέπει να εμφανίζεται πλήρης και χωρίς απώλειες στην μορφοποίησή της ανεξαρτήτως προγράμματος περιήγησης. Είναι λογικό πως όσο πιο παλιά είναι η έκδοση του περιηγητή αλλά και του συστήματός μας τόσο πιο αργά θα εμφανίζεται η ιστοσελίδα μας, ενώ συχνό είναι το φαινόμενο της κατάρρευσης. Υπάρχουν πολλά προγράμματα που βοηθούν τους προγραμματιστές να εξετάσουν εάν οι ιστοσελίδες που κατασκευάζουν μπορούν αν εκτελεστούν σε διαφορετικά λειτουργικά, με διαφορετικούς browsers και διαφορετικές εκδόσεις αυτών. Δεν καταφέραμε να εντοπίσουμε κάποιο πρόγραμμα που εκτελεί αυτούς τους ελέγχους, βρήκαμε ωστόσο αρκετά που προσφέρουν μια δοκιμαστική έκδοση όπως το [BrowserStack](#), το [Sauce Labs](#) το [CrossBrowserTesting](#) και πολλά άλλα. Για την μελέτη μας χρησιμοποιήσαμε το

πρώτο, αλλά εξετάσαμε και τα υπόλοιπα τα οποία έχουν παρόμοιες ικανότητες και λειτουργία.

4.4.1 Λειτουργία Ελεγκτή Συμβατότητας

Καταφέραμε να μελετήσουμε την λειτουργία του ελεγκτή με την δοκιμαστική έκδοση. Η χρήση του εργαλείου είναι πολύ εύκολη και περιλαμβάνει πολλές επιλογές ελέγχου. Μεταβαίνουμε στην αρχική σελίδα όπως κάνουμε εγγραφή και στην συνέχεια έχουμε μισή ώρα ελεύθερης χρήσης του προϊόντος. Ο χρήστης μπορεί να ελέγχει τόσο δημόσιες διευθύνσεις όσο και αρχεία ιστοσελίδων που βρίσκονται τοπικά στον υπολογιστή του κάνοντας upload τον φάκελο που περιέχει τα απαραίτητα έγγραφα.



Εικόνα 8

Όπως βλέπουμε το πρόγραμμα μπορεί να ελέγξει την συμβατότητα των 5 προγραμμάτων περιήγησης, IE, Firefox, Safari, Chrome, Opera στις περισσότερες εκδόσεις τους.

4.4.2 Μελέτη Ελεγκτή Συμβατότητας

Και σε αυτό το κομμάτι της μελέτης, χρησιμοποιήσαμε δημοφιλείς ιστοσελίδες από τον ελλαδικό χώρο. Λόγω του περιορισμένου χρόνου που διαθέταμε από την δοκιμαστική έκδοση, καταφέραμε να δούμε πως ανταποκρίνονται τρεις ιστοσελίδες, η μια εκ των οποίων είναι η ιστοσελίδα που κατασκευάσαμε για το προγραμματιστικό μέρος. Έτσι θα αναφέρουμε τα αποτελέσματα που προκύπτουν από τις δύο δημόσιες σελίδες .

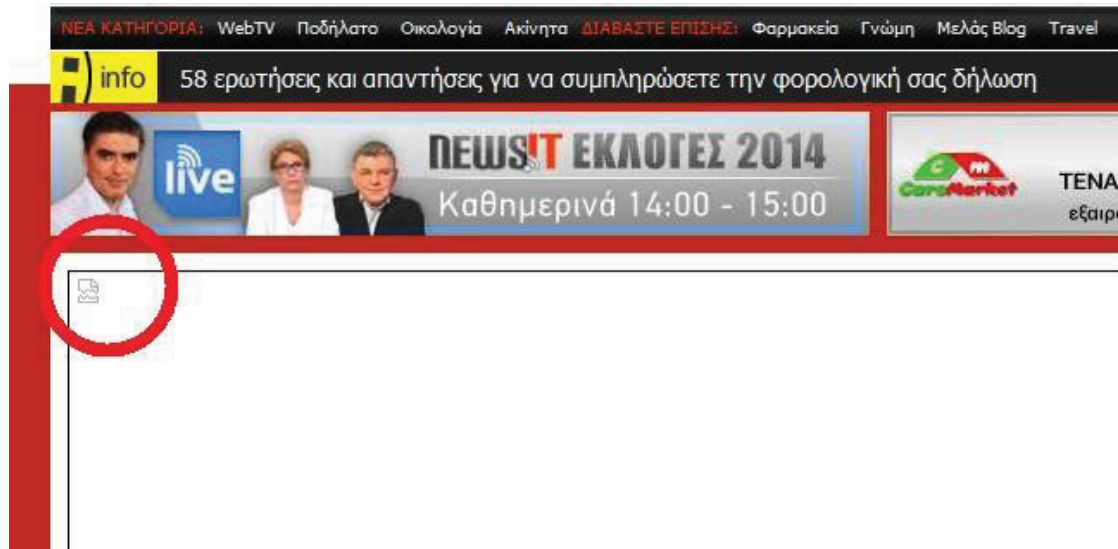
1. NewsIt.gr

Αρχικά εισάγαμε την διεύθυνση της σελίδας και περιμέναμε μέχρι να συνδεθεί με την υπηρεσία. Εξετάσαμε εάν η ιστοσελίδα ήταν συμβατή με την 27.0 έκδοση του Firefox ενώ το λειτουργικό σύστημα που εκτελούνταν η ιστοσελίδα ήταν Windows 7.



Εικόνα 9

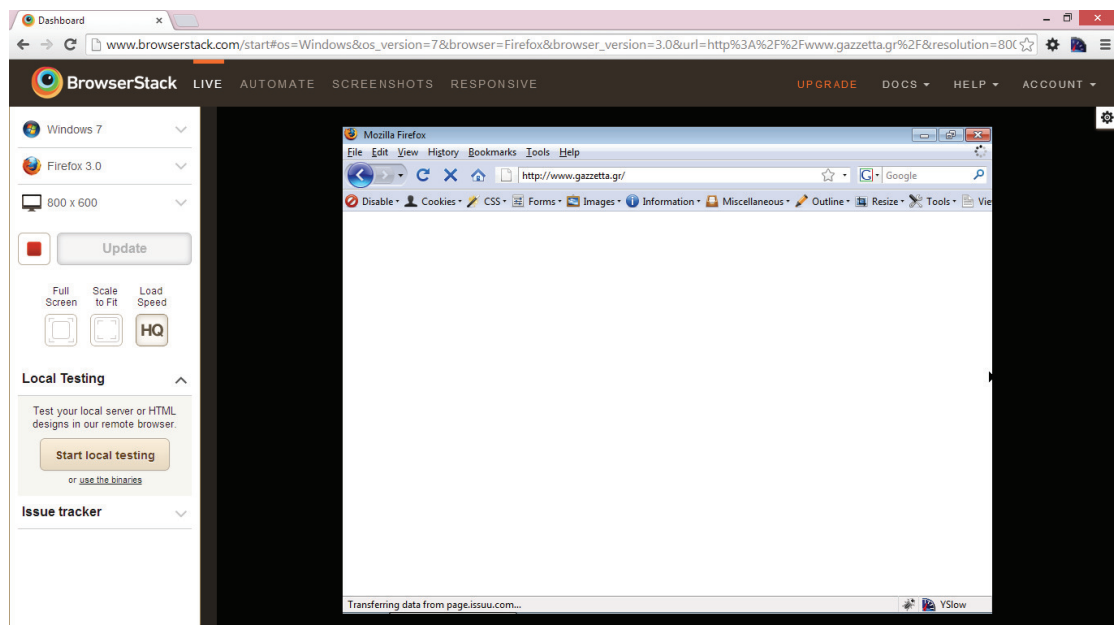
Παρατηρούμε προς μεγάλη μας έκπληξη, ότι η ιστοσελίδα δεν είναι συμβατή, γι' αυτό και δεν μπορεί να εμφανίσει όλα τα στοιχεία, στην συγκεκριμένη περίπτωση τις εικόνες του sidebar των πρόσφατων ειδήσεων



Εικόνα 10

2. In.gr

Η σελίδα In.gr δεν εμφάνισε κάποιο παρόμοιο πρόβλημα, αντιθέτως λειτούργησε και εμφάνισε όλο το περιεχόμενο της ιστοσελίδας χωρίς λάθη. Ωστόσο, κάναμε το τεστ για δύο εκδόσεις, η μία με την πιο πρόσφατη έκδοση του Firefox και η άλλη με την 3^η έκδοση, που είναι σαφώς παλαιότερη.



Εικόνα 11

Αυτό που παρατηρήσαμε είναι πως η απόδοση της ιστοσελίδας ήταν πολύ χαμηλή, μιας και περιμέναμε ένα αρκετά μεγάλο χρονικό διάστημα, για να εκτελεστεί στο επιλεγμένο σύστημα, της τάξεως ~30 sec. Η αναμονή αυτή δεν είναι ανεκτή από τον επισκέπτη ο οποίος πολύ πιθανόν να εγκατέλειπε την προσπάθεια πρόσβασης στον χώρο In.gr

4.5 Ελεγκτής Προσβασιμότητας

Στο προηγούμενο κεφάλαιο αναφέραμε πόσο σημαντικό είναι μια ιστοσελίδα να είναι προσβάσιμη από το μεγαλύτερο μέρος των χρηστών. Ο οργανισμός W3C μάλιστα, εξέδωσε δύο οδηγούς WCAG 1.0 και WCAG 2.0 για Web Content Accessibility Guidelines. Υπάρχουν διάφορα εργαλεία τα οποία ελέγχον αν μια ιστοσελίδα υπακούει στους οδηγούς αυτούς όπως το:

