



ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ
ΚΡΗΤΗΣ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.



ΤΙΤΛΟΣ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ :

ΑΝΑΠΤΥΞΗ ΣΥΣΤΗΜΑΤΟΣ ΠΑΡΟΧΗΣ ΥΠΗΡΕΣΙΩΝ ΔΡΟΜΟΛΟΓΗΣΗΣ-
ΚΑΘΟΔΗΓΗΣΗΣ ΤΩΝ ΧΡΗΣΤΩΝ ΓΙΑ ΤΙΣ ΚΑΘΗΜΕΡΙΝΕΣ ΤΟΥΣ
ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ

ΚΑΡΑΒΑ ΚΩΝΣΤΑΝΤΙΝΑ (ΑΜ:3514)

ΤΥΠΟΥ ΒΑΣΙΛΙΚΗ (ΑΜ:3572)

Επιβλέπων καθηγητής:

Ν. Παπαδάκης - Επίκουρος καθηγητής ΤΕΙ ΚΡΗΤΗΣ

ΗΡΑΚΛΕΙΟ 2016

ΕΥΧΑΡΙΣΤΙΕΣ

Θα θέλαμε να ευχαριστήσουμε θερμά τον επιβλέποντα της πτυχιακής εργασίας μας κ. Παπαδάκη Νικόλαο για την εμπιστοσύνη που μας παρείχε, αναθέτοντάς μας την εκπόνηση του συγκεκριμένου θέματος καθώς επίσης και τις οικογένειες μας για την εμπιστοσύνη, την κατανόηση και την αμέριστη συμπαράστασή τους καθ' όλη τη διάρκεια της συγγραφής της παρούσας πτυχιακής εργασίας.

ΠΕΡΙΛΗΨΗ

Η ραγδαία εξέλιξη της τεχνολογίας τα τελευταία χρόνια έχει επιφέρει σημαντική αλλαγή στην καθημερινή ζωή των ανθρώπων κυρίως του Δυτικού κόσμου. Πολλές από τις καθημερινές δραστηριότητες των ανθρώπων έχουν αλλάξει ριζικά ως προς τον τρόπο οργάνωσής τους... Παλαιότερα, προκειμένου να γίνει ένας κανονισμός ή μια υπενθύμιση μιας δραστηριότητας απαιτούνταν χειρόγραφες σημειώσεις και ημερολόγια με ημερομηνίες , πληροφορίες και τοποθετήσεις περί αυτής. Σήμερα, αυτό μπορεί να συμβεί σε λίγα μόλις δευτερόλεπτα και με πολύ εύκολο και απλό τρόπο. Ένα αρκετά μεγάλο μέρος του πληθυσμού , έρχεται καθημερινά σε επαφή με ένα πλήθος νέων τεχνολογιών που αφορούν κυρίως στον κανονισμό και στη δρομολόγηση διαφόρων καθημερινών δραστηριοτήτων οι οποίες βασίζονται σε δορυφορικούς χάρτες αποστάσεων και συστήματα προσδιορισμού θέσης(G.P.S).

Σκοπός της παρούσας πτυχιακής εργασίας είναι η ανάπτυξη μιας εφαρμογής η οποία χρησιμοποιώντας, μεταξύ άλλων, και τις νέες τεχνολογίες που προαναφέρθηκαν, να δίνει τη δυνατότητα στον χρήστη να οργανώνει καλύτερο, γρηγορότερα και ευκολότερα τις δραστηριότητες του. Συγκεκριμένα η εφαρμογή θα αποτελεί κατά κάποιον τρόπο μια προσωπική ατζέντα-ημερολόγιο στο οποίο ο χρήστης θα μπορεί να εισάγει τα στοιχεία του ώστε να κανονίζει άμεσα την καθημερινότητα του(π.χ ψώνια σε supermarkets , κρατήσεις εισιτηρίων σε cinema, γέμισμα αυτοκινήτου σε πρατήρια καυσίμων), δρομολογώντας τον παράλληλα στα αντίστοιχα κοντινότερα του σε σχέση με την τρέχουσα τοποθεσία που βρίσκεται τη συγκεκριμένη χρονική στιγμή ή που θα βρίσκεται στο μέλλον.

ABSTRACT

The rapid development of technology in recent years has brought about significant change in the daily lives of people especially the Western world. Many of the daily activities of the people have changed radically in terms of their organization are... Previously, in order to make a regulation or a reminder of a required activity handwritten notes and calendars with dates, information and positions on this. Today, this can happen in just a few seconds and very easy and simple way. A sizable part of the population ,comes daily in contact with a host of new technologies mainly in the regulation and launching various daily activities which are based on satellite maps distances and positioning systems (G.P.S).

The aim of this thesis work is the development of an application which is using, among other things, new technologies mentioned above, to enable the user to organize better, faster and easier activities. Specifically, the application will be somehow a personal agenda-calendar where the user can enter the data in order to directly arrange the everyday life (e.g. shopping in supermarkets, tickets to cinema, car filling at gas stations) launching in parallel the corresponding closest in relation to the current location of the particular time or will be in future.

Περιεχόμενα

ΕΥΧΑΡΙΣΤΙΕΣ	2
ΠΕΡΙΛΗΨΗ	3
ABSTRACT	4
1. Παγκόσμιος ιστός (World Wide Web) και Διαδίκτυο(Internet)	8
1.1 Εισαγωγικές Έννοιες.....	8
1.2 Επιστήμη του Web	9
1.3 Οι έννοιες του Web Development και Web Design	10
2.1 Εισαγωγή	13
2.2 World Wide Web Consortium (W3C)	13
2.3 Εξέλιξη της HTML μέχρι σήμερα και η HTML5	13
2.4 Στοιχεία της HTML5	14
2.5. Χρήση της HTML στην εφαρμογή.....	16
2.5.1. Σήμανση <html>	16
2.5.2. Σήμανση <head>.....	16
2.5.3. Σήμανση <title>	16
2.5.4. Σήμανση <body>.....	17
2.5.5. Σήμανση 	18
2.5.6. Σήμανση <table>.....	18
2.5.7. Σήμανση <input>	19
2.5.8. Σήμανση <script>.....	19
2.5.9. Σήμανση <p>.....	19
2.5.10. Σήμανση <center>	20
2.5.11. Σήμανση	20
2.5.12. Σήμανση <a>.....	20
2.5.13. Σήμανση	21
2.5.14. Σήμανση <style>	21
2.5.15. Σήμανση <div>&	21
2.5.16. Σήμανση <form>.....	21
2.5.17. Σημάνσεις <select> και <option>	22
2.5.18. Σήμανση <link>	24
2.5.19. Σήμανση <textarea>	24
2.5.20. Σχόλια στην HTML	24
2.6 Αλλαγές από την HTML4 στην HTML5	25
2.7. Επίλογος	33
3. CSS	34

3.1 Εισαγωγή	34
3.2 Πλεονεκτήματα CSS.....	34
3.3 Σύνταξη των CSS	35
3.4 Ιδιότητες φόντου των CSS	36
3.4.1. Η Ιδιότητα color	36
3.4.2. Η Ιδιότητα background-color	37
3.4.3. Η Ιδιότητα background-image.....	37
3.4.4. Η Ιδιότηταbackground-repeat.....	38
3.4.5. Η Ιδιότηταbackground-attachment.....	39
3.4.6. Η Ιδιότηταbackground-position	40
3.4.7. Η Ιδιότηταbackground-position	42
3.5. Οι γραμματοσειρές(Fonts)	43
3.5.1. Η Ιδιότηταfont-family	43
3.5.2. Η Ιδιότηταfont-style	44
3.5.3. Η Ιδιότηταfont-variant	45
3.5.4. Η Ιδιότηταfont-weight.....	45
3.5.5. Η Ιδιότηταfont-size.....	46
3.5.6. Η Ιδιότηταfont	47
3.6. Ιδιότητες περιθωρίου των CSS	48
3.6.1. Το boxmodel.....	48
3.6.2. Το Margin και το Padding	50
3.6.3. Τα Περιγράμματα(Borders).....	53
3.7. Εφαρμογή των CSS	60
3.7.1.Ενσωματωμένα Στυλ	60
3.7.2.Εξωτερικά Στυλ.....	61
3.7.3.Σύνδεση εξωτερικού στυλ με σελίδα.....	61
3.7.4.Ένθετα στυλ.....	63
3.7.5.Τα πολλαπλά φύλλα στυλ	63
3.7.6.Υποστήριξη των CSS από Browsers	64
3.7.7.Επίλογος	65
4. JavaScript.....	66
4.1. Εισαγωγή	66
4.2. Java Vs JavaScript	66
4.3. Πως εισάγεται η JavaScript	67
4.4. Που εισάγεται η JavaScript.....	67
4.4.1. Script στο τμήμα head.....	68

4.4.2. Script στο τμήμα body	68
4.4.3. Script σε ξεχωριστά αρχεία	69
4.5. Χειριστήρια συμβάντος της JavaScript	69
4.6. Συναρτήσεις της JavaScript	70
4.7. Εντολές της JavaScript	72
4.8. Document Object Model (DOM)	79
4.8.1. Αντικείμενα του DOM, μέθοδοι & ιδιότητες	79
4.8.2. Δημιουργία Αντικειμένου	83
4.8.3. Τροποποίηση Αντικειμένου	83
5. Apache, MySQL & PHP	84
5.1. Εισαγωγή	84
5.2. Apache HTTP Web server	84
5.2.1. Τι είναι ένας Web Server	84
5.2.2. Hypertext Transfer Protocol (HTTP)	85
5.2.3. Apache HTTP Web server	86
5.3. MySQL	86
5.3.1. Είδη βάσεων δεδομένων	86
5.3.2. Χαρακτηριστικά της MySQL	87
5.3.3. Οι βάσεις δεδομένων της MySQL	88
5.3.4. Τύποι δεδομένων	89
5.4 PHP	90
5.4.1. Εξέλιξη της PHP μέχρι σήμερα	90
5.4.2. Ενσωμάτωση της PHP	91
5.4.3. PHPMyAdmin	91
6. ΤΕΧΝΟΛΟΓΙΕΣ ΑΝΑΠΤΥΞΗΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	93
6.1 XAMPP	93
6.2 DREAMWEAVER	94
6.3 NOTEPAD++	95
7. ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΧΡΗΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	96
7.1 Εισαγωγή	96
7.2 Αρχική σελίδα	96
7.3 Εγγραφή στο Calendar	97
7.4 Σύνδεση στο Calendar	97
7.5 Μενού δραστηριοτήτων	98
7.6 Σημειωματάριο	99
BIBΛΙΟΓΡΑΦΙΑ	100

1. Παγκόσμιος ιστός (World Wide Web) και Διαδίκτυο(Internet)

1.1 Εισαγωγικές Έννοιες

Η έννοια του ιστού δημιουργήθηκε το 1989 από τον Βρετανό Tim Berners-Lee , ο οποίος την εποχή εκείνη εργαζόταν στον Ευρωπαϊκό Οργανισμό Πυρηνικών Ερευνών (CERN) στη Γενεύη της Ελβετίας. Το όνομα που έδωσε στην εφεύρεση του ο ίδιος ο Lee ήταν WorldWideWeb , όρος ο οποίος είναι γνωστός στο κοινό ως "www". Το γεγονός που οδήγησε τον Lee στην εφεύρεση του αυτή, ήταν το όραμα του για έναν κόσμο στον οποίο ο καθένας θα μπορούσε να επικοινωνεί και να ανταλλάσσει πληροφορίες και ιδέες ανά τον κόσμο, άμεσα προσβάσιμες σε όλους.

Θα μπορούσαν να αποδοθούν αρκετοί ορισμοί για τον Παγκόσμιο Ιστό ένας εκ των οποίων είναι ο εξής: *« Ο Παγκόσμιος Ιστός είναι μια τεράστια συλλογή από ψηφιακά έγγραφα (τις γνωστές μας ιστοσελίδες) που βρίσκονται αποθηκευμένα σε υπολογιστές του Διαδικτύου »*

Πολλές φορές ταυτίζουμε τον Παγκόσμιο Ιστό με το Διαδίκτυο, έννοιες οι οποίες είναι διαφορετικές και δεν πρέπει να επικρατεί σύγχυση μεταξύ τους. Το Internet και το World Wide Web περιγράφουν τη σχέση του «όλου» και ενός «μέρος» του όλου. Θα μπορούσε κανείς να σκεφτεί το Διαδίκτυο σαν το μεγάλο δοχείο, και το Web σαν να αποτελεί ένα μέρος μέσα σε αυτό το δοχείο.

Συγκεκριμένα, το **Διαδίκτυο** (αγγλ. Internet) είναι παγκόσμιο σύστημα διασυνδεδεμένων δικτύων υπολογιστών, οι οποίοι χρησιμοποιούν καθιερωμένη ομάδα πρωτοκόλλων, η οποία συχνά αποκαλείται "TCP/IP" (αν και αυτή δεν χρησιμοποιείται από όλες τις υπηρεσίες του Διαδικτύου) για να εξυπηρετεί εκατομμύρια χρηστών καθημερινά σε ολόκληρο τον κόσμο. Οι διασυνδεδεμένοι ηλεκτρονικοί υπολογιστές ανά τον κόσμο, οι οποίοι βρίσκονται σε ένα κοινό δίκτυο επικοινωνίας, ανταλλάσσουν μηνύματα (πακέτα) με τη χρήση διαφόρων πρωτοκόλλων (τυποποιημένοι κανόνες επικοινωνίας), τα οποία υλοποιούνται σε επίπεδο υλικού και λογισμικού. Το κοινό αυτό δίκτυο καλείται Διαδίκτυο. Η επικοινωνία μέσω του διαδικτύου καθίσταται άμεση και αμφίδρομη. Δίνεται η δυνατότητα σε κάθε χρήστη ηλεκτρονικού υπολογιστή συνδεδεμένου στο Διαδίκτυο, να πληροφορηθεί αλλά και να πληροφορήσει ανταλλάσσοντας απόψεις μέσω ενός πιο συμμετοχικού και λιγότερο ελεγχόμενου διαύλου επικοινωνίας. Οι

χρήστες αποκτούν ολοένα και περισσότερο την ιδιότητα του παγκοσμίου πολίτη. Υπάρχει έντονη τάση, ήδη από την αρχή της εμφάνισής του διαδικτύου, να θεωρείται ένα άκρως δημοκρατικό μέσο μαζικής επικοινωνίας, το οποίο αποδιαμεσολαβεί την επικοινωνία και καθιστά ισχυρότερο τον μέσο άνθρωπο, καθώς δίνει στον τελευταίο τη δυνατότητα πρόσβασης σε μεγάλο όγκο πληροφοριών συγκεντρωμένων σε ένα "χώρο" και την δυνατότητα της προσωπικής επιλογής των πληροφοριών αυτών. Συνεπώς, η βασική θέση της προσέγγισης αυτής είναι ότι το Διαδίκτυο θα εκδημοκρατίσει την κοινωνία με το να βελτιώσει την επικοινωνία καταργώντας την ανάγκη για διαμεσολάβηση.

1.2 Επιστήμη του Web

Το WSRI (Web Science Research Initiative) ή Πρωτοβουλία Έρευνας Επιστήμης του Ιστού είναι μια κοινή προσπάθεια μεταξύ του Computer Science and Artificial Intelligence Laboratory (CSAIL) στο MIT και του Τμήματος επιστήμης των Υπολογιστών στο Πανεπιστήμιο του Southampton. Η στόχευση του WSRI είναι να οργανώσει και να συστηματοποιήσει τις ερευνητικές μεθόδους για τον μελλοντικό σχεδιασμό και τη χρήση του Παγκόσμιου Ιστού. Το 2009 στην Αθήνα, έλαβε μέρος το 1ο Διεθνές συνέδριο της επιστήμης του Web. Από το 2010 η συγκεκριμένη πρωτοβουλία έχει λάβει την νομική μορφή του φιλανθρωπικού οργανισμού με την ονομασία Web Science Trust Web Science Trust. Έλαβε τη σημαντικότερη επιστημονική αναγνώριση από την αρχαιότερη επιστημονική κοινότητα, τη Βασιλική Ακαδημία (Royal Society). Ειδικότερα, η επιστήμη του Web, αποτέλεσε την τιμώμενη επιστήμη στα πλαίσια των εορτασμών για τα 350 χρόνια από την ίδρυση της Ακαδημίας. Το 2011, κλείνοντας των πρώτο κύκλο δημιουργίας, το Web Science Trust ανακοίνωσε τον επίσημο ταξινομικό κατάλογο της επιστήμης του Web

Σύμφωνα με τον Tim Berners-Lee, η συζήτηση και η ερευνητική προσπάθεια για τον Ιστό θα πρέπει οπωσδήποτε να επικεντρωθεί σε δύο κατηγορίες θεμάτων:

1. *τι πρέπει να παραμείνει σταθερό και*
2. *τι πρέπει να αλλάξει*

Στόχος, επομένως, της επιστήμης του Ιστού είναι αφενός, να συστηματοποιήσει τους τρόπους με τους οποίους οι αποκεντρωμένοι πληροφοριακοί μηχανισμοί (όπως ο Παγκόσμιος Ιστός) μπορούν να εξυπηρετήσουν τις ανάγκες της επιστήμης και της επικοινωνίας και αφετέρου, να διερευνήσει τις αρχές και τις δομές που κυριαρχούν ή πρόκειται να κυριαρχήσουν σε αυτούς τους μηχανισμούς.

Υποστηρίζεται ότι η μελέτη σε βάθος των αποκεντρωμένων πληροφοριακών μηχανισμών θα διαδραματίσει ζωτικό ρόλο στην κατανόηση του τρόπου με τον

οποίο άτυποι και τυχαίοι σύνδεσμοι ανάμεσα σε ανθρώπους, δημόσιους και ιδιωτικούς φορείς, βάσεις δεδομένων και λοιπούς πόρους μπορούν να καλύψουν τις πληροφοριακές ανάγκες σημαντικών δραστηριοτήτων, όπως η ηλεκτρονική διακυβέρνηση, οι ηλεκτρονικές συναλλαγές και η ηλεκτρονική επιστήμη (e-science).

Ένα από τα βασικά ερωτήματα της επιστήμης του Ιστού εντοπίζεται στις μεθόδους μέσα από τις οποίες ένας αποκεντρωμένος πληροφοριακός μηχανισμός είναι αποδοτικός.

Πέρα από τα αμιγώς τεχνολογικά ερωτήματα σχετικά με τα πρωτόκολλα και τις γλώσσες προγραμματισμού στον Ιστό, ανακύπτουν και ερωτήματα κοινωνικής, οικονομικής, ψυχολογικής και πολιτικής φύσης. Για παράδειγμα, τι επιζητούν οι άνθρωποι και οι κοινότητες από τον Ιστό και ποια είναι η online συμπεριφορά, σε ατομικό και συλλογικό επίπεδο, που απαιτείται για να λειτουργήσει εύρυθμα; Ή αντίστοιχα υπάρχει όριο μέγιστης χρησιμότητας της ελευθερίας που έχει δημιουργήσει η αποκεντρωμένη φύση του Ιστού;

Καθώς ο αριθμός των χρηστών αυξάνεται, υπάρχει περίπτωση οι επιλογές ενός χρήστη να καταπατούν και να περιορίζουν τις διαθέσιμες επιλογές για τους υπόλοιπους; Ή μήπως ο φόβος αυτός αποτελεί απλώς μια αβάσιμη και αποτυχημένη προέκταση του πραγματικού κόσμου, ο οποίος χαρακτηρίζεται από αμετάβλητες χωρικές παραμέτρους; Επιπλέον, σε ένα αποκεντρωμένο και συνεχώς εξελισσόμενο Ιστό, όπου δεν υπάρχουν «ιδιοκτήτες», πώς μπορεί να είμαστε βέβαιοι ότι οι αποφάσεις που φαίνονται εύλογες για ένα χρήστη δεν βλάπτουν τα συμφέροντα των άλλων ως σύνολο;

Για την απάντηση των ερωτημάτων αυτών (αλλά και μιας μεγάλης σειράς προβληματισμών που προκύπτει από την πολύπλευρη χρήση του Ιστού) ανακύπτει η ανάγκη συνεργασίας ερευνητών των Μαθηματικών, της Φυσικής, της Βιολογίας, της Κοινωνιολογίας, της Κοινωνικής Ανθρωπολογίας, της Ψυχολογίας, της μελέτης των Μέσων Μαζικής Ενημέρωσης, της Νομικής και της Οικονομίας με μηχανικούς και επιστήμονες της Πληροφορικής έτσι ώστε να διευκολυνθεί η κατανόηση του τεράστιου αυτού αποκεντρωμένου πληροφοριακού συστήματος.

