

Τμήμα
Μηχανικών
Πληροφορικής τ.ε.

Τεχνολογικό Εκπαιδευτικό Ίδρυμα
Δυτικής Ελλάδας

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

«HTML5 & CSS3: Ανάπτυξη Εφαρμογής Παγκόσμιου
Ιστού για την Εξ'Αποστάσεως Εκμάθησή τους»

Καραμούζη Αθανασία

Επιβλέπων καθηγητής: Ιωάννης Τζήμας

Αντίρριο – Ιούνιος 2015

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή
Αντίρριο,

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1.

2.

3.

Αφιέρωση

Στους γονείς μου,

με πολλή αγάπη για την στήριξη και την προσευχή τους

για τις σπουδές μου όλα αυτά τα χρόνια.

Ευχαριστίες

Θέλω να ευχαριστήσω ιδιαίτερα τον επιβλέποντα καθηγητή μου κ. Τζήμα Ιωάννη για την άριστη συνεργασία και την πολύτιμη καθοδήγηση του.

Πρόλογος

Αναφερόμενοι σε ένα μεγάλο ποσοστό χρηστών του διαδικτύου συνήθως αντιλαμβανόμαστε ότι παρότι χρησιμοποιούμε το διαδίκτυο πλέον στην καθημερινότητα μας ως βασικό εργαλείο διασκέδασης, εκμάθησης, επικοινωνίας και ενημέρωσης, δεν έχουμε κατανοήσει πλήρως την δομή του και τον τρόπο λειτουργίας του.

Πολλές φορές θαυμάζουμε μια καλαίσθητη σχεδιαστικά ιστοσελίδα όμως δεν γνωρίζουμε τον τρόπο που έχει σχεδιαστεί και λειτουργεί.

Πλήθος χρηστών έχει στραφεί πλέον στον τομέα της σχεδίασης ιστοσελίδων και αφιερώνουν αμέτρητες ώρες στην εκμάθηση της γλώσσας προγραμματισμού Html και των διαδοχικών φύλλων στυλ Css. Ο τομέας αυτός παρέχει νέες θέσεις εργασίες για τους νέους ανθρώπους σε εταιρίες προγραμματισμού λογισμικού, δίνει την δυνατότητα να κατασκευάσει κάποιος την σελίδα που επιθυμεί όπως ακριβώς την έχει φανταστεί ενώ ταυτόχρονα αποτελεί ένα τρόπο διασκέδασης και δημιουργικότητας για τους νέους σχεδιαστές.

Όπως αναφέραμε προηγουμένως οι νέοι σχεδιαστές καταφεύγουν σε κάθε γωνιά του διαδικτύου και σε κάθε σύγγραμμα με σκοπό να αποκομίσουν όσο το δυνατόν περισσότερες γνώσεις για τον προγραμματισμό σε Html. Όμως αυτό πολλές φορές δεν αρκεί γιατί οι πληροφορίες αυτές είναι διασκορπισμένες στο διαδίκτυο χωρίς τάξη και οργάνωση με αποτέλεσμα οι σχεδιαστές να αντιμετωπίζουν δυσκολίες στην εξ' αποστάσεως εκμάθησή τους.

Τις δυσκολίες αυτές καλούμαστε να λύσουμε μέσα από την παρούσα πτυχιακή εργασία δημιουργώντας έναν κατάλληλο ηλεκτρονικό οδηγό – ιστοσελίδα για την εξ' αποστάσεως εκμάθηση της Html5 & Css3, συγκεντρώνοντας και οργανώνοντας τις διασκορπισμένες πληροφορίες έχοντας ως συνέπεια την παραγωγή ενός αξιόλογου και πλήρους εκπαιδευτικού υλικού.

Περίληψη

Στην παρούσα πτυχιακή εργασία παρουσιάζεται μια λίστα από ήδη υπάρχουσες ηλεκτρονικές πλατφόρμες εκμάθησης της γλώσσας Html5 και των διαδοχικών φύλλων στυλ Css3 ενώ παράλληλα αναλύονται τα πλεονεκτήματα και τα μειονεκτήματα τους μέσα απο μια σύντομη και περιεκτική αξιολόγηση.

Το κύριο μέρος όμως της εργασίας είναι η δημιουργία ενός οδηγού εκμάθησης μέσα από τον οποίο καταγράφονται τα βασικά στοιχεία της Html5 και του Css3, παρουσιάζονται οι βασικοί τρόποι δημιουργίας, σύνταξης και σύνδεσης των αντίστοιχων αρχείων μιας ιστοσελίδας ενώ ταυτόχρονα προβάλλεται μέσα από σχόλια και συμβουλές ο καλύτερος τρόπος δημιουργίας γραφικών, μορφοποίησης και οπτικοποίησης μιας ιστοσελίδας.

Τέλος με την βοήθεια του αναλυτικού αυτού έγγραφου οδηγού δημιουργούμε την ηλεκτρονική μορφή του δηλαδή κατασκευάζουμε μια ιστοσελίδα που σκοπό της έχει την εξ' αποστάσεως εκμάθηση για την δημιουργία ενός site. Ο χρήστης της σελίδας έχει την δυνατότητα να μάθει να προγραμματίζει σε γλώσσα Html μέσα από επιμορφωτικά video, πίνακες καταγραφής των βασικών στοιχείων της γλώσσας καθώς και ελεύθερο εκπαιδευτικό υλικό που παρέχεται στην σελίδα μέσω ηλεκτρονικής λήψης.

Λέξεις κλειδιά: Html5, Css3, site, μορφοποίηση, εκμάθηση, σύνταξη, πλατφόρμες.

Abstract

In this thesis a list of existing online learning platforms for HTML5 and CSS3 programming is presented, while the advantages and disadvantages of them are analyzed through a short and comprehensive evaluation.

The core part of the thesis is the creation of a tutorial aiming to analyze the basic elements of HTML5 and CSS3, to present ways for creating, editing and linking the corresponding files of a website and simultaneously to project through reviews and advice tips the best ways to create graphics, format and visualize a website.

Finally, based on the above material a digital version of the tutorial was developed in the form of a web site.

Key words: HTML5, CSS3, website, platforms, learning, training videos, tables, programming

Περιεχόμενα

Περιεχόμενα	8
1 Λίστα Πινάκων	11
2 Λίστα Εικόνων.....	11
1 Βασικές έννοιες της Html5 και Css3	12
2.1 Νέα χαρακτηριστικά της Html5	14
3 Διαδοχικά φύλλα στυλ Css3.....	16
3.1 Διαφορές ανάμεσα σε Html και Css	16
3.2 Εφαρμογή των Css σε Html έγγραφο	17
3.3 Πλεονεκτήματα και λόγοι που χρησιμοποιούμε Css	20
4 Αναφορά και κριτική στις κύριες εκπαιδευτικές πλατφόρμες	20
4.1 W3schools.com.....	20
4.2 Code School.....	22
4.3 Channel 9.....	23
4.4 The code player.....	24
4.5 WebDirectionsSouth	25
4.6 Html5 / Css3 Box.com	26
4.7 Tuts+	27
4.8 Hongkiat.com	29
4.9 Wlearn	30
4.10 Computerman.gr	31
4.11 StudentGuru.gr.....	32
5 Βασική δομή και σύνταξη ενός αρχείου Html5	33
5.1 Βασικά στοιχεία – tags του κώδικα Html	34
5.1.1 Το tag <html>	34
5.1.2 Τα tags <head>, <title>	35
5.1.3 Το tag <body>	35
5.2 Τα tags μορφοποίησης	36
6 Δημιουργία πίνακα / λίστας και στοιχεία ομαδοποίησης.....	39
6.1 Δημιουργία πίνακα.....	39
6.2 Δημιουργία λίστας	44
6.2.1 Τα tags της αριθμημένης λίστας.....	44
6.2.2 Τα tags της ταξινομημένης λίστας με κουκίδες	46

6.2.3	Τα tags της ταξινομημένης λίστας με επεξήγηση	48
6.2.4	Δημιουργία εμφωλευμένης λίστας	49
6.3	Δημιουργία στοιχείων ομαδοποίησης	50
6.3.1	Τα στοιχεία block.....	50
6.3.2	Τα στοιχεία inline	50
6.3.3	Το block tag <div>.....	52
6.3.4	Το inline tag 	54
7	Εισαγωγή εικόνας, υπερσυνδέσμου και πολυμέσων (media)	54
7.1	Εισαγωγή και μορφοποίηση εικόνων.....	55
7.1.1	Το attribute alt	55
7.1.2	Μορφοποίηση μεγέθους της εικόνας.....	56
7.1.3	Οι Screen Readers.....	57
7.1.4	Στοιχίση εικόνας έναντι κειμένου (ιδιότητα float)	58
7.1.5	Εικονοχάρτες (image maps)	59
7.2	Εισαγωγή υπερσυνδέσμων (links)	60
7.2.1	Μορφοποίηση υπερσυνδέσμων	61
7.2.2	Το attribute target.....	63
7.2.3	Χρήση εικόνας ως υπερσύνδεσμος.....	64
7.2.4	Δημιουργία σελιδοδείκτη (bookmark)	65
7.3	Εισαγωγή πολυμέσων (media)	66
7.3.1	Εισαγωγή ήχου (audio)	67
7.3.2	Εισαγωγή video	68
7.3.3	Εισαγωγή video από Youtube	68
8	Εισαγωγή αντικειμένων (object)	75
9	Δημιουργία κλάσεων στην Html (classes)	76
10	Δημιουργία υποβάθρου – δομής της σελίδας (Layout).....	78
10.1	Ανάλυση πηγαίου κώδικα του layout	82
10.1.1	Δημιουργία άρθρου (<article>)	88
11	Καθιέρωση των Css3.....	90
11.1	Σύνταξη ενός αρχείου Css.....	90
11.1.1	Η ιδιότητα id του selector	92
11.1.2	Οι κλάσεις στο αρχείο Css	93
11.1.3	Ομαδοποίηση των selectors.....	95
12	Μορφοποίηση των βασικών στοιχείων μέσω του Css	95
12.1	Μορφοποίηση υποβάθρου σελίδας / στοιχείου (background)	96

12.1.1	Η ιδιότητα background- color	96
12.1.2	Η ιδιότητα background-image	96
12.1.3	Η ιδιότητα background-size	98
12.1.4	Η ιδιότητα background-origin	100
12.1.5	Η ιδιότητα background-clip.....	101
12.2	Μορφοποίηση κειμένου και εισαγωγή νέας γραμματοσειράς	102
12.2.1	Αλλαγή και εισαγωγή νέων ειδών γραμματοσειράς.....	104
13	Μορφοποίηση πίνακα , λίστας μέσω του Css.....	108
13.1	Μορφοποίηση πίνακα	108
13.1.1	Μορφοποίηση περιγράμματος ενός πίνακα (border).....	109
13.1.2	Οι ιδιότητες padding και margin	110
13.1.3	Μορφοποίηση ύψους / πλάτους του πίνακα.....	111
13.2	Μορφοποίηση λίστας.....	111
13.2.1	Μετατροπή λίστας σε μενού πλοήγησης	112
13.2.2	Μορφοποίηση ταξινόμησης απλής λίστας.....	118
14	Οι ιδιότητες – attributes position, display και float.....	119
14.1	Το attribute position	119
14.2	Το attribute display	120
14.3	Το attribute float	121
15	Δημιουργία animation – Κανόνας @keyframes	121
16	Σύνδεση αρχείου Css με Html	125
17	Παράρτημα – Πίνακες	126
	Πίνακας 1: Τα νέα attributes της Html5 σε σχέση με την Html4	126
	Πίνακας 2: Χρωματολόγιο	129
18	Βιβλιογραφία.....	134

1 Λίστα Πινάκων

Πίνακας 1: Τα νέα attributes της Html5 σε σχέση με την Html4

Πίνακας 2: Χρωματολόγιο

2 Λίστα Εικόνων

Εικόνα 1: Η λειτουργία ενός web browser.

Εικόνα 2: Προβολή της γνωστής εκπαιδευτικής πλατφόρμας ηλεκτρονικής εκμάθησης
W3schools.com.

Εικόνα 3: Προβολή της πλατφόρμας ηλεκτρονικής εκμάθησης CodeSchool.

Εικόνα 4: Προβολή της πλατφόρμας ηλεκτρονικής εκμάθησης Channel 9.

Εικόνα 5: Προβολή της πλατφόρμας ηλεκτρονικής εκμάθησης TheCodePlayer.

Εικόνα 6: Προβολή της πλατφόρμας ηλεκτρονικής εκμάθησης WebdirectionsSouth.

Εικόνα 7: Προβολή της πλατφόρμας ηλεκτρονικής εκμάθησης HTML5/CSS3 Box.com.

Εικόνα 8: Προβολή της πλατφόρμας ηλεκτρονικής εκμάθησης tuts+.

Εικόνα 9: Προβολή της πλατφόρμας ηλεκτρονικής εκμάθησης HONGKIAT.COM.

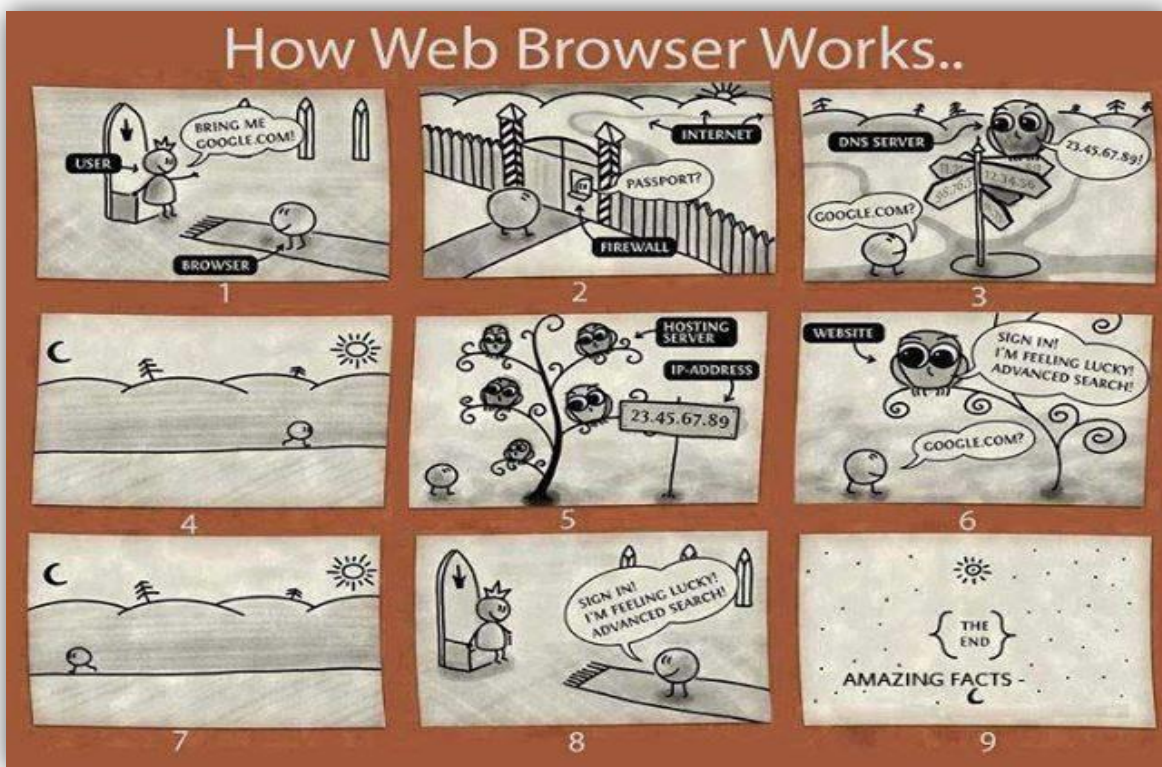
Εικόνα 10: Προβολή της πλατφόρμας ηλεκτρονικής εκμάθησης Wlearn.gr.

Εικόνα 11: Προβολή της πλατφόρμας ηλεκτρονικής εκμάθησης COMPUTERMAN.

1 Βασικές έννοιες της Html5 και Css3

Το ακρόνυμο HTML σημαίνει HyperText Markup Language και αποτελεί μια από τις σημαντικότερες γλώσσες σήμανσης (markup language) στο διαδίκτυο. Είναι ο μοναδικός τρόπος που υπάρχει για την δημιουργία ιστοσελίδων και είναι ένα είδος εγγράφου το οποίο προορίζεται για web browsers.

Η HTML δίνει την δυνατότητα της μετάδοσης της πληροφορίας από υπολογιστή σε υπολογιστή, με μεγάλη αποδοτικότητα. Όλες οι άλλες τεχνολογίες που υπάρχουν, όπως PHP, ASP, κτλ, μας δίνουν την δυνατότητα δημιουργίας δυναμικών εφαρμογών διαδικτύου, παρόλα αυτά μετά την επεξεργασία από τον web server το τελικό αποτέλεσμα που μεταδίδεται είναι HTML.



Εικόνα 1: Η λειτουργία ενός web browser.

Το web άλλαξε δραματικά από το 1999 όταν έγινε διαθέσιμη στο κοινό η προηγούμενη αναθεώρηση του προτύπου της html, η html 4.01.

Από τότε δημιουργήθηκαν νέες συνήθειες και τάσεις των χρηστών του web, όπως η ραγδαία αύξηση του αριθμού των οπτικοακουστικών μέσων (βίντεο και audio) που χρησιμοποιούνται στις ιστοσελίδες, η δημοσίευση κειμένων από εξωτερικά blogs και φόρουμ (content sharing), αλλά και η ανάγκη να γράφουν περισσότερες πληροφορίες για το περιεχόμενο του site τους ώστε να αυξάνουν τις πιθανότητες εμφάνισης του στις μηχανές αναζήτησης.

Όλες αυτές οι νέες τάσεις και κυρίως η όλο και αυξανόμενη ταχύτητα πρόσβασης στο Internet, δημιούργησε την ανάγκη ενός ευκολότερου τρόπου εισαγωγής τεχνολογιών στις ιστοσελίδες, χωρίς να χρειάζεται η ανάπτυξη μεγάλων κομματιών κώδικα JavaScript για την υλοποίησή τους. Οι νέες τάσεις χρήσης του web και η ανάγκες που οι ίδιες επέβαλαν, έγιναν αιτία να αναπτυχθεί ένα νέο πρότυπο της html, αυτό της html5. Οι συντάκτες της HTML5 είναι ο Ίαν Χίκσον της εταιρίας Google και ο Ντέιβ Χιάτ της εταιρίας Apple.

Η html5 είναι το νέο standard πρότυπο για την html, την Xhtml και την html DOM. Η ανάπτυξη της html5 έγινε με την συνεργασία της World Wide Web Consortium_(W3C) και της Web Hypertext Application Technology Working Group (WHATWG).

Η WHATWG εργαζόταν επάνω στις web φόρμες και τις web εφαρμογές, ενώ η W3C, η οποία δημιούργησε και διαχειρίζεται τα πρότυπα της html και της Xhtml, ασχολήθηκε με την ανάπτυξη του νέου προτύπου Xhtml 2.0. Το 2006 αποφάσισαν να συνεργαστούν για να δημιουργήσουν το νέο πρότυπο της html5.

Η html5 αναπτύχθηκε με βάση τους παρακάτω κανόνες:

1. νέα χαρακτηριστικά έπρεπε να προστεθούν στην δομή των HTML, CSS και JavaScript
2. μείωση των περιπτώσεων που χρειάζεται η εγκατάσταση plugins στον browser για κάποιου συγκεκριμένου τύπου στοιχεία (όπως βίντεο και audio)
3. καλύτερη διαχείριση σφαλμάτων
4. προσθήκη περισσότερων ετικετών οι οποίες θα αντικαταστήσουν κομμάτια κώδικα JavaScript που χρησιμοποιούσαν συχνά οι web designers
5. το νέο πρότυπο θα έπρεπε να είναι αυτόνομο χωρίς να χρειάζεται να καλεί κομμάτια κώδικα από άλλα πρότυπα

6. τα βήματα του σχεδιασμού και της ανάπτυξης του νέου προτύπου θα έπρεπε να είναι ορατά στο κοινό.

2.1 Νέα χαρακτηριστικά της Html5

Μερικά από τα νέα χαρακτηριστικά του νέου προτύπου είναι τα παρακάτω:

1. δυνατότητα σχεδιασμού γραφικών με χρήση JavaScript (νέα ετικέτα canvas)
2. αναπαραγωγή βίντεο και audio χωρίς να χρειάζεται η εγκατάσταση plugins (νέες ετικέτες video και audio)
3. προσθήκη νέων ετικετών που κάνουν την δημιουργία και την διαχείριση των ιστοσελίδων, ακόμη πιο εύκολη (νέες ετικέτες article, footer, header κτλ.)
4. νέα στοιχεία στις HTML φόρμες (calendar, date, time, search κτλ.). Πολλά νέα elements για καλύτερο και πιο semantic structure, όπως header, footer, nav, section, article, aside, dialog κτλ.
5. Πολλά βελτιωμένα elements για embedded content, όπως figure, audio και video. Στα δύο τελευταία δίνεται και η δυνατότητα ενσωμάτωσης user interface μέσω της markup αλλά και ενός API, έτσι ώστε ο χρήστης να μπορεί να ελέγχει το video/audio file. Το user interface θα δημιουργείται από την rendering engine του εκάστοτε user agent
6. time element, που θα αντιπροσωπεύει χρόνο (ημερομηνία, ώρα, κτλ.)
7. canvas element, για on-line rendering γραφικών
8. Templates elements όπως datatemplate, rule, and nest. Με αυτόν τον τρόπο η ίδια η markup αποκτά templating δυνατότητες
9. Form validation μέσω της markup (σε client-side επίπεδο φυσικά). Έτσι στα input elements, δηλώνοντας απλά τον τύπο τους (type) στην markup (όπως πχ. date, number, email, url, κτλ) θα αναγκάζεται ο user agent του browser να αναλαμβάνει το client-side validation

Μαζί με τα νέα χαρακτηριστικά, η καινούργια markup αφήνει πίσω της πολλά άχρηστα elements και attributes τα οποία είχαν παρατήρει εδώ και καιρό οι έμπειροι developers, όπως :

- font, αφού όλο το styling θα γίνεται αναγκαστικά μέσω CSS
- frame, frameset και noframes
- applet
- Όλες τις presentational attributes όπως : align, background, bgcolor, cellpadding, cellspacing, border, valign κτλ. Εάν κάποιος θέλει να τις χρησιμοποιήσει, θα πρέπει και πάλι να στραφεί στην CSS.

Τέλος αξίζει να σημειωθεί πως πάνω στην καινούργια markup θα χτιστούν και πολλά API για την ευκολότερη υλοποίηση web εφαρμογών, όπως :

1. 2D drawing API το οποίο θα χρησιμοποιείται για on-line rendering γραφικών μέσω του νέου canvas element
2. API για την δημιουργία interface στοιχείων (όπως play, pause, κτλ.) στα video και audio elements
3. Δυνατότητα client-side αποθήκευσης δεδομένων.
4. API για δημιουργία offline web εφαρμογών. Έτσι ακόμα και εάν δεν υπάρχει internet access, η web εφαρμογή θα συνεχίσει να λειτουργεί κανονικά. Σε συνδυασμό με το χαρακτηριστικό μιας client-side database καταλαβαίνουμε όλοι πως η βελτίωση είναι παραπάνω από “εμφανής”.
5. Drag and drop API, το οποίο θα μπορεί να γίνει attached σε σχεδόν οποιοδήποτε element μέσω της νέας draggable attribute
6. Editing API, το οποίο θα λειτουργεί με τα elements που έχουν την νέα contenteditable attribute κάτι το οποίο βελτιώνει αυτόματα και λειτουργία των Wiki-like και γενικότερα των social-based sites.
7. Network API
8. Διάφορα άλλα API μικρότερης σημασίας ή πιο εξειδικευμένων λειτουργιών

3 Διαδοχικά φύλλα στυλ Css3

Η CSS (Cascading Style Sheets-Διαδοχικά Φύλλα Στυλ ή Αλληλουχία Φύλλων Στύλ) 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. Πιο πρακτικά χρησιμοποιείται 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 ιστότοπου.

Η CSS είναι μια γλώσσα του υπολογιστή με σκοπό να αναπτύξει στυλιστικά μια ιστοσελίδα, δηλαδή να διαμορφώσει περισσότερα χαρακτηριστικά, χρώματα, στοίχιση καθώς δίνει πάρα πολλές δυνατότητες σε σχέση με την html. Για μια όμορφη και καλοσχεδιασμένη ιστοσελίδα η γλώσσα CSS είναι απαραίτητη. Τα Διαδοχικά Φύλλα Στυλ (CSS, Cascading Style Sheets) αποτελούν ένα πολύ καλό εργαλείο για να μπορούμε να αλλάζουμε την εμφάνιση και τη διάταξη (layout) των ιστοσελίδων μας. Μπορούν να μας γλυτώσουν από πολύ χρόνο και κόπο και μας δίνουν τη δυνατότητα να σχεδιάζουμε τις ιστοσελίδες μας με μια εντελώς καινούργια φιλοσοφία. Η κατανόηση των CSS απαιτεί να υπάρχει κάποια βασική εμπειρία με την HTML.

Για να δουλέψουμε με τα CSS δεν χρειάζεται να χρησιμοποιήσουμε κάποιο πρόγραμμα δημιουργίας ιστοσελίδων, όπως είναι τα γνωστά FrontPage, DreamWeaver ή και το Word, καθώς θα σταθούν εμπόδιο στην κατανόηση των CSS. Αυτό που χρειαζόμαστε είναι ένας απλός επεξεργαστής κειμένου, όπως είναι το Σημειωματάριο (Notepad) των Windows.

Μπορούμε να χρησιμοποιήσουμε όποιον φυλλομετρητή (browser) επιθυμούμε για να βλέπουμε πώς θα εμφανίζονται οι ιστοσελίδες που θα δημιουργούμε. Συνεπώς αυτό που χρειαζόμαστε είναι ένας φυλλομετρητής (browser) και ένας απλός επεξεργαστής κειμένου (text editor).

3.1 Διαφορές ανάμεσα σε Html και Css

Στο ερώτημα ποια είναι η διαφορά ανάμεσα στα CSS και την HTML, μπορούμε να πούμε ότι η HTML χρησιμοποιείται για να δομήσει το περιεχόμενο (content), ενώ τα CSS χρησιμοποιούνται για τη διαμόρφωση ή μορφοποίηση (formatting) του δομημένου περιεχομένου. Από το ξεκίνημά της, η HTML χρησιμοποιείτο μόνο για να μπορούμε να προσθέτουμε δομή στο κείμενο, όπως για να επισημαίνουμε ποια είναι κεφαλίδα (headline) ή ποια είναι παράγραφος (paragraph) με τα

γνωστά tags (ετικέτες) της HTML, όπως είναι τα <h1> και <p>. Όμως, καθώς η δημοτικότητα του Web αύξανε συνεχώς, οι σχεδιαστές των ιστοσελίδων άρχισαν να ψάχνουν για επιπλέον δυνατότητες, όπως προσθήκη διάταξης (layout) στα έγγραφα.

Για να γίνει αυτό, οι φυλλομετρητές επινόησαν καινούργια HTML tags, όπως για παράδειγμα το , τα οποία διέφεραν από τα αρχικά HTML tags καθώς όριζαν τη διάταξη και όχι τη δομή μιας ιστοσελίδας.

Στο ερώτημα τι έχουν να μας προσφέρουν τα CSS, μπορούμε να πούμε ότι τα CSS αποτέλεσαν μια επανάσταση στον κόσμο του Web design και τα μεγάλα πλεονεκτήματά τους είναι τα εξής :

- Δυνατότητα για τον ταυτόχρονο έλεγχο της διάταξης (layout) πολλών εγγράφων από ένα μόνο φύλλο στυλ (style sheet).
- Πιο ακριβής έλεγχος της διάταξης (layout) των ιστοσελίδων.
- Εφαρμογή διαφορετικής διάταξης σε διαφορετικές εξόδους, όπως screen, print κ.ά.
- Αναρίθμητες προηγμένες και εξεζητημένες τεχνικές.

3.2 Εφαρμογή των Css σε Html έγγραφο

Υπάρχουν τρεις τρόποι που μπορούμε να χρησιμοποιήσουμε για να εφαρμόσουμε ένα CSS σ' ένα HTML έγγραφο.

- **Μέθοδος 1 : In-line (attribute style).** Ο πρώτος τρόπος για να εφαρμόσουμε ένα CSS σ' ένα HTML κείμενο είναι να χρησιμοποιήσουμε την ιδιότητα (attribute) style της HTML. Για παράδειγμα το κόκκινο χρώμα φόντου μπορεί να εφαρμοσθεί ως εξής:

```
<html><head>  
  
<title>Παράδειγμα</title>  
  
</head>  
  
<body style="background-color: #FF0000;">  
  
<p>Αυτή είναι μια κόκκινη σελίδα</p>  
  
</body></html>
```

- **Μέθοδος 2 : Εσωτερικό-Internal (tag style)**

Ένας άλλος τρόπος είναι να γράψουμε τον κώδικα για τα CSS με το HTML tag <style>, ως εξής:

```
<html><head>

<title>Παράδειγμα</title>

<style type="text/css">

body {background-color: #FF0000;}

</style></head>

<body>

<p>Αυτή είναι μια κόκκινη σελίδα</p>

</body></html>
```

- **Μέθοδος 3 : Εξωτερικό-External (link to a style sheet)**

Ο τρίτος τρόπος, που είναι αυτός που συνιστάται να χρησιμοποιούμε, είναι ένας σύνδεσμος (link) προς ένα εξωτερικό αρχείο φύλλου στυλ (external stylesheet). Αυτή η μέθοδος θα χρησιμοποιηθεί σ' όλα τα παραδείγματα που ακολουθούν.

Ένα εξωτερικό αρχείο φύλλου στυλ (external style sheet) είναι απλά ένα αρχείο κειμένου (text file) που έχει επέκταση (extension).css. Μπορούμε να το τοποθετήσουμε (upload) στον Web server, όπως όλα τα άλλα αρχεία.

Για παράδειγμα, ας υποθέσουμε ότι το αρχείο μας style sheet έχει όνομα style.css και βρίσκεται στον φάκελο style. Για να δημιουργήσουμε έναν σύνδεσμο (link) από το HTML έγγραφο, που είναι το default.htm, προς το αρχείο του style sheet, που είναι το style.css, θα πρέπει να γράψουμε τον εξής κώδικα :

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

Η διαδρομή (path) προς το style sheet προσδιορίζεται με την ιδιότητα (attribute) href. Η παραπάνω γραμμή κώδικα πρέπει να τοποθετηθεί στο τμήμα header του HTML κώδικα, δηλ. ανάμεσα στα tags <head> και </head>, ως εξής :

```
<html> <head>
```

```
<title>Κύρια Ιστοσελίδα</title>
```

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

Ο παραπάνω σύνδεσμος (link) λέει στον φυλλομετρητή ότι θα πρέπει να χρησιμοποιήσει τη διάταξη (layout) που υπάρχει στο συγκεκριμένο CSS αρχείο όταν πρόκειται να εμφανίσει το περιεχόμενο του τρέχοντος HTML αρχείου.

Αυτό που είναι πολύ καλό με τη μέθοδο αυτή είναι ότι μπορούμε να συνδέσουμε πολλά HTML αρχεία με το ίδιο αρχείο style sheet. Μ' άλλα λόγια, ένα αρχείο CSS μπορεί να χρησιμοποιηθεί για να μπορούμε να ελέγχουμε το layout πολλών HTML εγγράφων ταυτόχρονα.

Για να το δούμε αυτό και στην πράξη, μπορούμε να δημιουργήσουμε δύο αρχεία, ένα HTML και ένα CSS, με το εξής περιεχόμενο :

default.htm

```
<html> <head>
```

```
<title>Κύρια Ιστοσελίδα</title>
```

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

```
</head> <body>
```

```
<h1>Το Πρώτο μας Φύλλο Στυλ</h1>
```

```
</body> </html>
```

style.css

```
body { background-color: #FF0000;}
```

Θα πρέπει να δημιουργήσουμε τα αρχεία και να τα αποθηκεύσουμε με τις σωστές επεκτάσεις .htm και .css και φυσικά να τα τοποθετήσουμε και στον ίδιο φάκελο.

3.3 Πλεονεκτήματα και λόγοι που χρησιμοποιούμε Css

Τα βασικά πλεονεκτήματα των διαδοχικών φύλλων στυλ θα μπορούσαν να καταγραφούν ως εξής:

1. Ταχύτητα
2. Καλύτερη διαχείριση του κώδικα
3. Προσβασιμότητα
4. Πιο γρήγορο redesign στην ιστοσελίδα μας

4 Αναφορά και κριτική στις κύριες εκπαιδευτικές πλατφόρμες

Στο κεφάλαιο αυτό παραθέτουμε τις κυριότερες πλατφόρμες εκπαιδευτικού υλικού που υπάρχουν στο διαδίκτυο , καθώς και τα πλεονεκτήματα και μειονεκτήματα τους όσον αφορά την αποτελεσματικότητα τους στην εξ' αποστάσεως εκμάθηση των χρηστών.

4.1 W3schools.com

The logo for W3schools.com, featuring the text 'W3schools.com' in a bold, sans-serif font. The 'W3' is in black, 'schools' is in black, and '.com' is in green.

Ίσως μπορεί να οριστεί ως η καλύτερη πλατφόρμα σε εκπαιδευτικό επίπεδο αφού περιλαμβάνει πίνακες καταγραφής για την επεξήγηση της χρήσης και της σύνταξης των εντολών και των μεθόδων που χρησιμοποιούνται στην γλώσσα html5 μέσα από παραδείγματα και κομμάτια έτοιμου κώδικα.

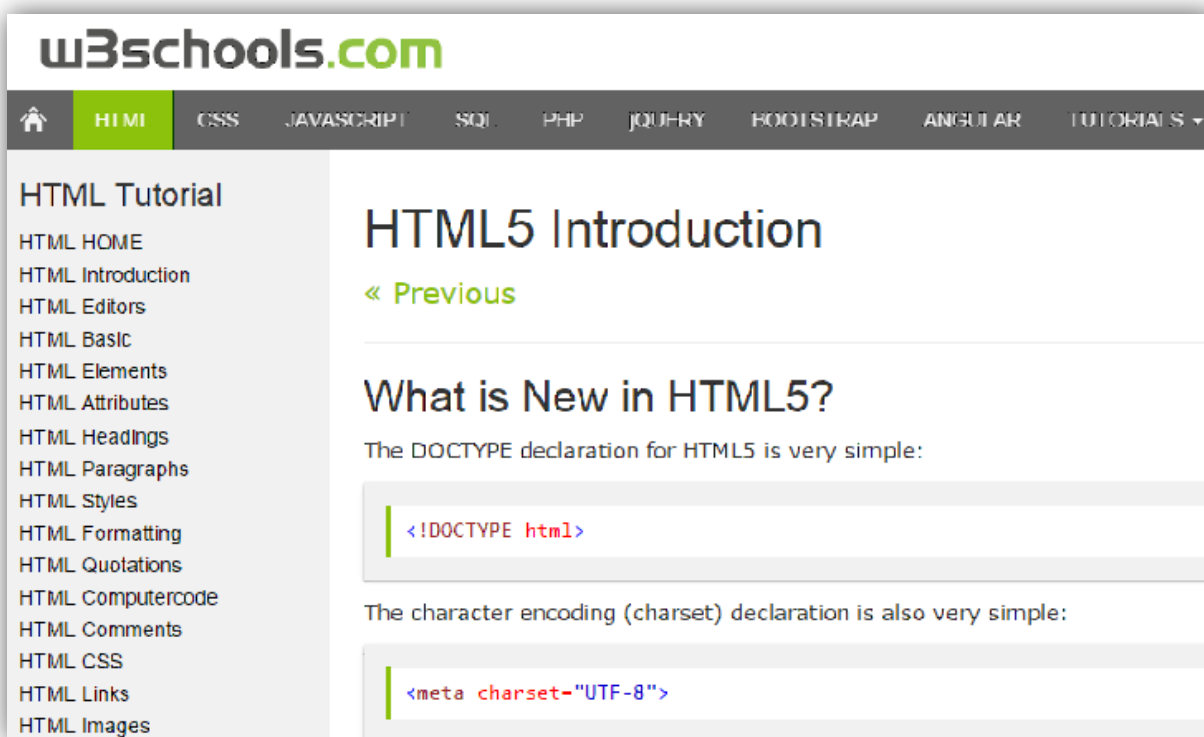
Βασικό πλεονέκτημα της συγκεκριμένης πλατφόρμας αποτελεί και η παροχή ειδικών quiz – ερωτήσεις γνώσεων οι οποίες παρέχονται όμως μετά από καταβολή συγκεκριμένου χρηματικού ποσού. Επίσης παρέχεται χρωματικός πίνακας με την αντίστοιχη ονομασία και κωδικοποίηση των χρωμάτων που μπορεί να χρησιμοποιήσει ένας προγραμματιστής ιστοσελίδων στον τομέα της σχεδίασης γραφικών της ιστοσελίδας.