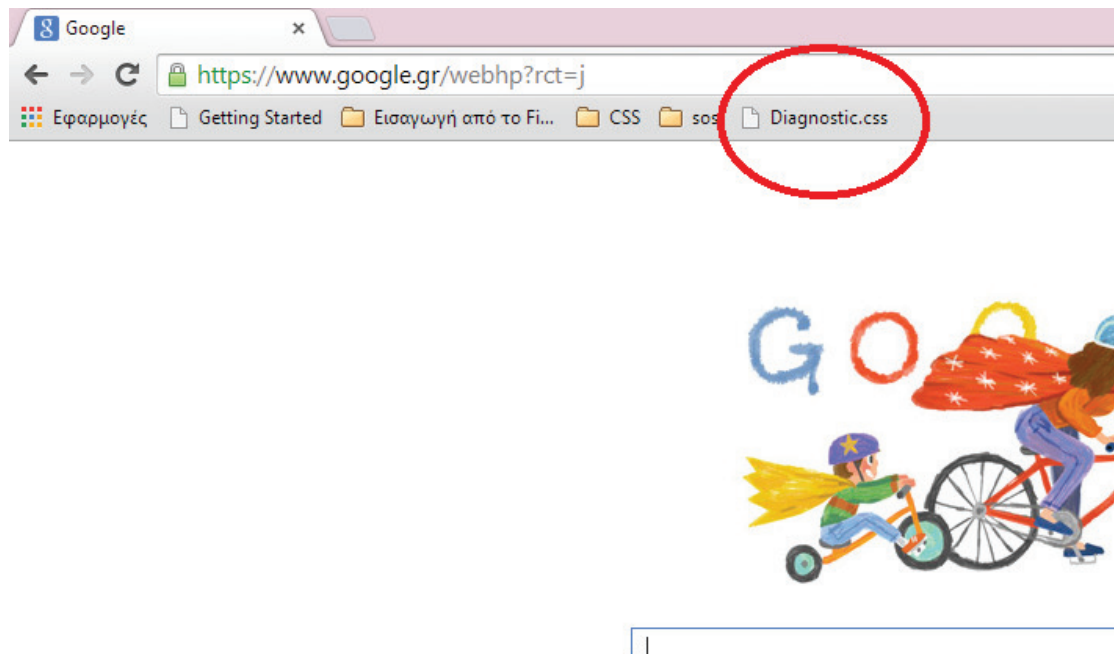
- Cynthia says – ελέγχει τις σελίδες βάσει του οδηγού WCAG 1.0 και των προτεραιοτήτων 1,2 και 3 καθώς και το τμήματος 508 του οδηγού προσβασιμότητας
- HiSoftware Site Quality check – παρέχει έναν ολοκληρωτικό έλεγχο για την προσβασιμότητα, την ποιότητα των σελίδων, την προστασία προσωπικών δεδομένων του χρήστη
- ATRC Web accessibility checker – παρέχει έλεγχο των σελίδων βάσει του οδηγού WCAG 2.0 αλλά και άλλων οδηγιών.
- Colorblind Web Page Filter – εργαλείο αξιολόγησης σελίδων που χρωματίζει τα λάθη που υπάρχουν μέσα στην ιστοσελίδα
- Diagnostic.css – εργαλείο που έχει κατασκευαστεί με την τεχνολογία CSS και τονίζει τα λάθη μέσα στην ιστοσελίδα

Για το κομμάτι της μελέτης των ιστοσελίδων χρησιμοποιούμε το τελευταίο εργαλείο, η χρήση του οποίου αναλύεται στην επόμενη ενότητα.

4.5.1 Λειτουργία Ελεγκτή Προσβασιμότητας

Στην σελίδα <http://www.karlgroves.com/2013/09/07/diagnostic-css-super-quick-web-accessibility-testing/> μπορούμε να βρούμε το εργαλείο Diagnostic.css, το οποίο κάνουμε σελιδοδείκτη για να μπορέσουμε να τον εφαρμόσουμε στην ιστοσελίδα που

θέλουμε να ελέγξουμε. Ένας άλλος τρόπος λειτουργίας είναι να προσθέσουμε τον κώδικα του εργαλείου στην σελίδα που κατασκευάζουμε για να εμφανιστούν οι παραβιάσεις των οδηγών προσβασιμότητας.



Εικόνα 12

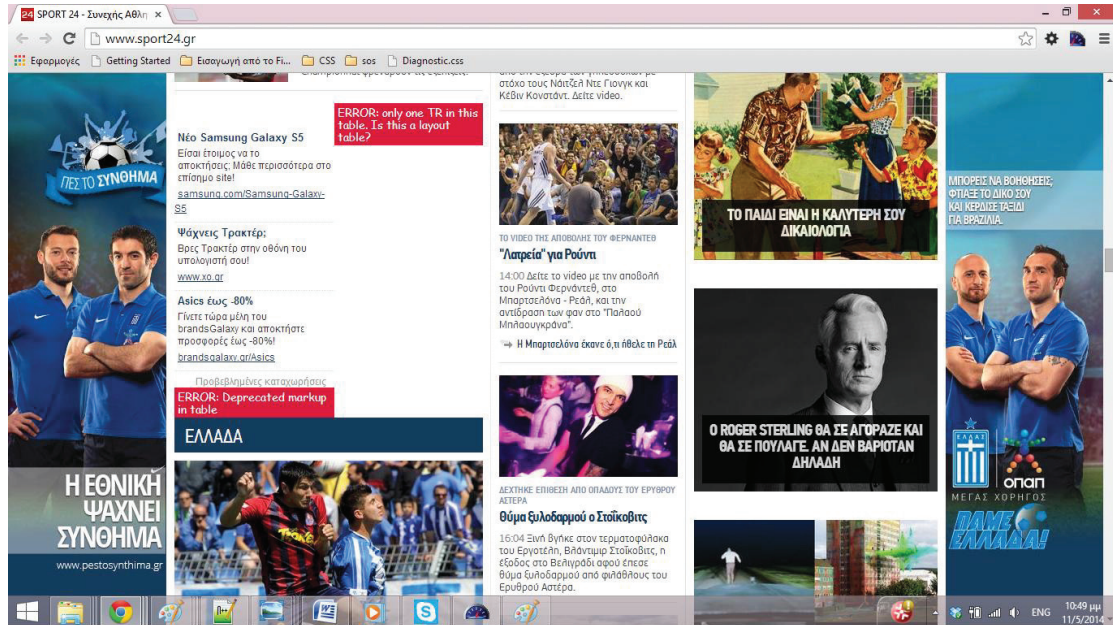
Αφού έχουμε κάνει σελιδοδείκτη το εργαλείο, απλά μεταβαίνουμε στην σελίδα που θέλουμε να μελετήσουμε.

4.5.2 Μελέτη Ελεγκτή Προσβασιμότητας

Η μελέτη με αυτό το εργαλείο ήταν αρκετά βατή. Εισάγαμε με την σειρά τις διευθύνσεις των σελίδων κι στην συνέχεια πατήσαμε τον σελιδοδείκτη του Diagnostic.css όπου τόνισε με κόκκινη γραμματοσειρά τα λάθη, δίνοντας μας την συγκεκριμένη ονομασία του λάθους

1. Sport24.gr

Η ιστοσελίδα sport24.gr περιέχει αθλητικά αλλά και ποικίλης ύλης θέματα. Μεταβήκαμε πληκτρολογώντας την διεύθυνση και παρατηρήσαμε ότι εμφανίζει αρκετά λάθη τα οποία επισημαίνονται μέσα σε κόκκινα πλαίσια.

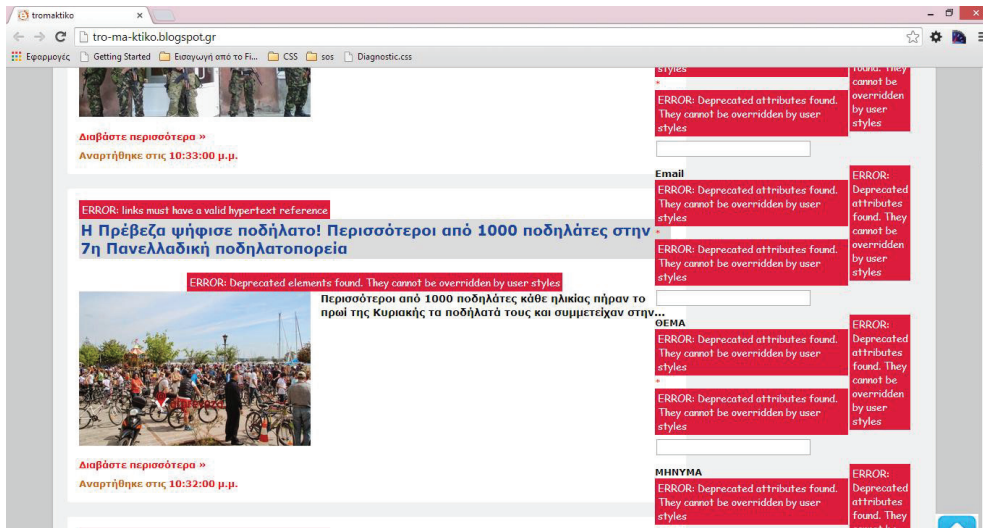


Εικόνα 13

Τα σχόλια είναι αρκετά κατατοπιστικά και βοηθάνε στην αποσφαλμάτωση του κώδικα CSS. Για παράδειγμα εμφανίζεται συχνά μήνυμα λάθους **ERROR: links must have a valid hypertext reference** με το οποίο καταλαβαίνουμε πως δεν έχουμε δηλώσει με σωστό τρόπο τις υπερσυνδέσεις.

2. Tro-ma-ktiko.blogspot.gr

Το <http://tro-ma-ktiko.blogspot.gr/> είναι μια σελίδα τύπου blog που περιέχει νέα της επικαιρότητας από την Ελλάδα και από όλο τον κόσμο. Παρατηρούμε ότι περιέχει πάρα πολλές παραβιάσεις των οδηγιών προσβασιμότητας.