1.3 Οι έννοιες του Web Development και Web Design

Η κατασκευή ιστοσελίδων (WebDevelopment) είναι η διαδικασία με την οποία κανείς μπορεί να οργανώσει δεδομένα και πληροφορίες (συνήθως υπερκειμένου πολυμέσων), οι οποίες προβάλλονται στον τελικό χρήστη του Διαδικτύου, μέσω ενός προγράμματος περιήγησης (browser-π.χ Google Chrome, Windows Internet Explorer, Mozilla Firefox) ή άλλων υπηρεσιών όπως διαδικτυακή τηλεόραση, τα ιστολόγια (blogs) και τα RSS Feeds.

Είναι μια μείξη τεχνικής και δημιουργικής διαδικασίας αλλά πλέον και ψηφιακής εμπορίας (digital marketing) που περιλαμβάνει τη χρήση πολλών ειδικοτήτων από ανθρώπινους πόρους (όπως web designers, web developers, digital marketers) κάθε ένας από τους οποίους, με την εξειδικευμένη γνώση του και την συνεισφορά του σε κώδικα αλλά και σε γνώσεις συνθέτουν στη δημιουργία και την κατασκευή ιστοσελίδων

Οι πιο συνήθεις τεχνολογίες που χρησιμοποιούνται στην ανάπτυξη ιστοσελίδων είναι οι εξής παρακάτω :

- Γλώσσες σελίδας: **HTML** και **XHTML**
- Φύλλα στυλ **CSS**
- Γλώσσες περιγραφής δεδομένων: **XML**, **JSON**
- Γλώσσες προγραμματισμού σεναρίων: **Perl**, **PHP**, **Python**, **Ruby**
- Πλατφόρμες: **ASP** και **ASP.NET** της Microsoft, **Java Enterprise** της Sun, **agile frameworks** όπως το Django και το Ruby on Rails
- Συγγραφή κώδικα στην πλευρά του πελάτη: **Javascript**

Η **ιστοσελίδα** είναι ένα ηλεκτρονικό αρχείο ή ένα σύνολο από ηλεκτρονικά αρχεία που υπάρχουν σε έναν ή και περισσότερους εξυπηρετητές (server/servers) και παρουσιάζει κείμενα και εφαρμογές πολυμέσων στον τελικό-χρήστη(user). Τέτοια αρχεία όπως κείμενο, εικόνες (σύμφωνα με τα πρότυπα SVG, BMP, GIF, JPEG ή PNG) και φόρμες οι οποίες μπορούν να τοποθετηθούν στη σελίδα με τη χρήση γλωσσών σήμανσης υπερκειμένου όπως **HTML/XHTML/XML**. Η αναπαραγωγή πιο σύνθετων πολυμέσων (ανυσματικών γραφικών, βίντεο, ήχων, γραφικών με ενσωματωμένο ήχο και εικόνα) μπορεί να γίνει με πρόσθετα (plug-ins) όπως το Flash, το QuickTime, το περιβάλλον χρόνου εκτέλεσης Java, κ.α ή με αναβαθμισμένες τεχνολογίες όπως οι γλώσσες σήμανσης (X) **HTML5**, **XML** και **MathML**, και το πρότυπο φύλλων στυλ **CSS 3**.

Οι καινούριες εκδόσεις των προγραμμάτων περιήγησης (Internet Explorer 7, 8, Firefox 3.6, safari, chromium κλπ) που ακολουθούν τα W3C πρότυπα οδήγησαν σε μια ευρεία αποδοχή και χρήση των XHTML/XML σε συνδυασμό με τα CSS (Cascading Style Sheets) για την τοποθέτηση και διαχείριση των στοιχείων και αντικειμένων της ιστοσελίδας. Τα τελευταία πρότυπα στοχεύουν στο να αποκτήσουν τα προγράμματα περιήγησης και προσφέρουν μια ευρεία γκάμα επιλογής πολυμέσων και πρόσβασης στους πελάτες χωρίς τη χρήση των προσθέτων (plug-ins).

Γενικότερα οι ιστοσελίδες διαχωρίζονται σε στατικές και δυναμικές:

- Οι στατικές ιστοσελίδες δεν αλλάζουν περιεχόμενο και διάταξη (layout) με οποιοδήποτε αίτημα εκτός και αν ο προγραμματιστής αναβαθμίσει (update) τη σελίδα. Μια απλή **HTML** σελίδα είναι παράδειγμα στατικού περιεχομένου.

- Οι δυναμικές ιστοσελίδες προσαρμόζουν το περιεχόμενο και/ή την εμφάνισή τους σύμφωνα με την καταχώρηση/αλληλεπίδραση ή τις αλλαγές του τελικού χρήστη στο περιβάλλον προγραμματισμού (χρήστης, ώρα, τροποποιήσεις στη βάση δεδομένων, κτλ) Το περιεχόμενο μπορεί να αλλάζει στον υπολογιστή του τελικού-χρήστη με τη χρήση των γλωσσών προγραμματισμού που εκτελούνται στον υπολογιστή του χρήστη (JavaScript, VBScript, Actionscript, κτλ). Το περιεχόμενο στις δυναμικές σελίδες συχνά μεταφράζεται στον εξυπηρετητή (server) μέσω γλωσσών προγραμματισμού που εκτελούνται σε αυτόν (Perl, PHP, ASP, JSP, ColdFusion, .NET κτλ).

Με την συνεχόμενη ειδίκευση στην τεχνολογία της Πληροφορικής δημιουργείται η ανάγκη διαχωρισμού του Σχεδιασμού Ιστοσελίδων από τον Προγραμματισμό Ιστοσελίδων.

Για τη διαδικασία σχεδιασμού μιας ιστοσελίδας, μιας εφαρμογής ή ενός πολυμέσου για το διαδίκτυο μπορεί να συνδυάζονται πολλοί κλάδοι όπως animation, συγγραφή, επικοινωνιακός σχεδιασμός, εταιρική ταυτότητα, σχεδιασμός γραφικών, διεπαφήχρήστη-υπολογιστή, αρχιτεκτονική υπολογιστών, σχεδιασμός αλληλεπίδρασης, marketing, φωτογραφία, βελτιστοποίηση μηχανών αναζήτησης και τυπογραφία.

Εξαιτίας της συνεχούς εξειδίκευσης στους τομείς της επικοινωνιακής Πληροφορικής, δημιουργείται μια σαφής τάση διαχωρισμού του σχεδιασμού στο Διαδίκτυο από την ανάπτυξη του διαδικτύου ως μέσου ροής της πληροφορίας και των αγαθών προς όλες τις διαδικτυακές υπηρεσίες.

Ο σχεδιασμός που αφορά το διαδίκτυο είναι κατά μία έννοια παρόμοια με την κλασσική τυπογραφία. Κάθε ιστοσελίδα είναι ένα σύνολο αναπαραγόμενων πληροφοριών, όπως ένα βιβλίο και κάθε σελίδα της ιστοσελίδας είναι το αντίστοιχο της σελίδας ενός βιβλίου. Ωστόσο στο web design χρησιμοποιείται ένα πλαίσιο εργασίας (framework) βασισμένο σε ένα ψηφιακό κώδικα και σε μια τεχνολογία απεικονίσεων για να δημιουργήσει και να διατηρήσει ένα περιβάλλον που θα διανέμει την πληροφορία σε πολλαπλές μορφές αρχείων (formats) όπως pdf, html, doc, κλπ. Οι δυνατότητες του web design το καθιστούν αδιαμφισβήτητο ως τον πλέον κομψό και σύνθετα αναπτυσσόμενο τρόπο επικοινωνίας στο σημερινό κόσμο

2. HTML5

2.1 Εισαγωγή

Κάθε σελίδα που εμφανίζεται στο Internet είναι ένα αρχείο γραμμένο με τη γλώσσα HTML που περιλαμβάνει το κείμενο της σελίδας, τη δομή της και τους συνδέσμους προς άλλα έγγραφα, εικόνες ή άλλα μέσα. Η HTML είναι ακρωνύμιο των λέξεων HyperText Markup Language (Γλώσσα Χαρακτηρισμού Υπερ-Κειμένου) και ο ορισμός της βασίζεται στην SGML (Standard Generalized Markup Language) που αποτελεί ένα πρότυπο για ορισμό άλλων γλωσσών σήμανσης κειμένου. Η HTML δεν είναι γλώσσα προγραμματισμού αλλά γλώσσα μορφοποίησης (markup language) και έχει τυποποιηθεί από τον παγκόσμιο οργανισμό τυποποίησης τεχνολογιών του Web, τον World Wide Web Consortium (W3C).

2.2 World Wide Web Consortium (W3C)

Η Κοινοπραξία του Παγκόσμιου Ιστού, γνωστή ως World Wide Web Consortium (W3C) έχει ως στόχο να οδηγήσει τον Παγκόσμιο Ιστό στο μέγιστο των δυνατοτήτων του, αναπτύσσοντας πρότυπα τεχνολογιών. Το W3C συστάθηκε το 1994 από τον εφευρέτη του Παγκόσμιου Ιστού, Sir Tim Berners-Lee, στο Πανεπιστήμιο MIT της Μασαχουσέτης των Ηνωμένων Πολιτειών. Το W3C διοικείται από μία κεντρική Ομάδα και έρχεται σε επαφή με τις κατά τόπους κοινότητες διεθνώς, μέσω των Γραφείων του. Στην ανάπτυξη των τεχνολογιών του συνεισφέρουν τα εγγεγραμμένα Μέλη του, καθώς και το ευρύ κοινό. Το όραμα του W3C για τον Ιστό περιλαμβάνει τη συμμετοχή, τη διανομή της γνώσης, και με αυτόν τον τρόπο τη δόμηση της εμπιστοσύνης σε μια παγκόσμια κλίμακα.

2.3 Εξέλιξη της HTML μέχρι σήμερα και η HTML5

Το 1990, ο Tim Berners-Lee από το Cern, το εργαστήριο φυσικής της Γενεύης, δημιούργησε ένα νέο πρωτόκολλο με το οποίο θα μπορούσαν να μεταφέρονται κάθε είδους αρχείων και αντικειμένων μέσα από το Internet. Το πρωτόκολλο αυτό ονομάστηκε HTTP (HyperText Transfer Protocol) και σηματοδότησε την αρχή του WWW όπως το ξέρουμε σήμερα. Οι σελίδες που ήταν η βάση του WWW ήταν γραμμένες στην πρώτη έκδοση της γλώσσας HTML. Μέχρι το 1993 η HTML (τότε στην αρχική της έκδοση) ήταν ένα περιβάλλον δημιουργίας υπερκειμένων (hypertext). Το 1994, αναπτύσσεται το πρότυπο HTML 2.0 από ένα διεθνή οργανισμό IETF (Internet Engineering Task Force) που δίνει νέες δυνατότητες στην HTML. Η επόμενη έκδοση η 3.0 δεν έγινε αποδεκτή από τις Microsoft και Netscape οπότε γρήγορα αντικαταστάθηκε από την έκδοση 3.2 (1996) που θεωρήθηκε σταθμός. Σήμερα η HTML βρίσκεται στην 4η έκδοση της και το W3C δεν θα συνεχίσει να αναπτύσσει την HTML. Η μελλοντική δουλειά του W3C θα επικεντρωθεί στην XHTML.

Η **HTML5** είναι μια υπό ανάπτυξη γλώσσα σήμανσης για τον Παγκόσμιο Ιστό που όταν ετοιμαστεί θα είναι η επόμενη μεγάλη έκδοση της HTML (Γλώσσα Υπερκειμένου, HyperText Markup Language). Η ομάδα Web Hypertext Application Technology Working Group (WHATWG) άρχισε δουλειά σε αυτή την έκδοση τον Ιούνιο του 2004 με το όνομα **Web Applications**. Το Φεβρουάριο του 2010 το πρότυπο ήταν ακόμη σε κατάσταση "Last Call" στο WHATWG. Η HTML5 προορίζεται για αντικατάσταση της HTML 4.01, της XHTML 1.0, και της DOM Level 2 HTML. Ο σκοπός είναι η μείωση της ανάγκης για ιδιόκτητα plug-in και πλούσιες διαδικτυακές εφαρμογές (RIA) όπως το Adobe Flash, το Microsoft Silverlight, το Apache Pivot, και η Sun JavaFX.

Οι ιδέες πίσω από την HTML5 εμφανίστηκαν αρχικά το 2004 από την ομάδα WHATWG. Η HTML5 εμπεριέχει το πρότυπο **Web Forms 2.0** που είναι επίσης της WHATWG.

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

Οι συντάκτες της HTML5 είναι ο Ian Hixson της εταιρίας Google και ο Dave Hyatt της εταιρίας Apple.

2.4 Στοιχεία της HTML5

Η HTML χρησιμοποιεί ειδικές **σημάνσεις** (tags) για να δώσει τις απαραίτητες οδηγίες στον περιηγητή Ιστού (Webbrowser). Οι σημάνσεις είναι εντολές προς τον

browser για τον τρόπο εμφάνισης του κειμένου και για την εμφάνιση πολυμεσικού περιεχομένου (εικόνες, video, ήχος). Ο καθένας μπορεί να δημιουργήσει ένα αρχείο HTML χρησιμοποιώντας απλώς έναν επεξεργαστή κειμένου. Οι οδηγίες είναι caseinsensitive, δηλαδή, δεν επηρεάζονται από το αν έχουν γραφτεί με πεζά (μικρά) ή κεφαλαία και ένα αρχείο HTML πρέπει να έχει κατάληξη htm ή html.

Η βασική δομή ενός HTML αρχείου είναι η παρακάτω:

```
<html> Όλος ο κώδικας περιέχεται στην βασική ετικέτα  
<HTML>  
<head> Εντολές προς τον υπολογιστή. Δεν εμφανίζονται  
στο χρήστη.  
</head>  
<body>  
  
Περιεχόμενο που εμφανίζεται στο χρήστη  
  
</body>  
</html>
```

Σε ένα έγγραφο HTML υπάρχουν αρκετές δεκάδες σημάνσεις οι οποίες είναι γραμμένες απευθείας σε μορφή απλού κειμένου στο HTML έγγραφο όπου μπορούν να διερμηνευτούν από το λογισμικό του υπολογιστή. Όταν ένας browser ανοίγει ένα αρχείο HTML ο browser αναγνωρίζει αυτόν τον τρόπο γραφής και εκτελεί τις εντολές που περιέχονται σε αυτόν. Διαφορετικά προγράμματα φυλλομετρητή μπορεί να μορφοποιούν και να εμφανίζουν το ίδιο αρχείο με διαφορετικό τρόπο, ανάλογα με τις δυνατότητες του συστήματος στο οποίο τρέχουν και τις επιλογές διαμόρφωσης του προγράμματος του φυλλομετρητή. Οι σημάνσεις αυτές καθ' εαυτές δεν παρουσιάζονται στον browser και είναι ξεχωριστές από το περιεχόμενο που περικλείουν. Επίσης, κάθε σήμανση (εκτός λίγων εξαιρέσεων π.χ.
) έχει ένα σημείο που ανοίγει (π.χ. <HTML>) και ένα σημείο που κλείνει (π.χ. </HTML>). Το κλείσιμο της σήμανσης περιέχει ένα / μετά το αρχικό <. Οι σημάνσεις μπορεί να περιέχουν ιδιότητες και κάθε ιδιότητα τιμές.

Η βασική δομή μιας σήμανσης είναι η παρακάτω:

```
< σήμανση [ιδιότητα ="τιμή"]>  
  
    περιεχόμενο  
  
</ σήμανση>
```

Παράδειγμα:

```
<body bgcolor="green" background="picture.jpg">  
    Μια απλή σελίδα  
</body>
```

2.5. Χρήση της HTML στην εφαρμογή

Παρακάτω παρουσιάζονται οι βασικότερες σημάνσεις που χρησιμοποιήθηκαν στην εφαρμογή καθώς και κάποια σχόλια για τον τρόπο εφαρμογής τους.

2.5.1. Σήμανση <html>

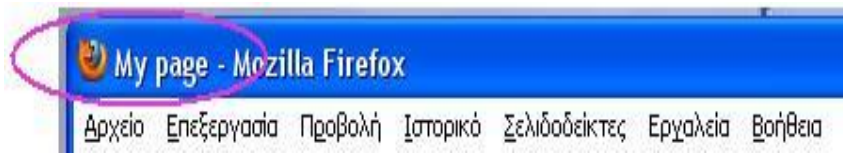
Το στοιχείο αυτό είναι το πρώτο το οποίο χρησιμοποιείται στο έγγραφο και περικλείει όλα τα υπόλοιπα μέσα σε αυτό. Με αυτό τον τρόπο πληροφορούμε το φυλλομετρητή ότι οι γραμμές που περικλείονται μέσα σε αυτές τις δύο σημάνσεις είναι κώδικας γραμμένος σε γλώσσα HTML. Επίσης, είναι γνωστό ως στοιχείο ρίζα (root element).

2.5.2. Σήμανση <head>

Είναι το κομμάτι κώδικα που παραμένει κρυφό στο χρήστη και περιέχει πληροφορίες σχετικά με το HTML έγγραφο. Περιλαμβάνει λέξεις κλειδιά που περιγράφουν την ιστοσελίδα καθώς άλλα και συνδέσμους με αρχεία τα οποία χρησιμοποιούνται από το έγγραφο όπως αρχεία CSS και JavaScript. Επίσης, μέσα στη σήμανση head περιέχεται και ο τίτλος του εγγράφου <title>.

2.5.3. Σήμανση <title>

Ο τίτλος του εγγράφου που εμφανίζεται στη γραμμή τίτλου του φυλλομετρητή. π.χ.
<title>My page</title>



Εικόνα 2-1 : Τίτλος σελίδας

2.5.4. Σήμανση <body>

Μέσα στη σήμανση <body> περιλαμβάνεται όλο το κείμενο και οποιοδήποτε άλλο περιεχόμενο, όπως σύνδεσμοι, εικόνες κ.ά.

2.5.5. Σήμανση

Η σήμανση
 αναγκάζει τον browser να σταματήσει να τυπώνει στην τρέχουσα γραμμή και έτσι το αμέσως επόμενο κείμενο ξεκινά από το αριστερό περιθώριο της επόμενης γραμμής. Το
 δεν προσθέτει επιπλέον χώρο πάνω ή κάτω από τη νέα γραμμή και δεν αλλάζει τη γραμματοσειρά ή το στυλ του κειμένου.

2.5.6. Σήμανση <table>

Το στοιχείο ενός πίνακα ξεκινά με την σήμανση <table> και τελειώνει με τη σήμανση </table>. Χρησιμοποιείται για να οργανώσουμε και να παρουσιάσουμε πληροφορίες σε μορφή πίνακα. Παράλληλα χρησιμοποιείται πάρα πολύ για τη μορφοποίηση των ιστοσελίδων αλλά αυτό έχει περιοριστεί αρκετά καθώς οι περισσότεροι πια κάνουν χρήση των CSS. Μέσα σε ένα πίνακα με τις σημάνσεις <tr></tr> και <td></td> καθορίζουμε την αρχή και το τέλος μιας γραμμής και μιας στήλης του πίνακα αντίστοιχα.

```
<table border="1" style="border-
color:fuchsia"><tr>
    <td> Επικεφαλίδα στήλης 1
</td><td> Επικεφαλίδα στήλης
2</td>
</tr>
<tr>
    <td>
δεδομένα</td><td>
> δεδομένα</td>
</tr>
</table>
```

Το παραπάνω κομμάτι κώδικα παράγει τον παρακάτω πίνακα:

Επικεφαλίδα στήλης 1	Επικεφαλίδα στήλης 2
Δεδομένα	Δεδομένα

Πίνακας 1 : Χρήση της σήμανσης <table>

2.5.7. Σήμανση <input>

Με τη σήμανση <input> ορίζουμε στοιχεία μιας φόρμας όπως textboxes, checkboxes, buttons και radiobuttons. Το είδος του στοιχείου που θα οριστεί στη φόρμα καθορίζεται από την ιδιότητα type, αν δεν υπάρχει η προκαθορισμένη τιμή είναι text.

Μια άλλη σημαντική ιδιότητα της <input> είναι η name με την οποία μπορούμε να κάνουμε αναφορά στο πεδίο. Επίσης, σημαντικές ιδιότητες της <input> είναι οι checked, disabled και maxlength. Με την checked, μπορούμε να δώσουμε αρχική

τιμή για τα checkbox και τα radiobuttons. Η disabled απενεργοποιεί το αντίστοιχο στοιχείο (η τιμή ενός απενεργοποιημένου στοιχείου δεν αποστέλλεται για επεξεργασία). Η ιδιότητα maxlength καθορίζει πόσους χαρακτήρες μπορεί να δεχτεί ένα πεδίο κειμένου. Για παράδειγμα ο παρακάτω κώδικας

```
<input name="text" type="text" size="10" maxlength="30">
```

ορίζει ένα πεδίο κειμένου μεγέθους 10 χαρακτήρων (το μέρος που φαίνεται) που δέχεται το πολύ 30 χαρακτήρες.

2.5.8. Σήμανση <script>

Με τη σήμανση <script> εισάγουμε κάποιο script στην σελίδα το οποίο είτε είναι γραμμένο με JavaScript είτε με κάποια άλλη γλώσσα.