Παράλληλα σαν προσόν της πλατφόρμας μπορεί να χαρακτηριστεί και η συμπληρωματική καταγραφή των φυλλομετρητών (browsers) που υποστηρίζουν κάθε φορά τις διαφορετικές που χρησιμοποιούνται από τον προγραμματιστή.

Ωστόσο υπάρχουν και μειονεκτήματα στην συγκεκριμένη πλατφόρμα που, αν θέλουμε να είμαστε αντικειμενικοί ως προς την κριτική της ιστοσελίδας, θα πρέπει να αναφερθούν.

Η έλλειψη επιμορφωτικών βίντεο και προσομοίωσης των παραδειγμάτων που υπάρχουν στην ιστοσελίδα και που σκοπός τους είναι η ανάδειξη της χρήσης των διάφορων εντολών ή έτοιμου κώδικα μέσα από τις εκάστοτε αλλαγές που θα πραγματοποιήσει ο χρήστης στις τιμές των αντίστοιχων tags και των εντολών – μεθόδων καθώς και η έλλειψη φόρμας σχολιασμού από τους χρήστες της ιστοσελίδας μπορούν να χαρακτηριστούν ως βασικά μειονεκτήματα.

Τέλος θα πρέπει να αναφέρουμε ότι ορισμένα πιο εξειδικευμένα άρθρα και παράγραφοι επιτρέπουν στον χρήστη να χρησιμοποιήσει το περιεχόμενο τους μόνο μετά την καταβολή του αντίστοιχου χρηματικού ποσού κάτι το οποίο δεν διευκολύνει τον χρήστη της πλατφόρμας για ευνόητους λόγους.



Εικόνα 2: Προβολή της γνωστής εκπαιδευτικής πλατφόρμας ηλεκτρονικής εκμάθησης W3schools.com.

4.2 Code School



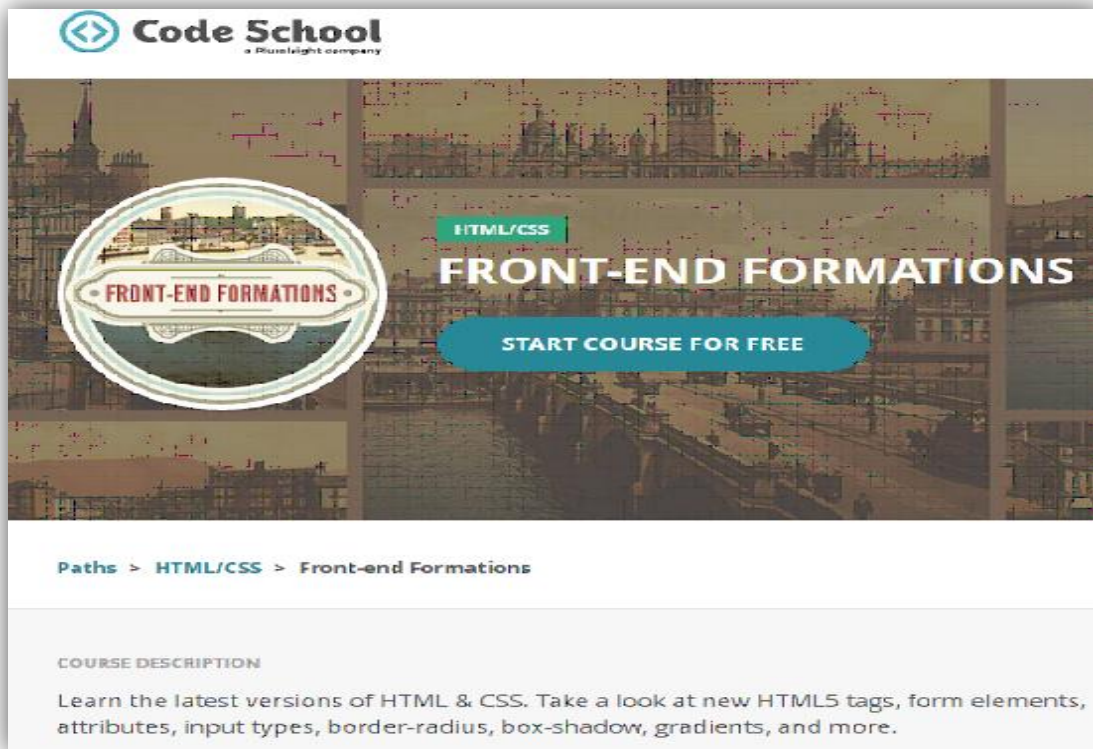
Η ιστοσελίδα Code School αποτελείται κυρίως από μαθήματα τα οποία είναι ελεύθερα προς τον χρήστη μέχρι ένα σημείο αφού για ανανεώσεις των μαθημάτων (updates) και για πιο εξειδικευμένες πληροφορίες όπως νέες τεχνολογίες στον προγραμματισμό ιστοσελίδας ζητείται η καταβολή του ανάλογου χρηματικού ποσού στο οποίο η μόνη διευκόλυνση στον χρήστη είναι η παροχή και η δυνατότητα έκπτωσης. Τα μαθήματα χωρίζονται σε κατηγορίες ανάλογα με το επίπεδο του προγραμματιστή και γενικότερα του χρήστη της ιστοσελίδας.

Βασικό πλεονέκτημα που έρχεται σε αντίθεση με την πλατφόρμα W3schools.com αποτελεί η ύπαρξη επιμορφωτικών video στα οποία μετά το πέρας τους υπάρχει ηλεκτρονική φόρμα προγραμματισμού στην οποία παρέχεται η δυνατότητα στον χρήστη να εφαρμόσει ότι παρακολούθησε στα video και έπειτα να δει άμεσα το αποτέλεσμα τους στο browser χωρίς να χρειαστεί να αποθηκεύσει το αρχείο του πηγαίου κώδικα στον υπολογιστή του.

Παρέχεται επίσης η δυνατότητα re-watch στα video κατά την διάρκεια της γραφής του κώδικα αλλά και η δυνατότητα λήψης των video και διαφανειών στον υπολογιστή του χρήστη.

Παρόλα αυτά τα πλεονεκτήματα που παρουσιάζει η ιστοσελίδα παρατηρείται έλλειψη ενός ολοκληρωμένου πίνακα καταγραφής της χρήσης και της επεξήγησης των εντολών και των μεθόδων που απαιτούνται για την συγγραφή του κώδικα.

Οι εντολές και οι μέθοδοι της html5 παρέχονται στην ηλεκτρονική φόρμα προγραμματισμού χωρίς όμως επεξήγηση και παρουσίαση της κύριας μορφής σύνταξης τους ενώ παράλληλα παρατηρείται η έλλειψη της αντίστοιχης σελίδας για τα ερωτήματα των χρηστών αλλά και τα σχόλια τους αφού το μόνο που υπάρχει είναι μια απλή υποτυπώδης φόρμα σχολίων στο κάτω μέρος της σελίδας μαθημάτων.



Εικόνα 3: Προβολή της πλατφόρμας ηλεκτρονικής εκμάθησης CodeSchool.

4.3 Channel 9

Channel 9

Μια πλατφόρμα η οποία ως περιεχόμενο της περιλαμβάνει σε ένα γενικότερο σύνολο την κατασκευή λογισμικού και ιστοσελίδων. Η ιστοσελίδα παρέχει μια σειρά από επιμορφωτικά 30λεπτα βίντεο στα οποία προβάλλεται ένας “ηλεκτρονικός” καθηγητής ο οποίος σκοπό του έχει να εξηγήσει στο χρήστη τα στάδια κατασκευής μιας ιστοσελίδας, ουσιαστικά δηλαδή αυτά τα βίντεο είναι ηλεκτρονικά μαθήματα για έναν αρχάριο αλλά και έναν πιο προχωρημένο σε επίπεδο γνώσης προγραμματιστή.

Παρατηρούμε την ύπαρξη αντίστοιχης φόρμας σχολιασμού ακριβώς από κάτω από τα βίντεο καθώς και συγκεκριμένης σελίδας που παραπέμπει σε forums στα οποία ένας χρήστης έχει την δυνατότητα να υποβάλλει ένα ερώτημα στο οποίο θα λάβει απάντηση είτε από άλλους χρήστες ή και από τους ίδιους τους καθηγητές / διαχειριστές της ιστοσελίδας.



Εικόνα 4: Προβολή της πλατφόρμας ηλεκτρονικής εκμάθησης Channel 9.

Στην εικόνα διακρίνεται η ανακοίνωση της ιστοσελίδας πως αποτελεί περιεχόμενο της ψηφιακής Ακαδημίας της Microsoft.

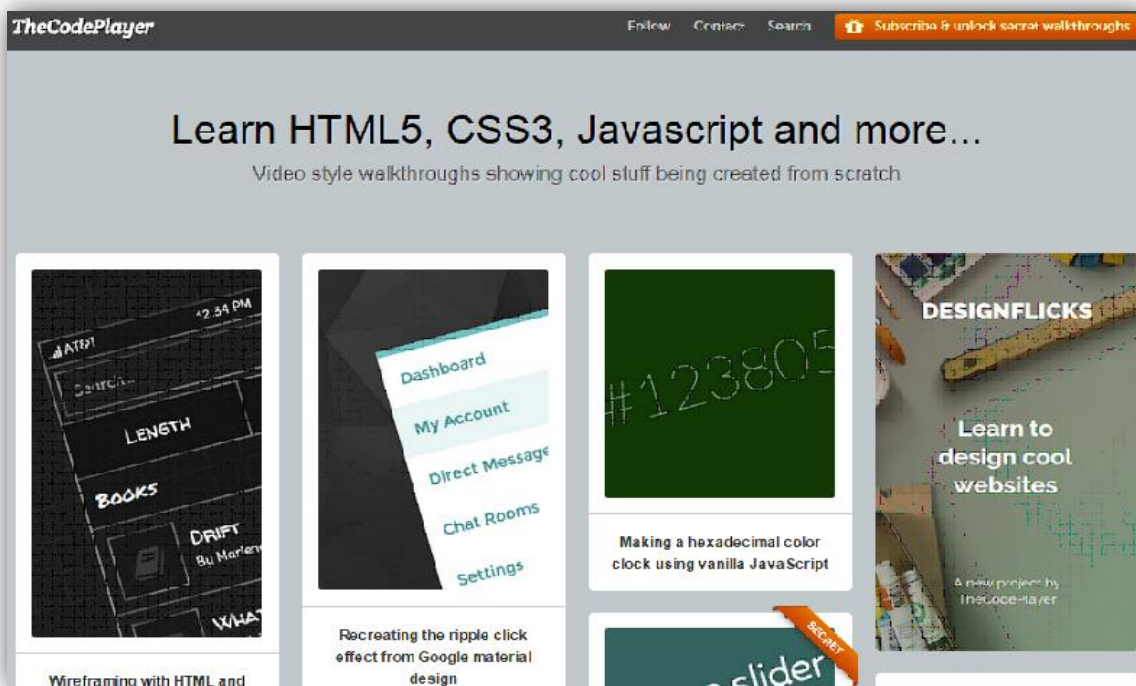
Ένα επιπλέον στοιχείο για την συγκεκριμένη ιστοσελίδα είναι και η παροχή ενημέρωσης για διάφορα events που μπορεί να πραγματοποιηθούν ανά τον κόσμο και στα οποία μπορούν αν θέλουν οι χρήστες να δηλώσουν συμμετοχή ή να τα παρακολουθήσουν αν έχουν την δυνατότητα. Τέλος πρέπει να αναφερθεί ότι η συγκεκριμένη σελίδα αποτελεί περιεχόμενο της ψηφιακής Ακαδημίας της Microsoft , κάτι το οποίο αναφέρεται ρητώς σε ανακοίνωση της ιστοσελίδας (βλέπε Εικόνα 3.3).

4.4 The code player

TheCodePlayer

Πλατφόρμα η οποία από την μορφή της μας θυμίζει τις διάσημες ιστοσελίδες Pinterest και Tumblr. Είναι μια σελίδα που όπως λέει και η ίδια στην επικεφαλίδα της θα βοηθήσει τον χρήστη να εκπαιδευτεί στις γλώσσες html5/css3/java script κ.α..

Εδώ διακρίνουμε την ύπαρξη ηλεκτρονικής φόρμας στην οποία μέσω εικονικής παρουσίασης παραδειγμάτων στην κατασκευή μιας ιστοσελίδας παρέχεται η δυνατότητα στον χρήστη να δει τον κώδικα που έχει χρησιμοποιηθεί για το κάθε παράδειγμα σε όλες τις γλώσσες html5/css3/javascript.



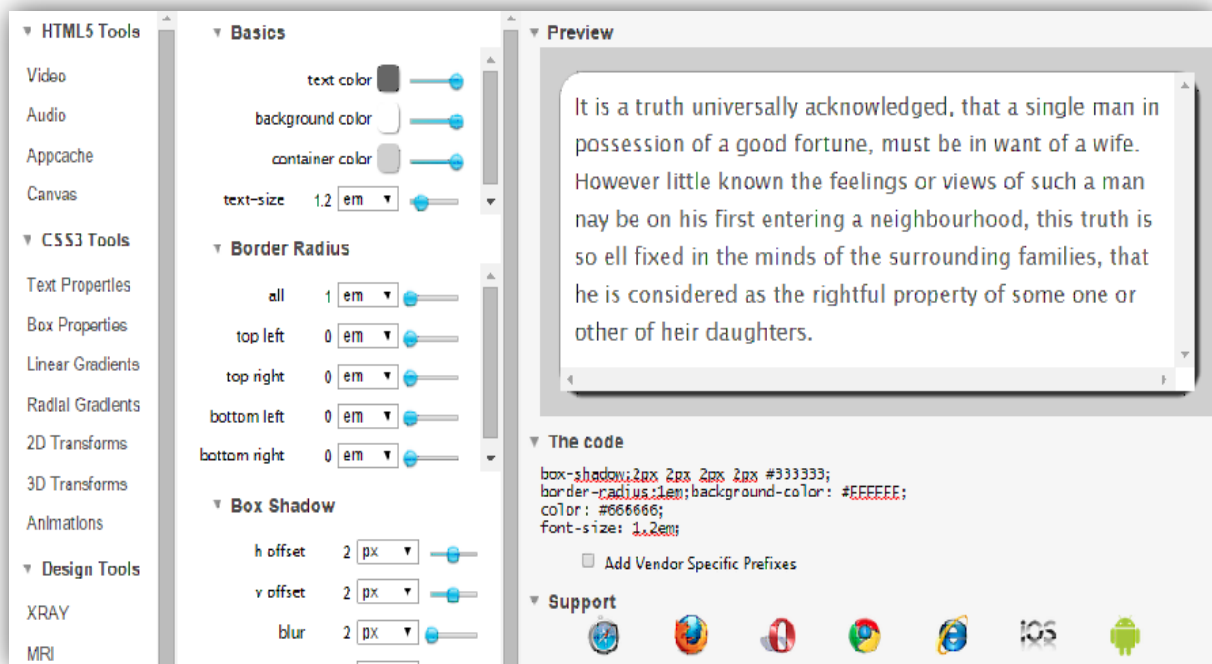
Εικόνα 5: Προβολή της πλατφόρμας ηλεκτρονικής εκμάθησης TheCodePlayer.

Παράλληλα υπάρχει κουμπί αναπαραγωγής (play) το οποίο δίνει την δυνατότητα στον χρήστη να παρακολουθήσει σε βίντεο την κατασκευή του παραπάνω έτοιμου κώδικα που έχει επιλέξει να δει διαμορφώνοντας επιπλέον τα βίντεο σε διαφορετική ταχύτητα ροής για ευκολότερη παρακολούθηση και ταυτόχρονη εξικοίωση με τον κώδικα.

4.5 WebDirectionsSouth



Η Webdirections.org είναι ιστοσελίδα η οποία περιλαμβάνει προσομοίωση ενός κώδικα στην οποία ο εκπαιδευόμενος έχει την δυνατότητα να αλλάξει τις τιμές στα διάφορα tags και εντολές της html5 και της css3 και να παρατηρήσει τις διάφορες αλλαγές που συντελούνται στην μορφοποίηση της σελίδας αλλά και την αλλαγή της ίδιας της εντολής στο κομμάτι του κώδικα.



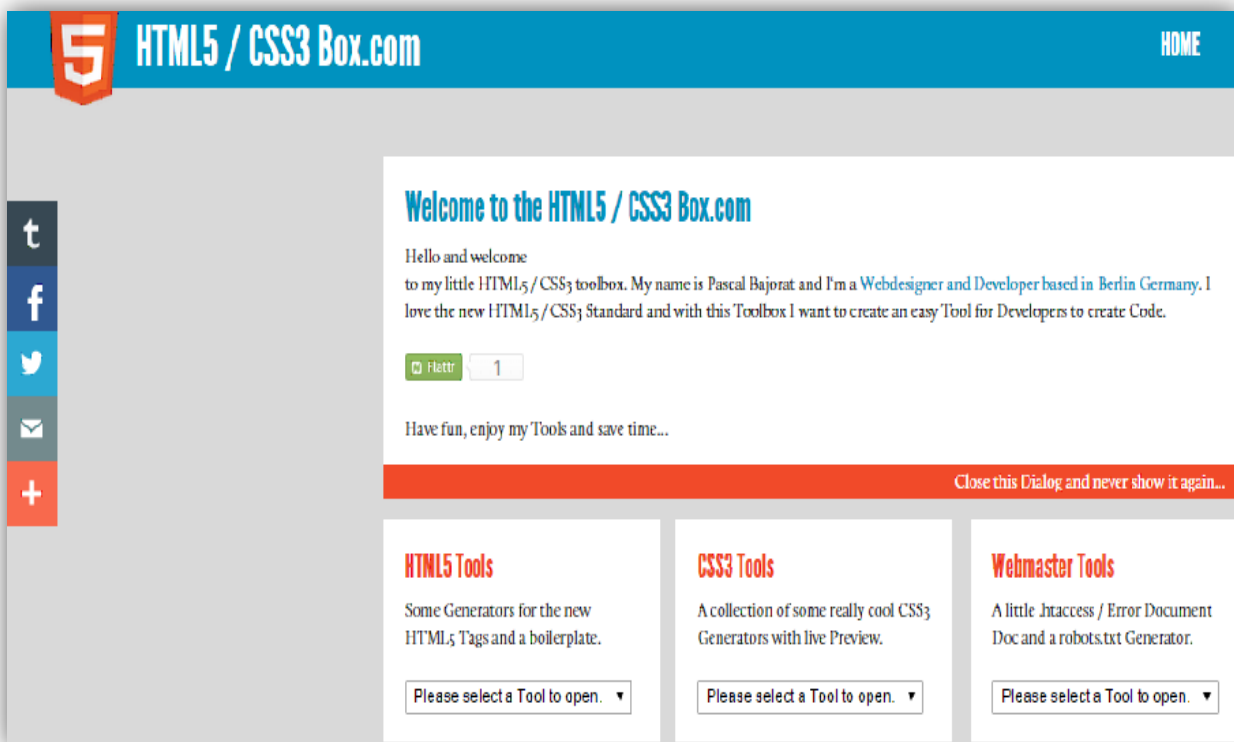
Εικόνα 6: Προβολή της πλατφόρμας ηλεκτρονικής εκμάθησης WebdirectionsSouth.

Δηστηγώς βασικό μειονέκτημα της σελίδας αποτελεί για ακόμη μια φορά η έλλειψη του πίνακα καταγραφής της σύνταξης και της έννοιας των εντολών, της φόρμα σχολιασμού από την μεριά των χρηστών καθώς και η έλλειψη αντίστοιχων επιμορφωτικών βίντεο που παρατηρούνται σε παραπάνω πλατφόρμες που αναφέρθηκαν.

4.6 Html5 / Css3 Box.com

HTML5 / CSS3 Box.com

Η συγκεκριμένη ιστοσελίδα δεν θα μπορούσε να σχολιαστεί πότε ως μια πλατφόρμα ωστόσο περιλαμβάνει ένα πολύ ενδιαφέρον στοιχείο το οποίο θα μπορούσε να βοηθήσει στην παρουσίαση αλλά και στην χρήση εκπαιδευτικού υλικού, αυτό είναι οι generators δηλαδή γεννήτριες έτοιμου κώδικα τον οποίο μπορεί να χρησιμοποιήσει ο χρήστης στην κατασκευή και την μορφοποίηση μιας ιστοσελίδας.



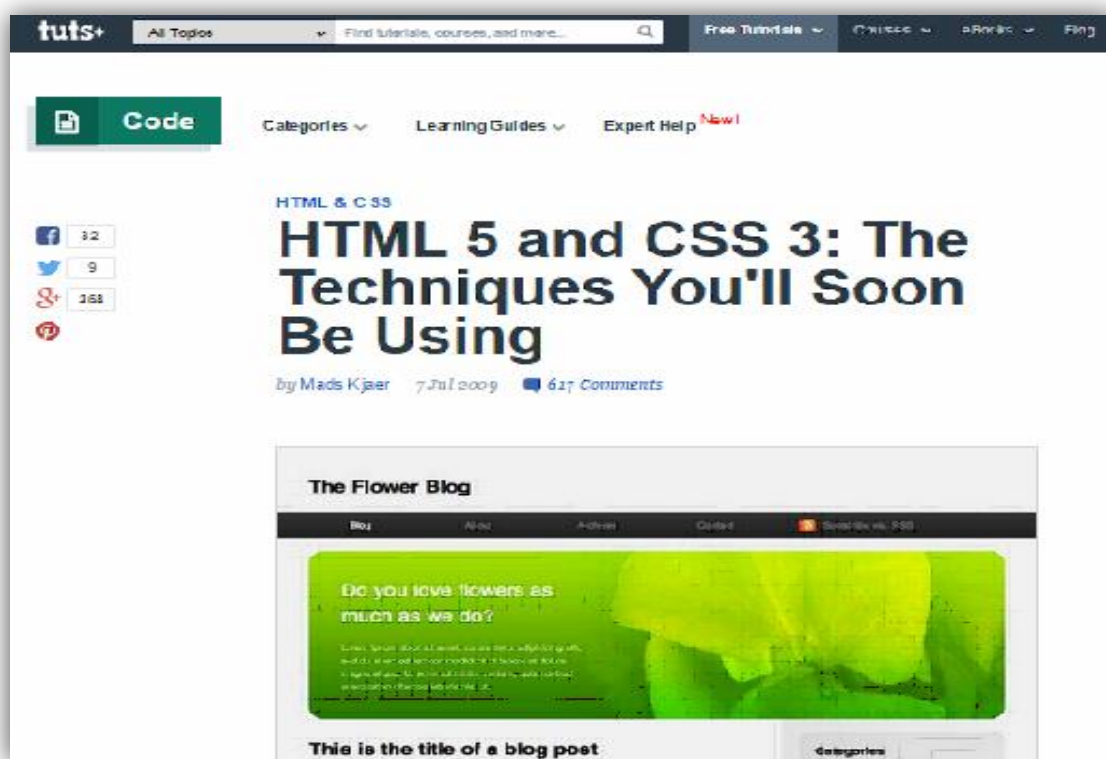
Εικόνα 7: Προβολή της πλατφόρμας ηλεκτρονικής εκμάθησης HTML5/CSS3 Box.com.

Οι γεννήτριες είναι κατά κάποιο τρόπο προσομοίωση του κώδικα των γλωσσών html5/css3 , η οποία μάλιστα διαφοροποιείται ανάλογα με τις τιμές που εισάγει ο χρήστης στις διάφορες εντολές και tags των γλωσσών. Φυσικά παρατηρείται έλλειψη εκπαιδευτικών βίντεο και άρθρων επεξήγησης της χρησιμότητας των εντολών καθώς και έλλειψη του πίνακα των εντολών και της φόρμας σχολιασμού των χρηστών.

4.7 Tuts+



Η tuts+ είναι μια ιστοσελίδα όπου διακρίνεται για την παροχή οδηγιών με εξειδικευμένα άρθρα για οτιδήποτε αφορά την κατασκευή ιστοσελίδων όπως έτοιμα κομμάτια κώδικα για διάφορα tags τα οποία βοηθούν τον χρήστη στη σύνταξη ενός κώδικα ιστοσελίδας. Επίσης η πλατφόρμα πλεονεκτεί σε αντίθεση με τις προαναφερθείσες ιστοσελίδες γιατί περιλαμβάνει σελίδα αναφοράς προβλημάτων και ιδιοτροπιών ορισμένων tags ως προς την υποστήριξη τους από τους φυλλομετρητές (browsers), την οποία σελίδα όμως ακολουθεί η προβολή διάφορων λύσεων των παραπάνω προβλημάτων.



Εικόνα 8: Προβολή της πλατφόρμας ηλεκτρονικής εκμάθησης tuts+.

Επιπλέον υπάρχει παροχή ηλεκτρονικών βιβλίων σε ορισμένα από τα οποία παρατηρείται και η δυνατότητα ηλεκτρονικού “κατεβάσματος”, λήψης δηλαδή σε διάφορες μορφές κειμένου. Ακόμη η ύπαρξη σελίδας μέσα από την οποία ο χρήστης έχει την δυνατότητα να βρει αγγελία εργασίας στον τομέα της κατασκευής ιστοσελίδας στην οποία μπορεί να αποστείλει το βιογραφικό του, αποτελεί ένα επιπλέον πλεονέκτημα της πλατφόρμας tuts+.

Αντίστοιχα παρέχονται ειδικά μαθήματα και οδηγίες στις οποίες τα αρχικά άρθρα είναι δωρεάν αλλά δυστυχώς αν θέλει κάποιος χρήστης να προχωρήσει και σε επιπλέον υλικό θα πρέπει να καταβάλλει και το αντίστοιχο χρηματικό πόσο.

Τέλος στη συγκεκριμένη ιστοσελίδα ο χρήστης μπορεί έπειτα από την εγγραφή του στον ιστότοπο και την επιλογή της γλώσσας που ομιλεί, να λαμβάνει τα άρθρα και τους διάφορους οδηγούς εκμάθησης της html5 και css3 στο email του μεταφρασμένα στην γλώσσα που έχει επιλέξει.

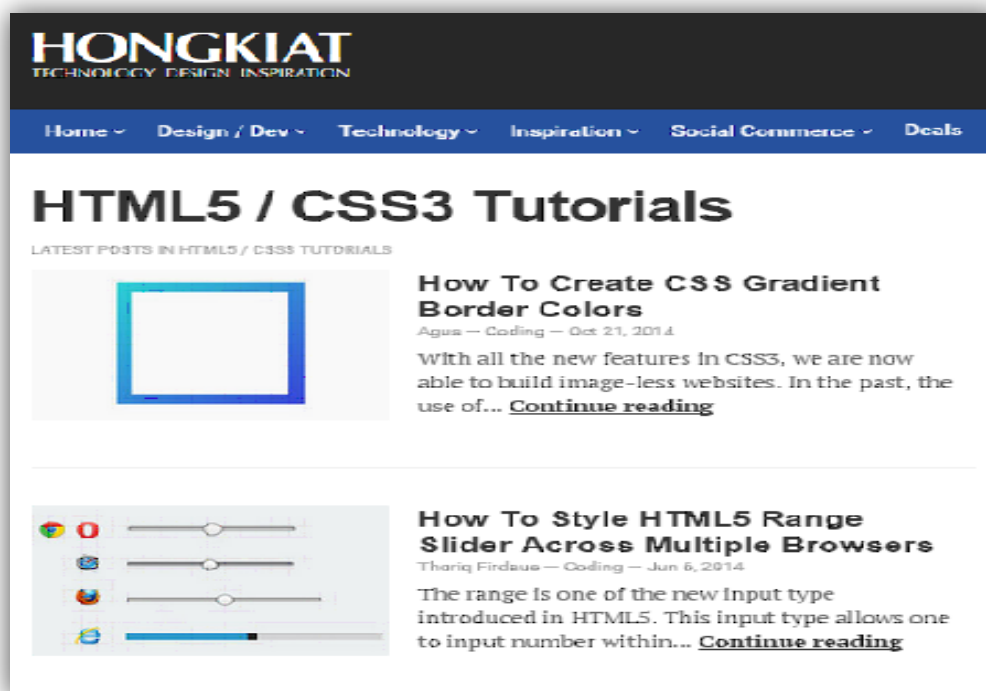
4.8 Hongkiat.com



Προσόν της ιστοσελίδας μπορεί να χαρακτηριστεί η προβολή ειδικών υπερσυνδέσμων (links) που οδηγούν σε σελίδες με έτοιμα κομμάτια κώδικα για την κατασκευή συγκεκριμένων στοιχείων σε μια σελίδα για παράδειγμα την κατασκευή menu, word-press tag-menu και διάφορων fonts και icons που μπορούν να χρησιμοποιηθούν σε μια σελίδα.

Στα πλεονεκτήματα της ιστοσελίδας μπορεί να προστεθεί η ύπαρξη του ειδικού πεδίου σχόλιων των χρηστών και η ύπαρξη πεδίου “know-how” που βοηθά σε απορίες που μπορεί να έχει σε διαφορετικά θέματα σχεδιασμού μιας ιστοσελίδας ένας χρήστης του διαδικτύου.

Αντίθετα παρατηρείται η συνήθης έλλειψη του πίνακα εντολών των γλωσσών html5/css3. Δυστυχώς πρέπει να αναφερθεί ότι η ιστοσελίδα “πάσχει” από έλλειψη οργάνωσης καθώς το περιεχόμενο της δεν αποτελείται μόνο από θέματα των γλωσσών html5/css3, αφού αντικείμενο της σελίδας αποτελεί και η γενικότερη θεματολογία που αναφέρεται στην βελτίωση ενός υπολογιστή ή σε διάφορα αλλά χρήσιμα εργαλεία όπως για παράδειγμα το photoshop.



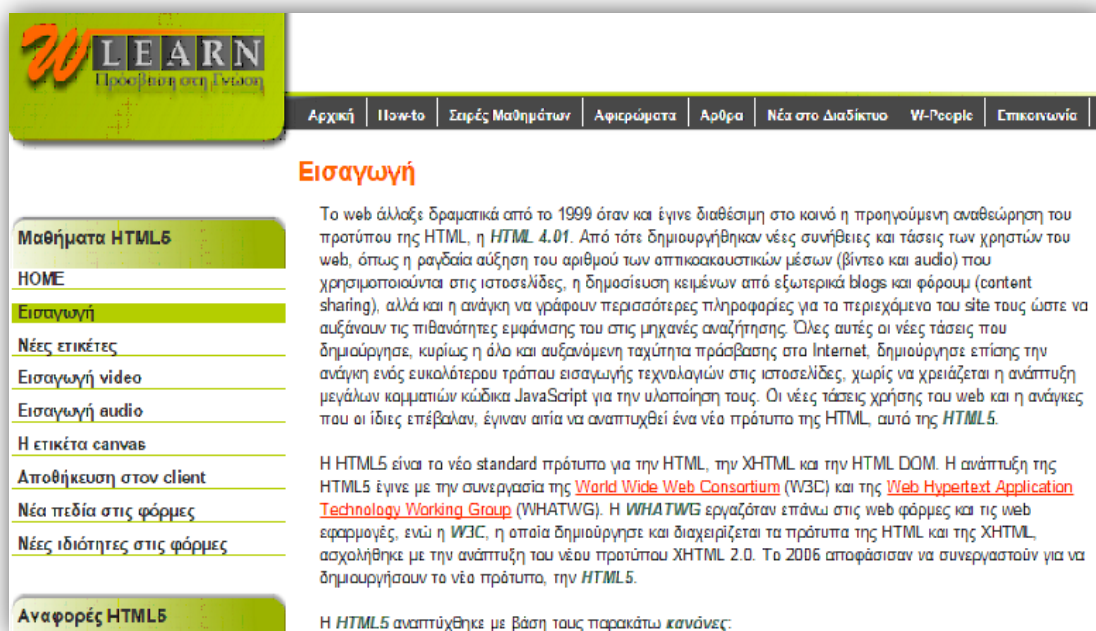
Εικόνα 9: Προβολή της πλατφόρμας ηλεκτρονικής εκμάθησης HONGKIAT.COM.

Επειδή το αντικείμενο της ιστοσελίδας ποικίλει και τα θέματα μεταξύ τους δεν “ταιριάζουν” πάντα με την εκμάθηση της δημιουργίας ιστοσελίδων, η συγκεκριμένη σελίδα δεν μπορεί να καθοριστεί απόλυτα ως “πλατφόρμα” εκμάθησης της html5/css3 παρόλη την βοήθεια που προσφέρει στον τομέα αυτό.

4.9 Wlearn



Ελληνικής κατασκευής ιστοσελίδα όπου περιλαμβάνει την παρουσίαση των βασικών χαρακτηριστικών της html5 και css3 όπως απλή καταγραφή και επεξήγηση της χρήσης και της συνταξης των εντολών και των μεθόδων καθώς και απλά παραδείγματα και εξομοίωση τους στον φυλλομετρητή.



Εικόνα 10: Προβολή της πλατφόρμας ηλεκτρονικής εκμάθησης Wlearn.gr.

Η ιστοσελίδα μειονεκτεί στο γεγονός ότι δεν πρόκειται για μια “καθαρή” πλατφόρμα εκμάθησης αφού μοιάζει πολύ στην ιστοσελίδα HONGKIAT.COM , δηλαδή το περιεχόμενο της δεν είναι απόλυτα ορισμένο στην κατασκευή και τον σχεδιασμό μιας ιστοσελίδας αλλά αποτελείται και από ποικίλη θεματολογία που στην προκειμένη περίπτωση δεν είναι το αντικείμενο της έρευνας μας.

4.10 Computerman.gr



Ακόμη μια ελληνικής δημιουργίας ιστοσελίδα, η οποία αν μπορεί να περηφανευτεί για ένα πλεονέκτημα της έναντι των άλλων ιστοσελίδων, είναι η προβολή εκπαιδευτικών βίντεο που έχουν ενσωματωθεί απο το youtube και μέσα από τα οποία οι δημιουργοί τους διδάσκουν τον χρήστη τα βασικά στοιχεία των γλωσσών html5/ css3/ joomla/ javascript/ php.

Η ιστοσελίδα δεν περιλαμβάνει πίνακα εντολών – μεθόδων αλλά παρέχει πεδίο παρατηρήσεων - σχολίων για τον χρήστη στο οποίο μπορεί να καταγράψει τις τυχόν απορίες του με σκοπό να λάβει κάποια σχετική απάντηση από τους διαχειριστές της σελίδας.

The screenshot shows the website interface for 'COMPUTERMAN VIDEO TUTORIALS'. The main content area displays an article titled 'HTML Basics: Εισαγωγή στην HTML5'. The article text explains that HTML (HyperText Markup Language) is a fundamental web technology used for creating web pages. It mentions that HTML5 is the latest version and that the article is part of a series. A red banner at the bottom of the article text reads: 'Τα επόμενα δυο μαθήματα θα αντικατασταθούν σύντομα με τα νέα!'. The sidebar on the right includes a search bar, social media links for Facebook and YouTube, a newsletter sign-up form, and a list of recent articles such as 'Εγκατάσταση LAMP Stack σε Linux Mint (Linux, Apache, MySQL, PHP)'.

Εικόνα 11: Προβολή της πλατφόρμας ηλεκτρονικής εκμάθησης COMPUTERMAN.

4.11 StudentGuru.gr



Ίσως η πιο ολοκληρωμένη πλατφόρμα εκμάθησης μετά την πλατφόρμα w3schools.com. Θα πρέπει να αναφερθεί ότι η σελίδα στην ουσία αποτελεί μια ελληνική εκπαιδευτική κοινότητα, συνέπεια της συνεργασίας της με τα Πανεπιστήμια και τα Τεχνολογικά ιδρύματα της χώρας. Στην περίπτωση της ιστοσελίδας studentguru.gr μερικά από τα πλεονεκτήματα που εντοπίζουμε είναι τα εξής:

- Παρουσίαση των βασικών στοιχείων (εντολές – μέθοδοι) των html5/css3 μέσω πινάκων και προβολής παραδειγμάτων και ταυτόχρονη εξωμοίωση τους στον φυλλομετρητή,
- Ύπαρξη επιμορφώτικων βίντεο που έχουν ενσωματωθεί στην σελίδα από το youtube,
- Ύπαρξη ειδικού forum μέσα στο οποίο οι χρήστες της σελίδας μπορούν να επικοινωνήσουν μεταξύ τους ανταλλάσσοντας τις γνώσεις τους και λύνοντας τυχόν απορίες,
- Παροχή ειδικού πεδίου downloads, που δίνει την δυνατότητα λήψης εκπαιδευτικού υλικού σε μορφή ανάγνωσης όπως επίσης και αντίστοιχο λογισμικό το οποίο επιπλέον είναι ελεύθερο καθώς ο χρήστης μπορεί να το κατεβάσει στον υπολογιστή του χωρίς να καταβάλει κάποιο χρηματικό ποσό,
- Η ιστοσελίδα συνεργάζεται με όλα τα ακαδημαϊκά εκπαιδευτήρια της χώρας με αποτέλεσμα το εκπαιδευτικό υλικό που παρουσιάζεται να είναι εγκεκριμένο και αξιόπιστο, ενώ παράλληλα από αυτήν την συνεργασία οι άμεσα ενδιαφερόμενοι σπουδαστές μπορούν να επικοινωνήσουν μεταξύ τους με σκοπό την καλύτερη διανομή του υλικού,
- Παροχή ενημέρωσης στο χρήστη για διάφορα events που μπορεί να πραγματοποιηθούν στα οποία αν θέλει μπορεί να λάβει μέρος και τα οποία σαν σκοπό τους έχουν την επιμόρφωση και την ενημέρωση στις νέες τεχνικές προγραμματισμού είτε είναι γενικός προγραμματισμός και σχεδιασμός λογισμικού είτε στην περίπτωση που μας αφορά εμάς προγραμματισμός και δημιουργία ιστοσελίδας.
- Η ιστοσελίδα διοργανώνει επίσης και διαγωνισμούς τεχνολογίας στους οποίους οι χρήστες μπορούν αν το επιθυμούν να πάρουν μέρος έχοντας την πιθανότητα να βραβευθούν στο τέλος του διαγωνισμού.