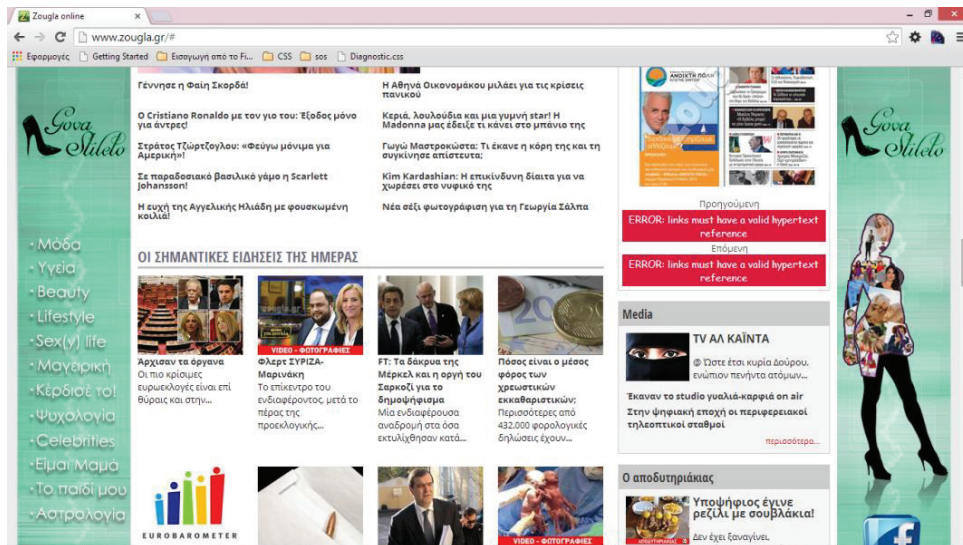


Εικόνα 14

Συχνή είναι η παραβίαση που γίνεται στην μορφοποίηση της ιστοσελίδας. Τα λάθη σε τέτοιου είδους μπλογκ είναι συχνά, μιας και ο χρήστης μπορεί να προσθέτει περιεχόμενο και να αλλάζει την μορφοποίηση μονός του. Είναι φυσικό λοιπόν, οι περισσότεροι χρήστες να μην γνωρίζουν τον σωστό τρόπο αλλαγής της μορφοποίησης μιας και είναι αδύνατο να έχουν βαθιά γνώση προγραμματισμού.

3. Zougla.gr

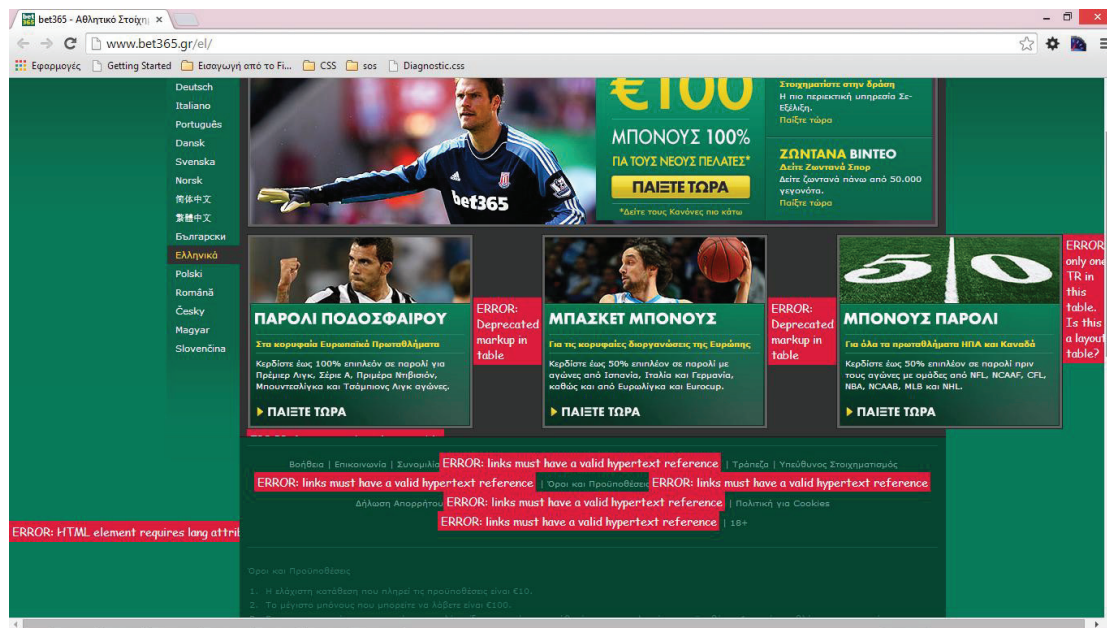
Η σελίδα zougla.gr είναι ακόμη ένας χώρος όπου ο χρήστης μπορεί να ενημερωθεί για νέα της Ελλάδας αλλά και όλου του κόσμου. Και εδώ εμφανίζονται αρκετά λάθη αλλά πολύ λιγότερα σε σχέση με την προηγούμενη που μελετήσαμε.



Εικόνα 15

4. Bet365.gr

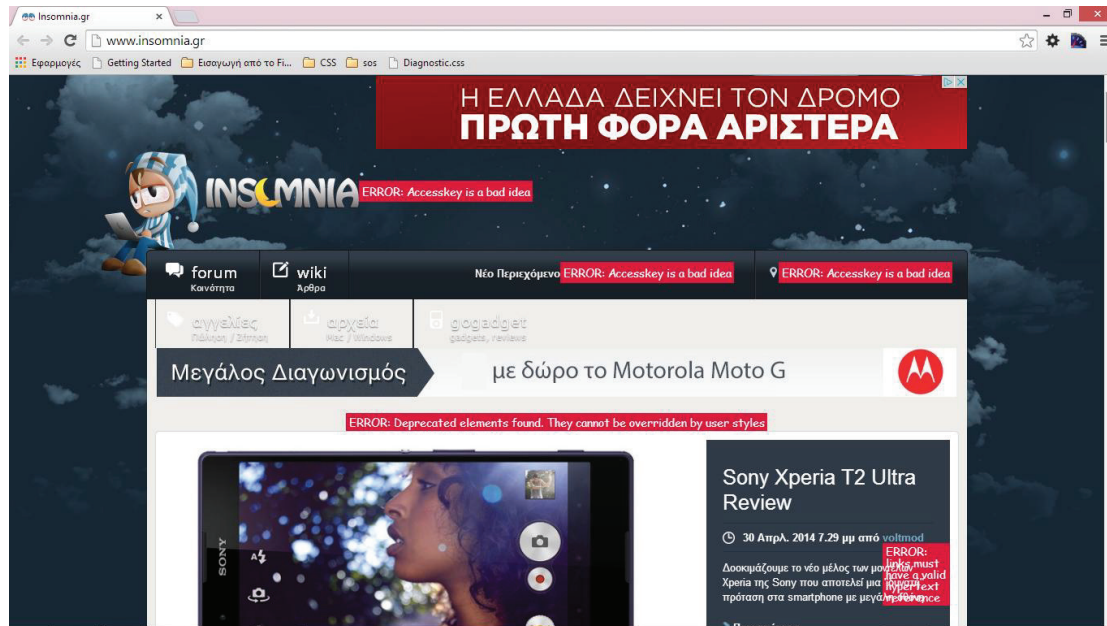
Η bet365.gr είναι μια ιστοσελίδα όπου χρήστες μπορούν να ενημερωθούν για την πορεία αθλητικών ομάδων ενώ έχουν την δυνατότητα να στοιχηματίσουν χρηματικά ποσά σε αγώνες. Και εδώ παρατηρούμε αρκετές παραβάσεις προσβασιμότητας.



Εικόνα 16

5. Insomnia.gr

Το insomnia.gr είναι ένας πολύ δημοφιλή ιστότοπος ο οποίος περιέχει νέα που αφορούν την τεχνολογία. Πρόκειται ουσιαστικά για ένα φόρουμ ανταλλαγής απόψεων πάνω σε θέματα τεχνολογίας υπολογιστών και κινητών ενώ οι χρήστες μπορούν να απολαύσουν μέχρι και μαθήματα προγραμματισμού



Εικόνα 17

Τα λάθη εμφανίζονται μόνο στην αρχή της σελίδας και είναι ελάχιστα.

4.6 Γεννήτρια Παραγωγής Κανόνων CSS

Έχουμε αναφέρει αρκετές φορές πόσο περίπλοκο είναι να γράφει ο προγραμματιστής κώδικα σε CSS. Ιδίως στην 3^η έκδοση όπου η γλώσσα αυξάνει τις δυνατότητες μορφοποίησης, αυξάνεται παράλληλα και η πολυπλοκότητα. Λύση σε αυτό το πρόβλημα έρχονται να δώσουν οι γεννήτριες παραγωγής κανόνων CSS. Με μια απλή αναζήτηση στον Ιστό βλέπουμε πως υπάρχουν πολλά online εργαλεία που κατασκευάζουν αυτόματα αντικείμενα CSS 3 ανάλογα με τις απαιτήσεις του χρήστη.

4.6.1 Λειτουργία Γεννήτριας Κανόνων CSS

Στην συνέχεια θα αναφέρουμε κάποιες σελίδες που περιέχουν γεννήτριες κανόνων καθώς και τις δυνατότητες που έχει η κάθε μία. Έτσι, κάποιες από αυτές είναι:

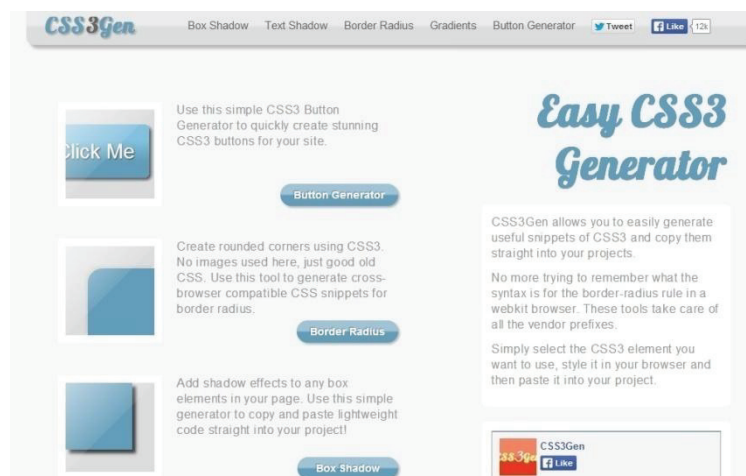
1. **CSS3 Generator:** Με αυτό το εργαλείο μπορούμε να κατασκευάσουμε κανόνες για Border Radius, Box Shadow, Text Shadow, RGBA, Font Face, Multiple Columns, Box Resize, Box Sizing, Outline, Transition, Transform και Gradient



Εικόνα 18

Επιλέγοντας ένα από τα παραπάνω ο χρήστης μεταβαίνει σε μια νέα σελίδα στην οποία δίνει στοιχεία και δημιουργείται βάσει αυτών ο κανόνας που επιθυμεί.