2.5.9. Σήμανση <p>

Με την σήμανση <p> διαχωρίζουμε προτάσεις κειμένου κατά την εμφάνισή τους από τον browser

2.5.10. Σήμανση <center>

Με τη σήμανση <center> μπορούμε να κεντράρουμε το περιεχόμενο της HTML που βρίσκεται μέσα στην σήμανση. Η σήμανση είναι συντόμευση της σήμανσης <div align=center> και λειτουργούν το ίδιο. Η σήμανση <center> τοποθετείται πριν από το κείμενο που θέλουμε να κεντράρουμε και η </center> μετά το τέλος του κειμένου.

```
<center>  
  
<h1> Αυτή η επικεφαλίδα θα είναι στο κέντρο της web σελίδας</h1>  
  
</center>
```

2.5.11. Σήμανση

Με την σήμανση μπορούμε να ρυθμίσουμε την γραμματοσειρά του περιεχομένου της HTML που βρίσκεται μέσα στην σήμανση. Με την ιδιότητα size ορίζουμε το μέγεθος των γραμμάτων. Η μπορεί να πάρει τιμές από 1 μέχρι 7 αλλά και σχετικές τιμές όπως (από -3 έως +4) που καθορίζουν το μέγεθος της σε σχέση με την προκαθορισμένη γραμματοσειρά που έχει οριστεί στον browser. Αν θέλουμε μπορούμε να σώσουμε ξεχωριστά χρώματα σε μεμονωμένες λέξεις ή χαρακτήρες με την ιδιότητα color.

2.5.12. Σήμανση <a>

Για να δημιουργήσουμε ένα σύνδεσμο χρειαζόμαστε δύο πράγματα:

1. Το όνομα του αρχείου στον τοπικό δίσκο ή το URL του αρχείου, για το οποίο θέλουμε να δημιουργήσουμε το σύνδεσμο.
2. Το κείμενο που θα εμφανίζεται στη σελίδα και στο οποίο εάν κάνουμε κλικ θα ακολουθήσουμε τον σύνδεσμο.

Για να δημιουργηθεί ο σύνδεσμος χρησιμοποιείται η σήμανση <a> και έχει την εξής μορφή:

```
<a name="link_1" href="http://www.google.gr" title="link">
```

Οι **ιδιότητες** name, href και title περιγράφουν τον ίδιο σύνδεσμο. Η href (**H**ypertext

Reference, αναφορά υπέρ-κειμένου) ορίζει το όνομα ή το URL του αρχείου στο οποίο δείχνει ο σύνδεσμος και είναι υποχρεωτική. Οι ιδιότητες `name`, `title` μπορούν να παραληφθούν. Το κείμενο ανάμεσα στη σήμανση αρχής `<a>` και τέλους `` είναι υπογραμμισμένο (εκτός και αν έχει καθορισθεί διαφορετικά) και πατώντας πάνω του ο τελικός χρήστης ενεργοποιεί το σύνδεσμο. Εκτός από κείμενο για την ενεργοποίηση του συνδέσμου μπορεί να υπάρχει και κάποιο γραφικό.

2.5.13. Σήμανση ``

Με τη σήμανση `` μπορούμε να προσθέσουμε γραφικά σε ένα HTML αρχείο. Η βασική ιδιότητα της σήμανσης `` είναι η `src` με την οποία καθορίζουμε τη διεύθυνση του αρχείου γραφικών.

2.5.14. Σήμανση `<style>`

Με την ετικέτα `<style>` μπορούμε να δημιουργήσουμε ένα ενσωματωμένο στυλ στο έγγραφο.

2.5.15. Σήμανση `<div>` & ``

Αυτές οι δύο σημάνσεις ομαδοποιούν το περιεχόμενο της HTML με σκοπό να δοθούν κοινά χαρακτηριστικά σε αυτό ή να αλλάξει αργότερα χαρακτηριστικά με την χρήση δυναμικής HTML και CSS.

Η διαφορά των δύο σημάνσεων είναι ότι η `` είναι στοιχείο το οποίο μπορεί να υπάρχει οπουδήποτε στη ροή του κειμένου χωρίς να προκαλεί αλλαγή γραμμής σε αντίθεση με την `<div>` που προκαλεί αλλαγή γραμμής.

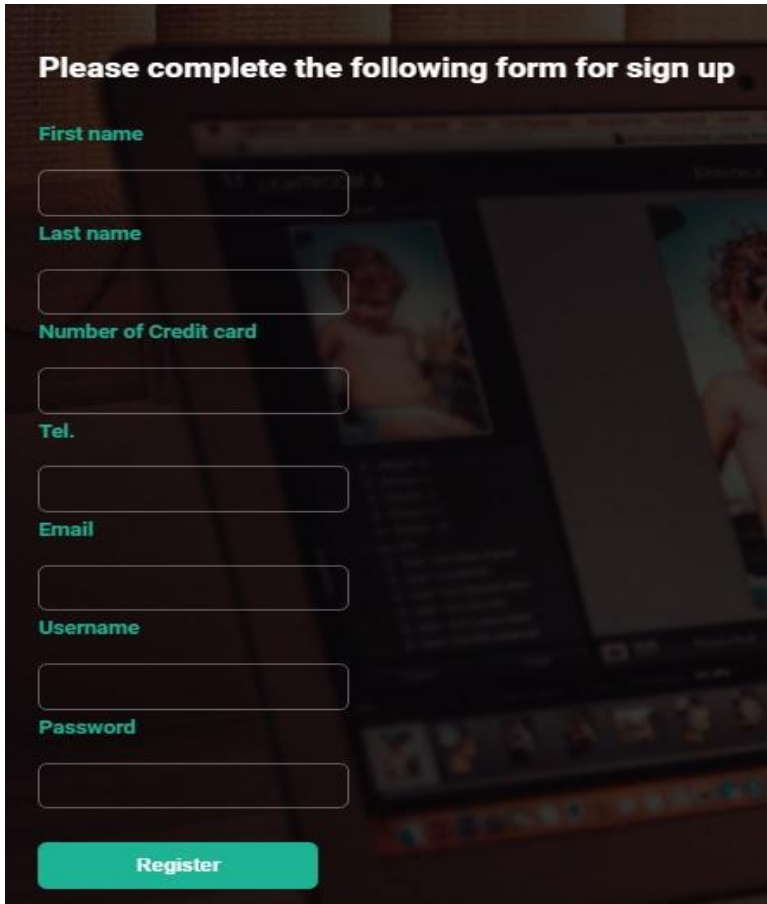
2.5.16. Σήμανση `<form>`

Με την σήμανση `<form>` έχουμε τη δυνατότητα να στείλουμε δεδομένα στον εξυπηρετητή (webserver). Οι φόρμες λειτουργούν ως εξής: ο χρήστης συμπληρώνει με δεδομένα τα πεδία μιας φόρμας και πατώντας το κουμπί υποβολής της φόρμας τα δεδομένα στέλλονται στον webserver. Ο webserver δέχεται τα δεδομένα και συνήθως τα επεξεργάζεται ή τα αποθηκεύει κάπου.

Μια φόρμα έχει δύο βασικές ιδιότητες, τις **method** και **action**.

Με την `action` καθορίζεται η url διεύθυνση του αρχείου που θα κληθεί για να λάβει τα

δεδομένα που θα αποσταλούν, ενώ με την method η οποία μπορεί να πάρει δύο τιμές (post και get) καθορίζεται ο τρόπος με τον οποίο τα δεδομένα στον web browser στέλνονται στον webserver.



Please complete the following form for sign up

First name

Last name

Number of Credit card

Tel.

Email

Username

Password

Register

Εικόνα1 : Φόρμα υποβολής στοιχείων

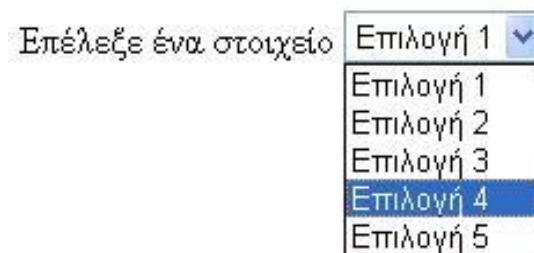
2.5.17. Σημάνσεις <select> και <option>

Με τις σημάνσεις <select> και <option> μπορούμε να δημιουργήσουμε ένα μενού επιλογών από τις οποίες ο χρήστης μπορεί να επιλέξει μία που αντιστοιχεί σε μία τιμή. Μπορούμε να δημιουργήσουμε απλά drop-downlistboxes όπου ο χρήστης επιλέγει μία από τις διαθέσιμες επιλογές, αλλά και multi-selectlistboxes στα οποία ο χρήστης έχει την δυνατότητα να επιλέξει πάνω από μία επιλογή. Οι επιλογές από τις οποίες διαλέγει ο χρήστης βρίσκονται μέσα στην σήμανση <option>.

Στο παρακάτω απόσπασμα κώδικα παρουσιάζεται ένα παράδειγμα χρήσεις των σημάνσεων:

```
<select>  
  
<option>Επιλογή 1 </option>  
  
<option>Επιλογή2</option>  
  
<option>Επιλογή3 </option>  
  
<option>Επιλογή4</option>  
  
<option>Επιλογή 5 </option>  
  
</select>
```

Το αποτέλεσμα του παραπάνω κώδικα είναι:



Εικόνα1: Ένα drop-down list box

2.5.18. Σήμανση <link>

Με τη σήμανση link μπορούμε να συνδέσουμε ένα έγγραφο HTML με εξωτερικά αρχεία. Η πιο κοινή του χρήση είναι να συνδέει εξωτερικά φύλλα στυλ (externalstylesheets) στο έγγραφο.

2.5.19. Σήμανση <textarea>

Με την σήμανση <textarea> δημιουργούμε ένα πεδίο πολλών γραμμών σε περίπτωση που το κείμενο προς εισαγωγή είναι πολύ μεγάλο για να χωρέσει σε ένα πεδίο μίας γραμμής κειμένου. Με τις ιδιότητες cols και rows μπορούμε να καθορίσουμε τις γραμμές και τις στήλες του <textarea>. Οι πιθανές τιμές που μπορεί να πάρει η <textarea> είναι:

Soft (Default)

Το κείμενο εμφανίζεται με αναδίπλωση κειμένου και όταν υποβάλλεται δεν υπάρχουν χαρακτηριστικές αλλαγής γραμμής

Hard

Το κείμενο εμφανίζεται με αναδίπλωση κειμένου και όταν υποβάλλεται υπάρχουν χαρακτηριστικές αλλαγής γραμμής

Off

Το κείμενο δεν εμφανίζεται με αναδίπλωση κειμένου αλλά ακριβώς όπως υποβάλλεται από τον χρήστη. Αν ο χρήστης πατήσει το πλήκτρο Enter μόνο τότε θα γίνει αλλαγή γραμμής

2.5.20. Σχόλια στην HTML

Τα σχόλια χρησιμοποιούνται για να γράφουμε σημειώσεις μέσα στον κώδικα. Δεν εμφανίζονται στην οθόνη του χρήστη. Ένα σχόλιο αρχίζει με το <!-- και τελειώνει με το -->

π.χ. <!-- Αυτό είναι ένα σχόλιο -->

2.6 Αλλαγές από την HTML4 στην HTML5

Στην νέα έκδοση της HTML προστέθηκαν τα παρακάτω στοιχεία:

- Section το στοιχείο αυτό είναι για να ομαδοποιήσει θεματικά ίδιο περιεχόμενο συνή-θως κάτω από μία κεφαλίδα. Παραδείγματα sections θα μπορούσε να είναι το κεφάλαιο ενός βιβλίου ή οι καρτέλες μιας σελίδας.
- Article ορίζει ένα αυτοτελές μέρος του εγγράφου το οποίο θα μπορούσε να ξαναχρη-σιμοποιηθεί μόνο του, όπως ένα άρθρο για παράδειγμα ή ένα widget.
- Aside είναι για να ορίσει ένα τμήμα με μικρότερη σχέση με το υπόλοιπο έγγραφο και διαφορετική ροή, όπως για παράδειγμα μενού.
- Hgroup είναι ομάδες από κεφαλίδες h1-h6 όταν το κείμενο έχει επικεφαλίδες, υποκε-φαλίδες, εναλλακτικούς τίτλους και γενικά χρειάζεται παραπάνω από ένα h στοιχείο.
- Header είναι ένα στοιχείο για την απεικόνιση εισαγωγικού περιεχομένου ή υλικό για την διευκόλυνση της περιήγησης.
- Footer ορίζει το υποσέλιδο σε ένα έγγραφο με πληροφορίες όπως δημιουργό, copy-rights κ.α.
- Nav το στοιχείο που ορίζει το τμήμα του εγγράφου που θα χρησιμοποιηθεί για τα κύ-ρια μενού περιήγησης.
- Figure αναπαριστά ένα αυτοδύναμο κομμάτι κειμένου που αντιμετωπίζετε ως ένα κομμάτι στη ροή του εγγράφου.
- Figcaption είναι για λεζάντες κυρίως μέσα σε figureelements π.χ.

```
<figure>  
  <video src="example.webm"  
  controls</video><figcaption>Example  
  Video</figcaption>  
</figure>
```

- Video&audio δύο νέα στοιχεία για την αναπαραγωγή των αντίστοιχων πολυμέσων, μπορούν να χρησιμοποιηθούν με τα στοιχεία ελέγχου του κάθε browser ή να γράψει ο κάθε ενδιαφερόμενος δικά του με την χρήση των προσφερόμενων APIs. Μπορεί να γίνει χρήση πολλών στοιχείων source μέσα σε αυτά τα στοιχεία αν υπάρχει ο πόρος

παραπάνω από μία φορές για λόγους συμβατότητας καθώς δεν υποστηρίζουν όλοι οι browsers τα ίδια formats πολυμέσων.

- Track επιτρέπει την αναπαραγωγή εξωτερικών χρονισμένων αρχείων κειμένου μαζί με στοιχεία video όπως για παράδειγμα υπότιτλους, από μόνο του σαν στοιχείο δεν μπορεί να σταθεί πρέπει πάντα να είναι μέσα σε ένα videoelement.
- Embed στοιχείο για την ενσωμάτωση περιεχομένου διαδραστικού ή μη για το οποίο συνήθως απαιτείται η χρήση plug-in.
- Mark για επισήμανση κειμένου με χρώμα.
- Progress στοιχείο αναπαριστά γραφικά την πρόοδο για την ολοκλήρωση μιας διεργασίας, είτε όταν η διεργασία απαιτεί άγνωστο χρόνο δείχνει ότι γίνεται πρόοδος, είτε όταν είναι πεπερασμένου χρόνου δείχνει κατά προσέγγιση την πρόοδο που έχει εκτελεστεί.
- Meter αναπαριστά μια αξία βαθμιαία ή κλασματικά.
- Time στοιχείο για την αναπαράσταση ημερομηνιών με τρόπο κατανοητό και από αν-θρώπους και από υπολογιστές.
- Ruby μαζί με τα στοιχεία rt και rp επιτρέπει για φωνητική απεικόνιση χαρακτήρων.
- Bdi αντιπροσωπεύει ένα κομμάτι κειμένου το οποίο πρέπει να διαχωριστεί από το υπόλοιπο κείμενο για την περίπτωση κειμένου με διαφορετικό προσανατολισμό.
- Wbr αναπαριστά την ευκαιρία για αλλαγή γραμμής
- Canvas στοιχείο για την απεικόνιση γραφικών bitmap με την χρήση scripts, μπορεί να χρησιμοποιηθεί για γραφήματα, παιχνίδια κ.α.
- Com και είναι μια εντολή την οποία μπορεί να καλέσει ο χρήστης.
- Details το στοιχείο παραθέτει στην διάθεση του χρήστη ένα widget, για την άντληση περαιτέρω πληροφοριών.
- Datalist μαζί με το καινούργιο στοιχείο inputlist μπορούν να χρησιμοποιηθούν για την δημιουργία comboboxes.
- Keygen στοιχείο για την έκδοση ιδιωτικού-δημοσίου κλειδιού, μπορεί να χρησιμοποιηθεί για πιστοποιήσεις.
- Output το στοιχείο αυτό θα περιέχει το αποτέλεσμα υπολογισμού.
- Το στοιχείο input έχει μερικά νέα attributes με στόχο την δημιουργία καινούργιων interfaces από τους webagents, όπως color και datepickers, sliders κ.α. Καθώς και καλύτερα διαμορφωμένα στοιχεία εισόδου για καλύτερο τοπικό έλεγχο της φόρμας πριν σταλεί αυτή στον server και κατά συνέπεια καλύτερη εμπειρία του χρήστη με

λιγότερη αναμονή σε περίπτωση λάθους σε κάποιο πεδίο εισόδου.

οΤείνεται ένα πεδίο κειμένου για την εισαγωγή τηλεφωνικών αριθμών.
οSearchείνεται ένα πεδίο κειμένου για τους όρους αναζήτησης οι διαφορές είναι κυρίως στιλιστικές.

οurlπεδίο κειμένου για την εισαγωγή ενόςurl.

οemailπεδίο κειμένου για την εισαγωγή ενόςemailοι κανόνες για την μορφή του πρέπει να έχει ένα email ισχύουν. οDatetimelocalστοιχείο ελέγχου για την επιλογή ημερομηνίας και ώρας σε παγκόσμια μορφή.

οDatelocalστοιχείο ελέγχου για την επιλογή ημερομηνίας.οMonthστοιχείο ελέγχου για την επιλογή μήνα.

οWeekστοιχείο ελέγχου για την επιλογή βδομάδας.
οTimeστοιχείο ελέγχου για την επιλογή ώρας.

- ο Datetime-local στοιχείο ελέγχου για την επιλογή ημερομηνίας και ώρας σε τοπικό επίπεδο.

- ο Number πεδίο κειμένου για την εισαγωγή ενός αριθμού. Υπάρχουν περιορισμοί για τον αριθμό και κανόνες για αριθμούς κινητής υποδιαστολής.

- οRangeδημιουργείsliderμε περιορισμούς για το βήμα της τιμής.οColorδημιουργεί στοιχείο ελέγχου για την επιλογή χρώματος.

Στην HTML5 δεν προστέθηκαν μόνο καινούργια στοιχεία αλλά πρόσθεσαν και attributes σε στοιχεία που προϋπήρχαν και στην HTML4.

- Στα στοιχεία area και a προστέθηκαν οι ιδιότητες media και επιπλέον στο area προστέθηκαν οι ιδιότητες type, hreflang και rel για λόγους ομοιομορφίας με το στοιχείο link.

- Στο στοιχείο base προστέθηκε η ιδιότητα target για λόγους συνοχής με το στοιχείο a.

- Το στοιχείο meta έχει την ιδιότητα charset για τον προσδιορισμό της κωδικοποίησης του εγγράφου. Η ιδιότητα αυτή είχε υλοποιηθεί και είναι ευρέως διαδεδομένη αλλά τώρα μπήκε στα standards.

- Η καινούργια ιδιότητα autofocus για το inputelement είναι ένας τρόπος για την επιλογή του στοιχείου της φόρμας που θέλουμε κατά την φόρτωση της ιστοσελίδας, προστέθηκε για την βελτίωση της εμπειρίας του χρήστη.

- Η ιδιότητα `placeholder` για τα στοιχεία `input` και `textarea` δίνει στον χρήστη ένα παρά-δειγμα ή μια εξήγηση για την μορφή που πρέπει να έχει η είσοδος π.χ. για ένα στοιχείο `email` θα είναι `<input type="email" placeholder="e.g. mymail@provider.com">`.
- Η ιδιότητα `form` για τα στοιχεία `input`, `output`, `select`, `textarea`, `button`, `label`, `object`, `fieldset` επιτρέπει στα στοιχεία αυτά να συσχετιστούν με μια φόρμα χωρίς την ανάγκη να είναι απόγονοι του `formelement` και μπορούν να τοποθετηθούν οπουδήποτε στη σελίδα.
- Η ιδιότητα `required` για τα στοιχεία `input`, `select` και `textarea` απαιτεί από τον χρήστη να συμπληρώσει το πεδίο για να καταθέσει την φόρμα.
 - Στο στοιχείο `fieldset` μπορεί να γίνει χρήση της ιδιότητας `disabled` η οποία απενεργοποιεί όλους τους απογόνους του `fieldset` και προστέθηκε η ιδιότητα `name` για την χρήση της σε `scripts`.
- Το στοιχείο `input` έχει μια σειρά καινούργιων `attributes` για των ορισμό ορίων, `autocomplete`, `min`, `max`, `multiple`, `pattern`, `step` και `list` το οποίο χρησιμοποιείται σε συνδυασμό με το στοιχείο `datalist`.
- Το στοιχείο `textarea` έχει δυο καινούργιες ιδιότητες `maxlength` και `wrap` τα οποία ελέγχουν το μήκος του κειμένου και την αναδίπλωση κειμένου αντίστοιχα.
- Στο στοιχείο `form` προστέθηκε η ιδιότητα `novalidate` για να μην περνάει η φόρμα ε-πικύρωση κατά την αποστολή της.
- Τα στοιχεία `button` και `input` έχουν τις ιδιότητες `formaction`, `formenctype`, `formmethod`, `formnovalidate` και `formtarget`, αν υπάρχουν παρακάμπτουν τα `action`, `enctype`, `method`, `novalidate` και `target` του στοιχείου `form` στο οποίο ανήκουν.
- Οι ιδιότητες `type` και `label` προστέθηκαν στο στοιχείο `menu` επιτρέπουν το στοιχείο να εμφανίζεται με τη μορφή των συνηθέστερων `UserInterface`.
 - Στο στοιχείο `style` η ιδιότητα `scoped` επιτρέπει την δημιουργία τοπικών κανόνων μορ-φοποίησης.
 - Η ιδιότητα `async` στο στοιχείο `script` επηρεάζει την φόρτωση και την εκτέλεση του `script`.
 - Το στοιχείο `html` έχει την καινούργια ιδιότητα `manifest` για να δείχνει στο `applicationcache` το οποίο χρησιμοποιείτε σε συνδυασμό με το `API` για `offlineWebApplications`.
 - Στο στοιχείο `link` προστέθηκε η ιδιότητα `sizes` για τον καθορισμό μεγέθους των εικο-νιδίων που μπορούν να συσχετιστούν με το στοιχείο `link` μέσω της

ιδιότητας rel.