Αντίθετα σαν μειονεκτήματα μπορούν να σημειωθούν τα εξής:

- Δυστυχώς παρόλο που η ιστοσελίδα μπορεί να χαρακτηριστεί ολοκληρωμένη ως προς τους τρόπους με τους οποίους προσφέρει το εκπαιδευτικό υλικό, δεν μπορούμε να πούμε το ίδιο και για το ίδιο το υλικό καθώς δεν μπορεί να συγκριθεί με τον πλούτο γνώσης της ιστοσελίδας w3schools.com είτε αυτό είναι πίνακες είτε είναι παραδείγματα
- Επίσης πρέπει να επισημάνουμε ότι η συγκεκριμένη ιστοσελίδα δεν αναφέρεται αποκλειστικά στις γλώσσα html5 και τα διαδοχικά φύλλα στυλ css3, αλλά περιλαμβάνει και ηλεκτρονικούς οδηγούς και προγράμματα εκμάθησης για τον γενικότερο προγραμματισμό και σχεδιασμό λογισμικού, κάτι το οποίο από μόνο του δεν μπορεί να χαρακτηριστεί ως μειονέκτημα αλλά όταν εμπλέκεται στο συγκεκριμένο αντικείμενο της σχεδίασης ιστοσελίδων προκαλεί σύγχυση σε έναν αρχάριο εκπαιδευόμενο χρήστη.
- Τέλος ένα ακόμα μειονέκτημα που μπορεί να προστεθεί στην λίστα είναι η έλλειψη ενός ειδικού τμήματος στην σελίδα το οποίο θα μπορούσε να περιλαμβάνει ελεύθερες προς λήψη χωρίς την καταβολή χρηματικού αντίτιμου ορισμένες βιβλιοθήκες της γλώσσας javascript για τον σχεδιασμό των menu μιας ιστοσελίδας ή άλλων πιο εξειδικευμένων γραφικών που δεν μπορούν να δημιουργηθούν από τα διαδοχικά φύλλα στυλ css3, και οι οποίες κρίνονται σημαντικές από τον χρήστη για το “χτίσιμο” μιας σελίδας.

5 Βασική δομή και σύνταξη ενός αρχείου Html5

Αρχικά θα πρέπει να καταλάβουν οι αναγνώστες του οδηγού με ποιόν τρόπο και μέσω ποιού εργαλείου μπορούμε να δημιουργήσουμε ένα αρχείο Html5.

Ένα αρχείο Html5 μπορεί να δημιουργηθεί είτε μέσω του απλού σημειωματάριου του υπολογιστή / Wordpad , είτε μέσω των παρακάτω Html5 Editors:

1. Aloha editor
2. Adobe Dreamweaver CS5 / Microsoft Office Publisher
3. Maqetta / MacFlux
4. BlueGriffon
5. Mercury Editor / Rendra
6. Aptana studio / codeLobster studio

Στην παρούσα πτυχιακή όμως το αρχείο Html που θα δημιουργήσουμε για να εξομοιώσουμε τα παραδείγματα του οδηγού , θα «γραφτεί» στο απλό σημειωματάριο των Windows.

Στις παρακάτω υποπαραγράφους του έκτου κεφάλαιου θα παρουσιαστούν τα βασικά μέρη ενός αρχείου Html, η σύνταξη τους καθώς και η περιγραφή της λειτουργίας τους.

5.1 Βασικά στοιχεία – tags του κώδικα Html

Τα κύρια βασικά στοιχεία που πρέπει να περιλαμβάνει ένας κώδικας σε Html είναι τα λεγόμενα tags.

Τα tags μπορούν να χαρακτηριστούν ως «ετικέτες» , δηλαδή είναι κάποιες συγκεκριμένες λέξεις που έχουν οριστεί από τους δημιουργούς της γλώσσας προγραμματισμού Html και οι οποίες είναι υπεύθυνες για την κατηγοριοποίηση των διαφόρων κομματιών κώδικα.

Με λίγα λόγια οι ετικέτες αυτές δηλώνουν την αρχή και το τέλος εισαγωγής ή μορφοποίησης ενός στοιχείου όπως για παράδειγμα τα tags:

<table> το οποίο δημιουργεί έναν πίνακα ή το tag **** το οποίο μορφοποιεί ένα κείμενο που θα εμφανίζεται στην σελίδα μετατρέποντας το σε έντονη γραμματοσειρά.

Όπως λοιπόν χρησιμοποιούμε ειδικό tag για την δημιουργία ενός πίνακα έτσι θα πρέπει να χρησιμοποιήσουμε και συγκεκριμένα tags μέσα στα οποία θα εμπεριέχεται ολόκληρος ο κώδικας Html.

Αυτά τα ειδικά tags είναι τα: **<html>** , **<head>** , **<title>** , **<body>**. Η αναλυτικότερη περιγραφή λειτουργίας του καθενός από τα παραπάνω συγκεκριμένα tags παρουσιάζεται στις παρακάτω υποπαραγράφους.

5.1.1 Το tag <html>

Το tag <html> δηλώνει πως παρακάτω εμπεριέχεται κώδικας γραμμένος σε γλώσσα html κάτι το οποίο βοηθά τον φυλλομετρητή μας να κάνει την ανάλογη εκτίμηση ανάλογα με το πως θα εμφανίσει το περιεχόμενο του κώδικα στην σελίδα, δηλαδή στην προκειμένη περίπτωση καταλαβαίνει από το tag ότι θα πρέπει να χρησιμοποιήσει τα πρότυπα της γλώσσας Html για να εμφανιστεί σωστά η σελίδα στην οθόνη μας.

Επίσης θα πρέπει να αναφέρουμε ότι για να καταλάβει ο φυλλομετρητής που αρχίζει και που τελειώνει ο κώδικας θα πρέπει το tag <html> να «κλείσει» δηλαδή στο τέλος του κώδικα να γραφτεί πάλι το tag με την διαφορά ότι πριν από το όνομα του tag θα γραφτεί μια κάθετος ως εξής:

< /html >

Έτσι λοιπόν καταλαβαίνουμε πως υπάρχουν τα open tags που δηλώνουν την αρχή ενός στοιχείου ή κομματιού κώδικα και τα αντίστοιχα close tags τα οποία δηλώνουν το τέλος της εισαγωγής ή της μορφοποίησης των στοιχείων.

5.1.2 Τα tags <head>, <title>

Τα tags <head> και </head> δηλώνουν πως ό,τι εμπεριέχεται σε αυτά αποτελεί τμήμα της επικεφαλίδας της σελίδας. Επίσης μέσα στο tag <head> μπορεί να υπάρχουν scripts (κώδικας γραμμένος σε php και javascript), styles (στοιχεία μορφοποίησης), meta informations (πληροφορίες σε μορφή σχολίων για ένα Html έγγραφο οι οποίες δεν εμφανίζονται στην σελίδα) καθώς και το απαραίτητο tag <title>.



Tip: Ενώ μέχρι τώρα ήταν απαραίτητη η χρήση του tag <head>, στην Html5 παρατηρούμε ότι μπορούμε να το παραλείψουμε σε αντίθεση με το tag <title> το οποίο καλό είναι να χρησιμοποιείται γιατί το να δίνουμε ένα τίτλο στην καρτέλα περιήγησης της σελίδας βοηθά και σε οπτικό επίπεδο τον χρήστη.

Τα tags <title> και </title> δηλώνουν πως ό,τι εμπεριέχεται σε αυτά θα εμφανιστεί ως τίτλος στο πάνω μέρος του προγράμματος περιήγησης, δηλαδή στην καρτέλα περιήγησης της σελίδας.

5.1.3 Το tag <body>

Τα tags <body> και </body> δηλώνουν πως ό,τι εμπεριέχεται σε αυτά αποτελεί τμήμα – περιεχόμενο της σελίδας που θα εμφανιστεί στην οθόνη μας. Μέσα σε αυτό το tag γίνεται και η όλη μορφοποίηση και κυρίως η εισαγωγή των στοιχείων που θέλουμε να υπάρχουν στην σελίδα μας. Επίσης το <body> είναι αυτό που δηλώνει τον κεντρικό «κορμό» της σελίδας.

5.2 Τα tags μορφοποίησης

Τα tags μορφοποίησης είναι και αυτά tags με την διαφορά ότι η λειτουργία τους δεν έχει να κάνει με την δημιουργία της «σωματοδομής» του Html εγγράφου αλλά με την μορφοποίηση του περιεχομένου της σελίδας που στις περισσότερες περιπτώσεις αποτελείται από κείμενο.

Μερικά απο αυτά είναι τα εξής:

1. Το tag : **...** το περιεχόμενο του tag εμφανίζεται στην οθόνη σε **bold** κατάσταση δηλαδή σε **εντονη γραμματοσειρά**.
2. Το tag: *<i>...</i>* το περιεχόμενο του tag εμφανίζεται στην οθόνη σε *italian* κατάσταση δηλαδή σε *πλάγια γραφή*.
3. Το tag: <u>...</u> το περιεχόμενο του tag εμφανίζεται στην οθόνη υπογεγραμμένο.
4. Το tag: ~~<strike>...</strike>~~ ή ~~<s>...</s>~~ προσθέτει μια ~~γραμμή~~ πάνω στο κείμενο που εμφανίζεται στην οθόνη συνήθως για να δείξει ότι η κατάσταση που περιγράφεται από το ίδιο το κείμενο έχει πάψει να ισχύει ή ότι δεν είναι αξιόπιστη.
5. Το tag:

```
<pre>...</pre>
```

 (preformatted text) χρησιμοποιείται για να δηλώσει ότι το κείμενο θα εμφανιστεί με την μορφοποίηση που θα χρησιμοποιούσαμε αν το γράφαμε σε κάποιον editor συμπεριλαμβάνοντας τα κενά και τα σημεία στίξης. Οι περισσότεροι browsers αν υπάρχουν κενά μεταξύ των λέξεων του κειμένου σε επίπεδο συγγραφής του κώδικα δεν τα «διαβάζουν» και κατά συνέπεια δεν τα εμφανίζουν και στην σελίδα.
6. Το tag: `<code>...</code>` χρησιμοποιείται για να μορφοποιήσει το κείμενο σε μορφή κώδικα δηλαδή σε συγκεκριμένο πλάτος γραμματοσειράς.
7. Το tag: `<tt>...</tt>` χρησιμοποιείται για να μορφοποιήσει το κείμενο σε μορφή γραφομηχανής με συγκεκριμένο πλάτος γραμματοσειράς.
8. Το tag: `<blockquote>...</blockquote>` καθορίζει ένα μεγάλο σε πλάτος απόσπασμα κειμένου το οποίο εμφανίζεται με ένα επιπλέον αριστερο περιθώριο.
9. Το tag: `<small>...</small>` χρησιμοποιείται όταν δεν είναι απαραίτητη η ιδιότητα font-size η οποία καθορίζει το μέγεθος της γραμματοσειράς , το tag `<small>` μετατρέπει το κείμενο που περικλείει ένα μικρότερο μέγεθος γραμματοσειράς από το υπόλοιπο κείμενο της σελίδας.
10. Το tag: ` ` καθορίζει το χρώμα της γραμματοσειράς του κειμένου που περικλείει. Τα 6 ερωτηματικά αντιπροσωπεύουν τον Hex κώδικο του κάθε χρώματος όπως φαίνεται στον πίνακα 2 του παραρτήματος.
11. Το tag: ` ` καθορίζει το μέγεθος της γραμματοσειράς παίρνοντας τιμές απο 1...7 με το 1 το μικρότερο μέγεθος και 7 το μεγαλύτερο.

12. Το tag: ` ` χρησιμοποιείται για την άμεση αλλαγή του μεγέθους της γραμματοσειράς σε σχέση με την γραμματοσειρά που προηγείται έξω από το tag. Επίσης αυξάνει ή μειώνει το μέγεθος από τον αριθμό που καθορίσαμε.
13. Το tag: ` ` χρησιμοποιείται για να ορίσουμε το όνομα της γραμματοσειράς που χρησιμοποιούμε.



Tip: Είναι προτιμότερο ένας σχεδιαστής να χρησιμοποιεί απλές και κοινής χρήσης γραμματοσειρές για δύο συνήθως λόγους: αρχικά γιατί οπτικά η πιο απλή σχεδίαση ιστοσελίδας είναι και πιο εμφανίσιμη αλλά κυρίως γιατί αν θέλουμε να χρησιμοποιήσουμε μια ποιο περίτεχνη γραμματοσειρά θα πρέπει να έχει εγκατασταθεί προηγουμένως στον υπολογιστή σχεδίασης της ιστοσελίδας κάτι το οποίο δεν είναι σίγουρο ότι θα ισχύει και στον υπολογιστή του χρήστη.

Με λίγα λόγια ο χρήστης της σελίδας μας δεν θα μπορεί να βλέπει την γραμματοσειρά που έχουμε χρησιμοποιήσει κατά την σχεδίαση της σελίδας γιατί μπορεί να μην έχει εγκατασταθεί στον υπολογιστή του.

Μερικές φορές το πρόβλημα λύνεται αν η περίτεχνη γραμματοσειρά που χρησιμοποιούμε είναι εξαρχής εγκατεστημένη όχι στον υπολογιστή μας αλλά σε κάποιον server και την καλούμε σαν εξωτερικό αρχείο, έτσι δεν θα χρειάζεται να εγκατασταθεί η γραμματοσειρά και στον υπολογιστή του τελικού χρήστη.

14. Το tag: `<center> </center>` στοιχίζει το κείμενο στο κέντρο της σελίδας.
15. Το tag: ` ` δίνει έμφαση σε ένα κείμενο εμφανίζοντας το συνήθως σε πλάγια γραφή (italian) αλλά αυτό εξαρτάται πάντα από τον browser που χρησιμοποιείται.
16. Το tag: ` ` δίνει έμφαση σε ένα κείμενο εμφανίζοντας το συνήθως σε έντονη γραφή (bold) αλλά αυτό εξαρτάται πάντα από τον browser που χρησιμοποιείται.

Παρακάτω εμφανίζεται ένα απλό Html έγγραφο στο οποίο το περιεχόμενο του tag `<body>` αλλάζει εμφάνιση ανάλογα με το tag μορφοποίησης στο οποίο περικλείεται:



Tip: Στον παρακάτω κώδικα με κόκκινο χρώμα επισημαίνονται τα βασικά tags , με πράσινο χρώμα επισημαίνονται τα tags μορφοποίησης και με γαλαζιο χρώμα τα προαιρετικά tags.

```

<html> <head> <title>Title of the document</title> </head>
<body>
<b>The content of the document</b> <br>
<i>The content of the document</i> <br>
<u>The content of the document</u> <br>
<strike>The content of the document</strike> <br>
<pre>The content of the document</pre> <br>
<code>The content of the document</code> <br>
<tt>The content of the document</tt> <br>
<blockquote>The content of the document</blockquote> <br>
<small>The content of the document</small> <br>
<font color="#A52A2A">The content of the document</font> <br>
<font size="3">The content of the document</font> <br>
<font size="-1">The content of the document</font> <br>
<font face="arial">The content of the document</font> <br>
<center>The content of the document</center> <br>
<em>The content of the document</em> <br>
<strong>The content of the document</strong>
</body> </html>

```

The content of the document

The content of the document

The content of the document

~~The content of the document~~

The content of the document

The content of the document

The content of the document

The content of the document

The content of the document

The content of the document

The content of the document

The content of the document

The content of the document

The content of the document

The content of the document

The content of the document

6 Δημιουργία πίνακα / λίστας και στοιχεία ομαδοποίησης

Στην ενότητα αυτή θα αναφερθούμε στην δημιουργία και στην εισαγωγή ενός πίνακα, μιας λίστας και ενός block, αφού είναι αυτονόητο πως το θέμα μιας ιστοσελίδας δεν θα προβάλλεται μόνο μέσα από ένα απλό κείμενο κάτι το οποίο πρέπει να αποφεύγεται κατά την σχεδίαση γιατί είναι πολύ κουραστικό οπτικά για τον χρήστη, αντ' αυτού μπορούμε να χρησιμοποιούμε πίνακες, λίστες που χρησιμοποιούνται συχνά και στην κατασκευή των menu της σελίδας και blocks.

6.1 Δημιουργία πίνακα

Αρχικά για να κατασκευάσουμε έναν πίνακα θα πρέπει να χρησιμοποιήσουμε και τα αντίστοιχα tags.

Το βασικό tag λοιπόν είναι το `<table>` είναι αυτό που δημιουργεί τον container του πίνακα όπως λέγεται δηλαδή το «κουτί» στην ουσία που θα περικλείει τις γραμμές και τις στήλες του πίνακα με το περιεχόμενο. Αντίστοιχα για την δημιουργία των γραμμών του πίνακα χρησιμοποιούμε το tag `<tr>` (από τα αρχικά του table rows) ενώ θα πρέπει να ξέρουμε πως οι γραμμές του πίνακα χωρίζονται σε κάθετα κελιά τις στήλες με το tag `<td>` (από τα αρχικά του table data) τα οποία μπορούν να αποτελούνται από κείμενο, εικόνες συνδέσμους ή ακόμα και έναν πίνακα. Επιπλέον θα πρέπει να ξέρουμε πως υπάρχει και το tag `<th>` το οποίο μας δίνει την δυνατότητα να βάλουμε μια επικεφαλίδα στον πίνακα. Έτσι λοιπόν ένας πίνακας χωρίζεται σε δύο ειδών κελιά:

1. Τα header cells – κελιά επικεφαλίδας στα οποία το περιεχόμενο εμφανίζεται σε έντονη (bold) γραφή και στοιχισμένο στο κέντρο
2. Και τα standard cells – «κλασσικά» κελιά στα οποία το περιεχόμενο δεν είναι bold και είναι στοιχισμένα στα αριστερά του κελιού.



Tip: Κατά την διάρκεια δημιουργίας ενός πίνακα ο σχεδιαστής θα πρέπει να προσέξει την σειρά με την οποία θα γραφτούν τα tags καθώς το tag `<tr>` γράφεται πάντα πριν από το tag `<td>`, όπως επίσης και να «κλείνει» τα tags στο τέλος του περιεχομένου τους.

Έτσι λοιπόν παρακάτω βλέπουμε την δημιουργία ενός πολύ απλού πίνακα 2x2 (2 γραμμές – 2 στήλες) και πως εμφανίζεται στην οθόνη. Το κόκκινο χρώμα της γραμματοσειράς δηλώνει τα tags των γραμμών του πίνακα ενώ το πράσινο τις στήλες και το μπλέ τον ίδιο τον πίνακα.

```
<html> <head> <title> tables </title> </head>
<body>
<table border="1" style="width:60%">
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
  <tr>
    <td>John</td>
    <td>Doe</td>
    <td>80</td>
  </tr>
</table>
</body> </html>
```

Jill	Smith	50
Eve	Jackson	94
John	Doe	80

Τα στοιχεία που βρίσκονται μέσα στο tag του πίνακα <table> και είναι υπογεγραμμένα με κίτρινο χρώμα ονομάζονται attributes του πίνακα και ορίζουν ότι κατά την εμφάνιση του πίνακα στην σελίδα το περίγραμμα του πίνακα θα έχει μέγεθος 1 (**border="1"**) και το πλάτος του πίνακα θα είναι το 60% της οθόνης (**style="width:60%"**).

Τα attributes είναι αυτά που μας δίνουν την δυνατότητα να μορφοποιήσουμε ένα στοιχείο όπως ένας πίνακας ή μια λίστα. Μερικά από αυτά για παράδειγμα είναι τα εξής:

1. Το **border** το οποίο μορφοποιεί το περίγραμμα του πίνακα δηλαδή το μέγεθος, το χρώμα, την διαφάνεια (δηλαδή πόσο διαφανές θα εμφανίζεται), το στυλ του αν θα είναι δηλαδή μια συνεχόμενη γραμμή ή διακοπτόμενη ή ακόμα και dotted (με τελείες «πουά»), παίρνοντας κάθε φορά διαφορετική κατάληξη στο attribute, ή ακόμα και αν θα υπάρχει κενό ανάμεσα στο περίγραμμα του πίνακα με το περίγραμμα του κελιού π.χ.:

- Για το **μέγεθος του περιγράμματος** θα είναι: border-width παίρνοντας τιμές σε px σε em ή % : border-width="15px" / border-width="15em" / border-width="15%".
- Για το **χρώμα του περιγράμματος** θα είναι: border-color παίρνοντας τιμές είτε σε κωδικοποίηση χρωμάτων Hex είτε σε κωδικοποίηση rgb: border-color="#ff0000" / border-color="(223,231,155)" / border-color="red green blue pink". Επιπλέον μπορούμε να έχουμε διαφορετικά χρώματα στις διάφορες πλευρές του περιγράμματος του πίνακα π.χ.:

1. border-color:red green blue pink;

- το περίγραμμα κορυφής είναι **κόκκινο**
- το περίγραμμα στα δεξιά είναι **πράσινο**
- το περίγραμμα στο κάτω μέρος είναι **μπλέ**
- το περίγραμμα στα αριστερά είναι **ροζ**

2. border-color:red green blue;

- το περίγραμμα κορυφής είναι **κόκκινο**
- τα περιγράμματα στα αριστερά και δεξιά είναι **πράσινα**
- το περίγραμμα στο κάτω μέρος είναι **μπλέ**

3. border-color:red green;

- τα περιγράμματα κορυφής και στο κάτω μέρος είναι **κόκκινα**
- τα περιγράμματα στα **αριστερά και δεξιά** είναι **πράσινα**

4. border-color:red;

- όλα τα περιγράμματα είναι **κόκκινα**
- για το **στυλ του περιγράμματος** θα είναι: border-style="..." όπου «...» παίρνει τις εξής τιμές:
 1. border-style="none" δεν εμφανίζει κανένα περίγραμμα
 2. border-style="hidden" το περίγραμμα είναι κρυμμένο
 3. border-style="dotted" το περίγραμμα αποτελείται από τελείες
 4. border-style="dashed" το περίγραμμα αποτελείται από παύλες
 5. border-style="solid" το περίγραμμα είναι μια συνεχόμενη γραμμή
 6. border-style="double" το περίγραμμα είναι δυο συνεχόμενες γραμμές
 7. border-style="groove" εμφανίζει ένα 3d περίγραμμα με σκιές πάνω στις γραμμές
 8. border-style="ridge" εμφανίζει ένα 3d περίγραμμα με σκιές στο πλάι των γραμμών
 9. border-style="inset" εμφανίζει ένα 3d περίγραμμα δημιουργώντας την αίσθηση πως εισάγεται προς τα «μέσα» στην σελίδα
 10. border-style="outset" εμφανίζει ένα 3d περίγραμμα δημιουργώντας την αίσθηση πως «βγαίνει» ένα επίπεδο προς τα έξω από την σελίδα

11. **border-style="initial"** εμφανίζει το περίγραμμα με στυλ σε standard – default μορφή

12. **border-style="inherit"** εμφανίζει το περίγραμμα με την μορφή του περιγράμματος του ανώτερου στοιχείου στην προκειμένη περίπτωση του πίνακα. Αν υπήρχαν εμφωλευμένοι πίνακες δηλαδή πίνακας μέσα σε πίνακα, και το περίγραμμα του εσωτερικού πίνακα είχε σαν στυλ το inherit σημαίνει ότι θα είχε το ίδιο στυλ με το περίγραμμα του εξωτερικού πίνακα.

2. Για την **ύπαρξη κενού ανάμεσα στα περιγράμματα** του πίνακα και των κελιών θα είναι:

1. **Border-collapse="separate"** εμφανίζει ξεχωριστά περιγράμματα για τα κελιά του πίνακα και τον ίδιο τον πίνακα προσθέτοντας κενό μεταξύ τους.

2. **Border-collapse="collapse"** ενώνει τα ξεχωριστά περιγράμματα των κελιών του πίνακα και του ίδιου του πίνακα εμφανίζοντας ένα μόνο περίγραμμα.

3. Επίσης ένα ακόμη attribute που είναι σημαντικό στην μορφοποίηση ενός πίνακα είναι και το **cellspacing** το οποίο αλλάζει το περιθώριο ανάμεσα στα κελιά του πίνακα και το **cellpadding** το οποίο αλλάζει το περιθώριο ανάμεσα στο περιεχόμενο του κελιού και το περίγραμμα του ίδιου κελιού.

4. Ένα επιπλέον attribute είναι το **bgcolor** το οποίο αλλάζει το χρώμα στο υπόβαθρο του πίνακα δηλαδή το background.



Tip: Κλείνοντας την παράγραφο της δημιουργίας του πίνακα θα πρέπει να αναφέρουμε ότι τα παραπάνω attributes πλέον έχουν καταργηθεί στην Html5 γιατί η μορφοποίηση σχεδιάζεται σε διαδοχικά φύλλα στυλ (Css) και όχι στο ίδιο το έγγραφο Html. Για την μορφοποίηση σε Css όμως θα μιλήσουμε στα παρακάτω κεφάλαια.

6.2 Δημιουργία λίστας

Όπως και για την δημιουργία του πίνακα υπάρχουν ειδικά tags έτσι υπάρχουν και τα αντίστοιχα για την δημιουργία λίστας , μόνο που στην περίπτωση της λίστας θα πρέπει να γνωρίζουμε ότι τα tags χωρίζονται σε τρεις κατηγορίες γιατί πολύ απλά υπάρχουν τρία διαφορετικά είδη λίστας.

Τα τρία είδη της λίστας είναι τα εξής:

- Ταξινομημένη λίστα με αριθμούς (Ordered list)
- Ταξινομημένη λίστα με τελείες (Unordered list)
- Επεξηγηματική λίστα (υπάρχει επεξήγηση κάτω από κάθε μέλος της λίστας)
- Εμφωλευμένη λίστα (όταν υπάρχει λίστα μέσα σε μια άλλη λίστα)

6.2.1 Τα tags της αριθμημένης λίστας

Για την δημιουργία της λίστας με αριθμούς χρησιμοποιούμε το tag που κατασκευάζει το «κουτί» της λίστας στο οποίο για να εισάγουμε ξεχωριστά τα μέρη της λίστας χρησιμοποιούμε το tag και δεν ξεχνούμε ποτέ να κλείνουμε τα tags στο τέλος του περιεχομένου.

Αν θα πρέπει να μιλήσουμε για ένα attribute της αριθμημένης λίστας αυτό είναι το type με το οποίο αλλάζουμε τον τύπο του εικονιδίου αρίθμησης, δηλαδή μπορούμε εκτός από την κλασσική δεκαδική αρίθμηση να έχουμε λατινική αρίθμηση σε κεφαλαία ή μικρά γράμματα ή αριθμούς.

Παρακάτω παρουσιάζεται η δημιουργία μιας λίστας με κόκκινο χρώμα γραμματοσειράς προβάλλονται τα tags δημιουργίας της λίστας, με πράσινο τα tags εισαγωγής του περιεχομένου της λίστας και με κίτρινη υπογράμμιση το attribute type.

```
<html> <head> <title> ordered list</title> </head>
<body>
<h2>Ordered list with different type</h2>
<ol type="1">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol> <br>
<ol type="A">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol> <br>
<ol type="a">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol> <br>
<ol type="I">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol> <br>
<ol type="i">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
</body> </html>
```

Ordered list with different type

1. Coffee
2. Tea
3. Milk

- A. Coffee
- B. Tea
- C. Milk

- a. Coffee
- b. Tea
- c. Milk

- I. Coffee
- II. Tea
- III. Milk

- i. Coffee
- ii. Tea
- iii. Milk

6.2.2 Τα tags της ταξινομημένης λίστας με κουκίδες

Από την άλλη λοιπόν μεριά έχουμε και τα tags για την λίστα με κουκίδες, τα οποία είναι: `` για την δημιουργία της λίστας και αντίστοιχα `` για την εισαγωγή των περιεχομένων της λίστας.

Τώρα για να συμπληρώσουμε το attribute της λίστας με κουκίδες θα πρέπει να αναφέρουμε το type το οποίο μπορεί να εμφανίσει στην θέση των μαύρων κουκίδων, τους κύκλους που είναι άσπροι στο κέντρο τους, τα τετράγωνα ή ακόμα μπορούμε να θέσουμε σαν τιμή στο type το “none” το οποίο δεν θα εμφανίσει κάποιο σχέδιο αρίθμησης.

Παρακάτω βλέπουμε την δημιουργία της λίστας με κουκίδες ενώ διακρίνουμε τα διαφορετικά tags από τα διαφορετικά χρώματα.

```
<html> <head><title>unordered list </title></head>
<body><h2>Unordered List with different Bullets</h2>
<ul style="list-style-type:disc">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul> <br>
<ul style="list-style-type:circle">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul> <br>
<ul style="list-style-type:square">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul> <br>
<ul style="list-style-type:none">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
</body></html>
```



Tip: Ενώ θα πρέπει ένας σχεδιαστής να γνωρίζει τις αλλαγές του attribute type παράλληλα θα πρέπει να ξέρει ότι πλέον έχει καταργηθεί για την ταξινομημένη λίστα με κουκίδες στην Html5 και χρησιμοποιείται μόνο στην ταξινομημένη λίστα με αριθμούς.

Unordered List with different Bullets

- Coffee
- Tea
- Milk

- Coffee
- Tea
- Milk

- Coffee
- Tea
- Milk

Coffee
Tea
Milk

6.2.3 Τα tags της ταξινομημένης λίστας με επεξήγηση

Για την δημιουργία της επεξηγηματικής λίστας τα tags είναι τα εξής:

- Το tag <dl> χρησιμοποιείται για την δημιουργία της λίστας
- Το tag <dt> χρησιμοποιείται για την εισαγωγή των μελών της λίστας
- Και τέλος το tag <dd> το οποίο χρησιμοποιείται για την εισαγωγή της επεξήγησης των μελών της λίστας.

Μια επεξηγηματική λίστα παρουσιάζεται παρακάτω:

```
<html> <head><title> description list </title> </head>
<body>
<h2>A Description List</h2>
<dl>
  <dt>Coffee</dt>
  <dd>- black hot drink</dd>
  <dt>Milk</dt>
  <dd>- white cold drink</dd>
</dl>
</body></html>
```


A Description List

Coffee
- black hot drink
Milk
- white cold drink

6.2.4 Δημιουργία εμφωλευμένης λίστας

Εμφωλευμένη λίστα σημαίνει λίστα μέσα σε ήδη υπάρχουσα λίστα. Η εμφωλευμένη λίστα δεν είναι κάτι δύσκολο για να δημιουργηθεί έτσι λοιπόν παρακάτω παρουσιάζεται ο τρόπος κατασκευής της:

```
<html> <head> <title> nested list </title></head>
<body>
<h2>A Nested List</h2>
<ul>
<li>Coffee</li>
<li>Tea
<ul>
<li>Black tea</li>
<li>Green tea</li>
</ul>
</li>
<li>Milk</li>
</ul>
</body></html>
```

A Nested List

- Coffee
- Tea
 - Black tea
 - Green tea
- Milk



Tip: Τελειώνοντας με το κεφάλαιο με την δημιουργία της λίστας θα πρέπει να αναφέρουμε πως οι λίστες εκτός από την χρήση τους ως τρόπο ταξινόμησης και εμφάνισης του περιεχομένου της σελίδας, χρησιμοποιούνται κυρίως και στην κατασκευή των μενού πλοήγησης της σελίδας, αυτό όμως είναι κάτι για το οποίο θα μιλήσουμε σε επόμενο κεφάλαιο.

6.3 Δημιουργία στοιχείων ομαδοποίησης

Με τον όρο στοιχεία ομαδοποίησης εννοούμε όλα εκείνα τα tags που είναι χρήσιμα για την ομαδοποίηση του περιεχομένου της σελίδας. Θα πρέπει να γνωρίζουμε ότι όλα τα στοιχεία της Html χωρίζονται σε δύο κατηγορίες:

- Τα στοιχεία block
- Και τα στοιχεία inline

Για τις παραπάνω κατηγορίες θα αναφερθούμε ξεχωριστά στις παρακάτω υποπαραγράφους.

6.3.1 Τα στοιχεία block

Τα στοιχεία block εμφανίζονται στην οθόνη σαν να έχουν μια κενή γραμμή πριν και μετά το περιεχόμενο. Για παράδειγμα τα tags: `<p>`, `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>`, ``, ``, `<dl>`, `<pre>`, `<hr>`, `<blockquote>` και `<address>` είναι κατεξοχήν στοιχεία block για τον λόγο αυτό όταν εμφανίζεται το περιεχόμενο τους στην οθόνη ξεκινά σε μια νέα γραμμή ενώ επίσης και με τα υπόλοιπα στοιχεία που ακολουθούν χωρίζονται με μια κενή γραμμή. Όπως καταλαβαίνουμε λοιπόν τα block στοιχεία αποτελούνται κυρίως από τα tags της δημιουργίας παραγράφου (`<p>`), επικεφαλίδων (`<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>`), αποσπάσματος (`<blockquote>`), μορφοποίησης κειμένου σαν διεύθυνση (`<address>`), κ.α..

6.3.2 Τα στοιχεία inline

Από την άλλη μεριά τα στοιχεία inline συνήθως μπορούν να εμφανιστούν μέσα σε προτάσεις κειμένου ενώ δεν είναι απαραίτητο να ξεκινούν σε νέα γραμμή καθώς τα περισσότερα inline στοιχεία έχουν σαν σκοπό την «επιτόπια» μορφοποίηση του κειμένου, και με τον όρο «επιτόπια» εννοούμε την μορφοποίηση στην γραμματοσειρά (μέγεθος , χρώμα κειμένου κ.τ.λ.) και όχι την

μετατροπή της κατάστασης του απλού κειμένου σε επικεφαλίδες ή αποσπάσματα άρθρων. Μερικά από τα inline στοιχεία είναι τα εξής tags:

, <i>, <u>, , , <sup>, <sub>, <big>, <small>, , <ins>, , <code>, <cite>, <dfn>, <kbd>, και το <var>.

Παρακάτω παρουσιάζονται η λειτουργία των στοιχείων block και inline:

```
<html><head> <title>block elements </title></head>
```

```
<body>
```

```
<h1>This is heading 1</h1>
```

```
<h2>This is heading 2</h2>
```

```
<h3>This is heading 3</h3>
```

```
<h4>This is heading 4</h4>
```

```
<h5>This is heading 5</h5>
```

```
<h6>This is heading 6</h6>
```

```
<pre>
```

Text in a pre element is displayed in a fixed-width font, and it preserves both spaces and line breaks

```
</pre>
```

```
<p>This is some text in a paragraph.</p>
```

```
<blockquote>
```

For 50 years, WWF has been protecting the future of nature. The world's leading conservation organization, WWF works in 100 countries and is supported by 1.2 million members in the United States and close to 5 million globally.

```
</blockquote>
```

```
<hr>
```

```
<address>
```

Written by Jon Doe.

Visit us at:

Example.com

Box 564, Disneyland

USA

```
</address>
```

```
</body></html>
```

6.3.3 To block tag <div>

Μπορεί τα στοιχεία block να περιλαμβάνουν τα tags μορφοποίησης της επικεφαλίδας , της παραγράφου και άλλων, αλλά θα πρέπει να ξέρουμε ότι δεν ασχολούνται μόνο με την μορφοποίηση αφού ένα από τα κυριότερα και χρήσιμα tags ομαδοποίησης στην Html είναι και το tag <div>. Επομένως όπως προαναφέραμε μέσω του tag <div> η ομαδοποίηση «αποκτά οικογένεια» στα στοιχεία block.

Ας πούμε λίγα λόγια λοιπόν για το tag <div>.

Το tag <div> είναι ένα από τα πιο χρήσιμα εργαλεία στην Html και αυτό γιατί παρέχει την δυνατότητα στον σχεδιαστή να ομαδοποιήσει πολλά Html στοιχεία μαζί, είτε είναι inline είτε είναι στοιχεία block. Επίσης παρέχει την δυνατότητα σχεδίασης του σχήματος ή αλλιώς υπόβαθρου της σελίδας (page layout) χωρίζοντας το σε 4 ίσα μέρη (top – right – left – bottom).