2. **CSS3Gen:** Παρόμοια λειτουργία με την προηγούμενη σελίδα προσφέρει και αυτή η γεννήτρια, αλλά με πιο περιορισμένες δυνατότητες. Η CSS3Gen κατασκευάζει Buttons, Border Radius, Box Shadows, CSS3 Gradients και Text Shadows



Εικόνα 19

3. **CSS 3 Generetor:** Διαθέτει ακριβώς τις ίδιες δυνατότητες με την πρώτη γεννήτρια μόνο που εδώ ο χρήστης μπορεί με την βοήθεια κινούμενων μπάρων να επιλέξει ακριβώς όπως επιθυμεί το χρώμα Gradient, το Border Radius, το Box Shadows κ.α.



Εικόνα 20

4.7 Οδηγός Συγγραφής Ποιοτικού Κώδικα CSS 3

1. Γράψε στην αρχή του κώδικα HTML την εντολή `<!DOCTYPE html>`
2. Πρόσθεσε inline το αρχείο CSS μέσα στο οποίο βρίσκεται η μορφοποίηση
3. Ακολούθησε τον Οδηγό Βελτίωσης της Απόδοσης για να γράψει πιο αποδοτικό CSS κώδικα
4. Ακολούθησε τον Οδηγό Προσβασιμότητας Ιστοσελίδων για να κατασκευάσεις προσβάσιμη ιστοσελίδα
5. Ακολούθησε τον Οδηγό Εγγραφής Αποτελεσματικών CSS επιλογέων για να κατασκευάσεις σωστούς επιλογείς
6. Ακολούθησε τον Οδηγό Επίλυσης Προβλημάτων Συμβατότητας για να έχεις συμβατές ιστοσελίδες
7. Ακολούθησε τον Οδηγό Επίτευξης Αφαιρετικότητας
8. Ακολούθησε τον Οδηγό Μέτρησης Πολυπλοκότητας
9. Επικύρωσε την σελίδα σου με τον CSS W3C
10. Έλεγξε με το εργαλείο Συμβατότητας
11. Έλεγξε με το εργαλείο Προσβασιμότητας
12. Κατασκεύασε έτοιμους κανόνες με CSS 3 Generators για μείωση της πολυπλοκότητας

4.8 Σύνοψη

Στο κεφάλαιο αυτό μελετήσαμε εργαλεία τα οποία βοηθάνε τους προγραμματιστές να κατασκευάζουν σελίδες που χαρακτηρίζονται από υψηλή ποιότητα. Έτσι, χρησιμοποιήσαμε το εργαλείο CSS Validator για να επικυρώσουμε τις σελίδες. Επικυρωμένες σελίδες, έχουν σαν αποτέλεσμα βελτίωση της απόδοσης των σελίδων. Μελετήσαμε στην συνέχεια, εργαλεία που τεκμηριώνουν την προσβασιμότητα των ιστοσελίδων, καθώς και τρόπους ελέγχου της συμβατότητας, με απώτερο σκοπό την κατασκευή ανταγωνιστικών σελίδων τις οποίες όλοι οι χρήστες, ακόμα και αυτοί που αντιμετωπίζουν κάποια μορφή αναπηρίας, μπορούν να χρησιμοποιήσουν. Τέλος, προκειμένου να μειώσουμε την πολυπλοκότητα του CSS κώδικα, κατασκευάσαμε αυτόματα και με την βοήθεια γεννητριών, κανόνες CSS.

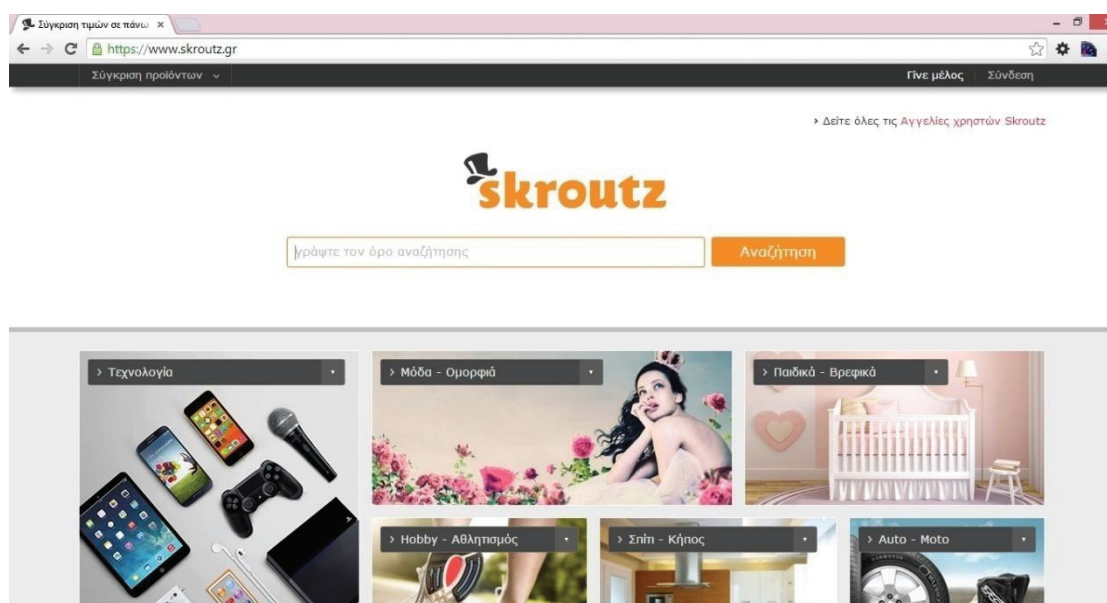
Κεφάλαιο 5^ο

5.1 Εισαγωγή

Στα πλαίσια της πτυχιακής εργασίας, ζητήθηκε να γραφτεί σε κώδικα CSS 3, μια εναλλακτική μορφή της αρχικής σελίδας, ενός από τα πιο δημοφιλή site της χώρας. Εμείς, επιλέξαμε να ανακατασκευάσουμε την αρχική σελίδα του Skroutz.gr το οποίο αποτελεί μια βάση αναζήτησης διάφορων προϊόντων ενώ παρουσιάζει τις καλύτερες τιμές που βρίσκει γι' αυτά. Για την δημιουργία, εμπνευστήκαμε από ένα έτοιμο template, ενώ προσθέσαμε και άλλες περισσότερες λειτουργίες που γίνονται μόνο με την 3^η έκδοση των CSS. Ακολουθήσαμε τέλος, τον οδηγό συγγραφής που παρουσιάστηκε σε προηγούμενο κεφάλαιο.

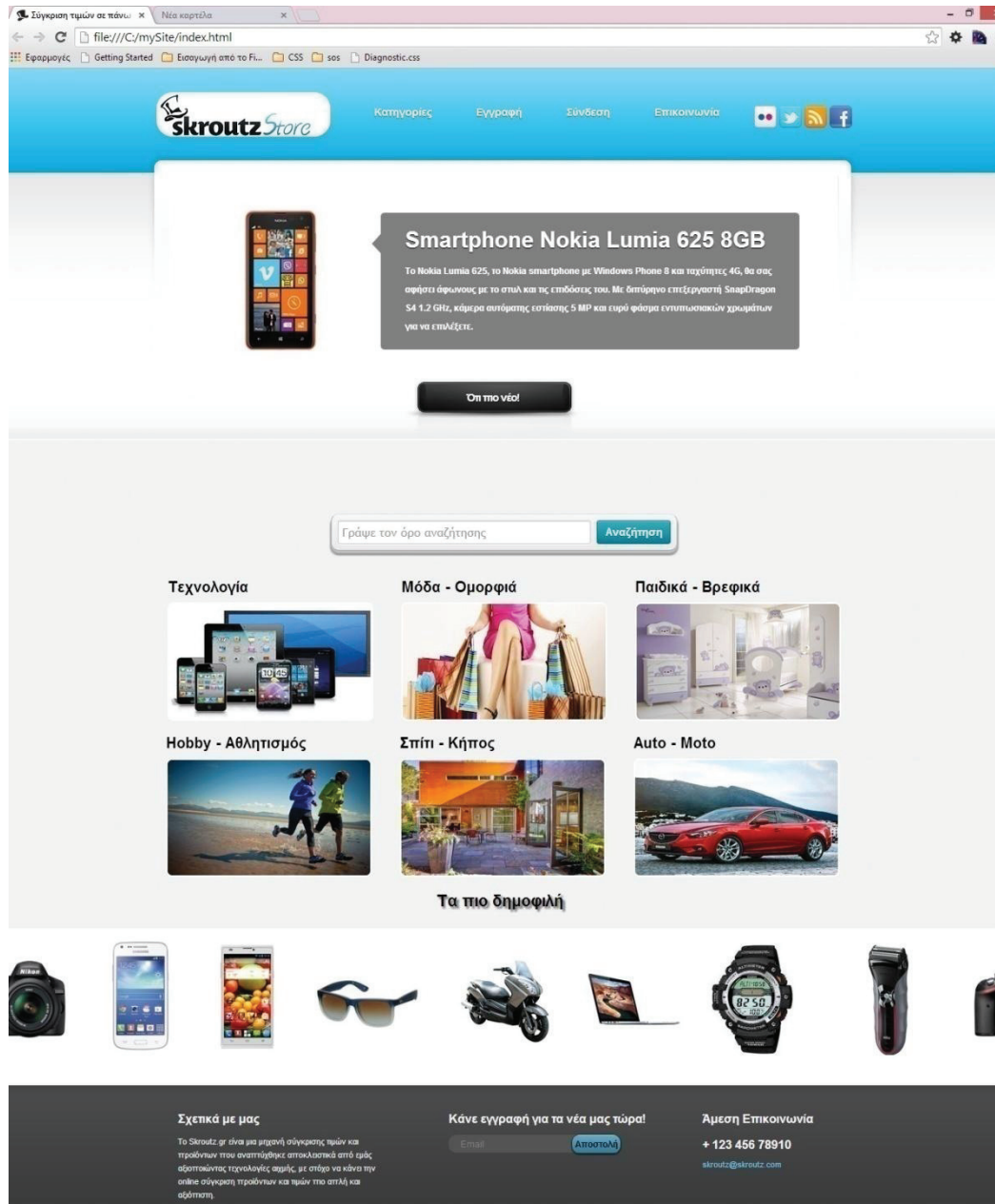
5.2 Κατασκευή Σελίδας με CSS 3

Η αρχική σελίδα του Skroutz είναι λειτουργική και καλαίσθητη ενώ ενημερώνει τον χρήστη για τις κατηγορίες των προϊόντων που μπορεί να βρει. Επιπλέον, διαθέτει μια μπάρα αναζήτησης για να διευκολύνει τους χρήστες, ενώ τους ενημερώνει για τα πιο δημοφιλή προϊόντα που έχει στην βάση του. Η αρχική σελίδα φαίνεται παρακάτω:



Εικόνα 21

Η σελίδα που κατασκευάσαμε εμείς έχει την εξής μορφή



Εικόνα 2

Στην συνέχεια θα αναλύσουμε τα νέα στοιχεία CSS 3 που προσθέσαμε.

- **Στοιχείο 1^ο** : Εισαγωγή εικόνας με στρογγυλεμένες άκρες. Χρησιμοποιήσαμε την γεννήτρια CSS 3 Generetor και κατασκευάσαμε την εικόνα με το λογότυπο του Skroutz
- **Στοιχείο 2^ο** : Κατασκευή περιστρεφόμενων εικόνων για την επικοινωνία των social media και όχι μόνο, δηλαδή Twitter, Facebook, RSS και Flickr.
- **Στοιχείο 3^ο** : Κατασκευή μπάρα αναζήτησης με στρογγυλεμένες άκρες και με χρωματισμό Gradient του CSS 3
- **Στοιχείο 4^ο** : Κατασκευή κυλιόμενης μπάρας εικόνων
- **Στοιχείο 5^ο** : Κατασκευή CSS 3 Button



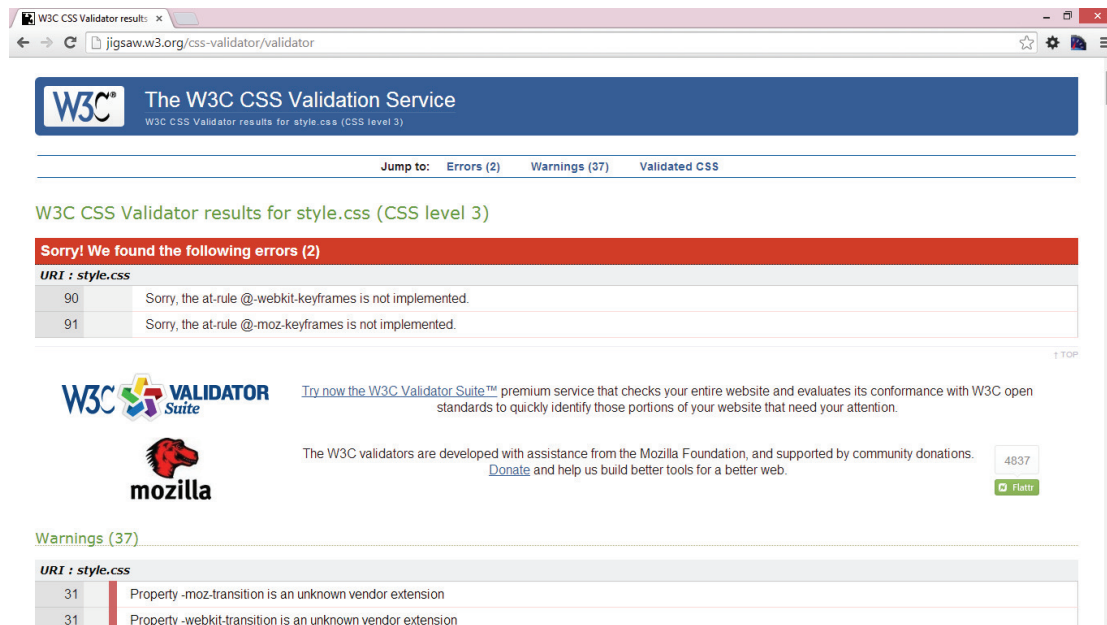
Στο ΠΑΡΑΡΤΗΜΑ Β ο αναγνώστης μπορεί να βρει τόσο τον κώδικα CSS3 που περιλαμβάνει την μορφοποίηση, όσο και τον κώδικα της HTML με το περιεχόμενο της σελίδας.

5.3 Μελέτη Σελίδας και έλεγχος με εργαλεία

Στο προηγούμενο κεφάλαιο αναπτύξαμε εργαλεία χρήσιμα για την συγγραφή σωστού κώδικα CSS 3. Στην συνέχεια μελετήσαμε την συμπεριφορά δημοφιλών σελίδων από τον ελλαδικό χώρο για το πώς ανταποκρίνονται. Την ίδια ακριβώς διαδικασία θα ακολουθήσουμε για να ελέγξουμε το site που κατασκευάσαμε.

5.3.1 Έλεγχος Επικυρωτή CSS 3

Ο CSS Validator μας δίνει την δυνατότητα να ελέγξουμε ένα αρχείο το οποίο βρίσκεται τοπικά στον υπολογιστή μας, αρκεί πρώτα να κάνουμε upload. Το αποτέλεσμα του Validation φαίνεται παρακάτω.



W3C CSS Validator results for style.css (CSS level 3)

Jump to: [Errors \(2\)](#) [Warnings \(37\)](#) [Validated CSS](#)

W3C CSS Validator results for style.css (CSS level 3)

Sorry! We found the following errors (2)

URI : style.css	
90	Sorry, the at-rule @-webkit-keyframes is not implemented.
91	Sorry, the at-rule @-moz-keyframes is not implemented.

Try now the [W3C Validator Suite™](#) premium service that checks your entire website and evaluates its conformance with W3C open standards to quickly identify those portions of your website that need your attention.

The W3C validators are developed with assistance from the Mozilla Foundation, and supported by community donations. [Donate](#) and help us build better tools for a better web.

Warnings (37)

URI : style.css	
31	Property -moz-transition is an unknown vendor extension
31	Property -webkit-transition is an unknown vendor extension

Εικόνα 3

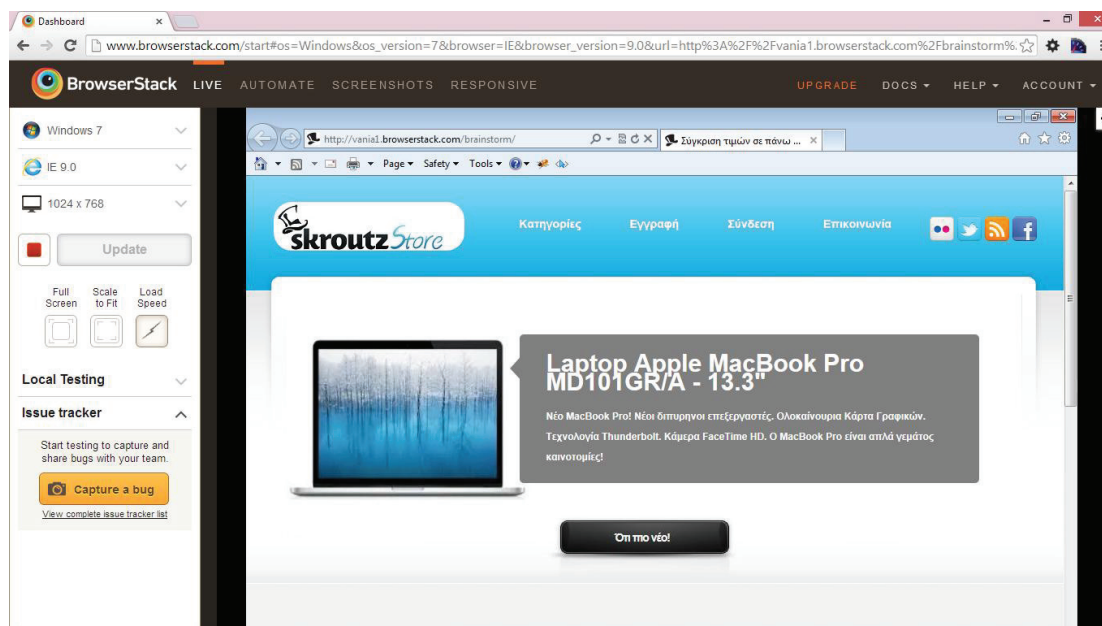
Παρατηρούμε ότι εμφανίζει 2 λάθη:

- Sorry, the at-rule @-webkit-keyframes is not implemented.
- Sorry, the at-rule @-moz-keyframes is not implemented.

Ήδη αναφέραμε στην ανάλυση του CSS Validator ότι τα λάθη αυτά εμφανίζονται γιατί οι προγραμματιστές εισάγουν τα λεγόμενα hacks για να κατασκευάσουν πιο αποτελεσματικές και καλαίσθητες σελίδες. Τα λάθη προέρχονται από την κατασκευή της κυλιόμενης μπάρας εικόνων και δεν επηρεάζουν την απόδοση της σελίδας.

5.3.2 Έλεγχος Συμβατότητας

Κάνοντας χρήση του εργαλείου [BrowserStack](#) μελετήσαμε πως δουλεύει η σελίδα μας σε διάφορους συνδυασμούς συστημάτων και εκδόσεων λογισμικού. Η σελίδα κατάφερε να περάσει όλα τα τεστ και να μην εμφανίζει κανένα πρόβλημα.