- Στο στοιχείο ol η νέα ιδιότητα δηλώνει ότι η λίστα είναι σε φθίνουσα κατάταξη.
- Στο στοιχείο iframe προστέθηκαν οι ιδιότητες skaiibox, seamless και srcdoc για την ασφαλή προβολή περιεχομένου.
- Μερικές ιδιότητες της HTML4 γίνανε global και τώρα μπορούν να εφαρμοστούν σε όλα τα στοιχεία, οι ιδιότητες αυτές είναι: accesskey, class, dir, id, lang, style, tabindex και title, επίσης η ιδιότητα xml:space πλέον επιτρέπεται σε όλα τα στοιχεία στην XHTML.

Επιπλέον προστέθηκαν και οι καινούργιες global μεταβλητές.

- Η ιδιότητα contenteditable δείχνει αν το στοιχείο είναι μεταβλητό από τον χρήστη.
- Οι ιδιότητες draggable και dropzone μπορούν να χρησιμοποιηθούν σε συνδιασμό με το νέο drag και dropAPI.
- Η ιδιότητα hidden δείχνει ότι το στοιχείο δεν είναι άλλο χρήσιμο ή δεν είναι σχετικό ακόμα.
- Η ιδιότητα spellcheck επιτρέπει την υπόδειξη διαθεσιμότητας ορθογραφικού ελέγχου για το συγκεκριμένο στοιχείο.

Η HTML5 παίρνει όλους τους χειριστές συμβάντων από την HTML4 και τους μετατρέπει σε global ιδιότητες και προσθέτει κάποιους χειριστές επιπλέον όπως για παράδειγμα του χειριστές για το video.

Με την προσθήκη των νέων στοιχείων κάποια στοιχεία άλλαξαν για να γίνουν πιο χρήσιμα ή για να γίνει πιο ξεκάθαρος ο λόγος χρήσης τους στο web.

- Το στοιχείο a χωρίς την ιδιότητα href τώρα είναι για να κρατάει χώρο για μελλοντική χρήση ή για να κρατάει περιεχόμενο για την ροή του εγγράφου.
- Το στοιχείο address έχει αλλαχτεί για να συμβαδίζει με την λογική των sections.
- Το στοιχείο b τώρα αναπαριστά ένα κομμάτι κειμένου όπου χρήζει προσοχής χωρίς όμως το κομμάτι αυτό να έχει κάποια ιδιαίτερη σημασία και χωρίς να υπονοεί αλλαγή ύφους ή διάθεσης, για παράδειγμα οι λέξεις κλειδιά σε μια εργασία.
- Το στοιχείο cite πλέον υπάρχει μόνο για την σήμανση ενός τίτλου κάποιου έργου.

- Το στοιχείο dl είναι μόνο για λίστες ονόματος – τιμής και δεν θεωρείτε πλέον κατάλληλη για την σήμανση διαλόγου.
- Στο στοιχείο head δεν επιτρέπεται πλέον η χρήση του στοιχείου object σαν απόγονος.
- Το στοιχείο hr αναπαριστά αλλαγή στυλ σε επίπεδο παραγράφου.
- Το στοιχείο i πλέον σημαίνει αλλαγή φωνής, διάθεσης ή αλλαγή από το σύνηθες ύ-φος του κειμένου.
- Το στοιχείο menu επαναπροσδιορίστηκε ώστε να χρησιμεύει σε γραμμές εργαλείων.
- Το στοιχείο s προσδιορίζει περιεχόμενα τα οποία δεν είναι πλέον σχετικά ή έγκυρα.
- Το στοιχείο small αναπαριστά σχόλια.
- Το στοιχείο strong πλέον αναπαριστά την σημαντικότητα αντί για την έμφαση.

Οι αλλαγές δεν μείνανε μόνο στα στοιχεία αλλά αλλάξανε και οι ιδιότητες αυτών. Η ιδιότητα value στα στοιχεία li δεν θεωρείτε περιττή, όπως επίσης και η ιδιότητα start στα στοιχεία ol. Η ιδιότητα target για τα στοιχεία a και area ξαναποκτά νόημα ιδιαίτερα στις εφαρμογές web και η ιδιότητα type στα στοιχεία script και style δεν χρειάζεται πλέον εφόσον η γλώσσα του script είναι η ECMAScript (javascript) και η γλώσσα του style είναι η CSS, τέλος η ιδιότητα border μπορεί να πάρει μόνο τις τιμές ένα ή άδειο αλφαριθμητικό.

Υπάρχει μια ομάδα ιδιοτήτων οι οποίες επιτρέπονται αλλά δεν συνιστάτε η χρήση τους και συστήνετε στους developers να χρησιμοποιούνται οι εναλλακτικές λύσεις:

- Η ιδιότητα border στο στοιχείο img πρέπει να έχει την τιμή μηδέν όταν αυτή υπάρχει οι developers συστήνετε να χρησιμοποιήσουν την CSS.
- Η ιδιότητα language στο script πρέπει να έχει σαν τιμή το αλφαριθμητικό "JavaScript" όταν υπάρχει και δεν μπορεί να έρχεται σε αντίθεση με την ιδιότητα type. Καλύτερα η ιδιότητα να παραλείπεται τελείως καθώς δεν προσφέρει κάτι.
- Αντί της ιδιότητας name στο στοιχείο a οι developers καλύτερα να χρησιμοποιούνε την ιδιότητα id.

- Οι ιδιότητες width και height στο στοιχείο img και άλλα στοιχεία δεν μπορούν να έ-χουν ποσοστά σαν τιμή.

Τα παρακάτω στοιχεία συνεχίζουν να υποστηρίζονται από τους browsers αλλά θα πρέπει να αποφεύγετε η χρήση τους καθώς είναι καθαρά για μορφοποίηση του εγγράφου και είναι σωστό αυτό να γίνετε με την χρήση της CSS: basfont, big, center, font, strike, tt. Υ-πάρχουν και στοιχεία που έχουν καταργηθεί τελείως στο specification της HTML5 γιατί η χρήση τους επηρεάζει αρνητικά την προσβασιμότητα και την χρηστικότητα του εγγράφου. Τα στοιχεία αυτά είναι: frame, frameset, noframes.

Υπάρχει και μια ομάδα στοιχείων που καταργήθηκε γιατί δεν ήταν ευρέως χρησιμοποι-ούμενα, ήταν δυσνόητα ή μπορούν να αντικατασταθούν από άλλα στοιχεία αναλυτικά τα στοιχεία είναι:

- Acronym η χρήση του ήταν δυσνόητη. Οι developers μπορούν να χρησιμοποιούνε το abbr για ακρωνύμια και συντομεύεις.
- Applet καταργήθηκε για το καλύτερο αντικείμενο object.
- Isindex αντικαταστάθηκε με την χρήση στοιχείων ελέγχου φόρμας.
- Dir αντικαταστάθηκε από το στοιχείο ul.
- Το στοιχείο noscript καταργήθηκε από τον xmlhttparser αλλά συνεχίζει να υποστηρίζετε από τον parser της HTML.

Δεν καταργήθηκαν μόνο στοιχεία αλλά και ιδιότητες. Τα specifications συνεχίζουν να ο-ρίζουν πως οι browsers πρέπει να χειρίζονται για παλαιότερα έγγραφα αλλά οι developers δεν πρέπει να συνεχίσουν να τα χρησιμοποιούνε και τα έγγραφα δεν θα περνάνε validation. Οι ιδιότητες είναι:

- Οι ιδιότητες Rev και charset στα στοιχείο link και a.
- Η ιδιότητα Shape και cords στο a.
- Η ιδιότητα Longdesc στα στοιχεία img και frame.
- Η ιδιότητα Target στο στοιχείο link.
- Η ιδιότητα Nohref στο στοιχείο area.
- Η ιδιότητα Profile στο στοιχείο head.
- Η ιδιότητα version στο στοιχείο html.

- Η ιδιότητα name στο στοιχείο img, πρέπει να γίνεται χρήση της ιδιότητας id.
- Η ιδιότητα scheme στο στοιχείο meta.
- Οι ιδιότητες archive, classid, codebase, codetype, declare και stcaibg στο στοιχείο ob-ject.
- Οι ιδιότητες valuetype και type στο στοιχείο param.
- Οι ιδιότητες axis και abbr στα στοιχεία td και th.
- Η ιδιότητα scope στο στοιχείο td.
- Η ιδιότητα summary στο στοιχείο table.

Επιπλέον οι παρακάτω ιδιότητες αφαιρέθηκαν καθώς είναι καθαρά για την μορφοποίηση του εγγράφου, κατά συνέπεια είναι καλύτερο να γίνει με την χρήση της CSS.

- Η ιδιότητα align στα στοιχεία caption, iframe, img, input, object, legend, table, hr, div, h1, h2, h3, h4, h5, h6, p, col, colgroup, tbody, td, tfoot, th, thead και tr.
- Οι ιδιότητες alink, link, text και vlink στο στοιχείο body.
- Η ιδιότητα bgcolor στα στοιχεία table, tr, td, th και body.
- Η ιδιότητα border στο στοιχείο object.
- Οι ιδιότητες cellpadding και cellspacing στο στοιχείο table.
- Οι ιδιότητες char και charoff στα στοιχεία col, colgroup, tbody, td, tfoot, th, thead και tr.
- Η ιδιότητα clear στο στοιχείο br.
- Η ιδιότητα compact στα στοιχεία dl, menu, ol και ul.
- Η ιδιότητα frame στο στοιχείο table.
- Η ιδιότητα frameborder στο στοιχείο iframe.

- Η ιδιότητα `height` στα στοιχεία `td` και `th`.
- Οι ιδιότητες `hspace` και `vspace` στα στοιχεία `img` και `object`.
- Οι ιδιότητες `marginheight` και `marginwidth` στο στοιχείο `iframe`.
- Η ιδιότητα `noshade` στο στοιχείο `hr`.
- Η ιδιότητα `nowrap` στα στοιχεία `td` και `th`.
- Η ιδιότητα `rules` στο στοιχείο `table`.
- Η ιδιότητα `scrolling` στο στοιχείο `iframe`.
- Η ιδιότητα `size` στο στοιχείο `hr`.
- Η ιδιότητα `type` στα στοιχεία `li`, `ol` και `ul`.
- Η ιδιότητα `valign` στα στοιχεία `col`, `colgroup`, `tbody`, `td`, `tfoot`, `th`, `thead` και `tr`.
- Η ιδιότητα `width` στα στοιχεία `hr`, `table`, `td`, `th`, `col`, `colgroup` και `pre`.

2.7. Επίλογος

Οι ετικέτες της HTML σχεδιάστηκαν αρχικά για να ορίσουν το περιεχόμενο ενός εγγράφου. Η δουλειά τους ήταν να λένε "Αυτή είναι μια επικεφαλίδα", "Αυτή είναι μια παράγραφος", "Αυτός είναι ένας πίνακας", χρησιμοποιώντας ετικέτες όπως `<h1>`, `<p>`, `<table>` κ.ο.κ. Η διάταξη (layout) του εγγράφου ήταν υπόθεση του φυλλομετρητή, χωρίς τη χρήση ετικετών μορφοποίησης (formattingtags).

Καθώς οι δύο κύριοι φυλλομετρητές, ο Netscape και ο Internet Explorer, συνέχισαν να προσθέτουν νέες HTML ετικέτες και χαρακτηριστικά (attributes), όπως η ετικέτα ``, στις αρχικές προδιαγραφές της HTML, γινόταν ολοένα και δυσκολότερη η δημιουργία Web γιατί οι σημάνσεις μορφοποίησης "μπλέκονταν" με τις σημάνσεις ορισμού κειμένου. Για να μπορέσει να λύσει αυτό το πρόβλημα, το W3C δημιούργησε τα ΣΤΥΛ (STYLES) σαν προσθήκη στην HTML 4.0.

3. CSS

3.1 Εισαγωγή

Τα **CSS (Cascading Style Sheets)** δηλαδή Διαδοχικά Φύλλα Στυλ δημιουργήθηκαν με σκοπό τη διαμόρφωση της παρουσίασης - εμφάνισης μιας ιστοσελίδας έτσι ώστε αυτή να διαχωριστεί τελείως από την HTML. Χρησιμοποιούνται κυρίως σε εφαρμογές γραμμένες σε HTML και XHTML αλλά μπορεί να χρησιμοποιηθούν επίσης και σε οποιοδήποτε έγγραφο της XML για τη μορφοποίησή του. Ο διαχωρισμός αυτός επιτρέπει παράλληλα σε πολλά έγγραφα να μοιράζονται την ίδια μορφοποίηση με αποτέλεσμα τη μείωση της πολυπλοκότητας και της επανάληψης κώδικα. Για την δημιουργία και επεξεργασία εγγράφων CSS, μπορεί να χρησιμοποιηθεί ένας επεξεργαστής (editor) απλού κειμένου ή κάποιο πρόγραμμα ειδικό για CSS, που προσφέρει αρκετές ευκολίες στην διαμόρφωση του κώδικα CSS. Ένα πρόγραμμα (open source, δωρεάν διαθέσιμο για download και χρήση) για την επεξεργασία εγγράφων CSS, είναι το CSSED.

3.2 Πλεονεκτήματα CSS

Τα πλεονεκτήματα από τη χρήση style sheets είναι αρκετά κ σημαντικά. Κατ' αρχήν ο προγραμματιστής μπορεί εύκολα να τροποποιήσει παράλληλα την εμφάνιση πολλών διαφορετικών HTML εγγράφων τροποποιώντας μόνο ένα αρχείο στυλ. Έτσι, οι ενδεχόμενες αλλαγές, γίνονται μόνο σε ένα έγγραφο και αυτόματα εφαρμόζονται σε όλες τις σελίδες που το χρησιμοποιούν. Αν έχουμε ποτέ προσπαθήσει να αλλάξουμε τη γραμματοσειρά (font) ή το χρώμα (color) όλων των επικεφαλίδων (headings) σ' όλες τις ιστοσελίδες μας, θα καταλάβουμε γιατί τα CSS μπορούν να μας γλυτώσουν από πολλή δουλειά. Επίσης, ο συνολικός κώδικας είναι σημαντικά λιγότερος με τη χρήση CSS με αποτέλεσμα το αρχείο να είναι πιο μικρό και ελαφρύ "κατεβαίνοντας" έτσι γρηγορότερα.

3.3 Σύνταξη των CSS

Η σύνταξη των CSS αποτελείται από τρία μέρη: έναν επιλογέα (selector), μια ιδιότητα (property) και μια τιμή (value) :

επιλογέας {ιδιότητα: τιμή}

```
selector {property: value}
```

Ο **επιλογέας** είναι συνήθως το στοιχείο/tag που θέλουμε να ορίσουμε, η **ιδιότητα** είναι το χαρακτηριστικό που θέλουμε να αλλάξουμε και η κάθε ιδιότητα μπορεί να πάρει μια **τιμή**. Η ιδιότητα και η τιμή ξεχωρίζουν από τον χαρακτήρα : και περικλείονται από τους χαρακτήρες { }, ως εξής :

```
body {color: red}
```

Αν η τιμή αποτελείται από πολλές λέξεις, πρέπει να τοποθετήσουμε εισαγωγικά :

```
p { font-family: "sans serif" }
```

Αν θέλουμε να ορίσουμε περισσότερες από μία ιδιότητες, πρέπει να ξεχωρίσουμε την κάθε ιδιότητα με τον χαρακτήρα ;. Το παρακάτω παράδειγμα δείχνει πώς μπορούμε να ορίσουμε μια κεντραρισμένη παράγραφο με χρώμα κειμένου πράσινο:

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

Για να κάνουμε τους ορισμούς των στυλ πιο ευανάγνωστους, μπορούμε να γράψουμε από μία ιδιότητα σε κάθε γραμμή, ως εξής :

```
p { text-align: center;
    color: black;
    font-family: arial
}
```

3.4 Ιδιότητες φόντου των CSS

Οι ιδιότητες φόντου (background properties) μάς δίνουν τη δυνατότητα να εφαρμόσουμε χρώματα (colors) και χρώματα φόντου (background colors) στις ιστοσελίδες μας. Θα δούμε επίσης και προηγμένες μεθόδους για την τοποθέτηση και τον έλεγχο των εικόνων φόντου (background images).

3.4.1. Η Ιδιότητα color

Η ιδιότητα color περιγράφει το χρώμα προσκηνίου (*foreground color*) ενός στοιχείου. Για παράδειγμα, ας υποθέσουμε ότι θέλουμε όλες οι επικεφαλίδες (headlines) ενός εγγράφου να έχουν χρώμα κόκκινο (red). Οι επικεφαλίδες χαρακτηρίζονται με το tag <h1> και ο παρακάτω κώδικας ορίζει σε κόκκινο το χρώμα των στοιχείων <h1> :

```
h1 {
    color: #ff0000;
}
```

Τα χρώματα μπορούν να γραφούν με τις δεκαεξαδικές τους τιμές, όπως στο παραπάνω παράδειγμα (#ff0000), ή μπορούμε να χρησιμοποιήσουμε τα ονόματα των χρωμάτων ("red") ή και τις τιμές RGB : rgb(255, 0, 0).

3.4.2. Η Ιδιότητα `background-color`

Η ιδιότητα `background-color` περιγράφει το *χρώμα φόντου* (*background color*) ενός στοιχείου. Ως γνωστόν, το στοιχείο `<body>` περιέχει όλο το κείμενο ενός HTML εγγράφου και έτσι για να αλλάξουμε το χρώμα φόντου μιας ιστοσελίδας, θα πρέπει να εφαρμόσουμε την ιδιότητα `background-color` στο στοιχείο `<body>`.

Μπορούμε επίσης να εφαρμόσουμε χρώμα φόντου και σ' άλλα στοιχεία, όπως σε επικεφαλίδες (*headlines*) και σε απλό κείμενο (*text*). Στο παρακάτω παράδειγμα, εφαρμόζουμε διαφορετικά χρώματα φόντου στα στοιχεία `<body>` και `<h1>`.

```
body {  
  
background-color: #FFCC66;  
  
}  
  
h1 {  
  
color: #990000;  
  
background-color: #FC9804;  
  
}
```

Βλέπουμε ότι για να εφαρμόσουμε δύο ιδιότητες σ' ένα στοιχείο, όπως το `<h1>`, τις ξεχωρίζουμε με το σύμβολο `;`.

3.4.3. Η Ιδιότητα `background-image`

Η ιδιότητα `background-image` χρησιμοποιείται για να εισάγουμε μια *εικόνα φόντου* (*background image*) σε μια ιστοσελίδα. Απλά εφαρμόζουμε την ιδιότητα `background-image` στο tag `<body>` και προσδιορίζουμε το όνομα και τη θέση του αρχείου της εικόνας.

```

body {

    background-color: #FFCC66;

    background-image: url("image.jpg");

}

h1 {

    color: red;

    background-color: blue;

}

```

Βλέπουμε ότι προσδιορίζουμε το όνομα και τη θέση της εικόνας με την έκφραση `url("image.jpg")`. Αυτό βέβαια προϋποθέτει ότι η εικόνα βρίσκεται στον ίδιο φάκελο με το φύλλο στυλ (style sheet).

Μπορούμε να αναφερόμαστε και σε εικόνες που βρίσκονται σ' άλλους φακέλους με την έκφραση `url("../images/image.jpg")` ή που βρίσκονται ακόμη και στο Internet, αρκεί να ορίσουμε την πλήρη διεύθυνση του αρχείου με την έκφραση `url("http://something.something.gr/image.jpg")`.