Tip: Το tag <div> δεν μεταβάλλει την μορφοποίηση των στοιχείων που περικλείει αλλά συμβάλει στην μορφοποίηση όταν χρησιμοποιηθεί μέσω των διαδοχικών φύλλων στυλ Css.

```
<html> <head> <title>HTML div Tag</title> </head>
<body>
<!-- First group of tags -->
<div style="color:red">
<h4>This is first group</h4>
<p>Following is a list of vegetables</p>
<ul>
<li>Beetroot</li>
<li>Ginger</li>
<li>Potato</li>
<li>Radish</li>
</ul>
</div>
```

```
<!-- Second group of tags -->
<div style="color:green">
<h4>This is second group</h4>
<p>Following is a list of fruits</p>
<ul>
<li>Apple</li>
<li>Banana</li>
<li>Mango</li>
<li>Strawberry</li>
</ul>
</div>
</body></html>
```

THIS IS FIRST GROUP

Following is a list of vegetables

- Beetroot
- Ginger
- Potato
- Radish

THIS IS SECOND GROUP

Following is a list of fruits

- Apple
- Banana
- Mango
- Strawberry

6.3.4 Το inline tag

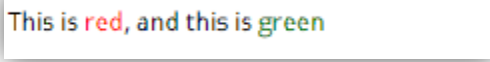
Όπως τα στοιχεία block έχουν στην «οικογένεια» τους ένα από τα πιο σημαντικά tags, το tag <div> έτσι και στο σύνολο των στοιχείων inline υπάρχει ένα πολύ σημαντικό tag, το .

Το tag σκοπό του έχει την ομαδοποίηση των inline στοιχείων ενώ μοιάζει πολύ με το tag <div> καθώς ούτε αυτό μεταβάλλει την μορφοποίηση του περιεχομένου αν ταυτόχρονα δεν συνδυαστεί με τα διαδοχικά φύλλα στυλ Css.

Παρακάτω παρουσιάζεται η λειτουργία του tag :

```
<html> <head> <title> HTML span Tag </title> </head>
<body>
<p>This is <span style="color:red"> red </span> and this is
<span style="color:green"> green </span></p>
</body>
</html>
```

Παρακάτω παρουσιάζεται το παράδειγμα όπως θα εμφανιζόταν στην οθόνη του χρήστη της σελίδας:



This is red, and this is green

Τα `style="color:red"` / `style="color:green"` είναι τα attributes του tag που μορφοποιούν το χρώμα της γραμματοσειράς του περιεχομένου που περικλείεται από το tag, σε κόκκινο.

7 Εισαγωγή εικόνας, υπερσυνδέσμου και πολυμέσων (media)

Μέχρι τώρα έχουμε δείξει διάφορους τρόπους μορφοποίησης, παρουσίασης κειμένου σε σελίδα αλλά και τρόπους ομαδοποίησης ορισμένων στοιχείων κειμένου όπως για παράδειγμα έχουμε αναφερθεί στους πίνακες και στις λίστες μέσω των οποίων παρουσιάζεται το περιεχόμενο της σελίδας αλλά και τα tags <div> / μέσω των οποίων ομαδοποιούμε τα παραπάνω tags. Σε μια σελίδα όμως δεν μπορούμε να έχουμε μόνο κείμενο.

Το περιεχόμενο της σελίδας μπορεί να προβάλλεται είτε μέσα από εικόνες, είτε μέσα από υπερσυνδέσμους και πολυμέσα δηλαδή video, ήχος κ.τ.λ..

Σε αυτήν λοιπόν την ενότητα θα ασχοληθούμε με τον τρόπο που εισάγουμε εικόνες, υπερσυνδέσμους (links) και video ή αρχεία ήχου.

7.1 Εισαγωγή και μορφοποίηση εικόνων

Για την εισαγωγή των εικόνων χρησιμοποιούμε το tag ``. Το tag `` είναι ένα άδειο tag το οποίο συμπληρώνεται με τα ανάλογα attributes. Επίσης θα πρέπει να αναφερθεί ότι το tag `` δεν «κλείνει» δηλαδή δεν έχει close tag. Για να εμφανίσουμε μια εικόνα θα πρέπει να εισάγουμε το μονοπάτι δηλαδή την διαδρομή – διεύθυνση που είναι αποθηκευμένη η εικόνα. Μια εικόνα μπορεί να εισάγεται στην σελίδα μας από τον τοπικό υπολογιστή του σχεδιαστή και τότε χρησιμοποιούμε την διεύθυνση του τοπικού φακέλου στον οποίο είναι αποθηκευμένη ή μπορεί η εικόνα να εισάγεται από το διαδίκτυο οπότε τότε χρησιμοποιούμε την url διεύθυνση. Το τοπικό μονοπάτι ή η url διεύθυνση της εικόνας εισάγεται με το attribute του tag ``, το src (από το source δηλαδή πηγή).

``

Με τον παραπάνω τρόπο συντάσσεται το tag `` και όπως βλέπουμε σαν attributes έχει δύο:

- Το `src="path/url"` που είπαμε ότι σκοπός του είναι να εισάγει την διεύθυνση της εικόνας
- Και το `alt="some_text"` του οποίου σκοπός είναι να δίνει ένα τίτλο στην εικόνα, ο οποίος μάλιστα εμφανίζεται όταν για κάποιο λόγο δεν μπορεί να εμφανιστεί η εικόνα στην σελίδα.

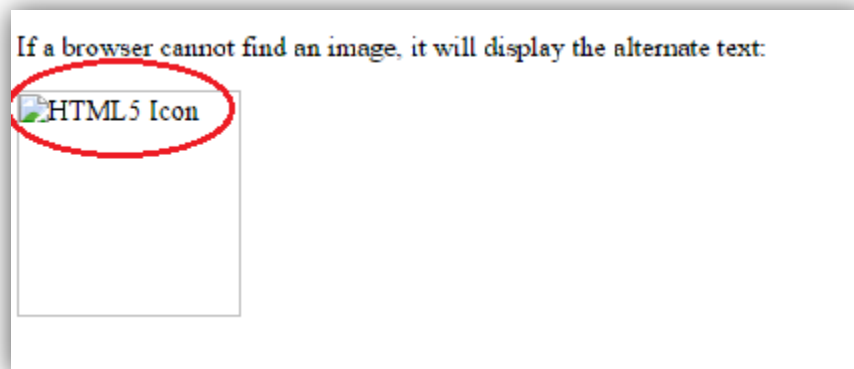
7.1.1 Το attribute alt

Όπως είπαμε και παραπάνω το attribute alt χρησιμοποιείται από τους σχεδιαστές για να δώσουν ένα μήνυμα ή ένα τίτλο σε μια εικόνα, το οποίο θα εμφανιστεί σε περίπτωση που δεν μπορέσει να προβληθεί η εικόνα στην σελίδα.

Το attribute alt λοιπόν παρέχει γενικές πληροφορίες για μια εικόνα η οποία αδυνατεί να προβληθεί στην σελίδα είτε λόγω χαμηλής ταχύτητας δικτύου, λάθους στην σύνταξη της διεύθυνσης της εικόνας είτε ακόμα λόγω της χρήσης των screen readers.

```
<html> <head><title> alt attribute </title></head>
<body>
<p>If a browser cannot find an image, it will display the alternate text:</p>

</body></html>
```



7.1.2 Μορφοποίηση μεγέθους της εικόνας

Όταν εισάγουμε μια εικόνα δεν είμαστε πάντα σίγουροι σε τι διαστάσεις θα προβληθεί στην σελίδα, γι' αυτό το λόγο για να μην προκαλούνται προβλήματα κατά την φόρτωση της σελίδας από την εμφάνιση των διαστάσεων της εικόνας είναι απαραίτητο κάθε φορά να μεταβάλλουμε σε ένα λογικό μέγεθος την εικόνα μας. Η μορφοποίηση του μεγέθους της εικόνας μπορεί να γίνει με δύο τρόπους:

- ή μέσω Css κάτι για το οποίο θα μιλήσουμε αργότερα
- ή γι' αυτό που θα μιλήσουμε τώρα, μέσω του attribute style με τις ιδιότητες width για το πλάτος και height για το ύψος της εικόνας.

Στον παρακάτω κώδικα βλέπουμε πως μπορούμε να εισάγουμε δύο εικόνες ρυθμίζοντας την εμφάνιση τους σε διαφορετικές διαστάσεις όπως φαίνεται στην εικόνα. Επίσης θα πρέπει να το προσέξουμε πως στην πρώτη εικόνα οι ιδιότητες του πλάτους και του ύψους τις εικόνας μεταβάλλονται έμμεσα μέσω του attribute style, σε αντίθεση με την δεύτερη εικόνα που οι δύο ιδιότητες του μεγέθους μεταβάλλονται άμεσα χωρίς να γίνεται χρήση του attribute style.


```
<html> <head> <title> different image size</title></head>
<body>


</body></html>
```



7.1.3 Οι Screen Readers

Μια απλή επεξήγηση μπορούμε να κάνουμε και για τους screen readers που αναφέρθηκαν παραπάνω. Οι screen readers ή αλλιώς «αναγνώστες οθόνης» είναι λογισμικό, προγράμματα δηλαδή που συνήθως χρησιμοποιούνται από χρήστες με ειδικές ανάγκες που πάσχουν συνήθως από τύφλωση, οπτική αστάθεια ή ακόμα δυσκολία στην ανάγνωση, γιατί πολύ απλά το λογισμικό αυτό παρέχει την δυνατότητα κατάλληλης μεγένθησης της σελίδας με στόχο την μεγαλύτερη ευκρίνεια.

Επομένως καταλαβαίνουμε ότι οι screen readers στην ουσία «διαβάζουν» το περιεχόμενο που πρέπει να εμφανιστεί στην σελίδα μέσα από τον κώδικα και έπειτα το προβάλλουν με τις κατάλληλες ρυθμίσεις για τους χρήστες που το έχουν επιλέξει.

Πολλές φορές όπως αναφέραμε παραπάνω στην ενότητα του attribute alt, μπορεί να προκαλέσουν οι screen readers προβλήματα στην εμφάνιση ορισμένων εικόνων στην σελίδα, αυτό όμως δεν σημαίνει ότι σαν λογισμικό δεν μπορούν να «διαβάσουν» στον κώδικα το attribute alt και να το προβάλλουν στην σελίδα έναντι της προβληματικής εικόνας.

7.1.4 Στοιχίση εικόνας έναντι κειμένου (ιδιότητα float)

Κατά την εισαγωγή της εικόνας εκτός από τις διαστάσεις της θα πρέπει να μας ενδιαφέρει και η στοιχίση της σε σχέση με το κείμενο της σελίδας, αν δηλαδή το κείμενο θα εμφανίζεται πάντα αριστερά ή δεξιά από την εικόνα. Για να το πετύχουμε αυτό χρησιμοποιούμε την ιδιότητα float στο attribute style παίρνοντας τιμές left για να στοιχιστεί το κείμενο στα δεξιά της εικόνας (άρα η εικόνα θα εμφανίζεται στο δεξιό μέρος της σελίδας) ή right για να στοιχιστεί στα αριστερά της εικόνας οπότε η εικόνα θα εμφανίζεται στο αριστερό μέρος της σελίδας.

Η ιδιότητα float παρουσιάζεται στον παρακάτω κώδικα:

```
<html> <head><title>float image </title></head>
<body>
<p><strong>Float the image to the right:</strong></p>
<p>

A paragraph with a floating image. A paragraph with a floating image. A paragraph with a
floating image.
</p>
<p><strong>Float the image to the left:</strong></p>
<p>

A paragraph with a floating image. A paragraph with a floating image. A paragraph with a
floating image.
</p>
</body></html>
```

Float the image to the right:

A paragraph with a floating image. A paragraph with a floating image. A paragraph with a floating image.



Float the image to the left:



A paragraph with a floating image. A paragraph with a floating image. A paragraph with a floating image.

7.1.5 Εικονοχάρτες (image maps)

Οι εικονοχάρτες όπως θα μπορούσαμε να μεταφράσουμε την φράση “image maps” είναι εικόνες που αποτελούνται από «ενεργές» περιοχές. Με τον όρο «ενεργές» εννοούμε ότι στις διαστάσεις μιας εικόνας υπάρχει μια περιοχή που έχουμε δώσει την δυνατότητα να παίρνει ένα τίτλο ή να μας εμφανίζει κάποιο κείμενο ή κάποια άλλη εικόνα, όταν κάνουμε κλικ με το δείκτη του ποντικιού πάνω σε αυτήν την περιοχή της εικόνας. Καταλαβαίνουμε ότι μια περιοχή σε μια εικόνα είναι clickable δηλαδή ενεργή όταν περνάει ο δείκτης του ποντικιού πάνω από την περιοχή αυτή και αμέσως από βέλος μετατρέπεται σε χεράκι. Για να δημιουργήσουμε μια ενεργή περιοχή πάνω σε μια εικόνα χρησιμοποιούμε το tag <map> το οποίο ταυτόχρονα συνδέεται με το attribute usemap του tag . Παρακάτω παρουσιάζουμε την δημιουργία των ενεργών περιοχών μιας εικόνας μέσω ενός παραδείγματος με σκοπό να καταλάβουμε την σύνταξη του tag <map> και την λειτουργία του.

```
<html> <head><title> image maps </title></head>
<body>
<p>Click on the sun or on one of the planets to watch it closer:</p>

<map name="planetmap">
  <area shape="rect" coords="0,0,82,126" alt="Sun" href="sun.htm">
  <area shape="circle" coords="90,58,3" alt="Mercury" href="mercur.htm">
  <area shape="circle" coords="124,58,8" alt="Venus" href="venus.htm">
</map>
</body></html>
```

Ας αναλύσουμε τώρα τον παραπάνω κώδικα.

Βλέπουμε πως με το tag εισάγουμε μια εικόνα με διεύθυνση `src="planets.gif"`, διαστάσεις `style="width:145px;height:126px;"` και τίτλο σε περίπτωση που δεν μπορέσει να εμφανιστεί η εικόνα τον εξής: `alt="Planets"`. Παρατηρούμε όμως πως μέσα στο tag υπάρχει ένα ακόμα attribute το usemap. Το usemap είναι το attribute που χρησιμοποιείται για να «καλέσει» το tag <map> που δημιουργεί τις ενεργές περιοχές στην εικόνα. Το tag <map> τώρα χρησιμοποιεί το attribute name με το οποίο στην ουσία δίνουμε ένα όνομα στο tag <map> το οποίο θα χρησιμοποιηθεί για το καλέσουμε με το usemap.

Μέσα στο <map> για να δημιουργήσουμε τις ενεργές περιοχές χρησιμοποιούμε το tag <area> το οποίο χρησιμοποιεί τα εξής attributes:

- Το **shape="rect"** το οποίο ορίζει ότι η περιοχή της εικόνας που θα είναι ενεργή θα έχει ορθογώνιο σχήμα,
- Το **coords="0,0,82,126"** με το οποίο δίνουμε τις συντεταγμένες της ορθογώνιας ενεργής περιοχής στην εικόνα
- Το **href="sun.htm"** το οποίο είναι ένα link δηλαδή ένας υπερσύνδεσμος στον οποίο μεταβαίνουμε αν κάνουμε κλικ στην ενεργή περιοχή της εικόνας.

Αρα σε μια πιο ελεύθερη ανάλυση του παραπάνω κώδικα έχουμε τα εξής:

Εισάγουμε μια εικόνα με το tag . Μέσα στο tag καλούμε με το usemap τον χάρτη των ενεργών περιοχών το tag <map> που έχει πάρει το όνομα «planetmap». Από την μεριά τώρα του χάρτη ορίζουμε σε κάποιες συντεταγμένες της εικόνας ορθογώνιες ή κυκλικές περιοχές στις οποίες όταν περνάει ο δείκτης του ποντικιού μετατρέπεται σε χεράκι και μας επιτρέπει να κάνουμε κλικ. Αν κάνουμε κλικ στις περιοχές αυτές μεταβαίνουμε μέσω των υπερσυνδέσμων **href** σε κάποιες σελίδες που έχουμε ορίσει εμείς και στις οποίες έχουμε δώσει ένα τίτλο που θα εμφανίζεται στην θέση τους σε περίπτωση που δεν μπορούν για κάποιο λόγο να προβληθούν στην οθόνη.

7.2 Εισαγωγή υπερσυνδέσμων (links)

Οι υπερσύνδεσμοι υπάρχουν παντού , τους συναντάμε σε όλες τις σελίδες του διαδικτύου καθώς είναι ο τρόπος μετακίνησης και πλοήγησης μεταξύ των ιστοσελίδων.

Ένας υπερσύνδεσμος μπορεί να είναι κείμενο, εικόνα, κουμπί κ.α.

Τα links κατασκευάζονται μέσω του tag <a> που χρησιμοποιεί το χρήσιμο attribute <href> όπως φαίνεται παρακάτω:

```
<a href="url">Link text</a>
```

Όπως παρατηρούμε και από την σύνταξη του tag <a> το σημαντικό είναι να δώσουμε τη σωστή διεύθυνση – url μέσω του attribute href ώστε να είναι δυνατή η μετάβαση από την μια σελίδα

στην άλλη ή από ένα σημείο της σελίδας σε ένα άλλο (bookmarks) δηλαδή δημιουργία σελιδοδεικτών. Για τους σελιδοδείκτες όμως θα μιλήσουμε παρακάτω.

Σαν παρατήρηση έχουμε να προσθέσουμε την διαφορά μεταξύ απόλυτης διεύθυνσης και σχετικής.

Χρησιμοποιούμε απόλυτη διεύθυνση όταν θέλουμε να μεταβούμε μέσω του υπερσυνδέσμου σε μια τελείως διαφορετική σελίδα σε κάποιον άλλο server. Στην περίπτωση αυτή το απόλυτο url συντάσσεται ως εξής:

«http://www.*.com/html/»

Διαφορετικά αν θέλουμε να μετακινηθούμε σε τοπικό επίπεδο δηλαδή σε μια διαφορετική σελίδα μέσα στον φάκελο στον οποίο έχουμε αποθηκεύσει τα αρχεία Html και τα οποία έχουμε ανεβάσει σε κάποιον server αρκεί να γράψουμε στην διεύθυνση το όνομα του αρχείου html της σελίδας δηλαδή στην οποία θέλουμε να μετακινηθούμε.

7.2.1 Μορφοποίηση υπερσυνδέσμων

Όταν ο δείκτης του ποντικιού περνάει πάνω από έναν υπερσύνδεσμο δύο πράγματα συμβαίνουν:

- ο δείκτης μετατρέπεται σε «χεράκι» που σου δίνει την δυνατότητα να πατήσεις πάνω στο link
- και το ίδιο το link αλλάζει χρώμα ανάλογα με την κατάσταση στην οποία βρίσκεται.

Εξ' ορισμού οι υπερσύνδεσμοι στην περίπτωση που ο χρήστης δεν έχει κάνει κλικ πάνω τους θα εμφανιστούν με υπογράμμιση και χρώμα γραμματοσειράς μπλέ στην αντίθετη περίπτωση παραμένουν υπογεγραμμένοι αλλά το χρώμα γραμματοσειράς μεταβάλλεται σε μωβ.

Επίσης υπάρχει και η περίπτωση του ενεργού υπερσυνδέσμου κατά την οποία το χρώμα γραμματοσειράς γίνεται κόκκινο.

Παρόλα αυτά ένας σχεδιαστής μπορεί να αλλάξει τις προεπιλογές εμφάνισης των υπερσυνδέσμων χρησιμοποιώντας το tag style όπως φαίνεται παρακάτω:

```
<html> <head> <title> style tag </title>
<style>
a:link {
    color:#000000;
    background-color:transparent;
    text-decoration:none;
}
a:visited {
    color:#000000;
    background-color:transparent;
    text-decoration:none;
}
a:hover {
    color:#ff0000;
    background-color:transparent;
    text-decoration:underline;
}
a:active {
    color:#ff0000;
    background-color:transparent;
    text-decoration:underline;
}
</style>
</head>
<body>
<p>You can change the default colors of links</p>
<a href="html_images.asp">HTML Images</a>
</body> </html>
```

Παρατηρούμε ότι το tag <style> γράφεται ανάμεσα στο tag <head> και όχι στο <body> γιατί πολύ απλά είναι tag μορφοποίησης και όχι attribute που θα πρέπει να εισάγεται μέσα σε άλλα tags του <body>.

Επιπλέον παρατηρούμε πως συντάσσονται οι διαφορετικές περιπτώσεις του tag <a> των υπερσυνδέσμων. Για παράδειγμα για να μορφοποιήσουμε τον υπερσύνδεσμο στην περίπτωση που κάνει κλικ ο χρήστης τότε θα γράψουμε:

```
a:visited {  
    color:#000000;  
    background-color:transparent;  
    text-decoration:none;  
}
```

Άρα το tag <a> μετατρέπεται στην ουσία σε attribute του tag <style> παίρνοντας τις παρακάτω τιμές:

- **visited** είναι η περίπτωση που αλλάζει η μορφοποίηση γιατί ο χρήστης έχει επισκεφτεί προηγουμένως την σελίδα,
- **active** όπου είναι η περίπτωση που έχει γίνει κλικ στο link από τον χρήστη και είναι ενεργός ο υπερσύνδεσμος,
- **link** είναι η αρχική μορφοποίηση του υπερσυνδέσμου
- και τέλος **hover** που είναι η περίπτωση στην οποία περνά απλα ο δείκτης του ποντικιού πάνω από το link και η μορφοποίηση του υπερσυνδέσμου αλλάζει

Στις παραπάνω λοιπόν περιπτώσεις όπου αλλάζει η κατάσταση στην οποία βρίσκεται ο υπερσύνδεσμος, μας δίνεται το δικαίωμα να αλλάξουμε και την εμφάνιση του link.

7.2.2 To attribute target

Το attribute target είναι αυτό που καθορίζει που θα ανοίξει για παράδειγμα η νέα σελίδα στην οποία θα οδηγηθούμε αν πατήσουμε τον υπερσύνδεσμο, αν δηλαδή η νέα σελίδα θα ανοίξει σε νέα καρτέλα ή σε νέο παράθυρο του browser.

```
<a href="http://www.some_page.com/" target="_blank">Visit some page!</a>
```

Οι τιμές που μπορεί να πάρει είναι οι εξής:

- **_blank**: ανοίγει τον υπερσύνδεσμο σε νέα καρτέλα ή παράθυρο

- **_self:** ανοίγει τον υπερσύνδεσμο στην frame που ανήκει δηλαδή αν το link είναι γραμμένο μέσα σε ένα πλαίσιο τότε και η σελίδα θα ανοίξει μέσα στο ίδιο πλαίσιο (αυτό είναι προεπιλογή)
- **_parent:** ανοίγει τον υπερσύνδεσμο στο ανώτερο πλαίσιο της σελίδας αν υπάρχουν εμφωλευμένα πλαίσια το ένα μέσα στο άλλο τότε η σελίδα θα ανοίξει στο ανώτερο πλαίσιο. Για τα πλαίσια όμως θα αναφερθούμε αναλυτικότερα σε επομενες παραγράφους.
- **_top:** ανοίγει τον υπερσύνδεσμο σε ολόκληρο το παράθυρο της σελίδας
- **_framename:** ανοίγει τον υπερσύνδεσμο στο πλαίσιο που έχουμε δημιουργήσει εμείς και που το καλούμε με το όνομα που του έχουμε δώσει.

7.2.3 Χρήση εικόνας ως υπερσύνδεσμος

Σε παραπάνω υποπαράγραφο αναφερθήκαμε στην εισαγωγή εικόνων σε μια σελίδα και μάλιστα είπαμε χαρακτηριστικά ότι μια εικόνα μπορεί να χρησιμοποιηθεί και ως υπερσύνδεσμος, δηλαδή να μπορούμε να κάνουμε κλικ σε μια εικόνα ώστε να μεταβούμε σε μια άλλη σελίδα.

Αυτό είναι πολύ εύκολο να πραγματοποιηθεί αρκεί να εισάγουμε το tag της εικόνας μέσα στο tag <a> του υπερσυνδέσμου όπως φαίνεται στο παράδειγμα:

```
<a href="default.asp">
  
</a>
```



Tip: Παρατηρούμε πως μέσα στο tag γίνεται χρήση ενός νέου attribute του border με τιμή 0 και αυτό γιατί έτσι δηλώνουμε πως αφού η εικόνα χρησιμοποιείται ως υπερσύνδεσμος δεν θέλουμε να εμφανιστεί γύρω από την εικόνα περίγραμμα κάτι που γίνεται συνήθως στις παλαιότερες εκδόσεις του browser Internet Explorer 9.

7.2.4 Δημιουργία σελιδοδείκτη (bookmark)

Σε αυτήν την παράγραφο θα μιλήσουμε για ακόμη ένα θέμα που έχει ήδη αναφερθεί και αυτό είναι η δημιουργία των σελιδοδεικτών. Οι σελιδοδείκτες όπως και στα βιβλία είναι πολύ χρήσιμοι γιατί μας παρέχουν την δυνατότητα να μεταφερθούμε στο σημείο του περιεχομένου που θέλουμε απευθείας χωρίς να χρειάζεται να προσπελάσουμε όλο το βιβλίο. Έτσι και στις ιστοσελίδες το να έχουμε σελιδοδείκτες είναι ένα πλεονέκτημα γιατί μπορούμε να μεταφερθούμε κατευθείαν στο σημείο της σελίδας που επιθυμούμε, ειδικά όταν η σελίδα είναι πολύ μεγάλη σε περιεχόμενο.

Για να αποκτήσει η σελίδα σελιδοδείκτη θα πρέπει πρώτα να τον κατασκευάσουμε και έπειτα να τον συνδέσουμε με έναν υπερσύνδεσμο. Όταν ο χρήστης κάνει κλικ στον υπερσύνδεσμο τότε η σελίδα θα κυλίσει στο σημείο που έχουμε ορίσει τον σελιδοδείκτη.

Ένας σελιδοδείκτης μπορεί να οριστεί σε οποιοδήποτε σημείο της σελίδας π.χ. σε μια επικεφαλίδα, ή στην αρχή μιας παραγράφου... την κατασκευή ενός σελιδοδείκτη σε μια επικεφαλίδα βλέπουμε στο παρακάτω παράδειγμα:

```
<h2 id="bookmark1">how to make bookmarks</h2>
```

Στον παραπάνω κώδικα ορίζουμε μέσω του attribute id έναν σελιδοδείκτη ο οποίος θα λέγεται «bookmark1» και η θέση του θα βρίσκεται στην συγκεκριμένη επικεφαλίδα της σελίδας.

Τώρα όμως θα πρέπει να συνδέσουμε τον σελιδοδείκτη με έναν υπερσύνδεσμο ως εξής:

```
<a href="#bookmark1">connect the bookmark with the link</a>
```

Στην γραμμή αυτή του κώδικα διακρίνουμε ότι η σύνδεση επιτυγχάνεται χρησιμοποιώντας το tag <a> του υπερσυνδέσμου δίνοντας σαν διεύθυνση του attribute href το όνομα που έχουμε δώσει στον σελιδοδείκτη. Έτσι καταλαβαίνουμε ότι αν ο χρήστης κάνει κλικ στον υπερσύνδεσμο, αυτός αυτόματα καλεί με το όνομα τον σελιδοδείκτη που έχουμε ορίσει εμείς και η σελίδα κυλάει στο σημείο που βρίσκεται ο σελιδοδείκτης δηλαδή στην συγκεκριμένη επικεφαλίδα.



Tip: Εκτός από το να χρησιμοποιούμε τους σελιδοδείκτες για να μας μεταφέρουν σε ένα σημείο της σελίδας στην οποία βρισκόμαστε, μας δίνεται η δυνατότητα μέσα από την χρήση των σελιδοδεικτών να μετακινηθούμε και σε ένα σημείο άλλης σελίδας. Η διαφορά υπάρχει στην διεύθυνση που θα δώσουμε στον υπερσύνδεσμο. Δηλαδή αν θέλουμε να κινηθούμε στην ίδια σελίδα που βρισκόμαστε αρκεί να βάλουμε σαν διεύθυνση του link το όνομα του σελιδοδείκτη έπειτα από το σύμβολο “ # ”, αν όμως θέλουμε να μεταβούμε σε κάποιο σελιδοδείκτη άλλης σελίδας τότε σαν διεύθυνση γράφουμε το όνομα της σελίδας ακολουθούμενο από το σύμβολο “ # ” και το όνομα του σελιδοδείκτη:

```
<a href="example.html#bookmark1">bookmark in another page</a>
```

Μερικές φορές οι σχεδιαστές αντιμετωπίζουν προβλήματα με την σύνδεση του σελιδοδείκτη με τον υπερσύνδεσμο, αυτό συμβαίνει γιατί χρειάζεται συνήθως να εισάγουμε σαν διεύθυνση ολόκληρο το url μιας σελίδας ώστε να είναι πιο εύκολο στην εύρεση της σελίδας.

7.3 Εισαγωγή πολυμέσων (media)

Με τον όρο πολυμέσα δεν εννοούμε μόνο εικόνα αλλά επίσης video και ήχο. Σε μια σελίδα λοιπόν εκτός από εικόνες μπορεί να παρουσιάζεται το περιεχόμενο της σελίδας μέσω video και αρχείων ήχου, δηλαδή μπορούμε να έχουμε για παράδειγμα μουσική στο υπόβαθρό της σελίδας με έτοιμο player όπου θα παρέχει την δυνατότητα στον χρήστη να μεταβάλλει την ένταση του ήχου, την ταχύτητα του ρυθμού ή ακόμα και να σταματήσει την μουσική αν δεν θέλει να ακούει. Εμείς συμβουλεύουμε πάντως πως όσο ωραίο και να είναι να ακούει την μουσική ο χρήστης όταν μπαίνει στην σελίδα αφού χρησιμοποιείται σαν effect στην σελίδα, καλό είναι να μην το επιχειρεί ο χρήστης γιατί συνήθως λειτουργικά «βαραίνει» κατά την φόρτωση την σελίδα αλλά και πρακτικά δεν είναι βολικό για τον χρήστη αν δεν του αρέσει η συγκεκριμένη μουσική.

Θα πρέπει σε αυτό το κεφάλαιο να τονίσουμε την μεγάλη σημασία που έχει η νέα έκδοση της Html. Πριν την Html5 δεν υπήρχε κάποιος standard τρόπος για την εισαγωγή πολυμέσων σε μια σελίδα γι' αυτό και τις περισσότερες φορές χρησιμοποιούσαν τα λεγόμενα plug-in όπως τα flash, κάτι που πολλές φορές είχε προβλήματα υποστήριξης από τους διάφορους browsers ή προβλήματα φόρτωσης στην σελίδα ειδικά αν η ταχύτητα του διακομιστή ήταν ιδιαίτερα χαμηλή.

7.3.1 Εισαγωγή ήχου (audio)

Όπως αναφέραμε και παραπάνω χάρη στην Html5 μπορούμε να έχουμε έναν standard πλέον τρόπο για την εισαγωγή ήχου σε μια σελίδα. Αυτός ο τρόπος είναι το tag <audio> το οποίο συντάσσεται ως εξής:

```
<audio controls>  
  <source src="file.ogg" type="audio/ogg">  
  <source src="file.mp3" type="audio/mpeg">  
Your browser does not support the audio element.  
</audio>
```

Οι παρατηρήσεις που έχουμε να κάνουμε για αυτό το tag είναι το attribute “controls” που χρησιμοποιεί μέσω του οποίου δημιουργείται το μενού του player δηλαδή προσθέτει τα κουμπιά της αναπαραγωγής του κομματιού, της παύσης και της ακουστικής έντασης (play, pause, volume).

Επιπλέον διακρίνουμε πως μέσα στο tag <audio> χρησιμοποιείται και το tag <source> μέσω του οποίου δίνουμε το όνομα και την διεύθυνση του αρχείου ήχου που θέλουμε να προστεθεί στην σελίδα. Ενώ μέσω του attribute type δίνουμε στον browser να καταλάβει τι τύπος αρχείου είναι το αρχείο ήχου που προσθέτουμε. Συνήθως καλό είναι να εισάγουμε το αρχείο ήχου και με τους δύο τύπους αρχείων το .ogg και το .mpeg έτσι ώστε όποιον browser και αν χρησιμοποιεί ο χρήστης να μπορεί να αναγνωρίσει το αρχείο ήχου και να μπορεί να το αναπαράγει στην σελίδα. Στην περίπτωση που δεν μπορεί να το αναγνωρίσει ο browser τότε εμφανίζει στην θέση του player το μήνυμα που έχει γραφτεί ανάμεσα στα tags <audio>, στην προκειμένη περίπτωση το μήνυμα: «**Your browser does not support the audio element**».

Τέλος θα πρέπει να ξέρουμε ότι όταν έχουμε πολλά <source> tags με διαφορετικά ονόματα και διευθύνσεις αρχείων ήχου ο browser θα επιλέξει να αναπαράγει αυτό που αναγνωρίζει πρώτο όσον αφορά τον τύπο του αρχείου.

7.3.2 Εισαγωγή video

Αντίστοιχα με την εισαγωγή του ήχου σε μια σελίδα έτσι και κατά την εισαγωγή video θα έχουμε το εξής tag:

```
<video width="320" height="240" controls autoplay>  
  <source src="movie.mp4" type="video/mp4">  
  <source src="movie.ogg" type="video/ogg">  
Your browser does not support the video tag.  
</video>
```

Το attribute controls και στο tag <video> κάνει την ίδια δουλειά όπως και στο tag <audio> προσθέτει δηλαδή το μενού του player ή αλλιώς τα κουμπιά του play, του pause και του volume.

Όπως βλέπουμε όμως στο tag <video> έχουν προστεθεί τα attributes του ύψους (height) και του πλάτους (width) τα οποία με την σειρά τους ορίζουν τις διαστάσεις του player.



Tip: Αν δεν οριστούν οι διαστάσεις του player τότε η σελίδα θα αντιμετωπίσει πρόβλημα καθώς θα αρχίσει ή να κινείται προς τα κάτω (scrolling) ή να «τρέμει» (flicker) κατά την διάρκεια φόρτωσης του video.

Το attribute autoplay όπως βλέπουμε χρησιμοποιείται και αυτό στο tag <video> ενώ η λειτουργία του έχει να κάνει με την αυτόματη αναπαραγωγή του video κατά την φόρτωση της σελίδας. Άρα αν χρησιμοποιήσουμε το autoplay όταν θα φορτώνει η σελίδα το video θα ξεκινήσει μόνο του αυτόματα χωρίς να χρειάζεται να πατήσουμε το play. Θα πρέπει να γνωρίζουμε ότι το attribute autoplay δεν αναγνωρίζεται και δεν λειτουργεί στους browsers Safari και Opera αλλά και στις συσκευές ipad και iphone.

7.3.3 Εισαγωγή video από Youtube

Στην προηγούμενη παράγραφο μιλήσαμε για την εισαγωγή των video σε μια σελίδα, video όμως τα οποία πρέπει συνήθως να τα μετατρέπουμε σε συγκεκριμένες μορφές αρχείων ώστε να τα αναγνωρίζει ο browser μέσα από το tag <video>.

Για να μην μπαίνουμε λοιπόν στην χρονοβόρα διαδικασία της μετατροπής των video μπορούμε να τα «ανεβάζουμε» (upload) στην ηλεκτρονική βιβλιοθήκη του youtube και από εκεί να τα ενσωματώσουμε στην σελίδα μας.

Μια url διεύθυνση ενός video στο Youtube αποτελείται από ένα id δηλαδή ένα αναγνωριστικό όπως για παράδειγμα τον κωδικό που φαίνεται παρακάτω με κόκκινη γραμματοσειρά:

«<https://www.youtube.com/watch?v=9gTw2EDkaDQ>»

Το id αυτό είναι ένα πολύ σημαντικό στοιχείο κατά την διαδικασία της ενσωμάτωσης του video στη σελίδα. Για την ενσωμάτωση του video χρησιμοποιούμε το tag <iframe> του οποίου η σύνταξη είναι ως εξής:

```
<iframe width="420" height="315"
src="http://www.youtube.com/embed/XGSy3_Czz8k?autoplay=1">
</iframe>
```

Το tag <iframe> δημιουργεί ένα πλαίσιο στην σελίδα μέσα στο οποίο μας δίνεται η δυνατότητα να ενσωματώσουμε τα πολυμέσα που θέλουμε και κυρίως τα video. Και το tag <iframe> χρησιμοποιεί τα attributes των διαστάσεων, το height δηλαδή του ύψους και το weight του πλάτους. Το σημαντικό όμως κομμάτι του <iframe> είναι η εισαγωγή της url διεύθυνσης του video μέσω του src.

Για την εισαγωγή της διεύθυνσης κάνουμε αντιγραφή το url του video στο youtube και έπειτα μέσω της επικόλλησης το προσθέτουμε στο src με την διαφορά ότι στην θέση του τμήματος της διεύθυνσης «[watch?v=](https://www.youtube.com/watch?v=)» εισάγουμε τον όρο «[embed/](#)».

Επίσης μετά το id του url χρησιμοποιούμε ένα ερωτηματικό το οποίο συνήθως ακολουθείτε από τις παραμέτρους του Youtube όπως λέγονται για παράδειγμα το «[autoplay=1](#)» που φαίνεται και στον παραπάνω κώδικα.