Εικόνα 4

Όπως βλέπουμε και στην εικόνα, όλα τα στοιχεία εμφανίζονται ακριβώς με τον ίδιο τρόπο που τα κατασκευάσαμε χωρίς κάποια απώλεια στην μορφοποίηση.

5.3.3 Έλεγχος Προσβασιμότητας

Με το εργαλείο Diagnostic.css εξετάσαμε εάν το αρχείο μας ακολουθεί όλους τους κανονισμούς που έχουν τεθεί από τους οδηγούς WCAG 1.0 και WCAG 2.0 και παρατηρήσαμε ότι δεν εμφανίζεται κανένα μήνυμα λάθους (Mesbah Ali, 2012), (Zhu Jiang, 2013), (Geneves Pierre, 2011), (Cyntrica, 2012), (Zakraoui Jesia, 2012), (Zakas, 2013).

Βιβλιογραφία

- [1].Adewumi Adewole, M.S..I.-O.N., 2012. Complexity Metrics for Cascading Style Sheets. In *ICCSA 2012.*, 2012.
- [2].Ahmad Rashid, L.Z.A.F., 2006. Measuring ‘Navigational Burden’, School of Software Engineering. In *SERA '06.*, 2006.
- [3].Bailey John, B.E., 2006. What is the current state of Web Accessibility? In *WSE'06.*, 2006.
- [4].Cornelia, B., 2002. Determination and Evaluation of Web Accessibility Research Institute for Software Evolution. In *Proceedings of the Eleventh IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'02).*, 2002.
- [5].Cyntrica, E., 2012. A Framework for Detecting and Diagnosing Configuration Faults in Web Applications.
- [6].Geneves Pierre, L.N..Q.V., 2011. On the Analysis of Cascading Style Sheets.
- [7].Guarino Reid Loretta, S.-W.A., 2008. WCAG 2.0: A Web Accessibility Standard for the Evolving Web. In *Google, Inc./IBM, W4A2008 - Technical.*, 2008.
- [8].Hall, A.C., 2009. Web Presentation Layer Bootstrapping for Accessibility and Performance.
- [9].Hao Zheng, H.H., 2012. A Web Design Mode for browsers to CSS compatibility issues.
- [10]. James, F., 2007. CSS 3 Takes On the Ongoing Problem of Fonts. *The Seybold Report.*
- [11]. Keller Matthias, N.M., 2009. Cascading Style Sheets: A Novel Approach Towards Productive Styling with Today’s Standards.
- [12]. Keller Matthias, N.M., 2010. *CSS Code Quality: A Metric for Abstractness Or Why Humans Beat Machines in CSS Coding.*
- [13]. Kepler, J., 2005. 16th International Workshop on Database and Expert Systems Applications. In *DEXA '05.*, 2005.
- [14]. Leitner Michael, S.S.S.T., 2005. Practical Application of WAI Guidelines in Web Design, Institute of Informatics in Business and Government.

- [15]. Lie, H.W., 2005. Cascading Style Sheets, Faculty of Mathematics and Natural Sciences.
- [16]. Lourdes Moreno, P.M.B.R., 2008. Guiding accessibility issues in the design of Websites.
- [17]. Mesbah Ali, M.S., 2012. Automated Analysis of CSS Rules to Support Style Maintenance.
- [18]. Oehler Matthias, B.S., 2008. Web Sites Communication Performance: a Methodological Approach.
- [19]. Sikos, L.F., 2011. *Web Standards, Mastering HTML5, CSS3, and XML*. Apress.
- [20]. Steve, S., 2009. *Even faster websites*. 1005 Gravenstein Highway Nort: O'Reilly Media.
- [21]. Tammy, E., 2013. An overview of techniques to speed page loading.
- [22]. Termens Miquel, R.M.P.M.B.M.S.A.P.P., 2009. Web Content Accessibility Guidelines: from 1.0 to 2.0.
- [23]. Zakas, N.C., 2013. Building Web sites that perform well on mobile devices remains a challenge.
- [24]. Zakraoui Jesia, Z.W., 2012. A Method for Generating CSS to Improve Web Accessibility for Old Users., 2012.
- [25]. Zhu Jiang, C.D.S..S.P.M.N.P.H.H.B.F., 2013. Improving Web Sites Performance Using Edge Servers in Fog Computing Architecture.
- [26]. Zihou, Z., 2009. Using a Practical Quality Model.
- [27]. Κομβούτη Βέρου, Μ., n.d. Εισαγωγή στο CSS.
- [28]. Μητσαράκης, Κ., 2009. Δημιουργία ενός δυναμικού κοινωνικού ιστοχώρου για τους υποψηφίους των εξετάσεων του ΑΣΕΠ (online social network site) με χρήση PHP, MySQL και JavaScript.
- [29]. Μπαλής, Π., n.d. *Πληροφορική VI: Δυναμικές Εφαρμογές Παγκόσμιου Ιστού*. Υπουργείο Εθνικής Παιδείας και Θρησκευμάτων.

ΠΑΡΑΡΤΗΜΑ Α

Ιδιότητα	IE	Firefox	Chrome	Safari	Opera
<u>align-content</u>		28	21		12.1
<u>align-items</u>	11	20	21	-webkit- 7	12.1
<u>align-self</u>		20	21		12.1
<u>@keyframes</u>	10	16	-webkit-	-webkit-4	12.1
<u>animation</u>	10	16	-webkit-	-webkit-4	12.1
<u>animation-name</u>	10	16	-webkit-	-webkit-4	12.1
<u>animation-duration</u>	10	16	-webkit- 3	-webkit-4	12.1
<u>animation-timing-function</u>	10	16	-webkit-		12.1
<u>animation-delay</u>	10	16	-webkit-	-webkit-4	12.1
<u>animation-iteration-count</u>	10	16	-webkit-	-webkit-4	12.1
<u>animation-direction</u>	10	16	-webkit-	-webkit-4	12.1
<u>animation-play-state</u>	10	16	-webkit-		12.1
<u>backface-visibility</u>	-ms-10	16	-webkit- 12		15
<u>background-clip</u>	09	4	4	3	10.5
<u>background-origin</u>	09	4	4	3	10.5
<u>background-size</u>	09	4	4	4.1	10
<u>border-bottom-left-radius</u>	09	4	4	5	10.5
<u>border-bottom-right-radius</u>	09	4	4	5	10.5
<u>border-image</u>	11	15	16	6	15
<u>border-image-outset</u>	11	15	15	6	
<u>border-image-repeat</u>	11	15	15	6	
<u>border-image-slice</u>	11	15	15	6	
<u>border-image-source</u>	11	15	15	6	
<u>border-image-width</u>	11	13	15	6	
<u>border-radius</u>	09	4	4	5	10.5
<u>border-top-left-radius</u>	09	4	4	5	10.5
<u>border-top-right-radius</u>	09	4	4	5	10.5
<u>box-decoration-break</u>					
<u>box-shadow</u>	09	-moz-4	-webkit-10	-webkit- 5.1	10.5
<u>box-sizing</u>	08	-moz-1	09	5.1	7
<u>break-after</u>	10				11.1
<u>break-before</u>	10				11.1
<u>break-inside</u>					
<u>column-fill</u>		-moz- 13			

<u>column-gap</u>	10		-webkit-	-webkit-3	11.1
<u>column-rule</u>	10	-moz-3.5	-webkit-	-webkit-3	11.1
<u>column-rule-color</u>	10	-moz-3.5	-webkit-	-webkit-3	11.1
<u>column-rule-style</u>	10	-moz-3.5	-webkit-	-webkit-3	11.1
<u>column-rule-width</u>	10		-webkit-	-webkit-	11.1
<u>column-span</u>	10		-webkit-	-webkit-	11.1
<u>column-width</u>	10	-moz-	-webkit-	-webkit-3	11.1
<u>columns</u>	10	-moz-9.9	-webkit-	-webkit-3	11.1
<u>column-count</u>	10	-moz-		3	11.1
<u>fit</u>					
<u>flex</u>	11	20	29		12.1
<u>flex-basis</u>		22	29		12.1
<u>flex-direction</u>		20	29		12.1
<u>flex-flow</u>		28	29		12.1
<u>flex-grow</u>		20	29		12.1
<u>flex-shrink</u>		20	29		12.1
<u>flex-wrap</u>	11	28	29	-webkit-6.1	12.1
<u>@font-face</u>	09	3.6	4	3	10
<u>font-feature-setting</u>	10	-moz-4	-webkit-16		
<u>@font-feature-values</u>					
<u>font-kerning</u>					
<u>font-language-override</u>					
<u>font-synthesis</u>					
<u>font-variant-alternates</u>					
<u>font-variant-caps</u>					
<u>font-variant-east-asian</u>					
<u>font-variant-ligatures</u>					
<u>font-variant-numeric</u>					
<u>font-variant-position</u>					
<u>font-size-adjust</u>		3			
<u>font-stretch</u>	09	9			
<u>hanging-punctuation</u>					
<u>hyphens</u>	-ms-10	-moz-6	-webkit-13	-webkit-5.1	
<u>icon</u>					
<u>image-orientation</u>		26			
<u>image-rendering</u>		-moz-			11.6
<u>image-resolution</u>					
<u>justify-content</u>		20	29		12.1
<u>line-break</u>					
<u>Mark</u>					

mark-after					
mark-before					
marks					
marquee-direction			-webkit-	-webkit-	
marquee-play-count			-webkit-	-webkit-	
marquee-speed			-webkit-	-webkit-	
marquee-style			-webkit-	-webkit-	
<u>nav-down</u>					11.5
<u>nav-index</u>					11.5
<u>nav-left</u>					11.5
<u>nav-right</u>					11.5
<u>nav-up</u>					11.5
object-fit					
object-position					
<u>opacity</u>	09	2	4	3.1	9
<u>order</u>		20	29		12.1
<u>outline-offset</u>	5.5	3.5			10.5
overflow-wrap	09	3.5		3	9.5
<u>overflow-x</u>	09	3.5		3	9.5
<u>overflow-y</u>	09			3	9.5
@page	08	19		5	6
<u>perspective</u>	10	16	-webkit-12	-webkit-5	
<u>perspective-origin</u>	10	16	-webkit-12	-webkit-	
<u>resize</u>		5	4	4	
rest					
rest-after					
rest-before					
ruby-align					
<u>tab-size</u>		-moz-4	21		15
<u>text-align-last</u>	5.5	-moz-12			
text-combine-horizontal					
<u>text-decoration-color</u>		-moz-6			
<u>text-decoration-line</u>		-moz-6			
<u>text-decoration-style</u>		-moz-6			
<u>text-justify</u>					
text-orientation					
<u>text-overflow</u>	06	7	4	3.1	11