3.4.4. Η Ιδιότητα `background-repeat`

Στο παραπάνω παράδειγμα, παρατηρούμε ότι εξ ορισμού η εικόνα επαναλαμβάνεται και προς την οριζόντια και προς την κατακόρυφη κατεύθυνση για να καλύψει ολόκληρη την οθόνη. Με την ιδιότητα `background-repeat` μπορούμε να ελέγχουμε την επανάληψη μιας εικόνας φόντου.

Η ιδιότητα `background-repeat` μπορεί να έχει τις εξής τέσσερις τιμές :

Τιμή	Περιγραφή
<code>background-repeat: repeat-x</code>	Η εικόνα επαναλαμβάνεται μόνο οριζόντια
<code>background-repeat: repeat-y</code>	Η εικόνα επαναλαμβάνεται μόνο κάθετα
<code>background-repeat: repeat</code>	Η εικόνα επαναλαμβάνεται οριζόντια και κάθετα
<code>background-repeat: no-repeat</code>	Η εικόνα δεν επαναλαμβάνεται

Πίνακας 2 : Ιδιότητα `background-repeat`

Για παράδειγμα, για να αποφύγουμε την επανάληψη μιας εικόνας φόντου (background image), θα πρέπει να γράψουμε τον εξής κώδικα :

```
body {  
  
    background-color: #FFCC66;  
  
    background-image: url("florina1.jpg");  
  
    background-repeat: no-repeat;  
  
}  
  
h1 {  
  
    color: green;  
  
    background-color: #FC9804;  
  
}
```

3.4.5. Η Ιδιότητα background-attachment

Με την ιδιότητα background-attachment μπορούμε να καθορίσουμε αν μια εικόνα φόντου θα είναι σταθερή (fixed) ή θα κυλάει (scrolls) μαζί με τα περιεχόμενα της ιστοσελίδας. Μια εικόνα φόντου που είναι σταθερή (fixed) δεν θα μετακινείται μαζί με το κείμενο όταν ο χρήστης κυλάει τα περιεχόμενα της ιστοσελίδας, ενώ μια μη κλειδωμένη εικόνα φόντου θα κυλάει μαζί με τα περιεχόμενα της ιστοσελίδας (εξ ορισμού).

Ο παρακάτω πίνακας παρουσιάζει τις δύο διαφορετικές τιμές που μπορεί να λάβει η ιδιότητα background-attachment :

Τιμή	Περιγραφή
background-attachment: scroll	Η εικόνα κυλάει μαζί με την υποσελίδα
background-attachment: fixed	Η εικόνα είναι κλειδωμένη

Πίνακας 3: Ιδιότητα background-attachment

Ο παρακάτω κώδικας κάνει την εικόνα φόντου της ιστοσελίδας να μην επαναλαμβάνεται (no-repeat) και να είναι κλειδωμένη (fixed) :

```
body {  
  
background-color: #FFCC66;  
  
background-image: url("florina1.jpg");  
  
background-repeat: no-repeat;  
  
background-attachment: fixed;  
  
}  
  
h1 {  
  
color: #990000;  
  
background-color: #FC9804;  
  
}
```

3.4.6. Η Ιδιότητα background-position

Εξ ορισμού, μια εικόνα φόντου τοποθετείται στην πάνω αριστερή γωνία της οθόνης. Με την ιδιότητα background-position μπορούμε να αλλάξουμε αυτή τη ρύθμιση και να τοποθετήσουμε την εικόνα φόντου οπουδήποτε θέλουμε στην οθόνη.

Υπάρχουν τρεις διαφορετικοί τρόποι για να προσδιορίσουμε τη θέση μιας εικόνας φόντου μέσα στην οθόνη. Για παράδειγμα, μπορούμε να ορίσουμε τις συντεταγμένες της εικόνας με τις τιμές '100px 200px', οπότε η εικόνα φόντου θα τοποθετηθεί σε απόσταση 100px (pixels) από την αριστερή πλευρά της οθόνης και 200px (pixels) από την κορυφή της οθόνης, στην ουσία του παραθύρου του φυλλομετρητή.

Ένας άλλος τρόπος είναι να δηλώσουμε την απόσταση σε ποσοστά, όπου οι τιμές 0% 0% σημαίνουν την πάνω αριστερή γωνία της οθόνης, οι τιμές 50% 50% το κέντρο της οθόνης και οι τιμές 100% 100% την κάτω δεξιά γωνία της οθόνης. Ένας τρίτος τρόπος είναι να χρησιμοποιήσουμε τις δεσμευμένες λέξεις top, bott

om, center, left και right και να κανουμε συνδυασμούς, όπως top left, center center, top right, bottom right κ.ά.

Ο παρακάτω πίνακας έχει μερικά χαρακτηριστικά παραδείγματα :

Τιμή	Περιγραφή
background-position: 2cm 2cm	Η εικόνα φόντου θα τοποθετηθεί 2 cm από τα αριστερά και 2 cm από την κορυφή
background-position: 50% 25%	Η εικόνα φόντου θα τοποθετηθεί στο κέντρο οριζόντια και κατά το ένα τέταρτο
background-position: top right	Η εικόνα φόντου θα τοποθετηθεί στην πάνω δεξιά γωνία (top-right) της οθόνης

Πίνακας 4: Ιδιότητα background-position

Ο παρακάτω κώδικας θα τοποθετήσει την εικόνα φόντου στην κάτω δεξιά γωνία (bottom right) της οθόνης :

```
body {  
  
    background-color: #FFCC66;  
  
    background-image: url("florina1.jpg");  
  
    background-repeat: no-repeat;  
  
    background-attachment: fixed;  
  
    background-position: right bottom;  
  
}  
  
h1 {  
  
    color: #990000;  
  
    background-color: #FC9804;  
  
}
```

3.4.7. Η Ιδιότητα `background-position`

Η ιδιότητα `background` αποτελεί μια συντόμευση γι' όλες τις ιδιότητες του αντικειμένου `background` που είδαμε προηγουμένως και μας δίνει τη δυνατότητα να γράψουμε όλες τις ιδιότητες σε μια γραμμή κώδικα.

Για παράδειγμα, ας δούμε τις επόμενες πέντε γραμμές κώδικα :

```
background-color: #FFCC66;  
background-image: url("image.jpg");  
background-repeat: no-repeat;  
background-attachment: fixed;  
background-position: rightbottom;
```

Με τη χρήση της ιδιότητας `background` μπορούμε να πετύχουμε το ίδιο αποτέλεσμα σε μία μόνο γραμμή κώδικα :

```
background: #FFCC66 url("image.jpg") no-repeat fixed right  
bottom;
```

Η σειρά αναγραφής των ιδιοτήτων είναι η εξής :

```
[background-color] | [background-image] | [background-repeat]  
| [background-attachment] | [background-position]
```

Αν παραλείψουμε κάποια ιδιότητα, θα εκληφθεί η προκαθορισμένη (default) τιμή της. Για παράδειγμα, αν παραλειφθούν οι δύο τελευταίες ιδιότητες, `background-attachment` και `background-position`, θα έχουμε το εξής :

```
background: #FFCC66 url("image.jpg") no-  
repeat;
```

3.5. Οι γραμματοσειρές (Fonts)

Θα δούμε τώρα τις γραμματοσειρές (fonts) και πώς μπορούμε να τις εφαρμόσουμε με τη βοήθεια των CSS.

3.5.1. Η Ιδιότητα `font-family`

Η ιδιότητα `font-family` χρησιμοποιείται για να ορίσει μια λίστα προτεραιότητας από γραμματοσειρές που θα χρησιμοποιηθούν για να εμφανισθεί ένα συγκεκριμένο στοιχείο ή μια ιστοσελίδα. Αν η πρώτη γραμματοσειρά που υπάρχει στη λίστα δεν είναι εγκατεστημένη στον υπολογιστή από τον οποίο γίνεται η πρόσβαση στο site, θα δοκιμασθεί η επόμενη γραμματοσειρά κοκ μέχρι να βρεθεί κάποια που να ταιριάζει. Υπάρχουν δύο είδη ονομάτων που χρησιμοποιούνται για να κατηγοριοποιηθούν οι γραμματοσειρές : τα `family-names` και τα `generic families`.

Παραδείγματα ενός `family-name`, που τα ξέρουμε και ως font, είναι για παράδειγμα τα "Arial", "Times New Roman", "Tahoma" κ.ά. Τα `generic families` είναι ομάδες από `family-names` με παρόμοια εμφάνιση. Ένα παράδειγμα είναι το `sans-serif`, που είναι μια συλλογή από γραμματοσειρές χωρίς πόδια (feet).

Στο generic family `serif` ανήκουν οι γραμματοσειρές *Times New Roman*, *Garamond* και *Georgia*, όπου όλες έχουν πόδια (feet). Στο generic family `sans-serif` ανήκουν οι γραμματοσειρές *Trebuchet*, *Arial* και *Verdana*, όπου όλες δεν έχουν πόδια (feet). Στο generic family `monospace` ανήκουν οι γραμματοσειρές *Courier*, *Courier New* και *Andale Mono*, όπου όλοι οι χαρακτήρες τους έχουν το ίδιο πλάτος.

Όταν εμφανίζουμε τις γραμματοσειρές για την ιστοσελίδα μας, θα πρέπει να ξεκινάμε με τις πιο δημοφιλείς γραμματοσειρές και μετά με κάποιες εναλλακτικές. Αποτελεί καλή πρακτική να ολοκληρώνουμε τη λίστα με μια γραμματοσειρά από generic family. Έτσι, η ιστοσελίδα θα εμφανισθεί με μια γραμματοσειρά της ίδιας οικογένειας αν δεν βρεθεί κάποια από τις προηγούμενες γραμματοσειρές.

Ένα παράδειγμα χρήσης της ιδιότητας font-family μπορεί να είναι το εξής :

```
h1 {  
  
    font-family: arial, verdana, sans-serif;  
  
}  
  
h2 {  
  
    font-family: "Times New Roman", serif;  
  
}
```

Σύμφωνα με τον παραπάνω κώδικα, οι επικεφαλίδες (headlines) που είναι μαρκαρισμένες με το tag <h1> θα εμφανισθούν με τη γραμματοσειρά "Arial". Αν αυτή η γραμματοσειρά δεν είναι εγκατεστημένη στον υπολογιστή του χρήστη, θα χρησιμοποιηθεί η "Verdana" και αν καμία απ' αυτές δεν είναι διαθέσιμη, θα χρησιμοποιηθεί μια γραμματοσειρά από την οικογένεια sans-serif, ώστε να εμφανισθεί σωστά το κείμενο των επικεφαλίδων.

Επειδή το όνομα της γραμματοσειράς "Times New Roman" περιέχει κενούς χαρακτήρες, χρησιμοποιήσαμε διπλά εισαγωγικά για να το προσδιορίσουμε.

3.5.2. Η Ιδιότητα font-style

Η ιδιότητα font-style κάνει τη μορφοποίηση της επιλεγμένης γραμματοσειράς να είναι *normal*, *italic* ή *oblique*. Στο επόμενο παράδειγμα, όλες οι επικεφαλίδες που είναι μαρκαρισμένες με το tag <h2> θα εμφανισθούν με πλάγια γράμματα.

```
h1 {  
  
    font-family: arial, verdana, sans-serif;  
  
}  
  
h2 {  
  
    font-family: "Times New Roman", serif; font-style: italic;  
  
}
```

3.5.3. Η Ιδιότητα `font-variant`

Η ιδιότητα `font-variant` χρησιμοποιείται για να επιλέξουμε ανάμεσα στις παραλλαγές *normal* ή *small-caps* μιας γραμματοσειράς. Μια γραμματοσειρά του τύπου *small-caps* χρησιμοποιεί μικρού μεγέθους κεφαλαία γράμματα αντί για μικρά (πεζά) γράμματα. Στην περίπτωση που η ιδιότητα `font-variant` είναι ορισμένη στην τιμή *small-caps* και δεν υπάρχει διαθέσιμη κάποια γραμματοσειρά που να την υποστηρίζει, ο φυλλομετρητής είναι πολύ πιθανό να εμφανίσει το κείμενο με κανονικά κεφαλαία γράμματα.

Ακολουθούν παραδείγματα :

```
h1 {  
    font-variant: small-caps;  
}  
  
h2 {  
    font-variant: normal;  
}
```

3.5.4. Η Ιδιότητα `font-weight`

Η ιδιότητα `font-weight` περιγράφει πόσο έντονα θα πρέπει να εμφανίζεται μια γραμματοσειρά. Μια γραμματοσειρά μπορεί να είναι είτε *normal* ή *bold*. Μερικοί φυλλομετρητές υποστηρίζουν ακόμη και αριθμούς από το 100 έως το 900 για να προσδιορίσουν με ακρίβεια το πόσο έντονη θέλουμε να είναι η γραμματοσειρά.

Ακολουθούν παραδείγματα :

```
p {  
  
    font-family: arial, verdana, sans-serif;  
  
}  
  
td {  
  
    font-family: arial, verdana, sans-serif; font-weight: bold;  
  
}
```

3.5.5. Η Ιδιότητα font-size

Η ιδιότητα font-size ορίζει το μέγεθος μιας γραμματοσειράς. Υπάρχουν πολλές μονάδες, όπως pixels, points και εκατοστά, που μπορούμε να χρησιμοποιήσουμε για να προσδιορίσουμε το μέγεθος μιας γραμματοσειράς.

Ακολουθούν παραδείγματα :

```
h1 {  
  
    font-size: 30px; }  
  
h2 {  
  
    font-size: 12pt;}  
  
h3 {  
  
    font-size: 120%; }  
  
p {  
  
    font-size: 1em; }
```

Με τις μονάδες px και pt προσδιορίζουμε το μέγεθος με απόλυτες τιμές, ενώ με τις μονάδες % και em προσδιορίζουμε το μέγεθος με σχετικές τιμές.

3.5.6. Η Ιδιότητα font

Με την ιδιότητα font μπορούμε να προσδιορίσουμε όλες τις ιδιότητες font σε μία μόνο γραμμή. Για παράδειγμα, οι επόμενες τέσσερις γραμμές κώδικα περιγράφουν τις ιδιότητες γραμματοσειράς του tag <p> :

```
p {  
  
    font-style: italic;  
  
    font-weight: bold;  
  
    font-size: 30px;  
  
    font-family: arial, sans-serif;  
  
}
```

Με την ιδιότητα font μπορούμε να επιτύχουμε το ίδιο αποτέλεσμα, αλλά με μία μόνο γραμμή κώδικα :

```
p {  
  
    font: italic bold 30px arial, sans-serif;  
  
}
```

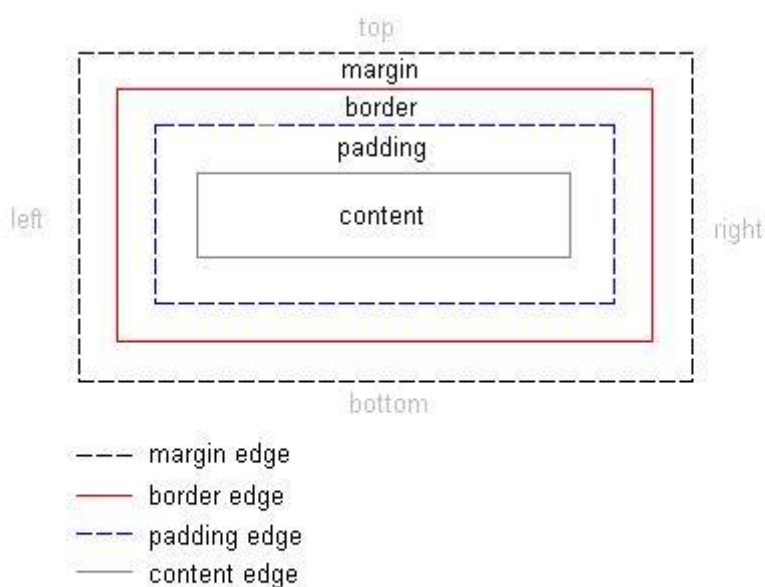
Η σειρά αναγραφής των τιμών των ιδιοτήτων για την ιδιότητα font είναι η εξής :

```
font-style | font-variant | font-weight | font-size | font-family
```

3.6. Ιδιότητες περιθωρίου των CSS

3.6.1. Το boxmodel

Το *boxmodel* στα CSS περιγράφει τα πλαίσια (boxes) που δημιουργούνται για τα στοιχεία (elements) της HTML. Το boxmodel περιέχει επίσης λεπτομέρειες σχετικά με τη ρύθμιση των τιμών των εξής στοιχείων: *margin* (περιθώριο), *border* (περίγραμμα), *padding* (γέμισμα) και *content* (περιεχόμενο). Το παρακάτω σχεδιάγραμμα δείχνει πώς είναι δομημένο το boxmodel μιας ιστοσελίδας :



Για να μπορέσουμε να κατανοήσουμε το παραπάνω σχεδιάγραμμα, θα δούμε ένα πρακτικό παράδειγμα με μια επικεφαλίδα (headline) και λίγο κείμενο (text). Ο HTML κώδικας που θα χρησιμοποιήσουμε θα περιέχει τη διακήρυξη των Δικαιωμάτων του Ανθρώπου και θα είναι ο εξής :

<h1>Article 1:</h1>

<p>All human beings are born free

and equal in dignity and rights.

They are endowed with reason and conscience

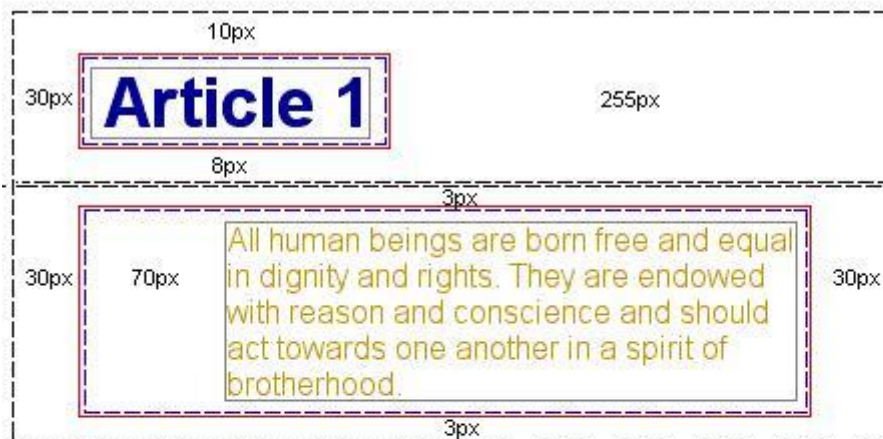
and should act towards one another in a spirit of brotherhood</p>

Προσθέτοντας λίγο χρώμα και ορίζοντας γραμματοσειρές, το κείμενο θα μπορεί να εμφανισθεί ως εξής :

Article 1

All human beings are born free and equal in dignity and rights. They are endowed with reason and conscience and should act towards one another in a spirit of brotherhood.

Το παραπάνω παράδειγμα περιέχει δύο στοιχεία, το <h1> και το <p>, και το box model γι' αυτά τα δύο στοιχεία μπορεί να απεικονισθεί ως εξής :



Από το παραπάνω σχεδιάγραμμα μπορούμε να δούμε πώς το κάθε στοιχείο της HTML περιβάλλεται από πλαίσια (boxes), τα οποία μπορούμε να ρυθμίσουμε με τα CSS.

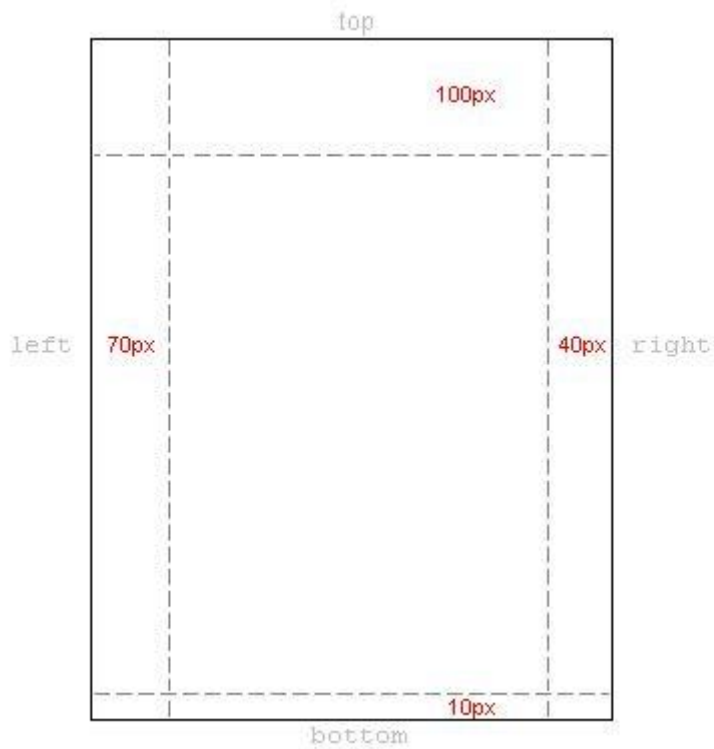
3.6.2. Το Margin και το Padding

Είδαμε νωρίτερα το box model και θα δούμε τώρα το πώς μπορούμε να αλλάξουμε την εμφάνιση των στοιχείων, δίνοντας τιμές στις ιδιότητες margin και padding.