Μερικές από τις παραμέτρους του Youtube είναι οι εξής:

- Το **autohide** που έχει να κάνει με την ορατότητα των κουμπιών του player μετά την έναρξη της αναπαραγωγής του video και μπορεί να πάρει τις τιμές:
 - **Τιμή 0**: τα κουμπιά του player είναι πάντα ορατά αφού αρχίσει η αναπαραγωγή του video
 - **Τιμή 1**: τα κουμπιά του player «κρύβονται» αυτόματα όταν το video ξεκινάει
 - **Τιμή 2**: (είναι και η προεπιλεγμένη τιμή) αν η κλίμακα διαστάσεων – ανάλυσης (ratio) του video είναι 16:9 τότε ισχύουν οι ρυθμίσεις της τιμής 1 δηλαδή τα κουμπιά του player «κρύβονται» αυτόματα όταν το video ξεκινάει, στην περίπτωση όμως που το ratio είναι 4:3 τότε τα κουμπιά είναι πάντα ορατά όπως ισχύει στην τιμή 0.
- Το **autoplay** που έχει να κάνει με την αυτόματη αναπαραγωγή του video κατά την φόρτωση της σελίδας και παίρνει τις τιμές:
 - **Τιμή 0**: (είναι και η προεπιλεγμένη τιμή) το video δεν ξεκινάει αυτόματα όταν φορτώνει η σελίδα και χρειάζεται να πατήσουμε το κουμπί αναπαραγωγής
 - **Τιμή 1**: το video ξεκινάει αυτόματα όταν φορτώνει η σελίδα
- Το **controls** που έχει να κάνει με την εμφάνιση των κουμπιών του player κατά την φόρτωση του και όχι κατά την αναπαραγωγή του video στην οποία χρησιμοποιούμε το autohide. Το controls παίρνει τις τιμές:
 - **Τιμή 0**: τα κουμπιά του player δεν εμφανίζονται αλλά ο player φορτώνει απευθείας με την φόρτωση της σελίδας
 - **Τιμή 1**: τα κουμπιά του player εμφανίζονται και φορτώνονται απευθείας – παράλληλα με τον player
 - **Τιμή 2**: τα κουμπιά εμφανίζονται και φορτώνονται κανονικά αλλά ο player φορτώνει μετά από «απαίτηση» του χρήστη



Tip: Οι τιμές 1 και 2 της παραμέτρου controls παρέχουν την ίδια εμπειρία χρήσης του player από τον χρήστη με την μικρή διαφορά όμως ότι με την τιμή 2 υπάρχει μια μικρή βελτίωση των επιδόσεων του player για τις ενσωματώσεις των video μέσω των iframes. Επίσης πολλές φορές οι δύο αυτές τιμές δημιουργούν οπτικές διαφορές στον player όπως το μέγεθος της γραμματοσειράς του τίτλου του video. Παρόλα αυτά όταν ο σχεδιαστής ορισμένες φορές επιλέξει να είναι διαφανής η οπτικοποίηση και η εμφάνιση των τιμών αυτών έναντι του χρήστη τότε οι τιμές μπορούν να εναλλάσσονται χωρίς να γίνεται αντιληπτή η εναλλαγή αυτή.

- Το **disablekb** που έχει να κάνει με την σύνδεση των κουμπιών του player με τα κουμπιά του πληκτρολογίου, δηλαδή να μπορεί ο χρήστης να χειριστεί τον player μέσω του πληκτρολογίου
 - **Τιμή 0:** το πληκτρολόγιο δεν ανταποκρίνεται και είναι απενεργοποιημένο
 - **Τιμή 1:** το πληκτρολόγιο είναι ενεργοποιημένο οπότε κάθε φορά που πατάμε τα παρακάτω πλήκτρα εκτελούνται και οι ανάλογες λειτουργίες στον player:
 - Κουμπί **spacebar** : αναπαραγωγή / παύση του video
 - Κουμπί **δεξί βελάκι** : το video προχωράει κατά 10% μπροστά στην αναπαραγωγή του
 - Κουμπί **αριστερό βελάκι** : το video πηγαίνει κατά 10% προς τα πίσω για την αναπαραγωγή του
 - Κουμπί **βελάκι προς τα πάνω** : αυξάνει την ένταση του ήχου
 - Κουμπί **βελάκι προς τα κάτω** : μειώνει την ένταση του ήχου
- Το **end** το οποίο μετράει την διάρκεια του video από την αρχή μέχρι και το τέλος και εμφανίζει έναν θετικό πάντα αριθμό
- Το **fs** το οποίο δίνει την δυνατότητα να μπει το video σε fullscreen όπως λέγεται δηλαδή οι διαστάσεις του video υπόκεινται σε μεγένθυση τόσο μεγάλη όσο και οι διαστάσεις τις οθόνης. Παίρνει τις τιμές 0 για απενεργοποιημένη την δυνατότητα fullscreen και την τιμή 1 για την ενεργοποίηση της δυνατότητας

- Το **loop** το οποίο δίνει την δυνατότητα της αυτόματης επανάληψης του video και παίρνει τις τιμές:
 - **Τιμή 0:** (είναι και η προεπιλεγμένη) το video θα παίξει μια φορά και για να ξεκινήσει πάλι θα πρέπει χειροκίνητα ο χρήστης να επιλέξει την αρχή του video και να πατήσει αναπαραγωγή
 - **Τιμή 1:** το video μπαίνει σε επανάληψη και κάθε φορά που τελειώνει ξεκινάει αυτόματα να ξαναπαίζει από την αρχή
- Το **playlist** που προσθέτει ένα σύνολο από συναφή videos που συγγενεύουν στην url διεύθυνση τους αφού συνήθως είναι videos που έχουν «ανέβει» στο Youtube από το κανάλι του ίδιου ιδιοκτήτη.



Tip 1: Κάθε χρήστης του Youtube θα έχει προσέξει ότι τα video έχουν δικό τους σύνολο κουμπιών στον player παρόλο που τα video έχουν ενσωματωθεί και αυτά στην σελίδα μέσω του iframe. Αυτό συμβαίνει γιατί τα αρχικά κουμπιά του player που εισάγονται από την παράμετρο controls έχουν κρυφτεί μέσω της παραμέτρου autohide. Για να έχουμε πρόσβαση στα κουμπιά αυτά μπορούμε αν πατήσουμε διπλό δεξί κλικ με το ποντίκι πάνω στο video να επιλέξουμε στην καρτέλα που εμφανίζεται την επιλογή «Show Controls» με την οποία θα εμφανιστούν τα αρχικά κουμπιά του player καθώς θα μας δωθεί και η δυνατότητα της επανάληψης του video στην καρτέλα που θα ανοίξει πατώντας την επιλογή «Loop».



Tip 2: Ένα από τα βασικά προβλήματα που αντιμετωπίζει ένας σχεδιαστής και κατά συνέπεια ένας χρήστης της σελίδας είναι η αργή φόρτωση στην οποία υπόκεινται η σελίδα όταν έχουν ενσωματωθεί σε αυτήν πολλά video μέσω του iframe. για το πρόβλημα λοιπόν αυτό παραθέτουμε μια πολύ απλή λύση που στην πραγματικότητα κάνει μια πολύ σοβαρή δουλειά. Αυτό που έχουμε να κάνουμε είναι να προσθέσουμε δύο γραμμές κώδικα όπως φαίνονται παρακάτω οι οποίες είναι υπεύθυνες για την εισαγωγή μιας βιβλιοθήκης javascript η οποία «διαβάζει» κατά κάποιον τρόπο τον Html κώδικα της σελίδας μας και όπου βρεί tag <iframe> απενεργοποιεί την άμεση εισαγωγή του video στην σελίδα.

Ουσιαστικά μετατρέπει τα video σε πολυμέσα on-demand που σημειώνει ότι αντικαθιστά τα video με σχετικά εικονίδια στην σελίδα στα οποία εμφανίζεται ένα κουμπί αναπαραγωγής και μόνο αν ο χρήστης πατήσει το κουμπί ο server ζητά σύνδεση με το youtube ώστε να αναπαράγει το video. Καταλαβαίνουμε λοιπόν ότι στην σελίδα δεν υπάρχουν οι άμεσες και μόνιμες συνδέσεις του server με το Youtube αλλά στην θέση τους εμφανίζονται τα έμμεσα και κατ' απαίτηση του χρήστη, σχετικά εικονίδια των video. Στην εικόνα παρουσιάζονται οι δύο γραμμές κώδικα που προσθέτουμε στον Html κώδικα της σελίδας καθώς και το οπτικό αποτέλεσμα τους.

```
<html>
<head>
  <script src="gplus-youtubeembed.js"></script>
<title> youtube_videos </title>
</head>
<body>
  <iframe width="420" height="345"
  src="http://www.youtube.com/embed/9gTw2EDkaDQ">
  </iframe>
  <script type="text/javascript">optimizeYouTubeEmbeds(</script>
</body>
</html>
```

Η γραμμή του κώδικα «`<script src="gplus-youtubeembed.js"></script>`» προστίθεται ανάμεσα στο tag `<head>` γιατί είναι αυτό που συνδέει το αρχείο javascript με το αρχείο Html. Προσέχουμε πάντα ποιιά διεύθυνση εισάγουμε ώστε ο browser να μην έχει προβλήματα εύρεσης του αρχείου. Αν δίνουμε σαν διεύθυνση μόνο το όνομα του αρχείου τότε προυποθέτει ότι το αρχείο με το οποίο θέλουμε να συνδεθούμε είναι στον ίδιο φάκελο με το αρχείο Html.

Επίσης η γραμμή του κώδικα «`<script type="text/javascript">optimizeYouTubeEmbeds(</script>`» εισάγεται πριν το κλείσιμο του tag `<body>` και είναι υπεύθυνη για το «κάλεσμα» και την εκτέλεση της βιβλιοθήκης javascript.

Τέλος θα πρέπει να ξέρουμε ότι προηγουμένως θα πρέπει να έχουμε «κατεβάσει» στον υπολογιστή μας την βιβλιοθήκη javascript που απαιτείται για τον κώδικα.

Ενδεικτική σελίδα που μπορούμε να βρούμε την βιβλιοθήκη είναι η εξής:
«<http://trgjtrymntyune.pp.ua/1940/download.php?id=1940&name=Gplus+youtube+embed+js>»



8 Εισαγωγή αντικειμένων (object)

Εκτός όμως από πολυμέσα με την έννοια του video και του ήχου (audio) μπορούμε να εισάγουμε σε μια σελίδα και άλλα αντικείμενα όπως αναγνώστες αρχείων pdf, εφαρμογές java (applets), χάρτες, flash players, ακόμη να εισάγουμε και μια σελίδα Html μέσα σε μια άλλη.

Για την εισαγωγή αυτών των αντικειμένων χρησιμοποιούμε το tag <object>.

Μέσω του tag <object> μπορούμε να εισάγουμε και προγράμματα που βοηθούν τον χρήστη στην περιήγηση του στην σελίδα και που παράλληλα αυξάνουν την λειτουργικότητα της σελίδας στον browser. Τα προγράμματα αυτά ονομάζονται plug – ins και συνήθως χρησιμοποιούνται για την εμφάνιση χαρτών, την αντιμετώπιση ιών που έχουν προσχρήσει στην σελίδα και ακόμα για την επικύρωση του λογαριασμού των χρηστών π.χ. τραπεζικοί λογαριασμοί σε σελίδες τραπεζών ή λοιπών καταστημάτων.

Για παράδειγμα αν θέλουμε να εισάγουμε ένα αρχείο pdf στην σελίδα μας όπου θα εμφανίζεται μαζί με τον αναγνώστη των pdf αρχείων (pdf reader) θα έχουμε τον παρακάτω κώδικα:

```
<html> <head> <title> pdf object </title> </head>
<body>
<object width="100%" height="500px" data="file.pdf"></object>
</body> </html>
```

Για την εισαγωγή αντικειμένων όμως μπορούμε εκτός από το tag <object> να χρησιμοποιήσουμε και ένα άλλο tag το οποίο έχει το πλεονέκτημα ότι υποστηρίζεται από όλους τους μεγάλους browsers. Το tag αυτό λέγεται <embed>, δηλαδή ενσωμάτωση, και συντάσσεται κάπως έτσι:

```
<html> <head> <title> pdf object </title> </head>
<body>
<embed width="100%" height="500px" src="file.pdf">
</body> </html>
```

Επίσης πρέπει να προσέξουμε ότι για την πρόσθεση της διεύθυνσης του αρχείου που θέλουμε να εισάγουμε τα δύο tags έχουν διαφορετικό attribute που η λειτουργία τους όμως είναι η ίδια.

Για το tag <object> έχουμε το attribute “data” ενώ για το tag <embed> έχουμε το attribute “src”. Τέλος άξιο προσοχής χαρακτηρίζεται το γεγονός ότι το tag <embed> δεν έχει close tag, δηλαδή δεν κλείνει καθώς δεν έχει την δυνατότητα να εμφανίσει κάποιο μήνυμα σε περίπτωση που δεν μπορεί να εμφανιστεί το αρχείο που έχει ενσωματωθεί στην σελίδα.

9 Δημιουργία κλάσεων στην Html (classes)

Αρχικά θα πρέπει να μάθουμε ότι οι κλάσεις δεν είναι ένα tag αλλά ένα attribute που μπορεί να συνδυαστεί με τα περισσότερα tags της Html. Κυρίως όμως τις κλάσεις τις συναντάμε συνήθως στα tags <div> και γιατί κατά κύριο λόγο οι κλάσεις χρησιμοποιούνται για την μορφοποίηση ενός στοιχείου συνήθως κειμένου.

Για την δημιουργία μιας κλάσης χρησιμοποιούμε το tag <style> για να ορίσουμε στην κλάση τι θέλουμε να κάνει και έπειτα την «καλούμε» να εκτελεστεί ως attribute μέσα στο tag που θέλουμε να μορφοποιηθεί σύμφωνα με την κλάση.

Παρακάτω παρουσιάζεται η όλη διαδικασία:

```
<html><head>
<style>
div.example {
    background-color:black;
    color:white;
} </style>
</head>
<body>
<div class="example">
<h2>Example 1</h2>
<p>In this example we will show how to create and use a class.</p>
</div>
</body></html>
```

Στην γραμμή του κώδικα «`<div class="example">`» κατασκευάζουμε και παράλληλα «καλούμε» την κλάση να εκτελεστεί. Με το attribute class δίνουμε ένα όνομα στην κλάση. Η κάθε κλάση έχει ένα μοναδικό όνομα για να ξεχωρίζεται από τις υπόλοιπες και θα πρέπει κάθε φορά που την χρειαζόμαστε και την καλούμε να γράφουμε το όνομα της όπως έχει οριστεί στο attribute διαφορετικά η Html δεν θα μπορεί να την αναγνωρίσει. Επίσης πρέπει να γνωρίζουμε ότι μια κλάση μπορεί να χρησιμοποιηθεί πάνω από μια φορές στο ίδιο Html αρχείο και από διαφορετικά tags.

Ταυτόχρονα στο τμήμα του κώδικα:

```
<style>
div.example {
    background-color:black;
    color:white;
} </style>
```

Μέσα στο tag `<style>` που και το ίδιο βρίσκεται μέσα στο tag `<head>`, ορίζουμε την λειτουργία της κλάσης όταν θα κληθεί στο `<body>` tag να εκτελεστεί. Μέσα λοιπόν στο `<style>` γράφουμε το tag μέσα στο οποίο έχει κατασκευαστεί η κλάση, που στην περίπτωση μας είναι το `<div>`, το οποίο ακολουθεί μια τελεία “.” και έπειτα το όνομα της κλάσης όπως έχει οριστεί με το attribute class. Τέλος μέσα σε αγκύλες γράφουμε την λειτουργία της κλάσης με λίγα λόγια στο παράδειγμα μας η κλάση όταν θα κληθεί να εκτελεστεί θα μορφοποιήσει το υπόβαθρο (**background-color:black;**) του `<div>` μαύρο και το χρώμα της γραμματοσειράς του κειμένου (**color:white;**) που θα υπάρχει μέσα στο `<div>` θα μορφοποιηθεί σε λευκό.

Στο παρακάτω παράδειγμα βλέπουμε ακόμα μια κλάση μόνο που αυτή την φορά εκτελείται σε tag `` και η λειτουργία της είναι να μετατρέψει το μέγεθος της γραμματοσειράς του κειμένου που βρίσκεται μέσα στο `` (**font-size:24;**).

```
<html> <head> <style>
span.note {font-size:24;}
</style> </head>
<body>
<p>My <span class="note">Important</span> Heading</p>
</body></html>
```

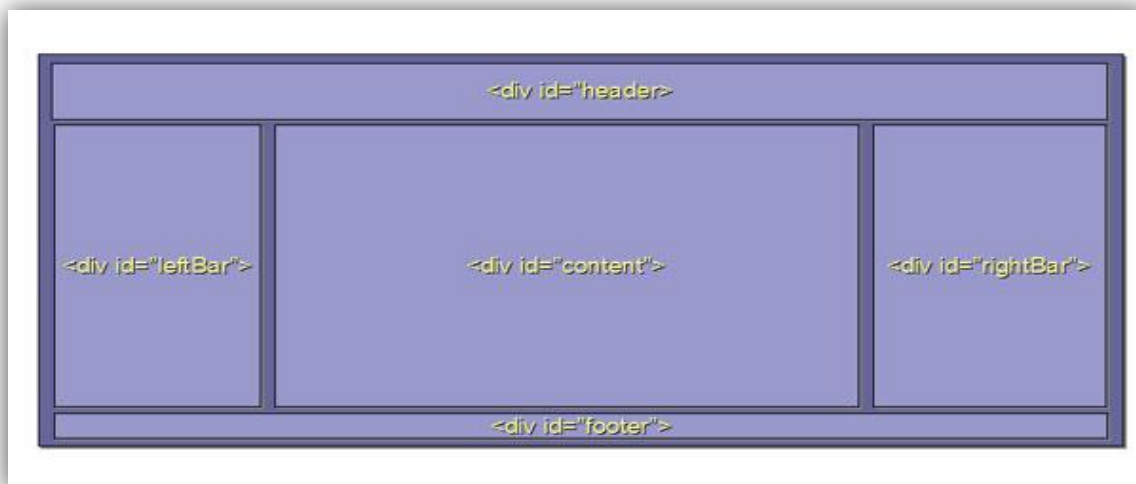
10 Δημιουργία υποβάθρου – δομής της σελίδας (Layout)

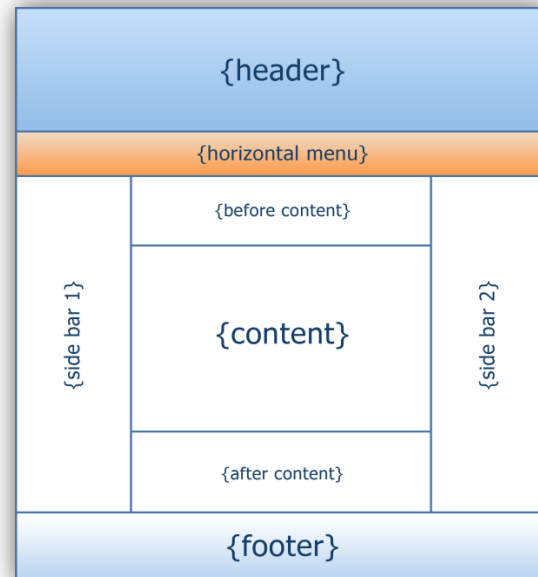
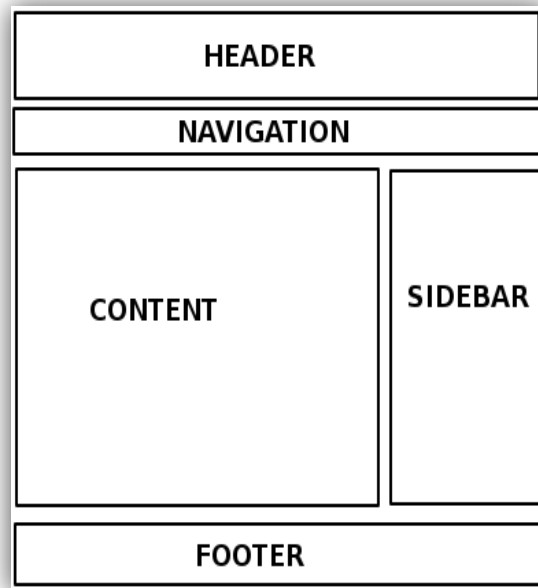
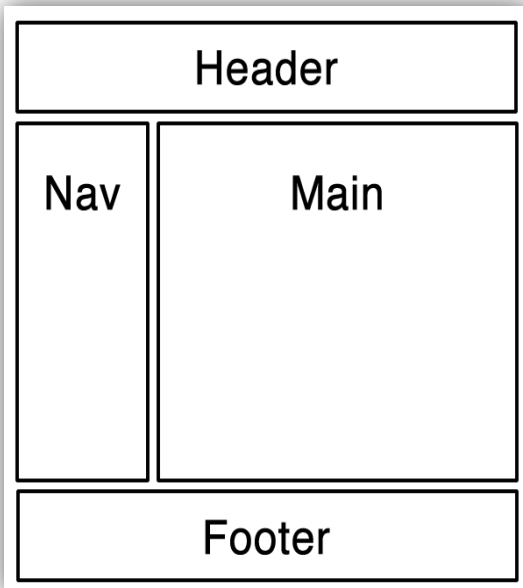
Η δημιουργία υποβάθρου – δομής (layout) ίσως είναι το πιο σημαντικό κομμάτι κατά την σχεδίαση μιας σελίδας. Το layout σκοπό του έχει να κατηγοριοποιήσει την σελίδα κατονομάζοντας τα διάφορα μέρη από τα οποία θα αποτελείται το site, δηλαδή για παράδειγμα μια σελίδα μπορεί να αποτελείται από το τμήμα των επικεφαλίδων (header) , ή το τμήμα των μενού πλοήγησης (nav) όπως επίσης και από το τμήμα της υποσελίδας (footer).

Τα διάφορα αυτά τμήματα στα οποία χωρίζεται η σελίδα χρησιμοποιούνται κυρίως για την βοήθεια του browser να κατανοήσει την δομή του site ώστε να βελτιωθεί η καλύτερη εύρεση του περιεχομένου της σελίδας από τις λέξεις – κλειδιά (keywords) μέσω των μηχανών αναζήτησης. Τα τμήματα αυτά προστέθηκαν με την καινούργια έκδοση την Html5 ενώ παράλληλα θα πρέπει να ξέρουμε ότι μπορούν να εφαρμοστούν στον κώδικα είτε σαν αυτοτελή tags είτε σαν attributes μέσω του tag <div>.

Επιπλέον θα πρέπει να γνωρίζουμε ότι αυτά τα tags / attributes δεν είναι εμφανίσημα δηλαδή δεν παρουσιάζονται οπτικά στον χρήστη, είναι διαφανή αφού προορίζονται κυρίως για την ομαδοποίηση των στοιχείων της σελίδας, για παράδειγμα αν έχουμε κάποιες λίστες οι οποίες κατασκευάζουν ένα μένου πλοήγησης στην σελίδα όλα τα tags του μενού θα εισαχθούν σε ένα tag <div> που θα ονομαστεί nav και που θα αποτελεί το τμήμα πλοήγησης του layout της σελίδας.

Παρακάτω παρουσιάζονται μερικά παραδείγματα διαφορετικών layout – υποβάθρων σε διαφορετικές σελίδες.





Στην επόμενη σελίδα παρουσιάζεται ένα παράδειγμα layout που κατασκευάσαμε και δίνεται ο πηγαίος κώδικας και η ανάλυση του.

```
<html>
<head>
<style>
header {
    background-color:orange;
    color:white;
    text-align:center;
    padding:5px;
}
nav {
    line-height:30px;
    background-color:#A0A0A0;
    height:350px;
    width:35%;
    float:left;
    padding:5px;
}
section {
    width:350px;
    float:left;
    padding:10px;
}
footer {
    background-color:orange;
    color:white;
    clear:both;
    text-align:center;
    padding:5px;
}
</style>
<title> layout elements </title>
</head>
```



```

<body>
<header>
<h1>Layout</h1>
</header>

<nav>
<a href url="index1.html"> Layout elements </a><br>
<a href url="index2.html"> Layout tags / attributes </a><br>
<a href url="index3.html"> Description and usage of layout </a><br>
</nav>

<section>
<h1>Layout elements</h1>

<p> This is an example to show you how to use the layout elements as attributes or tags in the
HTML file.</p>

<p> Layout elements are a very useful tool for the structure of the site because we use them
to group all the other elements. for an example the nav layout element is used to group all
the tags that have been used to create a navigation menu for the site.Layout is also very im-
portant because the grouping of the elements help the search engines to find easier the site
with the keywords that are used to the different sections of the layout. </p>

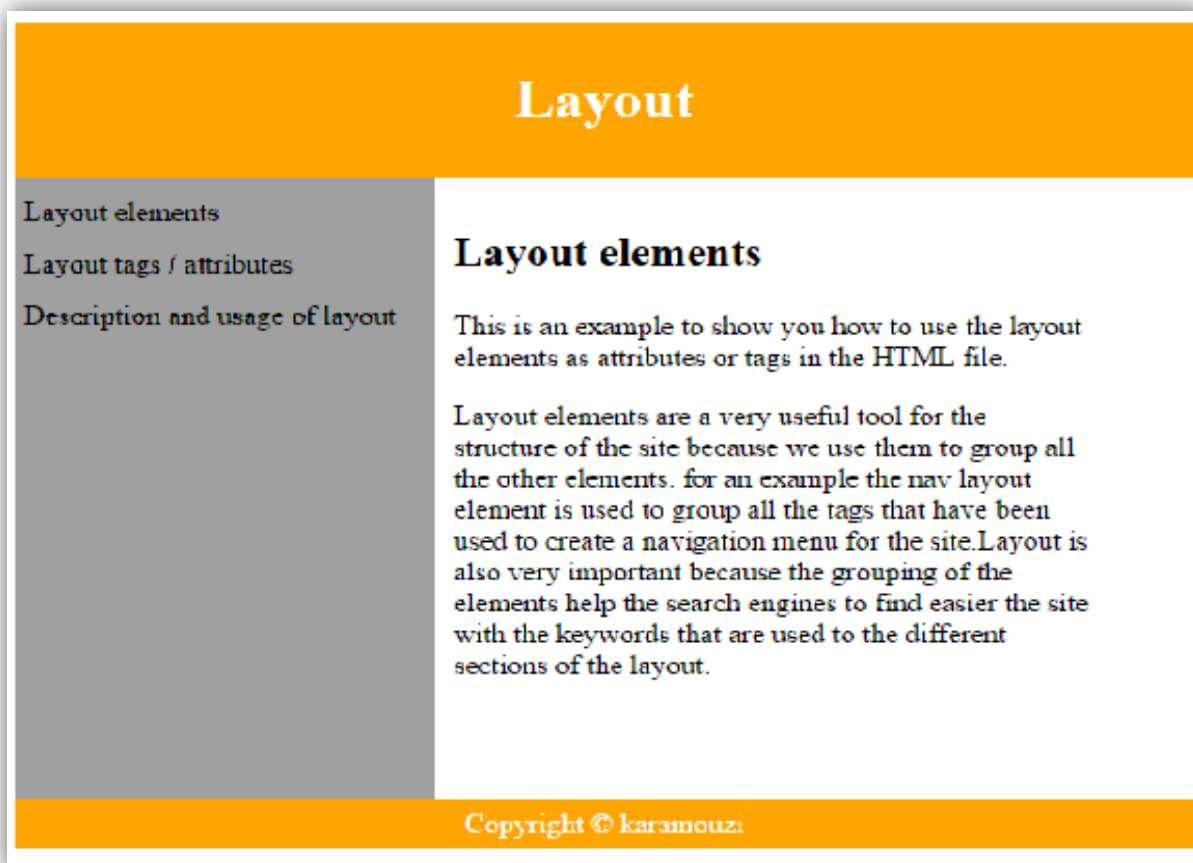
</section>

<footer>
Copyright © karamouzi
</footer>
</body>
</html>

```

Ο παραπάνω κώδικας έχει κατασκευάσει μια σελίδα με ένα απλό layout το οποίο μορφοποιείται μέσω του tag <style> στο ίδιο το αρχείο Html. Όπως θα δούμε παρακάτω η μορφοποίηση ενός layout ή γενικότερα των στοιχείων μιας σελίδας δεν είναι διαχειρίσιμη όταν γίνεται μέσω του tag <style> μέσα στο ίδιο αρχείο Html γι'αυτό τον λόγο και για άλλους πιο σημαντικούς χρησιμοποιούμε τα διαδοχικά φύλλα στυλ Css3 για την μορφοποίηση της σελίδας.

Το αποτέλεσμα του παραπάνω πηγαίου κώδικα φαίνεται στην παρακάτω εικόνα:



10.1 Ανάλυση πηγαίου κώδικα του layout

Βλέποντας τον παραπάνω κώδικα έχουμε αρκετές παρατηρήσεις να κάνουμε μέσα από τις οποίες θα προσπαθήσουμε να αναλύσουμε τον τρόπο με τον οποίο κατασκευάζεται ένα layout σε μια σελίδα.

Αρχικά διακρίνουμε ότι έχουμε χωρίσει τα διαφορετικά τμήματα του κώδικα για κάθε μέρος του layout με διαφορετική υπογράμμιση για την καλύτερη και ευκολότερη κατανόηση του κώδικα.

Ας αναφερθούμε λοιπόν πρώτα στην κατασκευή του layout και επομένως στην σύνταξη των αντίστοιχων tags. Η κατασκευή γίνεται με τα tags <header> που χρησιμοποιείται για το τμήμα των επικεφαλίδων, <nav> που έχει να κάνει με το τμήμα της πλοήγησης της σελίδας, <section> που έχει να κάνει με το κεντρικό «σώμα» του site και <footer> που έχει να κάνει με το τμήμα της υποσελίδας.

Επίσης πρέπει να γνωρίζουμε ότι η κατασκευή του layout γίνεται μέσα στο tag `<body>` ενώ τα tags του layout μορφοποιούνται μέσω του tag `<style>` που βρίσκεται στο tag `<head>`.

Το tag `<header>` αποτελεί το πάνω μέρος της σελίδας και αποτελείται από τα tags των επικεφαλίδων.

Layout

Συντάσσεται όπως φαίνεται παρακάτω:

```
<header>
```

```
<h1>Layout</h1>
```

```
</header>
```

Ενώ μορφοποιείται μέσα στο `<style>` ως εξής:

```
header {
```

```
    background-color:orange;
```

```
    color:white;
```

```
    text-align:center;
```

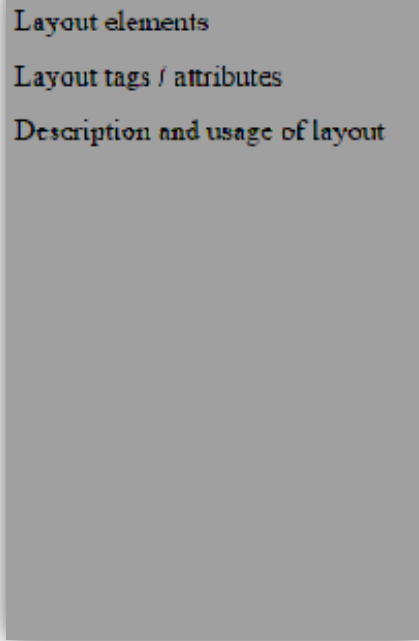
```
    padding:5px;
```

```
}
```

Η μορφοποίηση του `<header>` μέσα στο `<style>` περιλαμβάνει την αλλαγή του χρώματος του υποβάθρου σε πορτοκαλί (`background-color:orange;`) ενώ το χρώμα της γραμματοσειράς των επικεφαλίδων γίνεται άσπρο (`color:white;`) και το κείμενο στοιχίζεται στο κέντρο (`text-align:center;`). τέλος μέσω του attribute `padding` δημιουργείται ένα περιθώριο μεταξύ του κειμένου και του περιγράμματος του τμήματος `<header>` που είναι ίσο με `5px` (`padding:5px;`).

Το tag `<nav>` αποτελεί το αριστερό μέρος της σελίδας στο παράδειγμα μας και περικλείει τα tags των υπερσυνδέσμων δηλαδή της πλοήγησης της σελίδας αν είχαμε και μια λίστα – μενού στη σελίδα τότε τα tags της λίστας θα ανήκαν και αυτά στο tag `<nav>` γιατί θα είχαν να κάνουν με την μετάβαση μεταξύ των σελίδων.

Το tag <nav> ομαδοποιεί όλα τα tags που η λειτουργία τους είναι η πλοήγηση και η μεταφορά στην σελίδα.



```
Layout elements  
Layout tags / attributes  
Description and usage of layout
```

Κατασκευάζεται στο tag <body> ως εξής:

```
<nav>
```

```
<a href url="index1.html"> Layout elements </a><br>
```

```
<a href url="index2.html"> Layout tags / attributes </a><br>
```

```
<a href url="index3.html"> Description and usage of layout </a><br>
```

```
</nav>
```

Ενώ μορφοποιείται στο <style> με τον παρακάτω κώδικα:

```
nav {
```

```
    line-height:30px;
```

```
    background-color:#A0A0A0;
```

```
    height:350px;
```

```
    width:35%;
```

```
    float:left;
```

```
    padding:5px;
```

```
}
```

Όπως καταλαβαίνουμε από τα attributes που υπάρχουν μέσα στο <nav> το χρώμα του υποβάθρου του τμήματος πλοήγησης γίνεται μια απόχρωση του γκρι σε χρωματοκώδικα HEX (`background-color:#A0A0A0;`), το ύψος του τμήματος είναι 350px (`height:350px;`) ενώ το πλάτος του τμήματος καταλαμβάνει το 35% του πλάτους της σελίδας (`width:35%;`). το τμήμα πλοήγησης θα βρίσκεται πάντα στο αριστερό μέρος της σελίδας (`float:left;`).

Αν θέλουμε το τμήμα να βρίσκεται στο δεξί μέρος της σελίδας αρκεί να αλλάξουμε την τιμή του attribute float σε “right”. Τέλος το περιθώριο ανάμεσα σε όλες τις πλευρές του κειμένου και το περίγραμμα του τμήματος πλοήγησης είναι 5px (`padding:5px;`) ενώ το ύψος της κάθε γραμμής κειμένου είναι 30px (`line-height:30px;`).



Tip: Όταν δίνουμε τιμή πλάτους % τότε δίνουμε να καταλάβει στον browser ότι ανάλογα σε ποιά οθόνη θα εμφανίσει την σελίδα θα χρησιμοποιήσει ένα ποσοστό του συνολικού πλάτους της οθόνης για την εμφάνιση ενός στοιχείου. Για παράδειγμα στο τμήμα πλοήγησης που έχουμε παραπάνω έχουμε ορίσει να έχει πλάτος το 35% του συνολικού πλάτους της σελίδας.

Η τιμή του πλάτους ενός στοιχείου είναι πολύ χρήσιμη αν είναι σε ποσοστό % γιατί ανεξάρτητα από το πλάτος της οθόνης στην οποία εμφανίζεται η σελίδα κάθε φορά το μήκος του στοιχείου προσαρμόζεται ανάλογα με το πλάτος της οθόνης καταλαμβάνοντας πάντα το καθορισμένο ποσοστό μήκους.

Συνιστάται να χρησιμοποιούμε ποσοστιαίες τιμές πλάτους στο layout της σελίδας γιατί έτσι θα εμφανίζεται το ίδιο σωστά οπτικά σε κάθε είδους οθόνη είτε είναι οθόνη ενός tablet είτε είναι οθόνη ενός επιτραπέζιου υπολογιστή.

Το tag <section> αποτελεί το κεντρικό σώμα μίας σελίδας και αποτελείται από το κεντρικό περιχόμενο του site που μπορεί να είναι είτε παράγραφοι κειμένου, είτε πολυμέσα όπως video ή ακόμα και άρθρα.

Ένα άρθρο δημιουργείται από το tag <article> αλλά αυτό παρουσιάζεται παρακάτω.

Στο δικό μας παράδειγμα το tag <section> αποτελείται από δύο παραγράφους κειμένου.

Layout elements

This is an example to show you how to use the layout elements as attributes or tags in the HTML file.

Layout elements are a very useful tool for the structure of the site because we use them to group all the other elements. for an example the nav layout element is used to group all the tags that have been used to create a navigation menu for the site. Layout is also very important because the grouping of the elements help the search engines to find easier the site with the keywords that are used to the different sections of the layout.

To `<section>` συντάσσεται ως εξής:

```
<section>
```

```
<h1>Layout elements</h1>
```

```
<p> This is an example to show you how to use the layout elements as attributes or tags in the HTML file.</p>
```

```
<p> Layout elements are a very useful tool for the structure of the site because we use them to group all the other elements. for an example the nav layout element is used to group all the tags that have been used to create a navigation menu for the site. Layout is also very important because the grouping of the elements help the search engines to find easier the site with the keywords that are used to the different sections of the layout. </p>
```

```
</section>
```

και μορφοποιείται κατά αυτό τον τρόπο:

```
section {
```

```
width:350px;
```

```
float:left;
```

```
padding:10px;
```

```
}
```

Το `<section>` έχει πλάτος 350px (`width:350px;`) και περιθώριο μεταξύ του περιεχομένου του τμήματος και του περιγράμματος του `<section>` είναι 10px (`padding:10px;`). Επιπλέον με το attribute float δηλώνουμε ότι το τμήμα section θα «κυλάει» θα στοιχίζεται δηλαδή στο αριστερό μέρος της σελίδας επομένως θα «ακουμπήσει» πάνω στο τμήμα πλοήγησης nav (`float:left;`).