<u>text-shadow</u>	10	3.5	4	4	9.5
<u>text-underline-position</u>					
<u>transform</u>	10	16	-webkit-	-webkit-3.1	-webkit-15
<u>transform-origin</u>	10	16	-webkit-	-webkit-3.1	-webkit-15
<u>transform-style</u>		16	-webkit-12	-webkit-5	-webkit-15
<u>transition</u>	10	16	26	-webkit-3	12.1
<u>transition-property</u>	10	16	26	-webkit-	12.1
<u>transition-duration</u>	10	16	26	-webkit-3	12.1
<u>transition-timing-function</u>	10	16	26		12.1
<u>transition-delay</u>	10	16	26	3	12.1
<u>word-break</u>	5.5	15	4	3	15
<u>word-wrap</u>	5.5	3.5	4	3.1	10.5
<u>writing-mode</u>	-ms-				

Πίνακας Α

Στον παραπάνω πίνακα (Πίνακας Α) παρουσιάζονται όλες οι ιδιότητες της CSS όπου υπάρχει αριθμός δηλώνει την έκδοση του browser που άρχισε να υποστηρίζει την αντίστοιχη ιδιότητα, ενώ όπου:

- ❖ **-ms-** ο Internet Explorer υποστηρίζει την ιδιότητα με το πρόθεμα -ms-
- ❖ **-mz-** ο Mozilla υποστηρίζει την ιδιότητα με το πρόθεμα -mz-
- ❖ **-webkit-** ο Chrome υποστηρίζει την ιδιότητα με το πρόθεμα -webkit-
- ❖ **-webkit-** ο Safari υποστηρίζει την ιδιότητα με το πρόθεμα -webkit-
- ❖ **-webkit-** ο Opera υποστηρίζει την ιδιότητα με το πρόθεμα -webkit-

ΠΑΡΑΡΤΗΜΑ Β

Κώδικας HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>Σύγκριση τιμών σε πάνω από 1250 online καταστήματα!
  Skroutz.gr</title>
  <link rel="stylesheet" href="css/style.css" type="text/css"
  media="all" />
  <link rel="shortcut icon" type="image/x-icon"
  href="css/images/skroutz.ico" />
  <link rel="stylesheet" href="css/flexslider.css" type="text/css"
  media="all" />

  <script src="js/jquery-1.7.2.min.js"
  type="text/javascript"></script>
  <!--[if lt IE 9]>
    <script src="js/modernizr.custom.js"></script>
  <![endif]-->
  <script src="js/jquery.flexslider-min.js"
  type="text/javascript"></script>
  <script src="js/functions.js" type="text/javascript"></script>
</head>
<body>
  <!-- wrapper -->
  <div id="wrapper">
    <!-- header -->
    <header class="header">
      <div class="shell">
        <h1 id="logo"><a href="#">BrainStorm</a></h1>
        <!-- navigation -->
        <nav id="navigation">
          <ul>
            <li>
              <a href="#">Κατηγορίες</a>
            </li>
            <li>
              <a href="#">Εγγραφή</a>
            </li>
            <li>
              <a href="#">Σύνδεση</a>
            </li>
            <li>
              <a href="#">Επικοινωνία</a>
            </li>
            <li>
              <div class="rotate">
                

```

```

        
        
        
    </div>
</li>
</ul>
<div class="cl">&nbsp;</div>

</nav>
<!-- end of navigation -->
</div>
</header>
<!-- end of header -->

<!-- slider-holder -->
<div class="slider-holder">
    <!-- shell -->
    <div class="shell">
        <span class="l"></span>
        <span class="r"></span>
        <span class="t"></span>
        <span class="b"></span>
        <a href="#" class="slider-btn">Ότι πιο νέο!</a>

    <!-- slider -->
    <div class="slider flexslider">
        <ul class="slides">
            <li>
                
                <div class="box-t"></div>
                <div class="box-c">
                    <div class="box-cnt">
                        <h2>Laptop Apple MacBook Pro
MD101GR/A - 13.3"</h2>
                        <p>Νέο MacBook Pro! Νέοι
διπυρήνοι επεξεργαστές. Ολοκαίνουρια Κάρτα Γραφικών. Τεχνολογία
Thunderbolt. Κάμερα FaceTime HD. Ο MacBook Pro είναι απλά γεμάτος
καινοτομίες!</p>
                    </div>
                    <div class="cl">&nbsp;</div>
                </div>
                <div class="box-b"></div>
            </div>
        </li>
        <li>
            
            <div class="slide-cnt">
                <div class="box-t"></div>
                <div class="box-c">
                    <div class="box-cnt">
                        <h2>Smartphone Nokia Lumia
625 8GB </h2>
                        <p>Το Nokia Lumia 625, το
Nokia smartphone με Windows Phone 8 και ταχύτητες 4G, θα σας αφήσει
άφωνους με το στυλ και τις επιδόσεις του. Με διπύρηνο επεξεργαστή

```



```

        </div>

        <div class="col">
            <h3><a href="#">Παιδικά - Βρεφικά</a></h3>
            <a href="#"></a>
        </div>
        <div class="cl">&nbsp;</div>

        <div class="col">
            <h3><a href="#">Hobby - Αθλητισμός</a></h3>
            <a href="#"></a>
        </div>

        <div class="col">
            <h3><a href="#">Σπίτι - Κήπος</a></h3>
            <a href="#"></a>
        </div>

        <div class="col">
            <h3><a href="#">Auto - Moto</a></h3>
            <a href="#"></a>
        </div>
        <div class="cl">&nbsp;</div><div
class="cl">&nbsp;</div><div class="cl">&nbsp;</div>
    </section>

    <h2>Τα πιο δημοφιλή</h2>
</div>
<!-- end of main -->
</div>

<div id="tech-slideshow">
    <div id="tech-slideshow-1"></div>
</div>
<!-- end of shell -->
<div id="footer-push"></div>
</div>
<!-- wrapper -->

<!-- footer -->
<div id="footer">
    <div class="shell">
        <!-- footer-cols -->
        <section class="footer-cols">
            <div class="col">
                <h3><strong>Σχετικά με μας</strong></h3>
                <p>Το Skroutz.gr είναι μια μηχανή σύγκρισης τιμών
και προϊόντων που αναπτύχθηκε αποκλειστικά από εμάς αξιοποιώντας
τεχνολογίες αιχμής, με στόχο να κάνει την online σύγκριση προϊόντων
και τιμών πιο απλή και αξιόπιστη.</p>
            </div>
            <div class="col">
                <h3><strong>Κάνε εγγραφή για τα νέα μας
τώρα!</strong></h3>
                <form method="post">

```

```
        <input type="text" class="field"
value="Email" title="Your Email Here" />
        <input type="submit" class="submit-btn"
value="Αποστολή" />
    </form>
</div>
<div class="col contact">
    <h3><strong>Άμεση Επικοινωνία</strong></h3>
    <h4>+ 123 456 78910</h4>
    <a href="#">skroutz@skroutz.com</a>
</div>
<div class="cl">&nbsp;</div>
</section>
<!-- end of footer-cols -->
<p class="copy">&copy; Skroutz. Design by <a
href="http://chocotemplates.com" target="_blank">Choco</a></p>
<div class="cl">&nbsp;</div>
</div>
</div>
<!-- end of footer -->
</body>
</html>
```


Κώδικας CSS 3

```

* { margin: 0; padding: 0; outline: 0; }

body, html { height: 100%; }

body {
    font-size: 11px;
    line-height: 18px;
    font-family: arial, sans-serif;
    color: #000;
    background: url(images/body.png) repeat 0 0;
    min-width: 940px;
}

a { color: #0252aa; text-decoration: none; cursor: pointer; }
a:hover { text-decoration: underline; }
a img { border: 0; }

article, aside, details, footer, header, menu, nav, section { display: block; }
input, textarea, select { font-size: 12px; font-family: arial, sans-serif; }
textarea { overflow: auto; }

.cl { display: block; height: 0; font-size: 0; line-height: 0; text-indent: -4000px; clear: both; }
.notext { font-size: 0; line-height: 0; text-indent: -4000px; }
.left, .alignleft { float: left; display: inline; }
.right, .alignright { float: right; display: inline; }

.shell { width: 940px; margin: 0 auto; }
/* header */
.header { background: url(images/header.png) repeat-x 0 0; height: 141px; border-bottom: 1px solid #FFF; }
.header .shell { position: relative; }
.rotate img { -moz-transition: all 0.6s ease-in-out; -webkit-transition: all 0.6s ease-in-out; -o-transition: all 0.6s ease-in-out; -ms-transition: all 0.6s ease-in-out; transition: all 0.6s ease-in-out; -moz-border-radius: 5px; border-radius: 5px; }
.rotate img:hover { -moz-transform: rotate(360deg); -webkit-transform: rotate(360deg); -o-transform: rotate(360deg); -ms-transform: rotate(360deg); transform: rotate(360deg); }
.searchbar .shell { position: relative; }

#logo { width: 235px; font-size: 0; line-height: 10; position: absolute; top: 32px; left: 0; }
#logo a { height: 60px; background: url(images/logo.jpg) no-repeat 0 0; display: block; text-indent: -4000px; -moz-border-radius: 22px; border-radius: 22px; }

#navigation { padding-top: 50px; float: right; font-size: 14px; font-weight: bold; text-shadow: rgba(0,0,0,0.5) 0px 1px 1px; padding-bottom: 10px; }

```