Η Ιδιότητα Margin

Ένα στοιχείο (element) έχει τέσσερις πλευρές : right (δεξιά), left (αριστερά), top (πάνω) και bottom (κάτω). Το margin (περιθώριο) είναι η απόσταση από την κάθε πλευρά μέχρι το γειτονικό στοιχείο ή μέχρι τα όρια (borders) του εγγράφου (document).

Ως πρώτο παράδειγμα, θα δούμε το πώς μπορούμε να ορίσουμε περιθώρια (margins) για το ίδιο το έγγραφο, δηλ. για το στοιχείο <body>. Το επόμενο σχεδιάγραμμα δείχνει τις τιμές που θέλουμε να έχουν τα περιθώρια στις ιστοσελίδες μας.



Ο CSS κώδικας για να δοθούν οι παραπάνω τιμές είναι ο εξής :

```
body {  
  
    margin-top: 100px;  
  
    margin-right: 40px;  
  
    margin-bottom: 10px;  
  
    margin-left: 70px;  
  
}
```

Μπορούμε να χρησιμοποιήσουμε και τον εξής κώδικα :

```
body {  
  
    margin: 100px 40px 10px 70px;  
  
}
```

Με τον ίδιο τρόπο μπορούμε να ορίσουμε περιθώρια (margins) για κάθε στοιχείο της HTML. Για παράδειγμα, μπορούμε να ορίσουμε περιθώρια για το κείμενο όλων των παραγράφων μιας ιστοσελίδας, δηλ. των στοιχείων που είναι μαρκαρισμένα με το tag <p> :

```
body {  
  
    margin: 100px 40px 10px 70px;  
  
}  
  
p {  
  
    margin: 5px 50px 5px 50px;  
  
}
```

Η Ιδιότητα Padding

Το padding μπορούμε να πούμε ότι είναι το γέμισμα (filling) και είναι μια ιδιότητα που δεν επηρεάζει την απόσταση ανάμεσα σε κάποια στοιχεία αλλά ορίζει την εσωτερική απόσταση ανάμεσα στο περίγραμμα (border) και τα περιεχόμενα (content) του στοιχείου.

Θα δούμε ένα απλό παράδειγμα όπου όλες οι επικεφαλίδες έχουν κάποιο χρώμα φόντου για να μπορούμε να τις ξεχωρίσουμε :

```
h1 {  
  
    background: yellow;  
  
}  
  
h2 {  
  
    background: orange;  
  
}
```

Δίνοντας τιμές στην ιδιότητα padding, από μια για top, right, bottom και left, για τα στοιχεία h1 και h2, μπορούμε να ορίσουμε πόσο γέμισμα (filling) θα υπάρχει γύρω από το κείμενο της κάθε επικεφαλίδας :

```
h1 {  
  
background: yellow;  
  
padding: 20px 20px 20px 80px;  
  
}  
  
h2 {  
  
background: orange;  
  
padding-left:120px;  
  
}
```

3.6.3. Τα Περιγράμματα(Borders)

Τα περιγράμματα (borders) μπορούν να χρησιμοποιηθούν σε πολλές περιπτώσεις, όπως για παράδειγμα ως διακοσμητικά στοιχεία ή για την υπογράμμιση του διαχωρισμού δύο πραγμάτων. Με τα CSS έχουμε πάρα πολλές επιλογές για να χρησιμοποιήσουμε τα περιγράμματα στις ιστοσελίδες μας.

Το πλάτος (border-width)

Το πλάτος (width) των περιγραμμάτων ορίζεται με την ιδιότητα border-width, η οποία μπορεί να πάρει μια τις τιμές thin, medium και thick ή και μια αριθμητική τιμή σε pixels. Το παρακάτω σχήμα δείχνει τις πιθανές τιμές της ιδιότητας border-width :



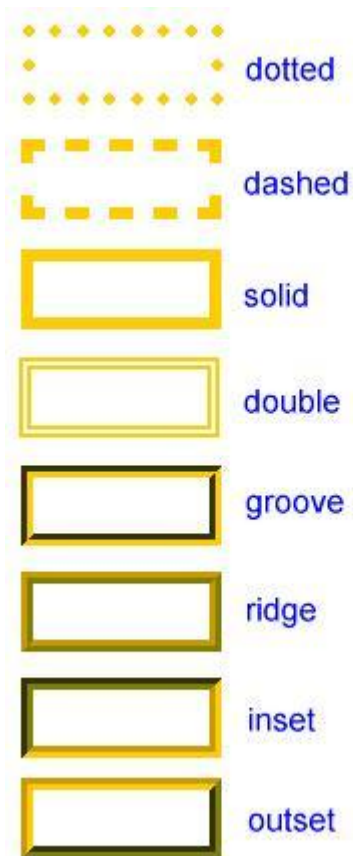
Το χρώμα (border-color)



Η ιδιότητα `border-color` ορίζει το χρώμα του περιγράμματος και οι τιμές που μπορεί να πάρει είναι οι γνωστές τιμές των χρωμάτων, όπως για παράδειγμα `"#123456"`, `"rgb(123, 123, 123)"` ή `"yellow"`.

Το στυλ (border-style)

Υπάρχουν διαφορετικοί τύποι (στυλ) περιγραμμάτων που μπορούμε να επιλέξουμε και παρακάτω φαίνονται 8 απ' αυτούς όπως τους ερμηνεύει ο Internet Explorer 5.5. Τα περιγράμματα αυτά έχουν χρώμα `"gold"` και πάχος `"thick"` αλλά μπορούν φυσικά να έχουν διαφορετικά χρώματα και διαφορετικά πάχη. Αν δεν θέλουμε να έχει περίγραμμα η ιστοσελίδα μας, μπορούμε να χρησιμοποιήσουμε μια από τις τιμές `none` ή `hidden`.



Παράδειγματα Ορισμού Περιγραμμάτων

Οι τρεις ιδιότητες που είδαμε παραπάνω μπορούν να ορισθούν όλες μαζί για το κάθε στοιχείο και να δημιουργήσουν έτσι ενδιαφέροντα περιγράμματα. Θα δούμε ένα παράδειγμα για ένα έγγραφο όπου θα ορίσουμε διαφορετικά περιγράμματα για τα στοιχεία `<h1>`, `<h2>`, `<p>` και `` :

```
h1 {  
  
    border-width: thick;  
  
    border-style: dotted;  
  
    border-color: gold;  
  
    }  
  
h2 {  
  
    border-width: 20px;  
  
    border-style: outset;  
  
    border-color: red;  
  
    }  
  
p {  
  
    border-width: 1px;  
  
    border-style: dashed;  
  
    border-color: blue;  
  
    }  
  
ul {  
  
    border-width: thin;  
  
    border-style: solid;  
  
    border-color: orange;  
  
    }
```

Μπορούμε επίσης να ορίσουμε διαφορετικές ιδιότητες για τις τέσσερις πλευρές ενός περιγράμματος, δηλ. *border-top*, *border-bottom*, *border-right* και *border-left*, όπως φαίνεται παρακάτω :


```
h1 {  
  
border-top-width: thick;  
  
border-top-style: solid;  
  
border-top-color: red;  
  
border-bottom-width: thick;  
  
border-bottom-style: solid;  
  
border-bottom-color: blue;  
  
border-right-width: thick;  
  
border-right-style: solid;  
  
border-right-color: green;  
  
border-left-width: thick;  
  
border-left-style: solid;  
  
border-left-color: orange;  
  
}
```

Όπως ισχύει σε πολλές άλλες ιδιότητες, μπορούμε να συνδυάσουμε πολλές ιδιότητες των περιγραμμάτων σε μία μόνο γραμμή κώδικα.

Για παράδειγμα, ο παρακάτω κώδικας :

```
p {  
  
border-width: 1px;  
  
border-style: solid;  
  
border-color: blue;  
  
}
```

Μπορεί να γραφεί και ως εξής :

```
p {  
  
    border: 1px solid blue;  
  
}
```

Το Ύψος (Height) και το Πλάτος (Width)

Θα δούμε τώρα το πώς μπορούμε να ορίσουμε τις διαστάσεις, δηλ. το ύψος (height) και το πλάτος (width) ενός στοιχείου (element).

Ορισμός του πλάτους (width)

Με την ιδιότητα *width* μπορούμε να ορίσουμε ένα συγκεκριμένο πλάτος για ένα στοιχείο. Ο παρακάτω κώδικας δημιουργεί ένα πλαίσιο (box) μέσα στο οποίο μπορεί να υπάρχει κάποιο κείμενο :

```
div.box {  
  
    width: 200px;  
  
    border: 1px solid black;  
  
    background: orange;  
  
}
```

Ο παραπάνω κώδικας γράφεται σ' ένα αρχείο css, ενώ ο HTML κώδικας μπορεί να είναι κάπως έτσι :

```
<div class="box">... το κείμενο γράφεται εδώ ... </div>
```

Ορισμός του ύψους (height)

Στο προηγούμενο παράδειγμα το ύψος του πλαισίου ορίζεται από το περιεχόμενο που υπάρχει στο πλαίσιο. Μπορούμε να προσδιορίσουμε το ύψος ενός στοιχείου με την ιδιότητα *height*, όπως φαίνεται παρακάτω :

```
div.box {  
  
    height: 500px;  
  
    width: 200px;  
  
    border: 1px solid black;  
  
    background: orange;  
}
```

3.7. Εφαρμογή των CSS

Τα Φύλλα Στυλ επιτρέπουν τον ορισμό των πληροφοριών στυλ με πολλούς τρόπους. Τα στυλ μπορούν να ορισθούν μέσα σ' ένα μόνο HTML στοιχείο, μέσα στο στοιχείο

<head> μιας HTML σελίδας ή σ' ένα εξωτερικό αρχείο CSS. Πολλά εξωτερικά Φύλλα Στυλ μπορούν να χρησιμοποιούνται μέσα απ' ένα μόνο HTML έγγραφο. Σε γενικές γραμμές μπορούμε να πούμε ότι όλα τα στυλ θα καταλήξουν σ' ένα νέο εικονικό (virtual) Φύλλο Στυλ σύμφωνα με τους παρακάτω κανόνες, όπου ο μεγαλύτερος αριθμός έχει και την υψηλότερη προτεραιότητα :

1. Προεπιλογή του φυλλομετρητή.
2. Εξωτερικό Φύλλο Στυλ (External Style Sheet).
3. Εσωτερικό Φύλλο Στυλ (Internal Style Sheet), μέσα στο τμήμα header του εγγράφου.
4. Inline Style, μέσα στο HTML στοιχείο.

3.7.1.Ενσωματωμένα Στυλ

Τα **ενσωματωμένα στυλ** (embedded styles) ορίζονται μέσα στη σήμανση <head> ανάμεσα στη σήμανση <style>. Από τη στιγμή που θα οριστεί ένα style για μια σήμανση κάθε εμφάνιση αυτής της σήμανσης θα μορφοποιείται σύμφωνα με το συγκεκριμένο style. Το μειονέκτημα των ενσωματωμένων στυλ είναι ότι εφαρμόζονται μόνο σε μία σελίδα, έτσι αν θέλουμε την ίδια μορφοποίηση σε μία άλλη σελίδα τότε πρέπει να ξαναγράψουμε τον ίδιο κώδικα.

Παράδειγμα ενός τέτοιου στυλ είναι το παρακάτω:

```
<html>
  <head>
    <title>Embedded CSS
    Example</title><style
    type="text /css">
    p {
    color:
    red; a {
    color:
    green; }
    </style>
  </head>
</html>
```

Το παραπάνω κομμάτι κώδικα θα κάνει όλες τις παραγράφους p της HTML ιστοσελίδας κόκκινες και όλους τους συνδέσμους a μπλε.

3.7.2.Εξωτερικά Στυλ

Τα **εξωτερικά φύλλα μορφοποίησης** (External Style Sheets) είναι πολύ χρήσιμα όταν θέλουμε να μορφοποιήσουμε ένα ολόκληρο site. Τα εξωτερικά φύλλα μορφοποίησης αποθηκεύονται σε ένα ξεχωριστό αρχείο ASCII (με κατάληξη css) και είναι της μορφής:

```
p { bold; color : red }
```

```
h1 { font-size : 16pt; color : green }
```

Το παραπάνω κομμάτι κώδικα θα κάνει όλες τις παραγράφους p της HTML ιστοσελίδας με έντονα και κόκκινα γράμματα και όλους τις επικεφαλίδες με πράσινα γράμματα μεγέθους 16.

3.7.3.Σύνδεση εξωτερικού στυλ με σελίδα

Αν το αρχείο CSS έχει όνομα style.css η κάθε σελίδα πρέπει να έχει έναν δεσμό (link) προς το φύλλο στυλ , ο οποίος βρίσκεται μέσα στο τμήμα head, ως εξής :

```
<html>

  <head>

    <title> External CSS Example</title>

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

  </head>

</html>

<body>
```

Έτσι με αυτό τον τρόπο μπορούμε να συνδέσουμε πολλές HTML σελίδες με ένα CSS αρχείο. Το CSS θα διαμορφώσει ομοιόμορφα όλες τις σελίδες.

Επίσης, πολλά CSS αρχεία μπορούν να ενσωματωθούν σε μία HTML σελίδα όπως για παράδειγμα:

```
<html>

  <head>

    <title> External CSS Example</title>

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

    <link href="style_2.css" rel="stylesheet" type="text/css">

  </head>

</html>

<body>
```

Ακόμη ένα CSS μπορεί να κάνει εισαγωγή (import) ένα άλλο CSS αρχείο.

```
@import: url (style_2.css)
```

Τέλος, μπορούμε να ομαδοποιήσουμε τους επιλογείς. Ξεχωρίζουμε τον κάθε επιλογέα με κόμμα. Στο παρακάτω παράδειγμα έχουμε ομαδοποιήσει όλα τα στοιχεία επικεφαλίδας (header elements). Το κάθε στοιχείο επικεφαλίδας θα είναι πράσινο :

```
h1, h2, h3, h4, h5, h6 { color: green }
```

3.7.4. Ένθετα στυλ

Ένα **ένθετο στυλ** (inline style) χάνει πολλά από τα πλεονεκτήματα των φύλλων στυλ αναμειγνύοντας το περιεχόμενο με την παρουσίαση. Πρέπει να χρησιμοποιούμε αυτή τη μέθοδο με προσοχή, όπως όταν ένα στυλ πρόκειται να εφαρμοσθεί σε μία μοναδική εμφάνιση ενός στοιχείου.

Για να χρησιμοποιήσουμε τα inline styles χρησιμοποιούμε το χαρακτηριστικό (attribute) style στο σχετικό tag. Το χαρακτηριστικό style μπορεί να περιέχει οποιαδήποτε ιδιότητα CSS. Το παρακάτω παράδειγμα δείχνει πώς μπορούμε να αλλάξουμε το χρώμα και το αριστερό περιθώριο μιας παραγράφου :

```
<p style="color: sienna; margin-left: 20px">  
  
Αυτή είναι μια παράγραφος  
  
</p>
```

3.7.5. Τα πολλαπλά φύλλα στυλ

Αν μερικές ιδιότητες έχουν ορισθεί για τον ίδιο επιλογέα (selector) σε διαφορετικά φύλλα στυλ, οι τιμές θα κληρονομηθούν από το γενικότερο φύλλο στυλ. Για παράδειγμα, ένα εξωτερικό φύλλο στυλ έχει αυτές τις ιδιότητες για τον επιλογέα h3:

```
h3 { color: red; text-align: left; font-size: 8pt }
```

Και ένα εσωτερικό φύλλο στυλ έχει αυτές τις ιδιότητες για τον επιλογέα h3 :

```
h3 { text-align: right; font-size: 20pt }
```

Αν η σελίδα με το εσωτερικό φύλλο στυλ συνδέεται επίσης στο εξωτερικό φύλλο στυλ, οι ιδιότητες του h3 θα είναι οι εξής :

```
color: red;  
  
text-align: right;  
  
font-size: 20pt
```

Το χρώμα κληρονομείται από το εξωτερικό φύλλο στυλ και η στοίχιση κειμένου και το μέγεθος γραμματοσειράς αντικαθίστανται από το εσωτερικό φύλλο στυλ.

3.7.6.Υποστήριξη των CSS από Browsers

Τα CSS υποστηρίζονται μόνο από τις τελευταίες εκδόσεις των browsers. Τόσο ο Internet Explorer 4 (υποστήριξε πρώτος, από την έκδοση 3, τα CSS) όσο και ο Netscape Navigator 4 υποστηρίζουν πλήρως τα CSS. Οσο περίεργο και αν ακούγεται, και οι δύο browsers είναι –σχεδόν– πλήρως συμβατοί. Είναι γνωστό σε όλους ότι οι δύο browsers σχεδόν ποτέ δεν συμβαδίζουν, με αποτέλεσμα να χρειάζεται άλλος τρόπος συγγραφής για τον πρώτο και άλλος για τον δεύτερο. Στα CSS όμως δεν ισχύει αυτός ο κανόνας. Ως αποτέλεσμα, τα CSS συχνά καλούνται ως η «νεκρήζωνη» για τους browsers τέταρτης γενιάς. Τι συμβαίνει όμως με τα CSS σε παλιότερους browsers; Ορισμένοι, όπως ο Netscape 2.0 και 3.0, είναι αρκετά «έξυπνοι» ώστε να αγνοήσουν style sheets που βρίσκονται μεταξύ του tag <STYLE>. Ορισμένοι ακόμα πιο παλιοί, όπως ο Internet Explorer 2 και ο Netscape 1, θα αγνοήσουν μεν το <STYLE> tag, αλλά θα εμφανίσουν τα περιεχόμενα μεταξύ των tags. Τα περιεχόμενα αυτά θα τυπωθούν στη κορυφή της σελίδας σε μία γραμμή – και αυτό δεν είναι καθόλου καλό. Για να λυθεί και αυτό το πρόβλημα, η λύση είναι να κάνουμε σχόλιο τα περιεχόμενα του tag με τη χρήση των <!-- -->, όπως στο ακόλουθο παράδειγμα:

```
<html>
  <head>
    <title>Κρύβονταςσε CSS σε παλιότερους
    browsers</title><style type="text /css">

      <!--
      2στοixei
      ο {
      position:
      absolute; z-
      index: 1;
      left:
      30px;
      top:
      30px; } --
      >
    </style>
  </head>
  <body>
    <div id=στοixείο>
      
    </div>
  </body>
</html>
```


Με αυτή την τεχνική, μπορούμε να είμαστε σίγουροι ότι δεν θα συμβεί κάποιο «ατύχημα» σε παλαιότερο browser.

3.7.7.Επίλογος

Ένα βασικό κομμάτι μιας σύγχρονης ιστοσελίδας είναι η αλληλεπίδραση του χρήστη με την σελίδα. Όταν ο χρήστης εκτελεί κάποια ενέργεια για την οποία έχει καθοριστεί μια λειτουργία, αντί να φορτώνεται από τη αρχή η σελίδα, εκτελείται η λειτουργία τοπικά και δίνεται η εντύπωση στο χρήστη της άμεσης ανταπόκρισης. Καθώς οι περισσότερες λειτουργίες είναι απλές, είναι άσκοπο για τον εξυπηρετητή να εκτελούνται όλες από αυτόν αν σκεφτούμε ότι σε μία σελίδα μπορεί να είναι ταυτόχρονα συνδεδεμένοι εκατοντάδες ή ακόμη και χιλιάδες χρήστες.

Έτσι, στα μέσα της δεκαετίας του 1990 άρχισε η ανάπτυξη κάποιων γλωσσών συγγραφής σεναρίων (script languages) προκειμένου να προστεθεί στις ιστοσελίδες διαδραστικότητα (interactivity). Η πρώτη client-side γλώσσα προγραμματισμού που εμφανίστηκε είναι η JavaScript και στη συνέχεια ακολούθησαν και άλλες όπως η Jscript, η VBScript κ.α.

4. JavaScript

4.1. Εισαγωγή

Η JavaScript δημιουργήθηκε το 1995 από τον Brendan Eich, έναν μηχανικό της Netscape και εκδόθηκε με τον Netscape 2 στις αρχές του 1996. Το αρχικό της όνομα ήταν LiveScript αλλά πιθανώς για λόγους μάρκετινγκ, καθώς η εξάπλωση της Java ήταν μεγάλη, μετονομάστηκε σε JavaScript. Είναι μία γλώσσα προγραμματισμού επηρεασμένη από αντικειμενοστραφείς γλώσσες προγραμματισμού όπως η C++ και Java, αλλά η ίδια δεν είναι αντικειμενοστραφής γλώσσα προγραμματισμού.