Τέλος το tag `<footer>` αποτελεί όπως είπαμε το τμήμα υποσελίδας δηλαδή το κάτω μέρος της σελίδας στο οποίο εισάγονται τα copyrights – η επικύρωση των πνευματικών δικαιωμάτων του σχεδιαστή, οι λέξεις – κλειδιά (keywords) που χρησιμοποιούνται από τις μηχανές αναζήτησης όπως επίσης και οι πληροφορίες του απορρήτου και του ιστορικού της σελίδας.



Copyright © karamouzi

Το tag `<footer>` συντάσσεται κάπως έτσι:

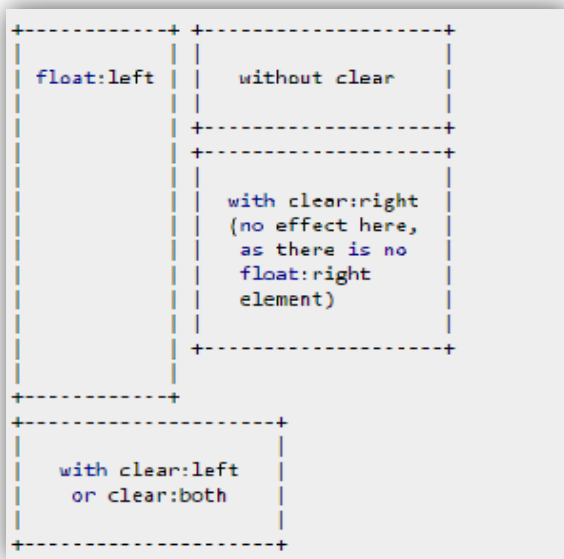
```
<footer> Copyright © karamouzi </footer>
```

Και μορφοποιείται ως εξής:

```
footer {  
    background-color:orange;  
    color:white;  
    clear:both;  
    text-align:center;  
    padding:5px;  
}
```

Στην μορφοποίηση του `<footer>` το χρώμα του υποβάθρου γίνεται πορτοκαλί (`background-color:orange;`) και το χρώμα της γραμματοσειράς γίνεται άσπρο (`color:white;`), ενώ παράλληλα το κείμενο στοιχίζεται στο κέντρο (`text-align:center;`) και το περιθώριο μεταξύ του κειμένου και του περιγράμματος του `<footer>` ορίζεται σε 5px (`padding:5px;`). Επιπλέον διακρίνουμε ότι χρησιμοποιείται ένα ακόμη άγνωστο μέχρι στιγμής attribute το `clear:both;`. Το attribute αυτό χρησιμοποιείται με σκοπό να δημιουργήσει κενό μεταξύ του στοιχείου που χρησιμοποιεί το attribute “clear” και των στοιχείων που μορφοποιούνται από το attribute “float”.

Με λίγα λόγια στο δικό μας παράδειγμα χρησιμοποιώντας το “clear” στο `<footer>` «κατεβάζουμε» το τμήμα υποσελίδας κάτω από τα τμήματα `<section>` και `<nav>` τα οποία χρησιμοποιούν το “float” για να στοιχίζονται στο αριστερό μέρος της σελίδας. αν δεν χρησιμοποιούσαμε το “clear” στο `<footer>` τότε το τμήμα υποσελίδας θα στοιχίζόταν αυτόματα στα δεξιά, δίπλα από το τμήμα `<section>`.



10.1.1 Δημιουργία άρθρου (<article>)

Όπως είπαμε και παραπάνω στο κεντρικό τμήμα του layout της σελίδας, το <section> μπορεί να αποτελείται από άρθρα. Σαν άρθρο ορίζεται ένα ανεξάρτητο, αυτοδύναμο σύνολο κειμένου που έχει ένα ολοκληρωμένο νόημα και για το λόγο αυτό μπορεί να διανεμηθεί αυτοδύναμα εκτός του υπόλοιπου περιεχομένου της σελίδας. Με λίγα λόγια ένα άρθρο μπορεί να είναι μια δημοσίευση σε ένα blog ή σε ένα forum, μια είδηση σε μια ενημερωτική σελίδα, ένα σχόλιο κ.α. Δηλαδή ένα ανεξάρτητο και αυτοτελές περιεχόμενο σελίδας, συνήθως είναι ένα κείμενο το οποίο έχει πλήρες νόημα ώστε όταν κάποιος το διανεμίει ηλεκτρονικά εκτός της σελίδας που το περιλαμβάνει στο περιεχόμενό της, ο χρήστης που θα το διαβάσει και θα το επεξεργαστεί δεν θα χρειαστεί να ανατρέξει στην αρχική σελίδα που το άρθρο εμπεριέχεται για να καταλάβει το νόημα του. Επιπλέον ένα tag <article> από το οποίο κατασκευάζεται ένα άρθρο, μπορεί να περιλαμβάνει και το ίδιο μια απλή μορφή layout όπως να αποτελείται για παράδειγμα από ένα τμήμα επικεφαλίδων <header>.

Για την κατασκευή ενός άρθρου χρησιμοποιούμε το tag <article> ως εξής:

<article>

<h1>An article example**</h1>**

<p>This is an article**</p>**

</article>



Tip 1: Θα πρέπει να διευκρινιστεί ότι μπορούμε να έχουμε και εμφωλευμένα άρθρα (ένα άρθρο μέσα σε ένα άλλο). Στην περίπτωση αυτή το εσωτερικό άρθρο δηλώνει ότι στην ουσία «συγγενεύει» με το εξωτερικό. Για παράδειγμα τα σχόλια μιας δημοσίευσης ενός blog αποτελούν ένα άρθρο και μάλιστα εσωτερικό γιατί συγγενεύουν με το εξωτερικό άρθρο την ίδια την δημοσίευση του blog.



Tip 2: Επιπλέον μπορούμε να δώσουμε σε ένα άρθρο και τις πληροφορίες του συγγραφέα του μέσω του tag `<address>` το οποίο εισάγεται εσωτερικά στο tag `<article>`. Το tag `<address>` όμως δεν μπορεί να χρησιμοποιηθεί στην περίπτωση των εμφωλευμένων άρθρων.

```
<article>
```

```
<h1>An article example</h1>
```

```
<p>This is an article</p>
```

```
<address>
```

You can contact author at:

```
<a href="http://www.somedomain.com/contact">contact</a> <br>
```

If you see any bugs, please

```
<a href="mailto:webmaster@somedomain.com">contact mail</a>.<br>
```

```
</address>
```

```
</article>
```



Tip 3: Τέλος εκτός από τις πληροφορίες και την επικοινωνία με τον συγγραφέα του άρθρου μέσω του tag `<address>` μπορούμε να εισάγουμε σε ένα άρθρο και την ημερομηνία δημοσίευσης του η οποία εισάγεται με το attribute “datetime” του tag `<time>` κάπως έτσι:

```
<article>
```

```
<h1>An article example</h1>
```

```
<p>This is an article</p>
```

```
Posted on <time datetime="2015-05-16 19:00">Jun 16</time> by Sonia.
```

```
</article>
```

11 Καθιέρωση των Css3

Μέχρι τώρα μιλούσαμε για την κατασκευή μιας σελίδας μέσω της Html αλλά και για την μορφοποίηση του περιεχομένου της ίδιας της σελίδας μέσω των tags μορφοποίησης κυρίως του <style>.

Η διαδικασία όμως της μορφοποίησης μέσω της Html και των διαφορετικών tags <style> δημιουργούν πονοκέφαλο στους σχεδιαστές γιατί πολύ απλά χρειάζονται τον διπλάσιο χρόνο για την ενσωμάτωση της ίδιας μορφοποίησης σε πολλαπλά αρχεία Html. το σημαντικό αυτό πρόβλημα έρχεται να λύσει η καθιέρωση των διαδοχικών φύλλων στυλ Css καθώς δίνεται η δυνατότητα πλέον στους σχεδιαστές να δημιουργήσουν ένα συγκεκριμένο στυλ μορφοποίησης και να το εφαρμόσουν παράλληλα σε πολλαπλά αρχεία Html καλώντας το είτε από ένα εξωτερικό αρχείο Css που συνδέεται μέσω link με το αρχείο Html, είτε να ενσωματωθεί μέσα στον κώδικα Html θυμίζοντας την μορφή σύνταξης μιας συνάρτησης σε κώδικα οποιασδήποτε γλώσσας προγραμματισμού.

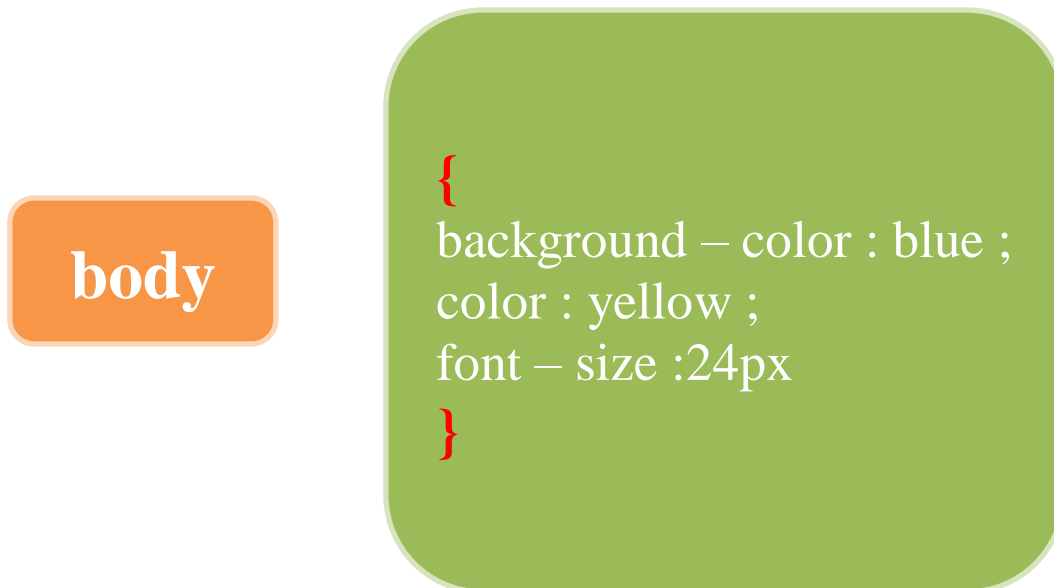
11.1 Σύνταξη ενός αρχείου Css

Αρχικά θα πρέπει να αναφέρουμε ότι ένα αρχείο Css μπορεί να συνταχθεί μέσω των ίδιων προγραμμάτων – editors που χρησιμοποιούμε στην σύνταξη των αρχείων Html. Εμείς στον οδηγό αυτό θα γράφουμε τον κώδικα των Css στο απλό σημειωματάριο των Windows όπως κάναμε και με την Html.

Η σύνταξη τώρα ενός Css δεν είναι μια δύσκολη διαδικασία απεναντίας είναι ευκολότερη και από τα αρχεία Html γιατί στα Css δεν υπάρχουν τα tags ορισμού της δομής της σελίδας. Με λίγα λόγια δεν υπάρχουν tags όπως τα <html>, <head> και <title>.

Αντ'αυτού για να αναφερθούμε στην μορφοποίηση αυτών των στοιχείων γράφουμε τα ονόματα των tags χωρίς τα εισαγωγικά, ακολουθούμενα από τις αγκύλες που περιλαμβάνουν τα attributes μορφοποίησης που χρησιμοποιούσαμε για την μορφοποίηση τους στην Html.

Η σύνταξη αυτή φαίνεται παρακάτω:



Ας αναλύσουμε τώρα το παραπάνω σχήμα. Το body αναφέρεται στο tag <body> του αρχείου Html με το οποίο έχει συνδεθεί το Css. Βλέπουμε ότι δεν υπάρχουν τα εισαγωγικά πριν και μετά από το body και αυτό είπαμε γιατί δεν χρησιμοποιούνται στο Css τα tags αλλά αναφερόμαστε σε αυτά. Το body τώρα ακολουθείται από μια ανοιχτή αγκύλη { η οποία περικλείει τα attributes του body ή αλλιώς όπως αναφέρονται στα Css, τις ιδιότητες.

Ο συγκεκριμένος κώδικας μορφοποιεί το χρώμα του υποβάθρου του body σε μπλε (**background – color : blue ;**), το χρώμα της γραμματοσειράς που θα υπάρχει μέσα στο body σε κίτρινο (**color : yellow ;**) και το μέγεθος των γραμμμάτων σε 24px (**font – size :24px**). Επιπλέον θα πρέπει να ξέρουμε ότι αν έχουμε ποικιλία από attributes μέσα στις αγκύλες τότε για να διαβάσει ο browser ξεχωριστά το κάθε attribute, τα χωρίζουμε με το ερωτηματικό της ελληνικής γραφής. Επίσης δεν ξεχνούμε ποτέ όταν τελειώσει η λίστα των attributes του κάθε στοιχείου κλείνουμε την αγκύλη }.

Τέλος να πούμε ότι το tag στο οποίο αναφερόμαστε στο Css και χρησιμοποιούμε το όνομα του ονομάζεται και αναφέρεται συνήθως στην ξένη βιβλιογραφία σαν «επιλογέας» (selector).

Στην παραπάνω περίπτωση κώδικα ο selector body αναφέρεται στο tag <body> του αρχείου Html με το οποίο έχει συνδεθεί το συγκεκριμένο Css. Ο browser αποτυπώνει την συγκεκριμένη μορφοποίηση που διαβάσει από το Css στο ίδιο tag της Html με το ίδιο όνομα. το όνομα body

είναι μοναδικό tag σε ένα αρχείο Html δηλαδή χρησιμοποιείται μόνο μια φορά σε ένα αρχείο Html και περικλείει όλο το περιεχόμενο της σελίδας. Τι θα γίνει όμως όταν χρησιμοποιήσουμε το tag <p> που δημιουργεί τις παραγράφους σε μια σελίδα σαν selector στο Css; Το πρόβλημα είναι που θα εφαρμόσει την μορφοποίηση ο browser όταν ο selector οδηγεί σε παραπάνω από ένα tag γιατί στην προκειμένη περίπτωση του selector p καταλαβαίνουμε ότι αναφέρεται στα πολλαπλά tags <p> των διαφορετικών παραγράφων της σελίδας. Με λίγα λόγια ο selector body άπευθυνόταν και μορφοποιούσε ένα και μοναδικό tag εδώ όμως για παράδειγμα με τον selector p αναφερόμαστε σε πολλά tags <p> γιατί μια σελίδα θα έχει πολλές παραγράφους. Σε ποιά λοιπόν από όλες τις παραγράφους θα εφαρμοστεί η μορφοποίηση του selector p; Η λύση στο πρόβλημα έρχεται με την ιδιότητα του selector id που αναλύεται στην παρακάτω υποπαραγράφο.

11.1.1 Η ιδιότητα id του selector

Όπως αναφέραμε και παραπάνω η ιδιότητα id του selector έρχεται για να λύσει το πρόβλημα της ενσωμάτωσης της μορφοποίησης σε διαφορετικά tags με το ίδιο όμως όνομα. δηλαδή αν θέλουμε να μορφοποιήσουμε μια μόνο παράγραφο από τις πολλαπλές που υπάρχουν σε μια σελίδα ή να μορφοποιήσουμε διαφορετικά τις διαφορετικές παραγράφους, τότε δεν θα χρησιμοποιήσουμε στο Css τον selector p γιατί αυτόματα το p θα αναφέρεται σε όλες τις παραγράφους οι οποίες για να κατασκευαστούν χρησιμοποιούν το ίδιο tag <p>.

Μια id ιδιότητα του selector θα πρέπει να είναι μοναδική για να μπορεί να ορίσει και να απευθυνθεί σε ένα μοναδικό στοιχείο.

Για την σύνταξη του ακολουθούμε την εξής διαδικασία:

- Στο αρχείο Html θα δώσουμε ένα μοναδικό όνομα στο tag που θέλουμε να μορφοποιήσουμε ξεχωριστά, μέσω του attribute id όπως φαίνεται με κίτρινη υπογράμμιση στον παρακάτω κώδικα:

```
<body>
```

```
<p id="para1"> id selector - example </p>
```

```
<p> This paragraph is formatted by id selector. </p>
```

```
</body>
```

- Έπειτα αφού δώσαμε ένα μοναδικό όνομα στο tag η επόμενη δουλειά που έχουμε να κάνουμε είναι να αναφερθούμε στο tag αυτό μέσω του Css με το όνομα που του έχουμε δώσει χρησιμοποιώντας πριν από το όνομα το σύμβολο της δέσης “#”. Αυτόματα στο Css η δέση ακολουθούμενη από το όνομα id του tag γίνεται selector οπότε για να μορφοποιήσουμε αυτό τον selector το μόνο που λείπει είναι να ανοίξουμε τις αγκύλες και να εισάγουμε τις ιδιότητες - attributes μορφοποίησης που θέλουμε. Η διαδικασία που περιγράψαμε φαίνεται παρακάτω:

```
#para1 {  
  
    text-align: center;  
  
    color: red;  
  
}
```

Ο παραπάνω κώδικας Css δηλώνει στον browser ότι όπου δεις στο αρχείο Html το tag το οποίο έχει πάρει το μοναδικό όνομα «para1» μέσω του attribute id, τότε εφαρμόσε την ψευδοπροσδιορισμένη μορφοποίηση που έχει οριστεί.



Tip: Συμβουλευουμε τους νέους σχεδιαστές να μην χρησιμοποιούν μοναδικά ονόματα στο attribute id που αρχίζουν από αριθμό γιατί υπάρχει περίπτωση να μην αναγνωρίζεται από τον browser.

11.1.2 Οι κλάσεις στο αρχείο Css

Οι κλάσεις όπως έχουμε προαναφέρει είναι ένα πολύ σημαντικό κομμάτι στην μορφοποίηση των στοιχείων Html. μέχρι τώρα έχουμε δει πως δημιουργούμε κλάσεις και πως εφαρμόζονται σε αρχείο Html αλλά τι γίνεται όταν θέλουμε να χρησιμοποιήσουμε τις κλάσεις μέσα από αρχείο Css;

Όταν χρησιμοποιούμε τις κλάσεις σε Css μπορούμε να εφαρμόσουμε το κομμάτι αυτό της μορφοποίησης σε πολλά και διαφορετικά tags της Html. για να κατασκευάσουμε μια κλάση αυτή την φορά θα πάμε την διαδικασία λίγο ανάποδα δηλαδή δεν θα δημιουργηθεί η κλάση μέσα στο αρχείο Html αλλά μέσα στο κώδικα του Css και έπειτα απλά θα την καλούμε στα διάφορα tags της Html που θέλουμε να μορφοποιήσουμε μέσω του attribute class.

- Για την κατασκευή της κλάσης μέσα στον κώδικα του Css εισάγουμε πρώτα το σύμβολο της τελείας ακολουθούμενο έπειτα από το όνομα της κλάσης που θέλουμε να δώσουμε. Έπειτα ανοίγουμε τις αγκύλες και εισάγουμε τις ιδιότητες – attributes της μορφοποίησης που θέλουμε να κάνουμε. Η διαδικασία της κατασκευής δίνεται παρακάτω μέσα από το παράδειγμα:

```
.center {
    text-align: center;
    color: red;
}
```

- Αφού κατασκευάσαμε την κλάση αυτό που μας μένει είναι να την καλέσουμε μέσα στα tags στα οποία θέλουμε να εφαρμόσουμε την μορφοποίηση της κλάσης. Για να γίνει αυτό το «κάλεσμα» χρησιμοποιούμε το attribute class που παίρνει σαν τιμή το όνομα της κλάσης που έχουμε δώσει μέσα στο Css:

```
<body>
```

```
<h1 class="center"> This heading will not be affected </h1>
```

```
<p class="center"> This paragraph will be red and center-aligned. </p>
```

```
</body>
```

Όπως παρατηρούμε μια κλάση μπορεί να μορφοποιήσει πολλά και διάφορα tags απλά καλώντας την μέσω του attribute class.

- Παράλληλα μας δίνεται η δυνατότητα να ορίσουμε για παράδειγμα όλα τα tags των παραγράφων της σελίδας να χρησιμοποιούν για την μορφοποίησή τους την ίδια κλάση για να το πετύχουμε αυτό απλά στο αρχείο Css εισάγουμε μπροστά από το σύμβολο της τελείας της κλάσης, το όνομα του tag που θέλουμε να μορφοποιείται με την ίδια κλάση κάθε φορά που εισάγεται στο αρχείο Html, στο παράδειγμα των παραγράφων θα εισάγουμε το όνομα του tag <p> κάπως έτσι:

```
p .center {text-align: center; color: red;}
```

11.1.3 Ομαδοποίηση των selectors

Ομαδοποίηση σημαίνει ότι αν έχουμε tags διαφορετικής λειτουργίας που χρησιμοποιούν τις ίδιες ιδιότητες – attributes μέσω των selector τους μπορούμε αντί να γράφουμε την ίδια μορφοποίηση ξανά και ξανά αλλάζοντας μόνο το όνομα του selector, να ομαδοποιήσουμε τους selectors εισάγοντας τα ονόματα τους μαζί δίπλα από το κομμάτι της μορφοποίησης ξεχωρίζοντας μεταξύ τους με το συμβολο του κόμματος όπως παρουσιάζεται παρακάτω:

Αντί για αυτό το κομμάτι κώδικα μέσα στο αρχείο Css:

```
h1 {  
  text-align: center;  
  color: red;  
}
```

```
h2 {  
  text-align: center;  
  color: red;  
}
```

```
p {  
  text-align: center;  
  color: red;  
}
```

μπορούμε να ομαδοποιήσουμε τους selectors κατά αυτόν τον τρόπο:

```
h1, h2, p {  
  text-align: center;  
  color: red;  
}
```

12 Μορφοποίηση των βασικών στοιχείων μέσω του Css

Σε αυτό το κεφάλαιο θα χωριστούν σε υποπαραγράφους η μορφοποίηση των διαφορετικών στοιχείων της Html μέσω των διαδοχικών φύλλων στυλ Css. Θα αναφερθούμε στις βασικότερες ιδιότητες – attributes που μπορούν να μορφοποιήσουν ένα υπόβαθρο σελίδας, ένα κείμενο και την γραμματοσειρά του, έναν πίνακα ή λίστα ή ακόμα και στοιχεία <div>.

12.1 Μορφοποίηση υποβάθρου σελίδας / στοιχείου (background)

Είπαμε πως μέσα από τα Css μπορούμε να μορφοποιήσουμε όλα τα στοιχεία μιας σελίδας. Το βασικότερο στοιχείο από το θέμα της οπτικής παρουσίασης μιας σελίδας είναι το υπόβαθρο της ή αλλιώς το background. Για να μορφοποιήσουμε το background υπάρχει μια λίστα από χρήσιμες ιδιότητες – attributes που είναι η εξής:

- background-color
- background-image
- background-repeat
- background-attachment
- background-position

12.1.1 Η ιδιότητα background-color

Με την ιδιότητα αυτή μπορούμε να αλλάξουμε το χρώμα του υποβάθρου παίρνοντας τιμές:

- είτε χρωματοκώδικα HEX με τιμή όπως αυτή: **"#ff0000"**
- είτε κώδικα χρωμάτων RGB με τιμή: **"rgb(255,0,0)"**
- ή ακόμα χρησιμοποιώντας το όνομα του ίδιου του χρώματος **"red"**

η ιδιότητα συντάσσεται στο Css ως εξής:

```
body { background-color: #b0c4de; }
```

παρατηρούμε ότι για να αλλάξουμε το χρώμα του υποβάθρου της σελίδας χρησιμοποιούμε σαν selector το body και αυτό γιατί θέλουμε να αναφερθούμε σε όλη την σελίδα. Αυτό όμως δεν σημαίνει ότι το συγκεκριμένο attribute δεν μπορεί να χρησιμοποιηθεί για να μορφοποιήσουμε και το υπόβαθρο άλλων στοιχείων για παράδειγμα αλλάζοντας το όνομα του selector από body σε p αλλάζουμε το χρώμα του υποβάθρου μόνο της παραγράφου και όχι όλης της σελίδας.

12.1.2 Η ιδιότητα background-image

Εκτός από το να αλλάξουμε το χρώμα του υποβάθρου ενός στοιχείου μπορούμε να εισάγουμε μια εικόνα η οποία θα χρησιμοποιείται σαν background του στοιχείου. Αυτή η δουλειά γίνεται μέσω του attribute background – image που συντάσσεται κάπως έτσι:


```
body { background-image: url("paper.gif"); }
```

όπως βλέπουμε το ίδιο το attribute χρησιμοποιεί την ιδιότητα url για να εισάγει την διεύθυνση της εικόνας.

Στην παραπάνω περίπτωση κώδικα έχουμε εισάγει μια εικόνα που χρησιμοποιείται σαν background σε ολόκληρη την σελίδα. Από προεπιλογή μια εικόνα που εισάγεται σαν background ενός στοιχείου επειδή είναι σχεδόν αδύνατον η εικόνα να έχει τις ίδιες διαστάσεις με το στοιχείο στο οποίο είναι υπόβαθρο, γι ' αυτό τον λόγο αυτόματα η εικόνα όταν είναι μικρότερη σε διαστάσεις από το στοιχείο επαναλαμβάνεται η εμφάνιση της παράλληλα στους άξονες x και y. Αν θέλουμε να επαναλαμβάνεται μόνο στον άξονα x ή μόνο στον άξονα y τότε χρησιμοποιούμε το attribute background – repeat με τιμές:

- ```
body {
 background-image: url("gradient_bg.png");
 background-repeat: repeat-x;
}
```

τιμή **repeat-x;** για την επανάληψη της εικόνας στον άξονα x

- ```
body {  
    background-image: url("gradient_bg.png");  
    background-repeat: repeat-y;  
}
```

τιμή **repeat-y;** για την επανάληψη της εικόνας στον άξονα y



Tip: Όταν χρησιμοποιούμε μια εικόνα για background μιας ολόκληρης σελίδας καλό είναι να ρυθμίζουμε τις διαστάσεις της και την στοίχιση της έτσι ώστε να μην ενοχλεί οπτικά το υπόλοιπο περιεχόμενο της σελίδας. Κατι τέτοιο γίνεται με το παρακάτω τμήμα του κώδικα:

```
body { background: url("images/back_logo.png") no-repeat center center fixed;  
  
    background-size: cover; overflow-x: hidden; overflow-y: }
```

Ας πούμε δυο λόγια για το παραπάνω τμήμα κώδικα. Όπως βλέπουμε εισάγει μια εικόνα σαν υπόβαθρο σελίδας μέσω του attribute (**background: url("images/back_logo.png")**) μπορεί να χρησιμοποιηθεί και σαν background:... χωρίς το image για την εισαγωγή μιας εικόνας. Μέσα στο ίδιο attribute η εικόνα

παίρνει και τις εξής ρυθμίσεις: (**no-repeat center center fixed;**) με τις οποίες η εικόνα δεν επαναλαμβάνεται σε κανένα άξονα (**no-repeat**), στοιχίζεται κεντρικά σύμφωνα με τους άξονες x και y (**center center**) και τέλος όταν η σελίδα θα έχει μπάρα κύλισης γιατί θα είναι πολύ μεγάλη όλο το περιεχόμενο θα κυλάει εκτός από την εικόνα όπου η θέση της θα είναι σταθερά κεντραρισμένη στην σελίδα (**fixed**). Με το δεύτερο attribute (**background-size: cover;**) και κυρίως με την τιμή που έχει στην ουσία «τεντώνουμε» την εικόνα ώστε οι διαστάσεις της να γίνουν τόσο μεγάλες ώστε να καλύπτει όλο το υπόβαθρο της σελίδας από την μια άκρη στην άλλη. Τα attributes overflow είναι αυτά που δίνουν στην σελίδα την δυνατότητα μεγάλωνει προς τα κάτω και να κυλάει όταν το αρχικό περιεχόμενο της ξεπερνάει τις αρχικές διαστάσεις της σελίδας.

12.1.3 Η ιδιότητα background-size

Όπως είδαμε και στο παραπάνω tip αναφερθήκαμε και στην ιδιότητα background – size. Η συγκεκριμένη ιδιότητα είναι πολύ σημαντική γιατί μας βοηθά να ορίσουμε νέες διαστάσεις σε μια εικόνα.

Το background – size μπορεί να πάρει τις εξής τιμές:

- **background-size: 100px 80px;**

Αριθμητικές τιμές που ορίζουν τις διαστάσεις τις εικόνας σε px. Ο αριθμός 100px ορίζει το μήκος της εικόνας και ο αριθμός 80px ορίζει το ύψος.

- **background-size: contain;**

Με την τιμή contain η εικόνα έχει το μέγιστο μήκος και ύψος που μπορεί να πάρει χωρίς όμως να ξεφεύγει έξω από τις διαστάσεις του πλαισίου στο οποίο βρίσκεται σαν υπόβαθρο. Για παράδειγμα αν μια εικόνα έχει οριστεί σαν υπόβαθρο σε ένα <div> στοιχείο τότε αν πάρει την τιμή contain το μήκος και το ύψος θα οριστούν στις μεγαλύτερες τιμές τους χωρίς όμως να υπερβαίνει τις διαστάσεις του <div>.

- **background-size: cover;**

Με την τιμή cover η εικόνα «τεντώνεται» τόσο πολύ ώστε το μήκος και το ύψος της να πάρουν τις ίδιες τιμές με τις διαστάσεις του πλαισίου στο οποίο είναι υπόβαθρο. Σε αντίθεση με την τιμή contain εδώ οι διαστάσεις της εικόνας μπορούν να υπερβούν τις διαστάσεις του πλαισίου. Η τιμή cover χρησιμοποιείται σε εικόνες υποβάθρου fullscreen.

Επίσης θα πρέπει να αναφέρουμε ότι μπορούμε να έχουμε σαν υπόβαθρο περισσότερες από μια εικόνες ενώ μπορούμε να ορίσουμε να τοποθετείται η μία πάνω στην άλλη.

Για παράδειγμα η εισαγωγή δυο εικόνων που η μια θα εμφανίζεται πάνω στην άλλη γίνεται με τον παρακάτω κώδικα:

```
body {  
    background-image: url(image1.gif), url(image2.gif);  
    background-position: right bottom, left top;  
    background-repeat: no-repeat, repeat;  
}
```

Στο τμήμα αυτού του κώδικα εισάγουμε δύο εικόνες των οποίων οι διευθύνσεις χωρίζονται μεταξύ τους με κόμμα. Με το attribute background – position ορίζουμε την θέση της κάθε εικόνας. Οι δύο πρώτες λέξεις (**right bottom**) ορίζουν την θέση της πρώτης εικόνας η οποία θα είναι στο δεξί και κάτω μέρος του body ενώ η θέση της δεύτερης εικόνας (**left top**) ορίζεται από τις άλλες δύο λέξεις που την τοποθετούν στο αριστερό και πάνω μέρος του body. Οι θέσεις όπως βλέπουμε χωρίζονται μεταξύ τους και αυτές με κόμμα. Επιπλέον με το attribute background – repeat απαγορεύουμε στην πρώτη εικόνα να επαναλαμβάνεται μέσω της τιμής (**no-repeat**), ενώ επιτρέπουμε στην δεύτερη εικόνα την επανάληψη με την τιμή (**repeat**).

Τέλος μπορούμε με μια γραμμή κώδικα να ρυθμίσουμε και τις διαστάσεις και των δύο εικόνων ταυτόχρονα ως εξής:

```
background-size: 50px, 130px, auto;
```

το 50px αναφέρεται στις διαστάσεις της πρώτης εικόνας και ορίζει το μήκος και το ύψος της με την ίδια τιμή 50px. Το 130px αναφέρεται στις διαστάσεις της δεύτερης εικόνας και ορίζει μήκος και ύψος στα 130px.επιπλέον αν είχαμε προσθέσει και μια τρίτη εικόνα τότε θα αναφερόμασταν στις διαστάσεις με έναν τρίτο αριθμό. Στο τμήμα όμως αυτού του κώδικα βλέπουμε ότι αναφέρεται στις διαστάσεις μιας τρίτης εικόνας χρησιμοποιώντας την λέξη auto που σημαίνει ότι οι διαστάσεις τις εικόνες ορίζονται αυτόματα και συνήθως παίρνουν τις αρχικές προεπιλεγμένες διαστάσεις της εικόνας.

12.1.4 Η ιδιότητα background-origin

Εκτός από την ιδιότητα background – position που ρυθμίζει την θέση της εικόνας που θα εμφανιστεί, η νέα έκδοση των Css η Css3 μας παρέχει άλλη μια ιδιότητα για τον ίδιο σκοπό, την ιδιότητα background – origin.

Η ιδιότητα αυτή μπορεί να πάρει τρεις τιμές:

- **background-origin: content-box;**

Με την τιμή content – box η θέση της εικόνας είναι να εμφανίζεται στο πάνω αριστερό μέρος του περιεχομένου του πλαισίου. Δηλαδή αν έχουμε ένα <div> στο οποίο έχουμε εισάγει ένα κείμενο τότε η εικόνα που βάζουμε για υπόβαθρο του <div> θα εμφανιστεί στο ύψος του κειμένου δηλαδή στην πάνω αριστερή άκρη που αρχίζει το κείμενο.

- **background-origin: padding-box;**

Με την τιμή padding – box η θέση της εικόνας είναι να εμφανίζεται στο πάνω αριστερό μέρος του που ορίζεται το περιθώριο padding μεταξύ του κειμένου και του περιγράμματος του πλαισίου. Με λίγα λόγια πιο πάνω έχουμε αναφέρει τον ορισμό του attribute padding το οποίο δημιουργεί ένα περιθώριο μεταξύ ενός κειμένου και του περιγράμματος του πλαισίου μέσα στο οποίο βρίσκεται το κείμενο. Ας πούμε ότι το πλαίσιο αυτό είναι ένα <div> και μέσα στο <div> έχουμε ένα κείμενο τότε αν χρησιμοποιήσουμε το padding θα δημιουργηθεί το περιθώριο μεταξύ του κειμένου και του περιγράμματος του <div>. Σε αυτήν την περίπτωση αν εισάγουμε μια εικόνα σαν υπόβαθρο του <div> τότε η θέση της εικόνας θα είναι στο πάνω αριστερό μέρος του padding.

- **background-origin: border-box;**

Με την τιμή αυτή η θέση της εικόνας τοποθετείται στο πάνω αριστερό μέρος του περιγράμματος του πλαισίου στο οποίο έχει εισαχθεί η εικόνα ως background. Αν έχουμε μια εικόνα σαν υπόβαθρο σε ένα <div> τότε η θέση της θα είναι στην επάνω αριστερή γωνία του περιγράμματος του <div>.

12.1.5 Η ιδιότητα `background-clip`

Και η ιδιότητα αυτή έχει προστεθεί με την καινούργια έκδοση την Css3. Πολλές φορές μπορεί να μην θέλουμε να εισάγουμε μια εικόνα ως υπόβαθρο ενός στοιχείου αλλά να θέλουμε να χρησιμοποιήσουμε κάποιο χρώμα – απόχρωση σαν background. Με το attribute αυτό ρυθμίζουμε την επιφάνεια του στοιχείου που θέλουμε να χρωματιστεί παίρνοντας τις παρακάτω τιμές:

- **`background-clip: content-box;`**

Με την τιμή `content – box` ορίζουμε ως επιφάνεια που θα χρωματιστεί ως υπόβαθρο την επιφάνεια του περιεχομένου του πλαισίου. Δηλαδή αν έχουμε ένα `<div>` πλαίσιο μέσα στο οποίο έχουμε εισάγει κάποιο κείμενο τότε η επιφάνεια που θα χρωματιστεί με το χρώμα του υποβάθρου θα είναι μόνο το `background` του κειμένου και όχι όλη η επιφάνεια του `<div>`.

- **`background-clip: padding-box;`**

Με την τιμή αυτή χρωματίζεται όλη η επιφάνεια του κειμένου και του `padding`. Με λίγα λόγια μέσα σε ένα `<div>` θα πάρει το χρώμα του υποβάθρου όλο το `background` του κειμένου που θα υπάρχει μέσα στο `<div>` καθώς και όλο το `background` του περιθωρίου ανάμεσα στο κείμενο και στο περίγραμμα του `<div>` δηλαδή όλο το `background` του `padding`.

- **`background-clip: border-box;`**

Με την τιμή `border – box` χρωματίζεται όλο το υπόβαθρο του στοιχείου ακόμη και του περιγράμματος του. Στο παράδειγμα λοιπόν του `<div>` θα χρωματιστεί όλο το `background` του κειμένου του `<div>`, όλο το `background` του `padding` αλλά και το υπόβαθρο του περιγράμματος του `<div>`.

12.2 Μορφοποίηση κειμένου και εισαγωγή νέας γραμματοσειράς

Αντίστοιχα και για την μορφοποίηση του κειμένου έχουμε μια λίστα από attributes που μπορούμε να χρησιμοποιήσουμε σε ένα αρχείο Css:

- color
- text - align
- text - decoration
- text - transform
- text - indent
- text - shadow

Το attribute color είναι αυτό που μας βοηθά να αλλάξουμε το χρώμα της γραμματοσειράς και συντάσσεται έτσι:

```
body { color: blue;}
```

```
h1 { color: #00ff00;}
```

```
h2 { color: rgb(255,0,0);}
```

όπως βλέπουμε και οι τιμές που παίρνει το color χωρίζονται στις γνωστές πλέον κατηγορίες των κωδικών χρωμάτων.