```
#navigation ul { list-style: none; list-style-position: outside; }
#navigation ul li { float: left; position: relative; margin: 0 0px 0
35px; }
#navigation li span { position: absolute; top: 25px; white-space:
nowrap; left: 14px; font-size: 11px; color: #fff; text-shadow: none;
font-weight: normal; }
#navigation li.active span { color: #47565d; }
#navigation a { color: #fff; border: 1px solid transparent; padding:
2px 11px 3px 12px; }
#navigation li a: hover,
#navigation li.active a { color: #14647d; text-shadow:
rgba(255,255,255,0.5) 1px 0px 1px; text-decoration: none; border: 1px
solid #36b0d9; border-radius: 5px; -moz-border-radius: 5px; -webkit-
border-radius: 5px; -o-border-radius: 5px; background:
url(images/nav-active.png) repeat-x 0 0; }

.slider-holder { background: url(images/slider-holder.png) repeat-x 0
0; border-bottom: 1px solid #fff; height: 359px; }
.slider-holder .shell { position: relative; top: -12px; }
.slider-holder span.t { background: url(images/slider-top.png) no-
repeat 0 0; position: absolute; top: -14px; left: 17px; width: 905px;
height: 31px; z-index: 100; }
.slider-holder span.b { background: url(images/slider-bottom.png) no-
repeat 0 0; position: absolute; bottom: -1px; left: 16px; width:
907px; height: 17px; z-index: 100; }
.slider-holder span.l { background: url(images/slider-left.png) no-
repeat 0 0; position: absolute; top: -14px; left: -11px; width: 28px;
height: 317px; z-index: 100; }
.slider-holder span.r { background: url(images/slider-right.png) no-
repeat 0 0; position: absolute; top: -14px; right: -7px; width: 25px;
height: 317px; z-index: 100; }
.slider-holder a.slider-btn { width: 237px; height: 67px; padding-top:
17px; background: url(images/slider-btn.png) no-repeat 0 0; text-
shadow: rgba(0,0,0,0.5) 1px 1px 2px; text-align: center; position:
absolute; bottom: -54px; left: 50%; letter-spacing: -1px; margin-
left: -134px; z-index: 120; display: block; color: #fff; font-size:
14px; font-weight: bold; }
.slider-holder a.slider-btn: hover { text-decoration: none; color:
#ddd; }
.slider-holder .shell { position: relative; }

.flexslider { width: 905px; height: 309px; position: relative; left:
16px; padding-top: 10px; }
.flexslider ul.slides { list-style: none; list-style-position:
outside; position: relative; }
.flexslider ul.slides li { float: left; width: 906px; height: 309px;
position: relative; z-index: 20; }
.flexslider ul.slides li img { width: 905px; height: 309px; }

.flexslider .slide-cnt { width: 588px; position: absolute; top: 48px;
right: 43px; }
.flexslider .slide-cnt h2 { font-size: 30px; line-height: 23px; color:
#fff; font-weight: bold; text-shadow: rgba(0,0,0,0.8) 0 1px 2px;
padding-bottom: 18px; }
```

```
.flexslider .slide-cnt p { color: #fff; font-size: 12px; font-weight:
bold; line-height: 25px; }
.flexslider .box-t { background: url(images/slide-cnt-t.png) no-repeat
0 0; height: 62px; font-size: 0; line-height: 0; text-indent: -
4000px; }
.flexslider .box-c { background: url(images/slide-cnt-c.png) repeat-y
0 0; padding: 0px 33px 14px 46px; position: relative; height: 100%; }
.flexslider .box-b { background: url(images/slide-cnt-b.png) no-repeat
0 bottom; height: 15px; width: 588px; font-size: 0; line-height: 0;
text-indent: -4000px; }
.flexslider .box-cnt { float: left; position: relative; margin-top: -
30px; }

.form-wrapper { width: 450px; padding: 8px; margin: 100px auto;
overflow: hidden; border-width: 1px; border-style: solid; border-
color: #dedede #bababa #aaa #bababa; box-shadow: 0 3px 3px
rgba(255,255,255,.1), 0 3px 0 #bbb, 0 4px 0 #aaa, 0 5px 5px #444;
border-radius: 10px; background-color: #f6f6f6; background-image:
linear-gradient(#f6f6f6, #eae8e8);}
.form-wrapper #search { width: 330px; height: 10px; padding: 10px 5px;
float: left; font: bold 16px 'lucida sans', 'trebuchet MS', 'Tahoma';
border: 1px solid #ccc; box-shadow: 0 1px 1px #ddd inset, 0 1px 0
#fff; border-radius: 3px;}
.form-wrapper #search:focus { outline: 0; border-color: #aaa; box-
shadow: 0 1px 1px #bbb inset; }
.form-wrapper #search::-webkit-input-placeholder { color: #999; font-
weight: normal;}
.form-wrapper #search:-moz-placeholder { color: #999; font-weight:
normal;}
.form-wrapper #search:-ms-input-placeholder { color: #999; font-
weight: normal;}
.form-wrapper #submit {float: right; border: 1px solid #00748f;
height: 32px; width: 100px; padding: 0; cursor: pointer; font: bold
15px Arial, Helvetica; color: #fafafa; background-color: #0483a0;
background-image: linear-gradient(#31b2c3, #0483a0); -moz-border-
radius: 3px; -webkit-border-radius: 3px; border-radius: 3px; text-
shadow: 0 1px 0 rgba(0, 0 ,0, .3);box-shadow: 0 1px 0 rgba(255, 255,
255, 0.3) inset, 0 1px 0 #fff;}
.form-wrapper #submit:hover,
.form-wrapper #submit:focus { background-color: #31b2c3; background-
image: linear-gradient( #0483a0, #31b2c3);}
.form-wrapper #submit:active {outline: 0; box-shadow: 0 1px 4px
rgba(0, 0, 0, 0.5) inset;}
.form-wrapper #submit::-moz-focus-inner { border: 0;}

#tech-slideshow { height: 200px; position: relative; overflow:
hidden; }
#tech-slideshow > div { height: 100px; width: 3500px; background:
url(images/slides.jpg); position: absolute; top: 0; left: 0; height:
100%; -moz-transition: opacity 0.5s ease-out; -o-transition: opacity
0.5s ease-out; -webkit-transition: opacity 0.5s ease-out; -ms-
transition: opacity 0.5s ease-out; -webkit-animation: moveSlideshow
60s linear infinite; -moz-animation: moveSlideshow 60s linear
infinite;}

@-webkit-keyframes moveSlideshow {0% { left: 0; } 100% { left: -
1684px; } }
```

```
@-moz-keyframes moveSlideshow { 0% { left: 0; } 100% { left: -1684px;
} }
.main { padding-top: 30px; padding-bottom: 25px; }
.cols { padding-bottom: 15px; position: relative; margin-right: -
22px; padding-left: 14px; width: 948px; }
.cols .col { width: 278px; float: left; padding-right: 38px; }
.main a img { display: block; border: 2px solid #fff; width: 274px;
height: 156px; margin-bottom: 11px; border-radius: 8px; -moz-border-
radius: 8px; -webkit-border-radius: 8px; -o-border-radius: 8px; }
.main h3 { font-size: 20px; line-height: 22px; font-weight: bold;
padding-bottom: 10px; }
.main h2 { font-size: 22px; line-height: 25px; font-weight: bold;
margin:auto; margin-left:auto; margin-right:auto; width:20%; text-
shadow: 3px 3px 3px #707070;}
.main h3 a { color: #000; }
.main p { color: #646458; font-size: 11px; line-height: 22px; padding-
left: 3px; }

/* footer */
#wrapper { min-height: 100%; height: auto !important; height: 100%;
margin: 0 auto -168px; }
#footer, #footer-push { height: 134px; padding-top: 34px; }
#footer { background: url(images/footer.png) repeat-x 0 0; }
#footer h3 { font-weight: normal; font-size: 16px; line-height: 22px;
color: #f1f1f1; padding-bottom: 10px; }

#footer .footer-cols { padding-bottom: 13px; }
#footer .footer-cols .col { width: 295px; float: left; padding-right:
43px; padding-left: 28px; }
#footer .footer-cols p { color: #f1f1f1; padding-bottom: 8px; }
#footer .footer-cols p a { color: #f1f1f1; padding-bottom: 8px; }
#footer .footer-cols form { position: relative; width: 232px; }
#footer .footer-cols form input.field { width: 138px; border: 0;
background-color: #555555; color: #808080; font-size: 12px; height:
14px; padding: 5px 26px 5px 16px; font-size: 12px; border-radius:
10px; -moz-border-radius: 10px; -webkit-border-radius: 10px; -o-
border-radius: 10px; }
#footer .footer-cols form input.submit-btn { width: 65px; height: 25px;
text-align: center; line-height: 25px; border-radius: 10px; -moz-
border-radius: 10px; -webkit-border-radius: 10px; -o-border-radius:
10px;
position: absolute; top: 0; right: 0; box-shadow: 0px 1px 2px 1px
#1c1d1d; -moz-box-shadow: 0px 1px 2px 1px #1c1d1d; -webkit-box-
shadow: 0px 1px 2px 1px #1c1d1d; -o-box-shadow: 0px 1px 2px 1px
#1c1d1d;
border: 0; font-size: 12px; color: #000; font-weight: bold; cursor:
pointer; background: url(images/submit-btn.png) repeat-x 0 0; }
#footer .footer-cols form input.submit-btn:hover { background:
url(images/submit-btn-hover.png) repeat-x 0 0; }

#footer .footer-cols .contact { padding: 0 0 0 0; width: 188px;
padding-left: 5px; }
#footer .footer-cols .contact h3 { padding-bottom: 15px; }
#footer .footer-cols .contact h4 { font-size: 16px; font-weight: bold;
color: #f1f1f1; padding-bottom: 8px; }
#footer .footer-cols .contact a { color: #69c9ef; }
#footer .footer-cols .contact a:hover { color: #f1f1f1; }
```

```
#footer p.copy { text-align: right; padding-bottom: 8px; color:
#818181; font-size: 9px; font-family: tahoma, arial, helvetica,
serif; padding-right: 3px; }
#footer p.copy a { color: #818181; text-decoration: underline; }
#footer p.copy a:hover { color: #f1f1f1; text-decoration: none; }
/* footer */
```