Ο κώδικας της JavaScript γράφεται σε καθαρό κείμενο (ASCII μορφή) και ενσωματώνεται μέσα στον κώδικα της HTML, μπορεί δε να εκτελεστεί αμέσως ή όταν λαμβάνει χώρα ένα συμβάν (event). Δεν γίνεται μεταγλώττιση (compilation) του κώδικα της JavaScript, αρκεί μόνο ο φυλλομετρητής (browser) να υποστηρίζει την JavaScript.

4.2. Java Vs JavaScript

Το όνομα της JavaScript αποτελεί μόνιμη πηγή σύγχυσης. Αν και ακούγονται ίδιες, η Java και η JavaScript δεν έχουν καμία απολύτως σχέση μεταξύ τους, ούτε στη σύνταξή τους σαν γλώσσες προγραμματισμού ούτε και στις εφαρμογές που χρησιμοποιούνται.

Java	Javascript
Η Java είναι περίπλοκη όσον αφορά τη χρήση της	Η Javascriptείναι σχετικά απλή στη χρήση της
Η Javaείναι σχετικά αργή στην εφαρμογή της	Η Javascriptεκτελείται πολύ γρήγορα
Η Java για να λειτουργήσει απαιτεί plug-in(όπως το JDK)	Η Javascriptαπαιτεί απλώς ένας συμβατό φυλλομετρητή για να λειτουργήσει
Τα προγράμματα στη Javaμεταγλωττίζονται σε .exeαρχεία και ενσωματώνονται σε htmlσελίδες με την ετικέτα <applet>	Στη Javascriptτα προγράμματα ενσωματώνονται σε σελίδες html με τη μορφή σεναρίου και δε χρειάζονται μεταγλώττιση

Η Javaείναι πλήρης γλώσσα προγραμματισμού με πολλαπλές δυνατότητες	Η Javascriptείναι κατάλληλη για μικρού μήκους και απλούστερες χρήσεις
Η Javaείναι προσανατολισμένη προς το αντικείμενο και είναι μια καθαρά αντικειμενοστραφής γλώσσα	Η Javascriptβασίζεται ναί μεν στο αντικείμενο αλλά δεν είναι μια πραγματική αντικειμενοστραφής γλώσσα προγραμματισμού
Η Javaείναι αυστηρή όσον αφορά τους κανόνες σύνταξης των μεταβλητών της	Η Javascriptχαρκτηρίζεται από χαλαρότητα όσον αφορά τους κανόνες σύνταξης της

Πίνακας 5 : Διαφορές Java-Javascript

4.3. Πως εισάγεται η JavaScript

Η JavaScript εισάγεται σε μια σελίδα HTML με την σήμανση `<script>` και το χαρακτηριστικό `language` ως εξής:

```
<script language="JavaScript">
```

Ανάμεσα στις σημάνσεις `<script>` και `</script>` γράφουμε τον κώδικα της JavaScript. Η εντολή για να εμφανίσουμε κάποιο κείμενο σε μια σελίδα είναι η `document.write` για παράδειγμα:

```
document.write ("Καλημέρα ! ")
```

Το σύμβολο `;` είναι προαιρετικό στο τέλος των εντολών

4.4. Που εισάγεται η JavaScript

Μπορούμε να τοποθετήσουμε ένα `script` στο τμήμα `head` ή στο τμήμα `body` μιας HTML σελίδας ή να γράψουμε ένα `script` σε ξεχωριστό αρχείο το οποίο στη συνέχεια θα ενσωματωθεί στην σελίδα.

4.4.1. Script στο τμήμα head

Τα scripts που πρέπει να εκτελεσθούν όταν κληθούν ή όταν προκαλείται ένα **συμβάν** (event) τοποθετούνται στο τμήμα head. Όταν τοποθετούμε το script στο τμήμα head τότε είναι σίγουρο ότι αυτό θα φορτωθεί πριν χρησιμοποιηθεί.

```
<html>
  <head>
    <script language="JavaScript">
      ...εντολές...
    </script>
  </head>
  <body>
    ...
  </body>
</html>
```

4.4.2. Script στο τμήμα body

Τα scripts που πρέπει να εκτελεσθούν όταν φορτώνεται η σελίδα τοποθετούνται στο τμήμα body. Όταν τοποθετούμε ένα script στο τμήμα body τότε αυτό συμμετέχει στη διαμόρφωση του περιεχομένου της σελίδας.

```
<html>
  <head>
  </head>
  <body>
    <script language="JavaScript">
      ...εντολές...
    </script>
    ...
  </body>
</html>
```

4.4.3. Script σε ξεχωριστά αρχεία

Τα scripts εκτός από το να γράφονται απευθείας μέσα στην HTML μπορούν να γραφτούν σε ξεχωριστό με κατάληξη .js, τα οποία καλούνται και εκτελούνται από τη σελίδα HTML με τη βοήθεια της ιδιότητας src .

```
<html>

  <head>

  </head>

  <body>

    <script language="JavaScript"
    src="script.js"></script>

    ...

  </body>

</html>
```

4.5. Χειριστήρια συμβάντος της JavaScript

Τα **χειριστήρια συμβάντος** (event handlers) είναι αυτά απ' όπου αντλεί τη δύναμή της η JavaScript. Παρακολουθώντας τις ενέργειες του χρήστη, μπορεί να γίνει επιβεβαίωση κάποιας ενέργειας του ανάλογα με τις καταχωρήσεις ή τις κινήσεις του. Οι ενέργειες του χρήστη σε μια ιστοσελίδα (HTML page) περιορίζονται στην κίνηση του ποντικιού και στις καταχωρήσεις στοιχείων στα πεδία των φορμών.

- **onBlur**- Το συμβάν αυτό συμβαίνει όταν χάνεται η εστίαση (lost focus), δηλ. όταν απομακρυνόμαστε είτε με κλικ με το ποντίκι είτε πατώντας το πλήκτρο tab σ' ένα πεδίο κειμένου (text field) ή σε μια περιοχή κειμένου (text area) ή και σ' ένα πλαίσιο λίστας (select) μιας φόρμας
- **onChange**- Το συμβάν αυτό συμβαίνει όταν τροποποιείται το περιεχόμενο ενός πεδίου κειμένου (text field) ή μιας περιοχής κειμένου (text area) ή και ενός πλαισίου λίστας (select) μιας φόρμας, πριν ακόμα απομακρυνθούμε από το στοιχείο αυτό.

- **onClick**- Το συμβάν αυτό συμβαίνει όταν κάνουμε κλικ με το ποντίκι σ' ένα αντικείμενο, όπως είναι για παράδειγμα ένα πλήκτρο εντολής (button) ή ένα πλαίσιο ελέγχου (check box).
- **onFocus**- Το συμβάν αυτό συμβαίνει όταν κάνουμε εστίαση (focus) μέσα σ' ένα πεδίο μιας φόρμας. Μπορούμε να εστιάσουμε (receive focus) σ' ένα στοιχείο (πεδίο) μιας φόρμας είτε πατώντας επανειλημμένα το πλήκτρο tab μέχρι να φθάσουμε στο επιθυμητό πεδίο της φόρμας ή κάνοντας κλικ με το ποντίκι μέσα στο συγκεκριμένο πεδίο. Η επιλογή κάποιου κειμένου μέσα σ' ένα πεδίο δημιουργεί το eventselect.
- **onLoad**- Το συμβάν αυτό συμβαίνει όταν τελειώσει το φόρτωμα μιας ιστοσελίδας σ' ένα παράθυρο ή όταν τελειώσει το φόρτωμα όλων των πλαισίων (frames) που βρίσκονται μέσα σ' ένα tag <frameset>
- **onMouseOver**- Το συμβάν αυτό συμβαίνει χώρα όταν ο δείκτης του ποντικιού τοποθετείται πάνω από ένα αντικείμενο ή και από έναν σύνδεσμο (link)
- **onSelect**- Το συμβάν αυτό συμβαίνει όταν επιλέγουμε ένα μέρος ή όλο το κείμενο ενός πεδίου κειμένου (textfield) ή μιας περιοχής κειμένου (textarea)
- **onSubmit**- Το συμβάν αυτό συμβαίνει όταν κάνουμε κλικ με το ποντίκι σ' ένα πλήκτρο υποβολής (submitbutton) για να υποβάλλουμε μια φόρμα. Το χειριστήριο συμβάντος πρέπει να επιστρέψει την τιμή true για να μπορέσουν να υποβληθούν (σταλούν) τα περιεχόμενα της φόρμας στον server. Αν επιστρέψει την τιμή false, δεν θα μπορέσει να γίνει η υποβολή των στοιχείων της φόρμας.
- **onUnload**- Το συμβάν αυτό συμβαίνει όταν φεύγουμε από ένα έγγραφο(ιστοσελίδα). Κανονικά θα πρέπει να τοποθετείται μέσα σ' ένα από τα tags <body> ή <frameset>

4.6. Συναρτήσεις της JavaScript

Μία συνάρτηση ομαδοποιεί ένα σύνολο εντολών υπό μια ονομαζόμενη υπορουτίνα. Αυτό επιτρέπει στο χρήστη να καλεί τη συνάρτηση αυτή όταν χρειαστεί τις λειτουργίες της. Οι συναρτήσεις είναι ένας βασικός τρόπος προγραμματισμού στα περισσότερα προγράμματα σε JavaScript. Φυσικά πριν την κλήση θα πρέπει να έχει γίνει ο ορισμός της συνάρτησης. Η χρήση των συναρτήσεων μπορεί να χωριστεί σε δύο λογικές κατηγορίες: ορισμός συναρτήσεων και κλήση συναρτήσεων. Η JavaScript παρέχει τη δυνατότητα δημιουργίας συναρτήσεων, οι οποίες, ουσιαστικά, είναι σενάρια τα οποία εκτελούνται όταν κληθούν με το όνομά τους, από κάποιο σημείο του HTML κώδικα, ή από κάποιο άλλο σενάριο JavaScript. Οι συναρτήσεις της JavaScript ενσωματώνονται ανάμεσα στις ετικέτες <HEAD> και </HEAD> και η δήλωσή τους γίνεται με την εντολή

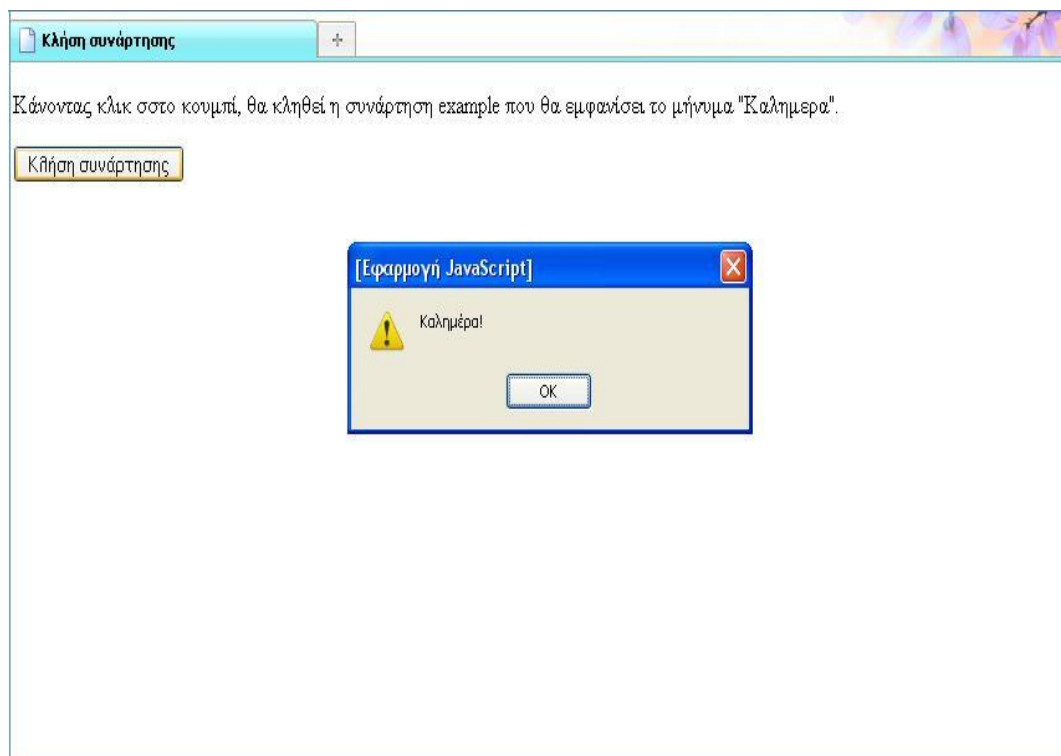
function Όνομα_Συνάρτησης (Όρισμα ή Ορίσματα)

Το όρισμα της συνάρτησης είναι προαιρετικό και παρέχει τη δυνατότητα μεταβίβασης συγκεκριμένων παραμέτρων στη συνάρτηση κατά την κλήση της. Στο ακόλουθο παράδειγμα, ορίζεται η συνάρτηση `test()`, η οποία στη συνέχεια καλείται πατώντας ένα `button` που έχει δημιουργηθεί με κώδικα HTML.

```
<html>
  <head>
    <title>Κλήσησυνάρτησης</title><script
      language="javascript">
      function test(){
        alert ("Καλημέρα!")
      }
    </script>
  </head>
  <body>
    <p>Κάνοντας κλικ στο κουμπί, θα κληθεί η συνάρτηση example που θα
      εμφανίσει το μήνυμα "Καλημερα".</p>

    <input type="button" value="Κλήσησυνάρτησης" onClick="test()">
  </body>
</html>
```

Αποθηκεύοντας το παραπάνω κείμενο ως αρχείο HTML και ανοίγοντας στη συνέχεια το αρχείο αυτό με το φυλλομετρητή, εάν πιέσουμε το button «Κλήση συνάρτησης», καλείται η συνάρτηση `example` του κώδικα JavaScript η οποία εμφανίζει ένα μήνυμα, όπως φαίνεται στο παρακάτω στιγμιότυπο:



Εικόνα 3 : Εμφάνιση μηνύματος με JavaScript

4.7. Εντολές της JavaScript

Για να εισάγουμε **σχόλια** (comments) στην JavaScript, υπάρχουν δύο τρόποι :

`//` αυτό είναι ένα σχόλιο μίας γραμμής

`/*` αυτό είναι ένα σχόλιο που μπορεί να συνεχίσει σε πολλές γραμμές και που πρέπει να ανοίξει και να κλείσει με τους ειδικούς χαρακτήρες `*/`

Με την εντολή **break** έχουμε την δυνατότητα να κάνουμε διακοπή της εκτέλεσης ενός βρόχου πριν αυτός ολοκληρωθεί και να συνεχίσουμε την εκτέλεση των εντολών με τις εντολές που ακολουθούν μετά τον βρόχο.

Ακολουθεί ένα παράδειγμα.

```
function
  myFunction()
  { var i

    var result

    for (i=1; i<20; i++) {

      result = confirm("Πατήστετο Cancel για
Break") if (!result) {

        break;

      //
end if }
// end
for

      alert("Μόλις ο βρόχος ολοκληρώθηκε, το i είναι ίσο με : " +
i) } // end function
```

Με την εντολή **continue** μπορούμε να συνεχίσουμε την εκτέλεση ενός βρόχου με την επόμενη επανάληψη και να αγνοήσουμε έτσι τις υπόλοιπες εντολές του βρόχου.

Ακολουθεί ένα παράδειγμα.

```
function
  myFunction(
) { var i = 0
var result
while
(i<20) {
  result = confirm("Πατήστετο Cancel για
Continue") if (!result) {
    continue;
  //
end
if i++
// end while
  alert("Ο βρόχος ολοκληρώθηκε, το i είναι ίσο με :
" + i) } // end function
```

Με την εντολή **break** έχουμε την δυνατότητα να κάνουμε διακοπή της εκτέλεσης ενός βρόχου πριν αυτός ολοκληρωθεί και να συνεχίσουμε την εκτέλεση των εντολών με τις εντολές που ακολουθούν μετά τον βρόχο.

Ακολουθεί ένα παράδειγμα.

```
function
  myFunction()
  { var i
    var result
    for (i=1; i<20; i++) {
      result = confirm("Πατήστετο Cancel για
Break") if (!result) {
        break;

      }//
    end if }
    // end
  for
    alert("Μόλις ο βρόχος ολοκληρώθηκε, το i είναι ίσο με :
" + i) }// end function
```

Με την εντολή **continue** μπορούμε να συνεχίσουμε την εκτέλεση ενός βρόχου με την επόμενη επανάληψη και να αγνοήσουμε έτσι τις υπόλοιπες εντολές του βρόχου.

Ακολουθεί ένα παράδειγμα.

```
function
  myFunction() { var i =
0
  var
  result
  while
  (i<20) {
    result = confirm("Πατήστετο Cancel για
Continue") if (!result) {
      continue;
    }//
  end
  if
  i++
}// end while
  alert("Ο βρόχος ολοκληρώθηκε, το i είναι ίσο
```

```
με : " + i) } // end function
```

Με την εντολή **for** μπορούμε να δημιουργήσουμε ένα βρόχο, οι εντολές που ακολουθούν εκτελούνται τόσες φορές όσες ορίζεται στη συνθήκη.

Ακολουθεί ένα παράδειγμα.

```
for (i = 0; i <= 5; i++) {  
    document.write("  
    Αριθμός : " + i)  
    document.write("  
    <br>")  
  
} // end for
```

Με την εντολή **for...in** έχουμε έναν πολύ χρήσιμο τρόπο πρόσβασης σ' όλες τις ιδιότητες ενός αντικειμένου ή σ' όλα τα στοιχεία ενός πίνακα.

Ακολουθεί ένα παράδειγμα.

```
for (i in navigator) {  
  
    document.write("Ιδιότητα : "+i+"<br>");  
    document.write("Τιμή : "+navigator[i]+  
    "<br><br>");  
  
} // end for
```

Με την εντολή **while** μπορούμε να δημιουργήσουμε ένα βρόχο που θα επαναλαμβάνεται όσο μια συνθήκη είναι αληθής. Ο έλεγχος της συνθήκης γίνεται

στην αρχή του βρόχου και έτσι υπάρχει περίπτωση να μην εκτελεστούν καθόλου οι εντολές που περιέχει ο βρόχος αν η συνθήκη είναι από την αρχή ψευδής.

Ακολουθεί ένα παράδειγμα.

```
function myFunction() { var  
    word = " "  
    while (word != "Exit") {  
        word = prompt("Γράψτε Exit για έξοδο", ""); } //  
    end while  
        alert("Ο βρόχος τελείωσε") } //  
end function
```

Με την εντολή **do...while** μπορούμε να δημιουργήσουμε ένα βρόχο που θα επαναλαμβάνεται όσο μια συνθήκη είναι αληθής. Ο έλεγχος της συνθήκης γίνεται στο τέλος του βρόχου και αφού έχουν εκτελεσθεί οι εντολές που περιέχει ο βρόχος μία τουλάχιστον φορά.

Ακολουθεί ένα παράδειγμα.

```
function myFunction() { var word  
    = " " do {  
  
        word = prompt("Γράψτε Exit για έξοδο", ""); } while  
        (word != "Exit")  
  
        alert("Ο βρόχος ολοκληρώθηκε") } // end  
function
```

Με την εντολή **if...else** γίνεται εκτέλεση μιας εντολής ή μια ομάδας εντολών αν είναι αληθής μια συνθήκη και μια άλλη εντολή ή μια άλλη ομάδα εντολών αν είναι ψευδής η ίδια συνθήκη. Η χρήση της λέξης κλειδί **else** είναι προαιρετική.

Ακολουθεί ένα παράδειγμα.

```
var d = new Date()

var time =
d.getHours() if
(time < 10) {

    document.write("<b>Καλημέρα</b>")

}

else {

    document.write("<b>Καλό Μεσημέρι</b>")

}
```

Με τον **τριαδικό τελεστή σύγκρισης** μπορούμε να αποφύγουμε τη χρήση της εντολής if ... else και να έχουμε το ίδιο αποτέλεσμα αλλά μ' έναν πιο σύντομο τρόπο.

Ακολουθεί ένα παράδειγμα.

```
var sText = (nSeconds > 30) ? "περισσότερο" : "λιγότερο"
```

Στην παραπάνω εντολή, αν η τιμή της μεταβλητής nSeconds είναι μεγαλύτερη από 30, τότε θα καταχωρηθεί η τιμή (συμβολοσειρά) "περισσότερο" στη μεταβλητή sText, ενώ αν η τιμή της μεταβλητής nSeconds είναι μικρότερη ή ίση από 30, τότε θα καταχωρηθεί η τιμή (συμβολοσειρά) "λιγότερο" στη μεταβλητή sText.

Με την εντολή **switch** μπορούμε να έχουμε ένα σύνολο από εντολές που θα εκτελεστούν ανάλογα με την τιμή μιας μεταβλητής.