Το δεύτερο attribute το text - align χρησιμοποιείται για την στοίχιση του κειμένου και παίρνει τις τιμές:

- `h1 {text-align: center;}`

το center στοιχίζει το κείμενο στο κέντρο

- `h2 {text-align: right;}`

το right στοιχίζει το κείμενο στα δεξιά

- `h2 {text-align: left;}`

το left στοιχίζει το κείμενο στα αριστερά

- `p {text-align: justify;}`

το justify είναι η λεγόμενη πλήρης στοίχιση του κειμένου κάθε γραμμή δηλαδή «τεντώνει» ώστε να έχει το ίδιο περιθώριο στο δεξί και στο αριστερό μέρος του κειμένου.

Το επόμενο attribute είναι το `text-decoration` αυτό μας δίνει την δυνατότητα να διακοσμήσουμε το κείμενο με μία γραμμή η οποία θα είναι είτε πάνω από το κείμενο:

- `h1 {text-decoration: overline;}`

είτε κάτω από το κείμενο:

- `h1 {text-decoration: underline;}`

ή ακόμα και πάνω στα ίδια τα γράμματα όπως φαίνεται και στο αποτέλεσμα παρακάτω:

- `h1 {text-decoration: line-through;}`

~~This is heading 1~~

Το attribute `text-transform` έχει σαν σκοπό του να μετατρέπει όλα τα γράμματα του κειμένου σε κεφαλαία:

- `p {text-transform: uppercase;}`

ή όλα τα γράμματα σε μικρά:

- `p {text-transform: lowercase;}`

και ακόμα να μετατρέπει μόνο τα αρχικά γράμματα της κάθε λέξης σε κεφαλαία:

- `p {text-transform: capitalize;}`

Τέλος το attribute `text-indent` μας βοηθά να δημιουργήσουμε ένα περιθώριο προς τα δεξιά σε μια γραμμή κειμένου για παράδειγμα αν το χρησιμοποιήσουμε το attribute σε μια παράγραφο τότε η πρώτη γραμμή θα μεταφερθεί προς τα μέσα τόσα px όσα του έχουμε δηλώσει στο attribute:

```
p {text-indent: 50px;}
```

Εκτός από τα πέντε παραπάνω attributes η έκδοση `Css3` έχει προσθέσει ακόμα τρία σημαντικά attributes μορφοποίησης τα οποία είναι τα εξής:

- `text-overflow`
- `word-wrap`
- `word-break`

Αρχικά το `text – overflow` είναι υπεύθυνο για το «κόψιμο» ενός μακροσκελούς κειμένου που υπερβαίνει το περίγραμμα του πλαισίου στο οποίο βρίσκεται. Δηλαδή αν έχουμε ένα κείμενο μέσα σε ένα `<div>` το οποίο έχει συγκεκριμένες διαστάσεις, αλλά το κείμενο είναι τόσο μεγάλο που δεν χωράει μέσα στο `<div>` τότε το `text – overflow` «κόβει» το κείμενο που περισσεύει ώστε να εμφανιστεί τόσο κείμενο όσο ακριβώς χωράει το `<div>` το attribute αυτό μπορεί στο σημείο που κόβεται το κείμενο ή να το αφήσει ακριβώς όπως έχει κοπεί παίρνοντας την τιμή (`text-overflow: clip;`) ή να προσθέσει τρεις τελείες σαν σύμβολο συνέχειας του κειμένου με την τιμή (`text-overflow: ellipsis;`).

Το attribute `word – wrap` είναι αυτό που έχει σαν δουλειά του να «κόβει» μια μακροσκελή λέξη η οποία δεν χωράει μέσα στο πλαίσιο (για παράδειγμα ένα `<div>`) και να συνεχίζει η λέξη στην επόμενη γραμμή του κειμένου. Η τιμή που παίρνει για αυτή την δουλειά είναι μια και είναι η εξής: `word-wrap: break-word;`

Επίσης το attribute `word – break` έχει ακριβώς την ίδια δουλειά με το `word – wrap` μόνο που αυτό παίρνοντας την τιμή (`word-break: keep-all;`) κόβει την γραμμή του κειμένου που υπερβαίνει το πλαίσιο κρατώντας ολόκληρες τις λέξεις χωρίζοντας όμως την φράση συνεχίζοντας στην επόμενη γραμμή σε αντίθεση με το (`word-break: break-all;`) το οποίο κόβει την λέξη σε οποιοδήποτε χαρακτήρα της χωρίς να νοιάζεται να την κρατήσει ολόκληρη στην ίδια γραμμή, οι υπόλοιποι χαρακτήρες συνεχίζουν στην επόμενη γραμμή.

Τέλος το attribute `text – shadow` μπορεί να δημιουργήσει σκιά στην γραμματοσειρά του κειμένου (`text-shadow: 2px 2px 5px red;`). Διακρίνουμε ότι παίρνει τρεις αριθμητικές τιμές και ένα χρώμα σκιάς. Ο πρώτος αριθμός ορίζει την οριζόντια σκιά ενώ ο δεύτερος την κάθετη σκιά. Ο τρίτος είναι ο αριθμός που προσθέτει μια θόλωση γύρω από τα γράμματα και ορίζει το πάχος της. Παρόλα αυτά σκιά μπορούμε να δώσουμε εκτός από τα γράμματα και σε ένα κουτί μέσω του attribute: (`box-shadow: 2px 2px 5px red;`)

12.2.1 Αλλαγή και εισαγωγή νέων ειδών γραμματοσειράς

Σε αυτή την μικρή υποπαράγραφο θα αναφερθούμε λίγο στα είδη των γραμματοσειρών, πως μπορούμε να αλλάξουμε την γραμματοσειρά σε μια σελίδα αλλά και πως να εισάγουμε ένα αρχείο μιας πιο περίτεχνης γραμματοσειράς.

Αρχικά θα πρέπει να ξέρουμε ότι υπάρχουν τρεις μεγάλες κατηγορίες – οικογένειες γραμματοσειρών:

- Η οικογένεια γραμματοσειρών Serif. Η κατηγορία αυτή χαρακτηρίζεται από τις καταλήξεις στο σχέδιο των γραμμάτων όπως φαίνεται στην εικόνα. Η Serif περιλαμβάνει γραμματοσειρές όπως η Times New Roman και η Georgia οι οποίες αποτελούνται από αυτές τις χαρακτηριστικές καταλήξεις στο σχέδιο των γραμμάτων τους.



- Η οικογένεια γραμματοσειρών Sans – Serif. Από το όνομα της καταλαβαίνουμε ότι η διαφορά της με την Serif θα είναι ότι τα γράμματα δεν περιλαμβάνουν τις καταλήξεις στο σχέδιο τους, γιατί το Sans σημαίνει στην γαλλική γλώσσα «χωρίς». Η Sans - Serif αποτελείται από γραμματοσειρές όπως οι Arial και Verdana.



- Και τέλος η οικογένεια Monospace. Χαρακτηριστικό της κατηγορίας αυτής είναι ότι όλοι οι χαρακτήρες έχουν το ίδιο πλάτος. Σε αυτή την κατηγορία συγκαταλέγονται γραμματοσειρές όπως οι Courier New και Lucida Console.

Με το attribute font – family εισάγουμε μια γραμματοσειρά κάπως έτσι:

```
p {font-family: "Times New Roman", Times, serif;}
```



Tip: Καλό θα είναι να έχουμε μια back – up γραμματοσειρά δηλαδή όταν εισάγουμε μια γραμματοσειρά θα πρέπει να βάζουμε μετά το κόμμα ένα δεύτερο όνομα γραμματοσειράς σε περίπτωση που ο browser δεν μπορέσει να αναγνωρίσει την πρώτη τότε να εμφανίσει την δεύτερη. Επίσης μπορούμε να προσθέσουμε και την κατηγορία της πρώτης γραμματοσειράς ώστε αν δεν εμφανιστεί να εισάγει αυτόματα ο browser μια παρόμοια γραμματοσειρά από την ίδια οικογένεια. Η διαδικασία αυτή φαίνεται στο παραπάνω τμήμα κώδικα.

Άλλο ένα χρήσιμο attribute είναι το font – size το οποίο μορφοποιεί το κείμενο χρησιμοποιώντας τις εξής τιμές:

- `p {font-style: normal;}`

με την τιμή normal το κείμενο εμφανίζεται στην φυσιολογική του μορφή.

- `p {font-style: italic;}`

με την τιμή italic το κείμενο εμφανίζεται σε πλάγια γραφή

- `p {font-style: oblique;}`

η μορφή της τιμής αυτής είναι περίπου ίδια με την μορφή της italic γιατί το κείμενο εμφανίζεται σε πλάγια γραφή αλλά δεν υποστηρίζεται τόσο εύκολα όσο η τιμή italic.

Επιπλέον μπορούμε να χειριστούμε και το πόσο έντονα θα εμφανίζονται τα γράμματα μέσα από το attribute font – weight το οποίο παίρνει τις παρακάτω τιμές:

- `p {font-weight: normal;}`

με την τιμή normal το κείμενο εμφανίζεται στην φυσιολογική του μορφή.

- `p {font-weight: bold;}`

με την τιμή bold το κείμενο εμφανίζεται με τα γράμματα σε πιο έντονη γραφή.

- `p {font-weight: 900;}`

με την αριθμητική αυτή τιμή τα γράμματα εμφανίζονται πιο έντονα και από την bold γραφή. Η τιμή 400 είναι η αντίστοιχη της τιμής normal ενώ η τιμή 700 αντιστοιχεί στην γραφή bold.

Τέλος το attribute font – size είναι αυτό που μας βοηθά να ρυθμίσουμε το μέγεθος της γραμματοσειράς με τρεις διαφορετικούς τρόπους:

- `p {font-size: 14px;}` μέγεθος γραμματοσειράς σε pixels (px).
- `p {font-size: 14em;}` μέγεθος γραμματοσειράς σε em.
- `p {font-size: 50%;}` μέγεθος γραμματοσειράς σε percent (%).



Tip: Η προεπιλεγμένη τιμή για φυσιολογικό μέγεθος γραμματοσειράς είναι τα 16px τα οποία ισοδυναμούν με 1em. Σε μια σελίδα μπορούμε για το μέγεθος της γραμματοσειράς των διαφόρων στοιχείων να έχουμε ένα συνδυασμό των τριών διαφορετικών μετρικών. Συνίσταται όμως όταν έχουμε ένα τέτοιο συνδυασμό να μορφοποιούμε το μέγεθος της γραμματοσειράς του στοιχείου <body> σε percent (%) για να μπορεί να αναγνωρίσει ο browser όλα τα διαφορετικά μεγέθη γραμματοσειρών που χρησιμοποιούνται στα υπόλοιπα διάφορα στοιχεία.

Όπως έχουμε αναφέρει και σε παραπάνω παραγράφους εκτός από τις standard γραμματοσειρές μπορεί να θέλουμε να μορφοποιήσουμε ένα κείμενο με μια πιο περίτεχνη γραμματοσειρά. Όμως σε αυτό το σημείο προκύπτει ένα πρόβλημα. Αν η γραμματοσειρά που θέλουμε να χρησιμοποιήσουμε δεν είναι εγκατεστημένη στον υπολογιστή του χρήστη της σελίδας μας τότε ο browser δεν θα μπορεί να την αναγνωρίσει και κατά συνέπεια να την εμφανίσει στην οθόνη.

Για να το λύσουμε αυτό το πρόβλημα μπορούμε να χρησιμοποιήσουμε από τις καλλιτεχνικές γραμματοσειρές οι οποίες βρίσκονται στο διαδίκτυο και μπορούμε να τις ενσωματώσουμε στο αρχείο Css που έχουμε με αποτέλεσμα όταν θα δημοσιοποιήσουμε την σελίδα μας στο διαδίκτυο μέσω ενός server, μαζί με τα αρχεία Html και Css θα ανέβει και το αρχείο της γραμματοσειράς. Έτσι κάθε φορά που ο χρήστης θα βλέπει την σελίδα μας αυτόματα ο browser αφού θα τσεκάρει ότι την συγκεκριμένη γραμματοσειρά δεν είναι εγκατεστημένη στον υπολογιστή θα την «κατεβάξει» και θα την εγκαθιστά κανοντας πραγματικότητα την εμφάνιση της στην σελίδα.

Για την διαδικασία αυτή χρησιμοποιούμε τον κανόνα που λέγεται @font – face.

```
@font-face {font-family: myFirstFont;  
             src: url(sansation_bold.woff);  
             font-weight: bold;  
             }
```

Με αυτό το τμήμα του κώδικα δίνουμε ένα όνομα που θέλουμε στην γραμματοσειρά που εισάγουμε μέσω της γραμμής (**font-family: myFirstFont;**), και εισάγουμε το αρχείο της γραμματοσειράς δίνοντας την διεύθυνση και το όνομα του αρχείου (**src: url(sansation_bold.woff);**).

Επίσης ορίζουμε ότι θα χρησιμοποιήσουμε την πλάγια γραφή της συγκεκριμένης γραμματοσειράς. Πρέπει να τονίσουμε ότι υπάρχουν πολλοί και διαφορετικοί τύποι αρχείων γραμματοσειράς. Για παράδειγμα υπάρχουν τα αρχεία ttf , τα woff, τα otf κ.α. τα πιο υποστηριζόμενα αρχεία γραμματοσειρών από όλους τους γνωστούς browsers είναι τα ttf γι' αυτό συνιστάται να χρησιμοποιούμε για την εισαγωγή μιας γραμματοσειράς τον τύπο αρχείου ttf.

Τέλος για να καλέσουμε την γραμματοσειρά που προσθέσαμε στα διάφορα στοιχεία που θέλουμε να μορφοποιήσουμε, πηγαίνουμε στους αντίστοιχους selectors και την καλούμε μέσω του attribute font – family που παίρνει σαν τιμή το όνομα που έχουμε δώσει κατά την εισαγωγή της στο αρχείο Css.

```
div {font-family: myFirstFont;}
```

13 Μορφοποίηση πίνακα , λίστας μέσω του Css

Σε αυτό το κεφάλαιο θα αναφερθούμε στην μορφοποίηση ενός πίνακα και μιας λίστας μέσω του αρχείου Css καθώς επίσης θα μιλήσουμε στις υποπαραγράφους και για ορισμένες ιδιότητες – attributes τα οποία χρησιμοποιούνται εκτός από την μορφοποίηση του πίνακα ή ης λίστας και άλλα στοιχεία όπως τα πολύ γνωστά και χρήσιμα <div>.

13.1 Μορφοποίηση πίνακα

Για την μορφοποίηση του πίνακα σαν selectors χρησιμοποιούμε τα ονόματα των αντίστοιχων tags δηλαδή τα: **table, tr, td, th.**

Σε ένα πίνακα μπορούμε να αλλάξουμε και να ρυθμίσουμε την εμφάνιση του περιγράμματος του ίδιου του πίνακα αλλά και την εμφάνιση του περιγράμματος των κελίων ξεχωριστά. Επίσης μας παρέχεται η δυνατότητα να προσθέσουμε ή να αλλάξουμε το περιθώριο μεταξύ του περιεχομένου του κελιού με το περίγραμμα του (padding), ή τα περιθώρια ανάμεσα στα περιγράμματα των κελιών και του ίδιου του πίνακα ή ακόμα το κενό ανάμεσα στο περίγραμμα του ίδιου του πίνακα σε σχέση με το υπόλοιπο περιεχόμενο της σελίδας. Αυτά τα περιθώρια ελέγχονται από το attribute margin. Επιπλέον οι διαστάσεις του πίνακα και των κελιών, η θέση και η στοίχιση του ίδιου του πίνακα σε σχέση με την σελίδα αλλά και του περιεχομένου των

κελιών καθώς επίσης και η μορφοποίηση του background του πίνακα αποτελούν μέρη της μορφοποίησης που καλούμαστε να κάνουμε σε ένα πίνακα. Πάμε λοιπόν να δούμε στις παρακάτω υποπαραγράφους ένα – ένα τα μέρη αυτά της μορφοποίησης.

13.1.1 Μορφοποίηση περιγράμματος ενός πίνακα (border)

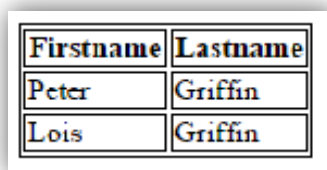
Το περίγραμμα ενός πίνακα μορφοποιείται με το attribute border το οποίο επεκτείνεται στο όνομα του ανάλογα με την δουλειά που πρέπει να κάνει. Για παράδειγμα παρακάτω παρουσιάζεται μια λίστα με την σύνταξη και την δουλειά που πρέπει να κάνει κάθε διαφορετική επέκταση του attribute border.

- `table, th, td {border: 1px solid black;}`

το απλό border ορίζει την αρχική εμφάνιση του περιγράμματος. Μπορεί να πάρει τρεις τιμές χωρίς να χωρίζονται μεταξύ τους με κόμμα, και αυτές ρυθμίζουν το πάχος του περιγράμματος (`1px`), το στυλ του αν θα είναι δηλαδή μια συνεχόμενη γραμμή, μια διακοπτόμενη, διπλή κ.α. (`solid`), καθώς επίσης και το χρώμα του περιγράμματος (`black`). Επίσης το χρώμα του περιγράμματος αν θέλουμε μπορεί να είναι διαφανές κάτι το οποίο το πετυχένουμε χρησιμοποιώντας σαν χρώμα την λέξη «`transparent`».

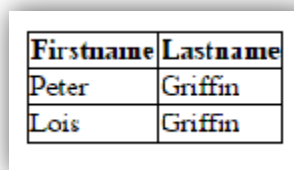
- `table {border-collapse: collapse;}`

με το attribute αυτό δηλώνουμε ότι θέλουμε τα περιγράμματα των κελιών να ενωθούν με το περίγραμμα του πίνακα γιατί εξ' αρχής τα περιγράμματα των κελιών και του πίνακα είναι ξεχωριστά μεταξύ τους.



A table with two columns: 'Firstname' and 'Lastname'. The first row contains 'Peter' and 'Griffin'. The second row contains 'Lois' and 'Griffin'. The table has a thick border and the borders of the individual cells are visible, creating a grid-like appearance.

Firstname	Lastname
Peter	Griffin
Lois	Griffin



A table with two columns: 'Firstname' and 'Lastname'. The first row contains 'Peter' and 'Griffin'. The second row contains 'Lois' and 'Griffin'. The table has a thick border and the borders of the individual cells are not visible, appearing as a single solid border around the entire table.

Firstname	Lastname
Peter	Griffin
Lois	Griffin

- `table, th, td {border-radius: 10px 25px 25px 10px;}`

με το border – radius μας δίνεται η δυνατότητα να στρογγυλέψουμε τις γωνίες του πίνακα και των κελιών. Μπορεί να πάρει έως και τέσσερις τιμές οι οποίες με τους αριθμούς που δίνουμε δηλώνουμε πόση έλλειψη θέλουμε να γίνει στις τέσσερις γωνίες.

Ο πρώτος αριθμός απευθύνεται στην επάνω αριστερή γωνία, ο δεύτερος αριθμός αναφέρεται στην επάνω δεξιά γωνία, ο τρίτος στην κάτω δεξιά και ο τέταρτος στην κάτω αριστερή γωνία. Αν χρησιμοποιήσουμε μόνο δύο τιμές τότε ο πρώτος αριθμός θα μορφοποιήσει την πάνω αριστερή και την κάτω δεξιά γωνία ενώ ο δεύτερος την πάνω δεξιά και την κάτω αριστερή. Τέλος αν έχουμε μόνο μια τιμή τότε σημαίνει ότι όλες οι γωνίες θα αλλάξουν με τον ίδιο αριθμό που έχουμε δώσει.

- `table, th, td {border-image: url(border.png);}`
μέσω του `border – image` μπορούμε να εισάγουμε μια εικόνα η οποία θα χρησιμοποιείται σαν περίγραμμα στον πίνακα και στα κελιά.

13.1.2 Οι ιδιότητες `padding` και `margin`

Παρακάτω παρουσιάζεται μια οπτική αναπαράσταση των attributes `padding` και `margin` σε σχέση με τα περιγράμματα του πίνακα και των κελιών ή άλλων στοιχείων.



Ο ορισμός του `padding` και του `margin` είναι όμοιος του `border – radius` αφού και αυτά παίρνουν τέσσερις τιμές δηλώνοντας τις τέσσερις κενές περιοχές που θέλουμε να δημιουργήσουμε ανάμεσα στο περιεχόμενο του στοιχείου με το περίγραμμά του και ανάμεσα στο περίγραμμά του στοιχείου και στο υπόλοιπο περιεχόμενο της σελίδας.

Έτσι έχουμε την παρακάτω σύνταξη του κώδικα:

- `table, th, td {padding: 10px 25px 25px 10px;}`
- `table {margin: 10px 25px 25px 10px;}`

μπορούμε όμως να ορίσουμε την κάθε περιοχή ξεχωριστά:

- ```
table, th, td { padding-top: 25px;
 padding-right: 50px;
 padding-bottom: 25px;
 padding-left: 50px;
 }
```
- ```
table { margin-top: 25px;
         margin-right: 50px;
         margin-bottom: 25px;
         margin-left: 50px;
       }
```

13.1.3 Μορφοποίηση ύψους / πλάτους του πίνακα

Για την αλλαγή του ύψους υπάρχει το attribute height ενώ για το πλάτος χρησιμοποιούμε το attribute width όπως παρουσιάζεται παρακάτω:

```
table, th, td { height: 100px;
                 width: 100px;
                 }
```



Tip: Το ύψος και το πλάτος σε ένα στοιχείο μπορούν να οριστούν σε px, σε em ή σε %. Στην περίπτωση που χρησιμοποιήσουμε percent % για τις διαστάσεις ενός εσωτερικού στοιχείου τότε οι διαστάσεις θα υπολογιστούν βάσει των διαστάσεων του ανώτερου εξωτερικού στοιχείου. Με λίγα λόγια αν το πλάτος του πίνακα είναι 80% τότε εννοούμε ότι το πλάτος καταλαμβάνει το 80% της επιφάνειας της σελίδας. Σε αυτήν την περίπτωση αν το πλάτος του κελιού του πίνακα οριστεί στο 20% τότε αυτόματα σημαίνει ότι το πλάτος του κελιού καταλαμβάνει το 20% του συνολικού πλάτους του ανώτερου στοιχείου δηλαδή του πίνακα.

13.2 Μορφοποίηση λίστας

Και αφού τελειώσαμε με την ανάλυση της βασικής μορφοποίησης ενός πίνακα πάμε να δούμε πως γίνεται και η μορφοποίηση της λίστας. Όπως έχουμε αναφέρει και παραπάνω μέσω μιας λίστας μπορούμε να δημιουργήσουμε ένα εύχρηστο μενού πλοήγησης για την σελίδα μας.

Στις επόμενες υποπαραγράφους θα δούμε πως μπορούμε να μετατρέψουμε μια λίστα σε ένα μενού μέσω Css.

13.2.1 Μετατροπή λίστας σε μενού πλοήγησης

Για την μετατροπή αυτή δεν χρειάζεται να κάνουμε κάτι στο αρχείο Html αρκεί να έχουμε φτιάξει μια λίστα με τα ονόματα που θέλουμε να δώσουμε στα κουμπιά του μενού. Θα πρέπει να ξέρουμε ότι τα κουμπιά στην ουσία θα είναι υπερσύνδεσμοι (links) που θα οδηγούν στις αντίστοιχες σελίδες άρα λοιπόν κάθε μέλος - tag της λίστας θα περικλείει ένα tag <a> των υπερσυνδέσμων. Παρακάτω παρουσιάζεται ο κώδικας μιας λίστας του αρχείου Html καθώς και η μορφοποίηση της μέσω του αρχείου Css. Αναφέρεται επίσης ότι η ουσιαστική μετατροπή της λίστας σε μενού γίνεται μέσω των attributes που χρησιμοποιούνται στο αρχείο Css.

```
<html> <head>
<link rel="stylesheet" href="css/index.css">
<title> navigation bar </title> </head>
<body>
<ul id='menu'>
  <li><a href='#'>Planets</a>
    <ul>
      <li><a href='#'>Mercury</a></li>
      <li><a href='#'>Venus</a></li>
      <li><a href='#'>Earth</a></li>
    </ul>
  </li>
  <li><a href='#'>Stars</a>
    <ul>
      <li><a href='#'>Sun </a></li>
      <li><a href='#'>Betelgeuse</a></li>
      <li><a href='#'>Bellatrix</a></li>
    </ul>
  </li>
  <li><a href='#'>Galaxies</a>
```



```

<ul>
  <li><a href='#>Milky Way </a></li>
  <li><a href='#>Andromeda</a></li>
  <li><a href='#>Antennae</a></li>
</ul>
</li>
</ul>
</body>
</html>

```

```

/*DROPDOWN FUNCTIONALITY*/

```

```

ul { padding:0; margin:0}
a { display:block; text-decoration: none;}
li { display:block; float:left; width:110px;}
li ul li { float:none;}
li ul { display:none; position:absolute; z-index:1}
li:hover ul { display:block;}

```

```

/*DROPDOWN STYLING:*/

```

```

#menu { height:30px}
#menu a { color:#024E67; padding:10px 15px 10px 15px;}
#menu a:hover { color:#ffffff;}
#menu li { background-color:#FFF; border:solid 1px #CCC; margin-left:-1px}
#menu li:hover { background-color:#61b4cf}

```

Το αποτέλεσμα του παραπάνω κώδικα είναι το εξής:

Planets	Stars	Galaxies
	Sun	
	Betelgeuse	
	Bellatrix	

Όμως ας πούμε δυο λόγια για το παραπάνω τμήμα κώδικα ο οποίος αποτελείται από δύο διαφορετικά αρχεία το Css και το Html. μέσα στο αρχείο Html κατασκευάζεται μια εμφωλευμένη λίστα που σημαίνει ότι τα μέλη της εξωτερικής λίστας περικλείουν τα μέλη κάποιας εσωτερικής. Η εμφωλευμένη αυτή λίστα μετατρέπεται σε μενού μέσω του αρχείου Css. Όπως θα αναλύσουμε και στο τελευταίο κεφάλαιο του οδηγού τα αρχεία Html ενώνονται με τα αντίστοιχα Css τα οποία μπορεί να είναι είτε εσωτερικά αρχεία Css είτε εξωτερικά. Στο παραπάνω παράδειγμα έχουμε ένα εξωτερικό αρχείο Css και το συνδέουμε με το αρχείο Html μέσω της παρακάτω γραμμής κώδικα:

```
<link rel="stylesheet" href="css/index.css">
```

Όμως αυτή την γραμμή κώδικα θα την αναλύσουμε στο τελευταίο κεφάλαιο.

Είπαμε λοιπόν ότι η λίστα είναι εμφωλευμένη δηλαδή υπάρχει μια λίστα μέσα σε μια άλλη. Παρατηρούμε ότι κάθε μέλος της λίστας περικλείει έναν υπερσύνδεσμο <a> ο οποίος χρησιμοποιείται για την πλοήγηση του μενού στις αντίστοιχες σελίδες:

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

Μια ακόμη παρατήρηση που μπορούμε να κάνουμε είναι η χρήση της ιδιότητας id του selector όπως έχουμε προαναφέρει.

```
<ul id='menu'>
```

Στην περίπτωση μας χρησιμοποιείται το id για να αποφύγουμε τυχόν προβλήματα αν θέλουμε να εφαρμόσουμε την μορφοποίηση μόνο στην λίστα του μενού και παράλληλα υπάρχει και κάποια άλλη λίστα πιο κάτω στον κώδικα. Για να μην υπάρξει λοιπόν σύγχυση και για να ξεχωρίσουμε, να κάνουμε ιδιαίτερη την λίστα που προορίζεται για μενού σε σχέση με υπόλοιπες λίστες που μπορεί να υπάρχουν κατασκευάζουμε έναν selector στον οποίο θα απευθυνόμαστε με το όνομα που έχουμε δώσει στο id.

Ας πάμε όμως να αναλύσουμε τον κώδικα του αρχείου Css. Όπως βλέπουμε ο κώδικας έχει χωρίσει νοητά από μια γραμμή σχολίων σε δύο μέρη.

Το πρώτο μέρος είναι αυτό που με τα attributes που χρησιμοποιεί μετατρέπει την λίστα σε μενού.

/*DROPDOWN FUNCTIONALITY*/

```
ul { padding:0; margin:0}
a { display:block; text-decoration: none;}
li { display:block; float:left; width:110px;}
li ul li { float:none;}
li ul { display:none; position:absolute; z-index:1 }
li:hover ul { display:block;}
```

Έχουμε να παρατηρήσουμε λοιπόν ότι μέσω του selector `ul` οι κενές περιοχές μεταξύ των μελών της λίστας και του περιεχομένου της σελίδας και μεταξύ του κειμένου των μελών και του περιγράμματος τους σε όλες τις πλευρές είναι 0 (**`ul { padding:0; margin:0}`**). Μέσω του selector `a` ρυθμίζουμε να μην υπάρχει καμία μορφοποίηση στο κείμενο των υπερσυνδέσμων (**`text-decoration: none;`**) και ορίζουμε ένα νοητό κουτί που περικλείει τους υπερσυνδέσμους και παίρνει την τιμή `block` κάτι που σημαίνει ότι το κουτί αυτό θα έχει τις ιδιότητες ενός στοιχείου `block` όπως για παράδειγμα το στοιχείο `<p>` (**`display:block;`**).



Tip: Όπως έχουμε αναφέρει κάθε στοιχείο έχει τον δικό του `display` τύπο δηλαδή ή θα είναι στοιχείο `inline` ή θα είναι `block` στοιχείο. Στην περίπτωση της λίστας που θ'έλουμε να μετατρέψουμε σε μενού χρειάζεται να μετατρέψουμε τον τύπο της λίστας σε `block` στοιχείο. Για να το πετύχουμε αυτό χρησιμοποιούμε το attribute `display`.

Έτσι λοιπόν μετατρέπουμε σε `block` στοιχείο και τα μέλη `li` της λίστας (**`display:block`**) και για να δημιουργήσουμε ένα οριζόντιο μενού τοποθετούμε τα μέλη της λίστας το ένα δίπλα στο άλλο μέσω του attribute `float` (**`float:left;`**). επίσης ορίζουμε το πλάτος των κουμπιών – μελών της λίστας (**`li`**) να έχουν πλάτος `110px` (**`width:110px;`**).

Στην εικόνα αυτή βλέπουμε οπτικά ότι ο selector **`li`** μορφοποιεί τα μέλη `li` της εξωτερικής λίστας. Πλέον τα μέλη αυτά έχουν μετατραπεί στα κεντρικά κουμπιά του μενού και το attribute (**`width:110px;`**) στο πλάτος του κάθε κουμπιού:



Από την άλλη μεριά τα μέλη της εσωτερικής λίστας που βρίσκονται μέσα στα μέλη της εξωτερικής και που στην οθονία αποτελούν το υπομενού των κεντρικών κουμπιών χρησιμοποιούν το attribute **float** παίρνοντας όμως την τιμή none γιατί δεν θέλουμε τα μέλη του υπομενού να βρίσκονται το ένα δίπλα στο άλλο αλλά να έχουν κάθετη στοίχιση. Άρα ο selector **li ul li** αναφέρεται στα μέλη της εσωτερικής λίστας όπως τα βλέπουμε στην παρακάτω εικόνα:



Ο selector **li ul** αναφέρεται στην εσωτερική λίστα σαν σύνολο και μορφοποιείται με τα attributes **display, position** και **z – index**. Όταν το attribute display παίρνει την τιμή none σημαίνει ότι ή θέλει να κρύψει ή να εμφανίσει ένα στοιχείο χωρίς να χρειάζεται να το διαγράψει ή να το ξαναδημιουργήσει. στην προκειμένη περίπτωση χρησιμοποιείται για να κρύψει το περίγραμμα της εσωτερικής λίστας και να δημιουργηθεί η ψευδαίσθηση ότι τα κουμπιά της εσωτερικής λίστας ενώνονται με τα κουμπιά της εξωτερικής λίστας.

Η εσωτερική λίστα δεν σταματά να υπάρχει ούτε έχει διαγραφεί από το περιεχόμενο της σελίδας απλά έχει κρυφτεί και δεν οπτικά στον χρήστη. Το attribute **position** είναι ένα από τα πλέον σημαντικά attributes και μάλιστα όταν παίρνει την τιμή absolute κατατάσσεται ως άξιο προσοχής από έναν σχεδιαστή.

Η τιμή **absolute** είναι και η πιο δύσκολη τιμή στο θέμα κατανόησης της λειτουργίας της. Μέσω της τιμής αυτής στοιχίζεται ένα στοιχείο στην ακριβή θέση που το θέλουμε αρκεί μόνο να βρίσκεται εμφωλευμένο μέσα σε ένα άλλο το οποίο χρησιμοποιεί και αυτό το attribute position. Αν το στοιχείο δεν βρίσκεται εμφωλευμένο μέσα σε κάποιο άλλο αλλά παρόλα αυτά χρησιμοποιεί το position με τιμή absolute τότε στοιχίζεται βάσει της σελίδας και κάθε φορά που η σελίδα κυλάει (**scroll**) προς τα κάτω μετακινείται μαζί με την σελίδα. Επίσης πρέπει να αναφέρουμε ότι με την τιμή absolute το στοιχείο δεν επηρεάζει την θέση των υπόλοιπων στοιχείων μέσα στην σελίδα. Με το attribute **z – index** χρησιμοποιείται για να δηλώσουμε ότι η εσωτερική λίστα με τα μέλη της πρέπει να μην υπερκαλύπτεται οπτικά από τα υπόλοιπα στοιχεία

της σελίδας. Με λίγα λόγια όταν θα ανοίγει το υπομενού θα πρέπει να είναι στην επιφάνεια της σελίδας και να καλύπτει το υπόλοιπο περιεχόμενο της σελίδας.

Τέλος υπάρχει ένας ακόμη selector ο **li:hover ul** ο οποίος αναφέρεται στην κατάσταση hover της εσωτερικής λίστας. Η κατάσταση hover σημαίνει την λειτουργία κατά την οποία συντελείται με το πέρασμα του δείκτη του ποντικιού πάνω από το κουμπί – μέλος της εξωτερικής λίστας. Δηλαδή αν ο δείκτης περάσει πάνω από το **li** της εξωτερικής λίστας τότε ανοίγει το υπομενού με τα μέλη της εσωτερικής λίστας. Το **hover** λοιπόν χρησιμοποιεί το attribute display με την τιμή block μετατρέποντας την κατάσταση αυτή σε block στοιχείο.

Το δεύτερο μέρος χρησιμοποιεί πολλά από τα attributes που έχουμε δει μέχρι τώρα για να μορφοποιήσει την λίστα.

```
/*DROPDOWN STYLING:*/
```

```
#menu { height:30px }
```

```
#menu a { color:#024E67; padding:10px 15px 10px 15px; }
```

```
#menu a:hover { color:#ffffff; }
```

```
#menu li { background-color:#FFF; border:solid 1px #CCC; margin-left:-1px }
```

```
#menu li:hover { background-color:#61b4cf }
```

Όπως βλέπουμε για την μορφοποίηση του μενού πλέον χρησιμοποιούμε την ιδιότητα id του selector. Δηλαδή δίνουμε ένα όνομα το οποίο θα το χρησιμοποιήσουμε για selector στο Css με σκοπό να μορφοποιήσουμε όπως θέλουμε την λίστα – μενού ξεχωρίζοντας την από τις υπόλοιπες λίστες που μπορεί να υπάρχουν στο περιεχόμενο της σελίδας.

Έτσι λοιπόν έχουμε:

Ο selector **#menu** με το attribute (**height:30px**) ρυθμίζει το ύψος των κεντρικών κουμπιών του μενού δηλαδή των μελών της εξωτερικής λίστας.

Ο selector **#menu a** ρυθμίζει το χρώμα της γραμματοσειράς των υπερσυνδέσμων των κεντρικών κουμπιών του μενού (**color:#024E67**) καθώς επίσης και το περιθώριο μεταξύ των γραμμμάτων και του περιγράμματος των κουμπιών (**padding:10px 15px 10px 15px;**).

Ο selector **#menu a:hover** ρυθμίζει το χρώμα της γραμματοσειράς των υπερσυνδέσμων των κουμπιών του μενού και των υπομενού που αλλάζει όταν περνάει ο δείκτης του ποντικιού (**color:#ffffff;**).

O selector **#menu li** ρυθμίζει το χρώμα υποβάθρου των κουμπιών του μενού και των υπομενού (**background-color:#FFF**), το χρώμα και το πάχος του περιγράμματος του μενού (**border:solid 1px #CCC**) όπως επίσης και το αριστερό περιθώριο μεταξύ των κουμπιών και του περιεχομένου της σελίδας (**margin-left:-1px**).

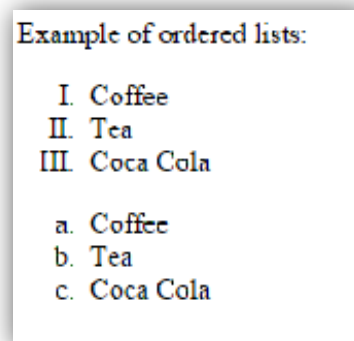
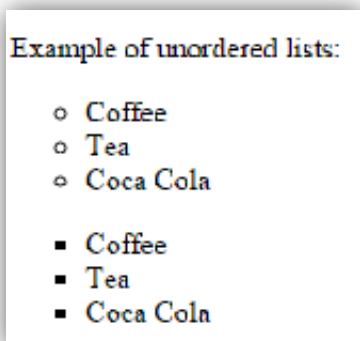
Τέλος ο selector **#menu li:hover** είναι αυτός που ρυθμίζει το χρώμα του υποβάθρου του μενού και των υπομενού που αλλάζει όταν ο δείκτης περνάει πάνω από το μενού (**background-color:#61b4cf**).

13.2.2 Μορφοποίηση ταξινόμησης απλής λίστας

Έχουμε αναφέρει στο κεφάλαιο της λίστας ότι υπάρχουν οι ταξινομημένες λίστες με αρίθμηση και οι ταξινομημένες λίστες με κουκίδες. Μπορούμε να ορίσουμε μέσω Css το τύπο της αρίθμησης ή τον τύπο της κουκίδας. Για να το πετύχουμε αυτό χρησιμοποιούμε το attribute list – style – type το οποίο συντάσσεται και παίρνει τιμές όπως φαίνεται παρακάτω:

```
ul {list-style-type: circle;}  
ul {list-style-type: square;}  
ol {list-style-type: upper-roman;}  
ol {list-style-type: lower-alpha;}
```

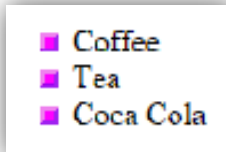
Οι τιμές που παίρνει παραπάνω μας δίνουν το παρακάτω αποτέλεσμα:



Εκτός όμως από τους αριθμούς και τα σύμβολα που μπορούμε να χρησιμοποιήσουμε για την ταξινόμηση της λίστας μας δίνεται η δυνατότητα για την χρήση εικόνας ως σύμβολο ταξινόμησης μέσω του attribute `list-style-image` – `style` – `image`:

```
ul { list-style-image: url('sqpurple.gif');}
```

Το αποτέλεσμα του attribute είναι το παρακάτω:



14 Οι ιδιότητες – attributes `position`, `display` και `float`

Στο κεφάλαιο αυτό θα αναλύσουμε τις λειτουργίες των παραπάνω attributes που μέχρι τώρα έχουμε δει πως λειτουργούν μόνο μέσα από παραδείγματα και μόνο με συγκεκριμένες τιμές. Γι' αυτό στο κεφάλαιο αυτό θα δούμε όλες τις τιμές που μπορούν να πάρουν το κάθε ένα από αυτά τα attributes.

14.1 Το attribute `position`

Με το attribute `position` ρυθμίζουμε την θέση ενός στοιχείου σε σχέση με το ανώτερο στοιχείο μέσα στο οποίο μπορεί να είναι εμφωλευμένο ή σε σχέση με την ίδια την σελίδα. Το attribute αυτό παίρνει τις εξής τιμές:

- `h1 {position: static;}`
με την τιμή `static` δεν στοιχίζεται κάπου το στοιχείο. Είναι η προεπιλεγμένη τιμή.
- `h1 {position: relative;}`
με την τιμή `relative` στοιχίζεται το στοιχείο όπως και με την τιμή `static` εκτός και αν χρησιμοποιήσουμε και τις ιδιότητες `top`, `left`, `right` και `bottom` που στην ουσία αυτές ορίζουν την θέση στοίχισης που θέλουμε. Η τιμή `relative` επηρεάζει την θέση του υπόλοιπου περιεχομένου της σελίδας αφού αν υπάρξει κενό ανάμεσα στο περιεχόμενο της σελίδας και το στοιχισμένο στοιχείο, δεν καταληφθεί από το περιεχόμενο.

- `h1 {position: absolute;}`

με την τιμή αυτή το στοιχείο στοιχίζεται βάσει είτε του ανωτέρου στοιχείου μέσα στο οποίο μπορεί να είναι εμφωλευμένο και χρησιμοποιεί και αυτό το attribute position έχοντας την δυνατότητα να πάρει όλες τις τιμές εκτός της static, είτε αν είναι ένα αυτόνομο στοιχείο στοιχίζεται βάσει της σελίδας. Αν η σελίδα κυλάει προς τα κάτω (scroll) τότε κυλάει και το στοιχείο με την τιμή absolute. Πρέπει να τονίσουμε ότι είναι μια τιμή που δεν επηρεάζει την θέση του περιεχομένου της σελίδας.

- `h1 {position: fixed;}`

τέλος με την τιμή αυτή το στοιχείο στοιχίζεται και παίρνει μια θέση standard σαν να «καρφιτσώνεται» πάνω στο παράθυρο του browser και μετακινείται μαζί με την σελίδα συνήθως χρησιμοποιούμε αυτή την τιμή στα μενού πλοήγησης ώστε να μένουν σταθερά στην θέση τους στον browser και να κυλούν προς τα κάτω μαζί με την σελίδα με σκόπο να μην χρειάζεται να «ανεβαινει» ο χρήστης προς τα πάνω στην σελίδα κάθε φορά που χρειάζεται να χρησιμοποιήσει το μενού.

14.2 To attribute display

Με την χρήση του attribute display καθορίζουμε αν ένα στοιχείο θα είναι block ή inline και φυσικά έχει τις δύο αυτές τιμές:

- `h1 {display: block;}`

η συγκεκριμένη τιμή μπορεί να μετατρέψει ένα inline στοιχείο σε block όπως το στοιχείο <div>

- `h1 {display: inline;}`

η τιμή inline αντίθετα μπορεί να μετατρέψει ένα block στοιχείο σε inline όπως τα στοιχεία της λίστας και το .

- `h1 {display: none;}`

με την τιμή αυτή μπορούμε να ορίσουμε αν θα κρυφτεί οπτικά ένα στοιχείο ή θα εμφανιστεί στην σελίδα χωρίς όμως να το διαγράψουμε ή να το ξαναδημιουργήσουμε. Όταν το στοιχείο κρύβεται με την τιμή none είναι σαν να μην υπάρχει στην σελίδα αφού δεν καταλαμβάνει κάποιο χώρο ανάμεσα στα υπόλοιπα στοιχεία της σελίδας.

Για να κρύψουμε ένα στοιχείο αλλά παράλληλα να συνεχίζει να υπάρχει στην σελίδα και να καταλαμβάνει ένα συγκεκριμένο χώρο μεταξύ των υπόλοιπων στοιχείων χρησιμοποιούμε το attribute `visibility` το οποίο παίρνει την τιμή `hidden`:

```
h1 {visibility: hidden;}
```

14.3 Το attribute `float`

Με το attribute `float` ορίζουμε ότι τα στοιχεία μεταξύ τους θα στοιχίζονται το ένα δίπλα στο άλλο δηλαδή θα κυλάνε και θα παίρνουν θέση κάθε στοιχείο το ένα δίπλα στο άλλο ανάλογα με την πλευρά που έχουμε ορίσει στην τιμή του attribute. Αν χρησιμοποιήσουμε την τιμή **right** τότε τα στοιχεία στοιχίζονται στην δεξιά πλευρα του διπλανού στοιχείου τους. Αντίστοιχα με την τιμή **left** στοιχίζονται στην αριστερή μεριά των διπλανών στοιχείων. Υπάρχει όμως και μια τιμή που λέγεται **clear:both** και η οποία απενεργοποιεί κάθε ιδιότητα `float` που μπορεί να έχει χρησιμοποιηθεί σε ένα στοιχείο και σε αυτή την περίπτωση το στοιχείο παίρνει θέση κάτω από τα υπόλοιπα στοιχεία της σελίδας.

- `img {float: right;}`
- `img {float: left;}`
- `img {clear: both;}`

15 Δημιουργία animation – Κανόνες `@keyframes`

Ένας λόγος που χρησιμοποιούμε τα `Css` είναι και η δυνατότητα που μας παρέχεται να δημιουργούμε animations δηλαδή κινούμενα σχέδια κάτι που δεν μπορούσαμε να κάνουμε στην `Html`. Αυτή η δυνατότητα μας δίνεται μέσω του κανόνα `@keyframes` με τα attributes που μπορεί να πάρει. Επίσης θα πρέπει να ξέρουμε ότι μέσω του `Css` κατασκευάζουμε animations χωρίς την χρήση της `Javascript` ή του `Flash`.

Ένας κανόνας `@keyframes` μπορεί να αλλάξει το στυλ ενός αντικειμένου σε συγκεκριμένες χρονικές στιγμές που θα του ορίσουμε και με συγκεκριμένη ταχύτητα. Επιπλέον ένας τέτοιος κανόνας θα πρέπει να αναφέρεται σε ένα animation το οποίο συνδέεται σε ένα στοιχείο για παράδειγμα ας κατασκευάσουμε ένα animation το οποίο θα συνδέεται σε ένα `<div>` και θα αλλάζει σταδιακά με συγκεκριμένη ταχύτητα το χρώμα του υποβάθρου του `<div>` από κόκκινο σε μπλέ.

Παρακάτω δίνεται ο κώδικας για το παράδειγμα:

```
/* The animation code */
@keyframes example {
  from {background-color: red;}
  to {background-color: blue;}
}
/* The element to apply the animation to */
div { width: 100px;
  height: 100px;
  background-color: red;
  animation-name: example;
  animation-duration: 4s;
}
```

το παραπάνω τμήμα Css κώδικα χωρίζεται σε δύο μέρη:

- Το πρώτο μέρος κατασκευάζει το animation μέσω του κανόνα @keyframes (@keyframes example) ο οποίος ορίζει ότι το χρώμα υποβάθρου του <div> με το οποίο συνδέεται το animation θα ξεκινάει με κόκκινο (from {background-color: red;}) και θα καταλήγει σε μπλέ (to {background-color: blue;}). Όπως παρατηρούμε ο κανόνας έχει το όνομα «example» και αυτό γιατί έτσι έχει ονομαστεί και καλείται το animation από το στοιχείο με το οποίο συνδέεται το <div>.
- Το δεύτερο μέρος μορφοποιεί το div το οποίο περιλαμβάνει το animation και του δίνει το όνομα «example» μέσω του attribute (animation-name: example;). μέσα σε αυτόν τον κώδικα ορίζουμε το πλάτος και το ύψος του div (width: 100px; height: 100px;) καθώς επίσης και το αρχικό χρώμα του υποβάθρου του πριν ξεκινήσει το animation (background-color: red;), ενώ παράλληλα μέσω του attribute (animation-duration: 4s;) ορίζεται η ταχύτητα με την οποία θα εναλλάσσονται τα χρώματα που έχουν οριστεί στο animation.

Μπορούμε όμως στο παραπάνω παράδειγμα να ορίσουμε την εναλλαγή περισσότερων χρωμάτων στο animation. Για να γίνει αυτό αντί να γίνει χρήση των ιδιοτήτων from και to στον κανόνα θα χρησιμοποιήσουμε ποσοστά όπως φαίνεται παρακάτω:

```
@keyframes example {
  0%   {background-color: red;}
  25%  {background-color: yellow;}
  50%  {background-color: blue;}
  100% {background-color: green;}
}
```

Ενώ παράλληλα μπορούμε μέσα από τον κανόνα να μετακινούμε το div ταυτόχρονα με την εναλλαγή των χρωμάτων του background του.

Στο παράδειγμα θα μετακινήσουμε το div διαγράφοντας μια νοητή τετραγωνική κίνηση ξεκινώντας από την πάνω αριστερή γωνία της σελίδας και περνώντας από τις τέσσερις γωνίες ενός νοητού τετραγώνου καταλήγοντας στην αρχική γωνία:

```
@keyframes example {
  0%   {background-color: red; left:0px; top:0px;}
  25%  {background-color: yellow; left:200px; top:0px;}
  50%  {background-color: blue; left:200px; top:200px;}
  75%  {background-color: green; left:0px; top:200px;}
  100% {background-color: red; left:0px; top:0px;}
}
```

Επιπλέον μπορούμε να ορίσουμε:

- την καθυστέρηση που θα έχει μέχρι να ξεκινήσει το animation με το attribute: `animation-delay: 2s;` που μετρείται σε δευτερόλεπτα
- πόσες φορές θα «παίξει» ένα animation: `animation-iteration-count: 3;` επίσης μπορούμε να χρησιμοποιήσουμε την τιμή `infinite` που σημαίνει ότι το attribute θα παίζει επ' άπειρον
- σε ποιά κατεύθυνση θα τρέξει το animation. Στο παράδειγμα μας μπορεί να θέλουμε να ορίσουμε την κίνηση του div να γίνει ανάποδα για να το πετύχουμε αυτό χρησιμοποιούμε το attribute: `animation-direction: reverse;` που μπορεί να πάρει και την τιμή `alternate` η οποία αλλάζει την κατεύθυνση κίνησης από την πλευρά στην άλλη.

- Την ταχύτητα και το στυλ της κίνησης μέσα από τα παρακάτω attributes:

`div {animation-timing-function: linear;}` το στυλ της linear κίνησης ορίζει την κίνηση με την ίδια ταχύτητα από την αρχή ως το τέλος της μετακίνησης.

`div {animation-timing-function: ease;}` το στυλ της ease κίνησης ορίζει την κίνηση που ξεκινάει με χαμηλή ταχύτητα αυξάνει στην μέση και στο τέλος φτάνει στον προορισμό του με χαμηλή πάλι ταχύτητα.

`div {animation-timing-function: ease-in;}` το στυλ της ease – in κίνησης δεν διαφέρει και πολύ από την απλή ease. Απλά ορίζεται με μια χαμηλή ταχύτητα στην αρχή της κίνησης.

`div {animation-timing-function: ease-out;}` το στυλ της ease – out κίνησης ορίζει χαμηλή ταχύτητα στο τέλος της μετακίνησης.

`div {animation-timing-function: ease-in-out;}` και τέλος το στυλ της ease – in – out κίνησης ορίζει χαμηλή ταχύτητα στην αρχή και στο τέλος της μετακίνησης.

16 Σύνδεση αρχείου Css με Html

Στο τελευταίο κεφάλαιο του οδηγού θα ασχοληθούμε με το σημαντικότερο μέρος του Css... πως να το συνδέσουμε με το αρχείο Html και να μορφοποιείται η σελίδα αποκλειστικά από το Css. Τις μεθόδους σύνδεσης τις έχουμε δει και παραπάνω [\[βλέπε σελίδα 17\]](#) αλλά εδώ θα πούμε δύο μόνο λόγια για μια γενικότερη επισήμανση.

Έχουμε αναφέρει πως ένα αρχείο Css μπορεί ή να είναι ενσωματωμένο μέσα στο κώδικα του αρχείου Html ή να είναι εξωτερικό αρχείο και να το συνδέουμε με την Html μέσω μιας γραμμής κώδικα.

Στην περίπτωση που είναι ενσωματωμένο το αρχείο Css μέσα στον Html κώδικα θεωρείται ότι είναι εσωτερικό αρχείο και γράφεται ο κώδικας του μέσα σε ένα tag `<style>` που και αυτό ανήκει μέσα στο tag `<head>` κάπως έτσι:

```
<head> <style>
body { background-color: linen;}
h1 { color: maroon; margin-left: 40px;}
</style> </head>
```

Αν όμως είναι εξωτερικό αρχείο τότε συνδέεται μέσω της παρακάτω γραμμής κώδικα:

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

Η οποία όπως βλέπουμε γράφεται μέσα στο tag `<head>` και καλεί το αρχείο Css με το attribute `href` το οποίο παίρνει σαν τιμή το όνομα του αρχείου Css.

Μέσα από αυτόν τον οδηγό συμβουλεύουμε τους επίδοξους σχεδιαστές να χρησιμοποιούν κυρίως εξωτερικά αρχεία Css γιατί παρέχουν ευελιξία και αυτονομία αφού ένα εξωτερικό αρχείο Css μπορεί να συνδεθεί με πολλαπλά αρχεία Html όπως επίσης ένα αρχείο Html έχει την δυνατότητα να μορφοποιείται από πολλά και διάφορα αρχεία Css.

17 Παράρτημα – Πίνακες

Πίνακας 1: Τα νέα attributes της Html5 σε σχέση με την Html4







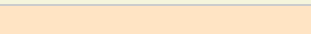


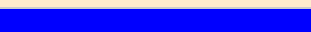


Ετικέτα	Περιγραφή	HTML 4+	HTML 5
<comment>	Ορίζει ένα σχόλιο	✓	✓
<DOCTYPE>	Ορίζει τον τύπο του εγγράφου	✓	✓
<a>	Ορίζει ένα υπέρ-σύνδεσμο	✓	✓
<abbr>	Ορίζει μια συντομογραφία	✓	✓
<acronym>	Ορίζει μια ακρόνυμο	✓	
<address>	Ορίζει ένα στοιχείο διεύθυνσης	✓	✓
<applet>	Ορίζει ένα applet	✓	
<area>	Ορίζει μια περιοχή μέσα σε ένα χάρτη εικόνας	✓	✓
<article>	Ορίζει ένα άρθρο		✓
<aside>	Ορίζει μια περιοχή δίπλα από το περιεχόμενο της σελίδας		✓
<audio>	Ορίζει μια περιοχή ήχου		✓
	Ορίζει ένα έντονα γραμμένο κείμενο	✓	✓
<base>	Ορίζει το βασικό URL για όλα τα URL της σελίδας	✓	✓
<basefont>	Είναι απαγορευμένη. Στη θέση της χρησιμοποιήστε CSS	✓	
<bdo>	Ορίζει την κατεύθυνση του κειμένου που εμφανίζει	✓	✓
<big>	Ορίζει μεγάλο κείμενο	✓	
<blockquote>	Ορίζει μια μεγάλη περιοχή εισαγωγικών	✓	✓
<body>	Ορίζει το στοιχείο κορμού	✓	✓
 	Εισάγει μια αλλαγή γραμμής	✓	✓
<button>	Εισάγει ένα κουμπί	✓	✓
<canvas>	Ορίζει μια περιοχή για γραφικά		✓
<caption>	Ορίζει μια επικεφαλίδα σε πίνακες	✓	✓
<center>	Ορίζει ένα κείμενο στοιχισμένο στο κέντρο	✓	
<cite>	Ορίζει μια παραπομπή	✓	✓
<code>	Ορίζει μια περιοχή που προβάλλει κώδικα υπολογιστών	✓	✓
<col>	Ορίζει ιδιότητες για στήλες σε ένα πίνακα	✓	✓
<colgroup>	Ορίζει ομάδες από στήλες σε ένα πίνακα	✓	✓
<command>	Ορίζει ένα κουμπί εντολής		✓
<datagrid>	Ορίζει δεδομένα σε λίστα δέντρου		✓

Ετικέτα	Περιγραφή	HTML 4+	HTML 5
<datalist>	Ορίζει μια αναδυόμενη λίστα δεδομένων		✓
<datatemplate >	Ορίζει πρότυπα δεδομένων		✓
<dd>	Ορίζει την περιγραφή ενός ορισμού	✓	✓
	Ορίζει ένα διαγραμμένο κείμενο	✓	✓
<details>	Ορίζει της λεπτομέρειες ενός στοιχείου		✓
<dialog>	Ορίζει ένα διάλογο		✓
<dir>	Ορίζει μια λίστα καταλόγου	✓	
<div>	Ορίζει ένα τμήμα σε ένα έγγραφο	✓	✓
<dfn>	Ορίζει ένα ορισμό για ένα όρο	✓	✓
<dl>	Ορίζει μια λίστα ορισμών	✓	✓
<dt>	Ορίζει ένα όρο ορισμού	✓	✓
	Ορίζει ένα κείμενο με έμφαση	✓	✓
<embed>	Ορίζει ένα εξωτερικό διαδραστικό στοιχείο ή κάποιο plugin		✓
<eventsources>	Ορίζει μια περιοχή «στόχο» για γεγονότα που αποστέλλονται από το web server		✓
<fieldset>	Ορίζει μια περιοχή πεδίων φόρμας	✓	✓
<figure>	Ορίζει μια ομάδα πολυμέσων και τους τίτλους τους		✓
	Ορίζει το όνομα το χρώμα και το μέγεθος της γραμματοσειράς	✓	
<footer>	Ορίζει το υποσέλιδο ενός τμήματος ή μιας σελίδας		✓
<form>	Ορίζει μια φόρμα	✓	✓
<frame>	Ορίζει ένα υπό-παράθυρο	✓	
<frameset>	Ορίζει ένα σύνολο από παράθυρα	✓	
<h1> – <h6>	Ορίζει επικεφαλίδες τάξης από 1 μέχρι 6	✓	✓
<head>	Ορίζει πληροφορίες σχετικά με το έγγραφο	✓	✓
<header>	Ορίζει την επικεφαλίδα ενός τμήματος ή της σελίδας		✓
<hr>	Ορίζει μια οριζόντια γραμμή	✓	✓
<html>	Ορίζει ένα έγγραφο html	✓	✓
<i>	Ορίζει πλάγια γράμματα	✓	✓
<iframe>	Ορίζει ένα υπό-παράθυρο	✓	✓
	Ορίζει μια εικόνα	✓	✓
<input>	Ορίζει ένα πεδίο εισαγωγής	✓	✓
<ins>	Ορίζει ένα κείμενο που προστέθηκε αργότερα	✓	✓
<isindex>	Ορίζει ένα πεδίο εισαγωγής μιας γραμμής	✓	
<kbd>	Ορίζει κείμενο γραμμένο με το πληκτρολόγιο	✓	✓

Ετικέτα	Περιγραφή	HTML 4+	HTML 5
<label>	Ορίζει μια ετικέτα για ένα στοιχείο εισαγωγής φόρμας	✓	✓
<legend>	Ορίζει τον τίτλο ενός συνόλου πεδίων	✓	✓
	Ορίζει ένα στοιχείο λίστας	✓	✓
<link>	Ορίζει την αναφορά σε μια πηγή	✓	✓
<mark>	Ορίζει ένα κείμενο με επισήμανση		✓
<map>	Ορίζει ένα χάρτη εικόνας	✓	✓
<menu>	Ορίζει μια λίστα μενού	✓	✓
<meta>	Ορίζει μετά πληροφορίες για το έγγραφο	✓	✓
<meter>	Ορίζει τη μέτρηση σε ένα προκαθορισμένο εύρος τιμών		✓
<nav>	Ορίζει συνδέσμους πλοήγησης		✓
<nest>	Ορίζει το σημείο ενθυλάκωσης σε ένα πρότυπο δεδομένων		✓
<noframes>	Ορίζει ένα τμήμα που δεν έχει υπό-παράθυρα	✓	
<noscript>	Ορίζει μια περιοχή χωρίς σενάρια εντολών	✓	✓
<object>	Ορίζει ένα ενσωματωμένο αντικείμενο	✓	✓
	Ορίζει μια ταξινομημένη λίστα	✓	✓
<optgroup>	Ορίζει ένα σύνολο επιλογών	✓	✓
<option>	Ορίζει μια επιλογή σε μια αναδιδόμενη λίστα	✓	✓
<output>	Ορίζει ορισμένους τύπους εξόδου		✓
<p>	Ορίζει μια παράγραφο	✓	✓
<param>	Ορίζει μια παράμετρο ενός αντικειμένου	✓	✓
<pre>	Ορίζει ένα κείμενο που είναι μορφοποιημένο εξ αρχής	✓	✓
<progress>	Ορίζει την πρόοδο κάθε είδους εργασίας		✓
<q>	Ορίζει σύντομα εισαγωγικά	✓	✓
<rule>	Ορίζει τους κανόνες αναβάθμισης ενός προτύπου		✓
<s>	Ορίζει ένα κείμενο με μια γραμμή στη μέση	✓	
<samp>	Ορίζει δείγμα κώδικα υπολογιστή	✓	✓
<script>	Ορίζει ένα σενάριο εντολών	✓	✓
<section>	Ορίζει ένα τμήμα		✓
<select>	Ορίζει μια λίστα επιλογών	✓	✓
<small>	Ορίζει ένα κείμενο με μικρότερη γραμματοσειρά	✓	✓
<source>	Ορίζει πηγές πολυμέσων		✓
	Ορίζει ένα τμήμα του εγγράφου	✓	✓
<strike>	Ορίζει ένα κείμενο με μια γραμμή στη μέση	✓	
	Ορίζει ένα έντονα γραμμένο κείμενο	✓	✓

Ετικέτα	Περιγραφή	HTML 4+	HTML 5
<style>	Ορίζει ένα ορισμό μορφοποίησης	✓	✓
<sub>	Ορίζει ένα κείμενο βάσης	✓	✓
<sup>	Ορίζει ένα κείμενο εκθέτη	✓	✓
<table>	Ορίζει ένα πίνακα	✓	✓
<tbody>	Ορίζει τον κορμό ενός πίνακα	✓	✓
<td>	Ορίζει ένα κελί του πίνακα	✓	✓
<textarea>	Ορίζει μια περιοχή κειμένου	✓	✓
<tfoot>	Ορίζει το υποσέλιδο ενός πίνακα	✓	✓
<th>	Ορίζει την επικεφαλίδα μια στήλης του πίνακα	✓	✓
<thead>	Ορίζει την επικεφαλίδα ενός πίνακα	✓	✓
<time>	Ορίζει μια ώρα/ημερομηνία		✓
<title>	Ορίζει τον τίτλο του εγγράφου	✓	✓
<tr>	Ορίζει μια γραμμή σε ένα πίνακα	✓	✓
<tt>	Ορίζει κείμενο τηλετύπου	✓	
<u>	Ορίζει ένα υπογραμμισμένο κείμενο	✓	
	Ορίζει μια αταξινόμητη λίστα	✓	✓
<var>	Ορίζει μια μεταβλητή	✓	✓
<video>	Ορίζει ένα video		✓
<xmp>	Ορίζει ένα κείμενο μορφοποιημένο εξ ορισμού	✓	

Πίνακας 2: Χρωματολόγιο

CSS Colour Name	HTML HEX Code	Sample Colour Box	Sample Word
AliceBlue	#F0F8FF		AliceBlue
AntiqueWhite	#FAEBD7		AntiqueWhite
Aqua	#00FFFF		Aqua
Aquamarine	#7FFFD4		Aquamarine
Azure	#F0FFFF		Azure
Beige	#F5F5DC		Beige
Bisque	#FFE4C4		Bisque
Black	#000000		Black
BlanchedAlmond	#FFEBCD		BlanchedAlmond
Blue	#0000FF		Blue
BlueViolet	#8A2BE2		BlueViolet
Brown	#A52A2A		Brown

BurlyWood	#DEB887		BurlyWood
CadetBlue	#5F9EA0		CadetBlue
Chartreuse	#7FFF00		Chartreuse
Chocolate	#D2691E		Chocolate
Coral	#FF7F50		Coral
CornflowerBlue	#6495ED		CornflowerBlue
Cornsilk	#FFF8DC		Cornsilk
Crimson	#DC143C		Crimson
Cyan	#00FFFF		Cyan
DarkBlue	#00008B		DarkBlue
DarkCyan	#008B8B		DarkCyan
DarkGoldenRod	#B8860B		DarkGoldenRod
DarkGray	#A9A9A9		DarkGray
DarkGreen	#006400		DarkGreen
DarkKhaki	#BDB76B		DarkKhaki
DarkMagenta	#8B008B		DarkMagenta
DarkOliveGreen	#556B2F		DarkOliveGreen
Darkorange	#FF8C00		Darkorange
DarkOrchid	#9932CC		DarkOrchid
DarkRed	#8B0000		DarkRed
DarkSalmon	#E9967A		DarkSalmon
DarkSeaGreen	#8FBC8F		DarkSeaGreen
DarkSlateBlue	#483D8B		DarkSlateBlue
DarkSlateGray	#2F4F4F		DarkSlateGray
DarkTurquoise	#00CED1		DarkTurquoise
DarkViolet	#9400D3		DarkViolet
DeepPink	#FF1493		DeepPink
DeepSkyBlue	#00BFFF		DeepSkyBlue
DimGray	#696969		DimGray
DodgerBlue	#1E90FF		DodgerBlue
FireBrick	#B22222		FireBrick
FloralWhite	#FFFAF0		FloralWhite
ForestGreen	#228B22		ForestGreen
Fuchsia	#FF00FF		Fuchsia
Gainsboro	#DCDCDC		Gainsboro
GhostWhite	#F8F8FF		GhostWhite
Gold	#FFD700		Gold
GoldenRod	#DAA520		GoldenRod
Gray	#808080		Gray
Green	#008000		Green
GreenYellow	#ADFF2F		GreenYellow
HoneyDew	#F0FFF0		HoneyDew
HotPink	#FF69B4		HotPink
IndianRed	#CD5C5C		IndianRed
Indigo	#4B0082		Indigo

Ivory	#FFFFFF0		Ivory
Khaki	#F0E68C		Khaki
Lavender	#E6E6FA		Lavender
LavenderBlush	#FFF0F5		LavenderBlush
LawnGreen	#7CFC00		LawnGreen
LemonChiffon	#FFFACD		LemonChiffon
LightBlue	#ADD8E6		LightBlue
LightCoral	#F08080		LightCoral
LightCyan	#E0FFFF		LightCyan
LightGoldenRodYellow	#FAFAD2		LightGoldenRodYellow
LightGrey	#D3D3D3		LightGrey
LightGreen	#90EE90		LightGreen
LightPink	#FFB6C1		LightPink
LightSalmon	#FFA07A		LightSalmon
LightSeaGreen	#20B2AA		LightSeaGreen
LightSkyBlue	#87CEFA		LightSkyBlue
LightSlateGray	#778899		LightSlateGray
LightSteelBlue	#B0C4DE		LightSteelBlue
LightYellow	#FFFFE0		LightYellow
Lime	#00FF00		Lime
LimeGreen	#32CD32		LimeGreen
Linen	#FAF0E6		Linen
Magenta	#FF00FF		Magenta
Maroon	#800000		Maroon
MediumAquaMarine	#66CDAA		MediumAquaMarine
MediumBlue	#0000CD		MediumBlue
MediumOrchid	#BA55D3		MediumOrchid
MediumPurple	#9370D8		MediumPurple
MediumSeaGreen	#3CB371		MediumSeaGreen
MediumSlateBlue	#7B68EE		MediumSlateBlue
MediumSpringGreen	#00FA9A		MediumSpringGreen
MediumTurquoise	#48D1CC		MediumTurquoise
MediumVioletRed	#C71585		MediumVioletRed
MidnightBlue	#191970		MidnightBlue
MintCream	#F5FFFA		MintCream
MistyRose	#FFE4E1		MistyRose
Moccasin	#FFE4B5		Moccasin
NavajoWhite	#FFDEAD		NavajoWhite
Navy	#000080		Navy
OldLace	#FDF5E6		OldLace
Olive	#808000		Olive
OliveDrab	#6B8E23		OliveDrab
Orange	#FFA500		Orange
OrangeRed	#FF4500		OrangeRed
Orchid	#DA70D6		Orchid

PaleGoldenRod	#EEE8AA		PaleGoldenRod
PaleGreen	#98FB98		PaleGreen
PaleTurquoise	#AFEEEE		PaleTurquoise
PaleVioletRed	#D87093		PaleVioletRed
PapayaWhip	#FFefd5		PapayaWhip
PeachPuff	#FFDAB9		PeachPuff
Peru	#CD853F		Peru
Pink	#FFC0CB		Pink
Plum	#DDA0DD		Plum
PowderBlue	#B0E0E6		PowderBlue
Purple	#800080		Purple
Red	#FF0000		Red
RosyBrown	#BC8F8F		RosyBrown
RoyalBlue	#4169E1		RoyalBlue
SaddleBrown	#8B4513		SaddleBrown
Salmon	#FA8072		Salmon
SandyBrown	#F4A460		SandyBrown
SeaGreen	#2E8B57		SeaGreen
SeaShell	#FFF5EE		SeaShell
Sienna	#A0522D		Sienna
Silver	#C0C0C0		Silver
SkyBlue	#87CEEB		SkyBlue
SlateBlue	#6A5ACD		SlateBlue
SlateGray	#708090		SlateGray
Snow	#FFFAFA		Snow
SpringGreen	#00FF7F		SpringGreen
SteelBlue	#4682B4		SteelBlue
Tan	#D2B48C		Tan
Teal	#008080		Teal
Thistle	#D8BFD8		Thistle
Tomato	#FF6347		Tomato
Turquoise	#40E0D0		Turquoise
Violet	#EE82EE		Violet
Wheat	#F5DEB3		Wheat
White	#FFFFFF		White
WhiteSmoke	#F5F5F5		WhiteSmoke
Yellow	#FFFF00		Yellow
YellowGreen	#9ACD32		YellowGreen

Επίλογος

Μέσα από την πτυχιακή αυτή εργασία προέκυψε η δημιουργία της ηλεκτρονικής της μορφής δηλαδή δημιουργήθηκε μια ιστοσελίδα που σκοπό της έχει την εξ' αποστάσεως εκμάθηση των χρηστών της πάνω στην τέχνη του προγραμματισμού της γλώσσας Html .

Η ιστοσελίδα λοιπόν αποτελεί έναν ηλεκτρονικό οδηγό που μέσα από τα άρθρα, τα επιμορφωτικά videos και την ποικιλία ελεύθερου εκπαιδευτικού υλικού, παρέχει την δυνατότητα στον χρήστη να εξοικιωθεί με την Html5 και τα διαδοχικά φύλλα στυλ Css3.

Το θέμα της πτυχιακής μου εργασίας έδωσε και σε μένα την δυνατότητα να μάθω καινούργια και χρήσιμα εργαλεία που έχουν έρθει στο προσκήνιο για την καλύτερη οπτικοποίηση και σχεδίαση μιας ιστοσελίδας. Παράλληλα λοιπόν με την κατασκευή της σελίδας κατάφερα να εμπλουτίσω τις γνώσεις μου, να μάθω καινούργια πράγματα σε βαθμό που να μπορώ εύκολα να πραγματοποιήσω αλλαγές σε ένα έτοιμο κώδικα Html ή ακόμα και να κατασκευάσω από την αρχή κάποια δική μου ιστοσελίδα που δεν θα έχει σε τίποτα να «ζηλέψει» τις καλύτερες σχεδιαστικά ιστοσελίδες του διαδικτύου.

Τελειώνοντας κύριος σκοπός μου μέσα από αυτή την πτυχιακή εργασία είναι ακριβώς η δημιουργία του καλύτερου τρόπου εκμάθησης της Html5 & Css3. Επιθυμώ οι χρήστες της του οδηγού αυτού να μπορέσουν να βρουν τις κατάλληλες οδηγίες για την σχεδίαση μιας σελίδας καταφέροντας να φτάσουν στο επίπεδο όχι μόνο του προγραμματισμού ενός αρχικού κώδικα Html , αλλά μέσα από τις γνώσεις που θα αποκομίσουν να έχουν την δυνατότητα και του ελέγχου, της διόρθωσης και της μεταβολής ενός ήδη έτοιμου κώδικα. Να γίνουν δηλαδή όχι μόνο οι σχεδιαστές μιας δικής τους σελίδας αλλά και οι ελεγκτές άλλων project. το σημαντικό όμως είναι να αγαπήσουν την Html μέσα από αυτό τον οδηγό και να αποκομίσουν όσο το δυνατόν περισσότερες γνώσεις.

18 Βιβλιογραφία

- [1] <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/article>
- [2] https://developers.google.com/youtube/player_parameters
- [3] https://developer.mozilla.org/en-US/docs/Web/HTML/Block-level_elements
- [4] <http://www.w3schools.com/html/default.asp>
- [5] <https://en.wikipedia.org/wiki/HTML5>
- [6] <http://www.w3.org/TR/html5/>
- [7] <http://html5doctor.com/>
- [8] <https://html5test.com/>
- [9] <http://www.microsoftvirtualacademy.com/training-courses/html5-css3-fundamentals-development-for-absolute-beginners>
- [10] <http://www.htmlgoodies.com/html5/client/learn-html5-in-5-minutes-.html#fbid=DNs1bBltpyP>
- [11] <http://www.html-5-tutorial.com/>
- [12] <http://learnlayout.com/>
- [13] <https://developer.mozilla.org/en-US/Learn/CSS>
- [14] <https://learnable.com/courses/learn-css3-203>
- [15] <http://www.codecademy.com/en/tracks/web>
- [16] <http://webdesign.tutsplus.com/tutorials/the-best-way-to-learn-css--webdesign-11906>
- [17] <http://www.csstutorial.net/>
- [18] <http://www.w3.org/Style/Examples/011/firstcss.en.html>
- [19] <http://html.net/>
- [20] <http://html5dog.com/guides/>
- [21] <http://www.stepbystephtml.com/pages/css.php>
- [22] <http://www.subcide.com/articles/creating-a-css-layout-from-scratch/>
- [23] Σύγγραμμα: Προγραμματισμός Internet & World Wide Web (4^η Έκδοση , Εκδόσεις: Μ. Γκιουρδας).