Ακολουθεί ένα παράδειγμα.

```
switch (theDay) {  
case 4:document.write("ΣήμεραείναιΠέμπτη!")  
break  
  
case 6: document.write("ΣήμεραείναιΣάββατο!")  
break  
  
case 0: document.write("ΣήμεραείναιΚυριακή!")  
break  
  
default:  
  
document.write("Περιμένω το Σαββατοκύριακο!")  
  
} // end switch
```

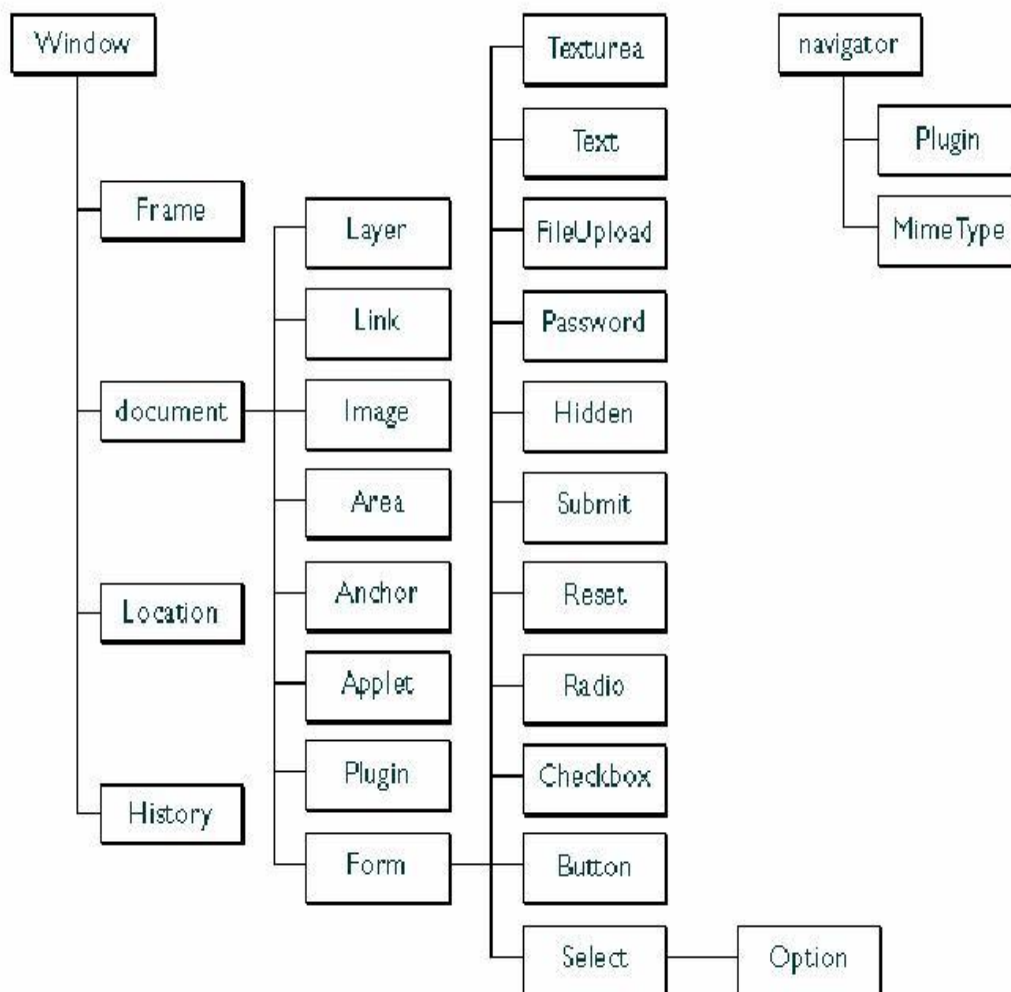
Στο παραπάνω παράδειγμα αν η τιμή της μεταβλητής theDay είναι ίση με 4 θα εμφανισθεί ένα συγκεκριμένο μήνυμα, αν είναι ίση με 6 θα εμφανισθεί ένα άλλο μήνυμα, αν είναι ίση με 0 θα εμφανισθεί ένα άλλο μήνυμα, ενώ αν έχει μια οποιαδήποτε άλλη τιμή, θα εμφανισθεί το μήνυμα που υπάρχει στο τμήμα default της εντολής switch. Η χρήση της εντολής break μέσα σε κάθε κομμάτι της εντολής switchείναι απαραίτητη αν θέλουμε να εκτελεστούν μόνο οι εντολές που ανήκουν στο συγκεκριμένο κομμάτι και όχι και οι υπόλοιπες.

Με την εντολή **return** έχουμε τη δυνατότητα να τερματίσουμε την εκτέλεση μιας συνάρτησης και να επιστρέψουμε, αν θέλουμε, μια τιμή στο όνομα της συνάρτησης. Η εντολή χρησιμοποιείται μόνο μέσα στον κώδικα των συναρτήσεων (functions).

Με την πρόταση **var** μπορούμε να δηλώσουμε μια μεταβλητή και να της αποδώσουμε και τιμή προαιρετικά. Αν και η χρήση της πρότασης var δεν είναι υποχρεωτική στη JavaScript, είναι καλό να τη χρησιμοποιούμε τουλάχιστον στις δηλώσεις των τοπικών μεταβλητών μιας συνάρτησης, για να μην υπάρξει έτσι σύγχυση με τις καθολικές μεταβλητές που δηλώνονται στο κυρίως σώμα ενός script.

4.8. Document Object Model (DOM)

Για να μπορέσουμε με τη χρήση της JavaScript (ή οποιαδήποτε άλλης γλώσσας) να κάνουμε αλλαγές σε μία HTML σελίδα θα πρέπει με κάποιο τρόπο τα διάφορα στοιχεία που αποτελούν την σελίδα να είναι προσπελάσιμα. Για αυτόν το λόγο υπάρχει το DOM, είναι δηλαδή ένα μοντέλο με βάση το οποίο κάθε στοιχείο της HTML σελίδας και ενός browser μπορεί να γίνει προσπελάσιμο σε μία γλώσσα όπως η JavaScript ως ένα αντικείμενο.



Εικόνα 4: Τα αντικείμενα του DOM

4.8.1. Αντικείμενα του DOM, μέθοδοι & ιδιότητες

Μερικά από τα πιο βασικά αντικείμενα του DOM, οι μέθοδοι και οι ιδιότητες που σχετίζονται με αυτά είναι τα παρακάτω:

Window-Το αντικείμενο `window` είναι το μεγαλύτερο αντικείμενο του DOM (top-level object) και με αυτό γίνεται αναφορά στο παράθυρο του browser. Δημιουργείται από τον φυλλομετρητή όταν φορτώνεται μια σελίδα, είναι προκαθορισμένο αντικείμενο και δεν χρειάζεται να αναφέρουμε το όνομα του παραθύρου (`window`) όταν αναφερόμαστε στα αντικείμενά του, στις ιδιότητές του ή στις μεθόδους του. Για παράδειγμα, οι επόμενες δύο γραμμές έχουν το ίδιο αποτέλεσμα, δηλαδή την εμφάνιση ενός μηνύματος στη γραμμή κατάστασης (status line) :

```
status("Hallo!")  
window.status("Hallo!")
```

Μπορούμε να ανοίξουμε ένα καινούργιο παράθυρο με τη μέθοδο `open` και να κλείσουμε με την μέθοδο `close`, για παράδειγμα :

```
NewWindow = window.open("http://www.it.tei.gr/", "height=400", "width=400")  
Window1 = window.close()
```

Επίσης μπορούμε να εμφανίσουμε μήνυμα με την μέθοδο `alert`.

```
window.alert("Καλημέρα!")
```

Document-Το αντικείμενο `document` δημιουργείται από τον browser όταν φορτώνεται μια σελίδα και περιέχει όλες πληροφορίες για το τρέχον έγγραφο, όπως είναι ο τίτλος του (title), το χρώμα φόντου (background-color) και οι φόρμες που περιέχει. Αυτές οι ιδιότητες ορίζονται μέσα στις σημάνσεις της HTML. Το αντικείμενο `document` είναι πολύ χρήσιμο διότι μέσω αυτού σχεδόν όλα τα στοιχεία μίας σελίδας είναι προσπελάσιμα.

History-Το αντικείμενο `history` περιέχει το ιστορικό επισκέψεων του browser με τις ιστοσελίδες που έχουμε επισκεφτεί νωρίτερα. Μερικές μέθοδοι είναι οι `back`, `forward` και `go`. Το παρακάτω παράδειγμα μας δείχνει τη χρήση των μεθόδων `back` και `forward` με τις οποίες έχουμε τη δυνατότητα να μετακινηθούμε στην προηγούμενη-επόμενη σελίδα.

```
window.history.back()  
window.history.forward()
```

Screen-Το αντικείμενο `screen` μας δίνει τη δυνατότητα να αναφερθούμε και να ανακτήσουμε τις ιδιότητες της οθόνης. Για παράδειγμα το παρακάτω κομμάτι κώδικα μας εμφανίζει το πλάτος της οθόνης.


```
<script type="text/javascript">

    document.write("Total
Width: " + screen.width); </script>
```

Form- Το αντικείμενο `form` είναι ιδιότητα του αντικειμένου `document`. Κάθε φόρμα σε ένα έγγραφο (`document`) είναι και ένα ξεχωριστό αντικείμενο (`object`) και μέσω κώδικα JavaScript μπορούμε να προσπελάσουμε τα στοιχεία της φόρμας. Το αντικείμενο `form` είναι ένας πίνακας (`array`) που δημιουργείται καθώς ορίζονται οι φόρμες (`forms`) μέσω των HTML tags. Αν η πρώτη φόρμα σ' ένα έγγραφο ονομασθεί `form1`, τότε θα μπορούμε να αναφερόμαστε σ' αυτήν ως `document.form1` ή και ως `document.forms[0]`. Αποτελεί ιδιότητα του αντικειμένου `document` και σχετικές ιδιότητες είναι οι `action`, `elements`, `encoding`, `forms`, `method`, `name` και `target`. Σχετική μέθοδος είναι η `submit` και σχετικό χειριστήριο συμβάντος είναι το `onSubmit`.

Navigator- Το αντικείμενο αυτό περιέχει πληροφορίες για την τρέχουσα έκδοση του `browser` που χρησιμοποιεί ο πελάτης. Σχετικές ιδιότητες είναι οι `appName`, `appCodeName`, `appVersion` και `userAgent`. Για παράδειγμα:

```
<script type="text/javascript">

    document.write("appName: " + navigator.appName);

    document.write("appVersion: " + navigator.appVersion);

    document.write("userAgent: " + navigator.userAgent);

</script>
```

Location - Το αντικείμενο `location` περιέχει πλήρεις πληροφορίες URL για το τρέχον έγγραφο (`document`), ενώ η κάθε ιδιότητά του αναφέρεται σ' ένα διαφορετικό κομμάτι του URL. Αποτελεί ιδιότητα του αντικειμένου `document`. Μερικές ιδιότητες είναι οι `hash`, `host`, `hostname`, `href`, `location`, `pathname`, `port`, `protocol`, `search` και `target`. Με την χρήση της ιδιότητας `href` μπορούμε να αλλάξουμε τη τρέχουσα σελίδα, δηλαδή να μεταφερθούμε σε μία νέα σελίδα.

```
window.location.href = "http://www.it.teicrete.gr"
```

String-Είναι μια σειρά από χαρακτήρες (characters) που ορίζεται με διπλά ή απλά εισαγωγικά, όπως για παράδειγμα :

```
firstname = "Nikos"  
lastname = "Papadopoulos"
```

Η παραπάνω εντολή επιστρέφει δύο αντικείμενα string με όνομα firstname και lastname με τιμές "Nikos", "Papadopoulos". Μπορούμε να χρησιμοποιήσουμε μεθόδους που να επιστρέφουν ένα κομμάτι του string, όπως για παράδειγμα η μέθοδος `firstname.toUpperCase()` που επιστρέφει την τιμή "NIKOS". Περιέχει επίσης μεθόδους που επιστρέφουν HTML παραλλαγές του string, όπως σε μορφή bold και italics. Σχετική ιδιότητα είναι η `length` και σχετικές μέθοδοι είναι οι `anchor`, `big`, `blink`, `bold`, `charAt`, `fixed`, `fontcolor`, `fontsize`, `indexOf`, `italics`, `lastIndexOf`, `link`, `small`, `strike`, `sub`, `substring`, `sup`, `toLowerCase` και `toUpperCase`.

Math-Το αντικείμενο αυτό περιέχει ιδιότητες για μαθηματικές σταθερές και μεθόδους που χρησιμοποιούνται σε συναρτήσεις. Με αυτό το αντικείμενο μπορούμε να κάνουμε μαθηματικούς υπολογισμούς. Για παράδειγμα, η δήλωση `Math.PI` αντιπροσωπεύει την τιμή του π σε μια εξίσωση ενώ με την μέθοδο `sin()` μπορούμε να υπολογίσουμε τα ημίτονα. Σχετικές ιδιότητες είναι οι `E`, `LN10`, `LN2`, `PI`, `SQRT1_2`, `SQRT2` και σχετικές μέθοδοι είναι οι `abs`, `acos`, `asin`, `atan`, `ceil`, `cos`, `exp`, `floor`, `log`, `max`, `min`, `pow`, `random`, `round`, `sin`, `sqrt` και `tan`.

Date-Αντικαθιστά έναν κανονικό τύπο ημερομηνίας (date type). Αν και δεν έχει καθόλου ιδιότητες (properties), διαθέτει πολλές μεθόδους (methods). Οι μέθοδοι για να λαμβάνουμε και να ορίζουμε πληροφορίες για την ημερομηνία και την ώρα χωρίζονται σε τέσσερις κατηγορίες : `set`, `get`, `to` και `parse/UTC`. Εκτός από τις ημέρες του μήνα, όλες οι άλλες αριθμητικές εκφράσεις ημερομηνιών ξεκινούν από το 0. Οι μήνες ξεκινούν από το 0 (Ιανουάριος) έως και το 11 (Δεκέμβριος). Σχετικές μέθοδοι είναι οι `getDate`, `getDay`, `getHours`, `getMinutes`, `getMonth`, `getSeconds`, `getTime`, `getTimezoneOffset`, `getYear`, `parse`, `setDate`, `setHours`, `setMinutes`, `setMonth`, `setSeconds`, `setTime`, `setYear`, `toGMTString`, `toLocaleString` και `toString`.

Button-Είναι ένα αντικείμενο που αποτελεί στοιχείο μιας φόρμας και μπορεί να χρησιμοποιηθεί για να εκτελεσθεί μια ενέργεια (action). Αποτελεί ιδιότητα του αντικειμένου `form`. Σχετικές ιδιότητες είναι η `name` και η `value`, σχετική μέθοδος είναι η `click` και σχετικό χειριστήριο συμβάντος είναι το `onClick`.

4.8.2. Δημιουργία Αντικειμένου

Για να δημιουργήσουμε ένα δικό μας αντικείμενο, υπάρχουν οι εξής λέξεις κλειδιά :

- **new**, δημιουργούμε ένα καινούργιο αντικείμενο.
- **this**, αναφερόμαστε στο τρέχον αντικείμενο.
- **with**, κάνουμε προεπιλεγμένο ένα αντικείμενο σε μια ομάδα εντολών για να μην είμαστε αναγκασμένοι να το γράφουμε συνέχεια.

Για να μπορέσουμε να δημιουργήσουμε ένα δικό μας αντικείμενο, γράφουμε πρώτα μια συνάρτηση δημιουργίας αντικειμένων (constructor), με την οποία δίνουμε τιμές στις ιδιότητες του αντικειμένου με τη λέξη κλειδί this, ως εξής :

```
function FullName (firstname, lastname) {  
  
    this.firstname = firstname;  
    this.lastname = lastname; }  

```

Μετά, για να μπορέσουμε να δημιουργήσουμε ένα στιγμιότυπο (instance) αυτού του αντικειμένου, χρησιμοποιούμε λέξη κλειδί new, ως εξής :

```
John01 = new FullName("Nikos", "Papadopoulos");
```

4.8.3. Τροποποίηση Αντικειμένου

Με τη λέξη κλειδί prototype μπορούμε να προσθέσουμε κι άλλες ιδιότητες στο πρωτότυπο ενός αντικειμένου, δηλ. στον ορισμό του, και έτσι όλα τα αντικείμενα αυτού του τύπου που θα δημιουργήσουμε θα περιλαμβάνουν τις ιδιότητες αυτές.

Για παράδειγμα, για να προσθέσουμε την ιδιότητα fathersname στο αντικείμενο FullName που δημιουργήσαμε προηγουμένως, γράφουμε την εξής εντολή :

```
FullName.prototype.fathersname = "George";
```

5. Apache, MySQL & PHP

5.1. Εισαγωγή

Προκειμένου να δημιουργήσουμε ένα database-driven Website, χρησιμοποιούμε μια γλώσσα συγγραφής σεναρίων στην πλευρά του διακομιστή, μια σχεσιακή βάση δεδομένων και έναν διακομιστή. Στην εφαρμογή χρησιμοποιήθηκε ως διακομιστής ο Apache HTTP Web Server, ως σχεσιακή βάση δεδομένων η MySQL και τέλος η PHP ως η γλώσσα συγγραφής σεναρίων από την πλευρά του διακομιστή.

5.2. Apache HTTP Web server

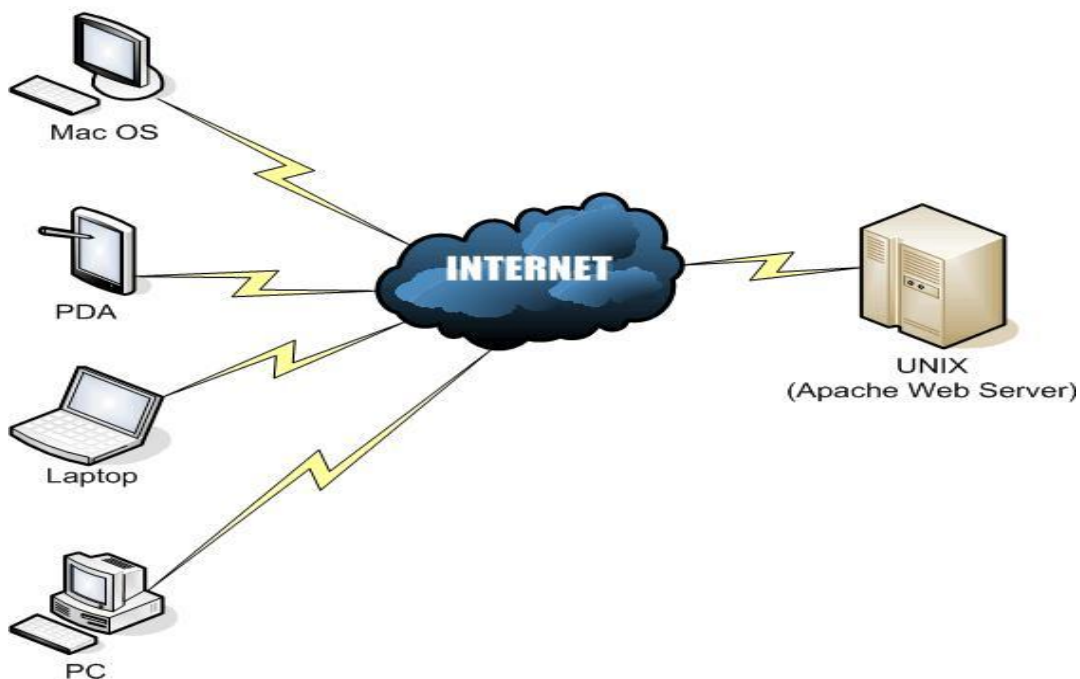
5.2.1. Τι είναι ένας Web Server

Κεντρικός υπολογιστής ή αλλιώς εξυπηρετητής ή **διακομιστής** (Web server) είναι ο υπολογιστής που ρόλος του είναι να παρέχει διάφορες υπηρεσίες σε υπολογιστές πελατών στον παγκόσμιο Ιστό. Είναι πιο αναβαθμισμένος όσον αφορά το λογισμικό και υλικό του εξοπλισμό, σε σχέση με κοινούς υπολογιστές.

Τα κύρια χαρακτηριστικά του είναι:

1. Υποστηρίζει επεξεργαστές που χρησιμοποιούνται για την επεξεργασία των πολλαπλών δεδομένων που δέχεται.
2. Υποστηρίζει μεγάλης ταχύτητας σκληρούς δίσκους
3. Υποστηρίζει ταχύτερες μνήμες.

Οι κεντρικοί υπολογιστές συνήθως συνδέονται με συστήματα διπλής τροφοδοσίας ή παροχή αδιάλειπτης ενέργειας (UPS) για μεγαλύτερη αξιοπιστία και σιγουριά στις παρερχόμενες υπηρεσίες.



Εικόνα 5- Η λειτουργία ενός webserver

5.2.2. Hypertext Transfer Protocol (HTTP)

Το πρωτόκολλο HTTP καθορίζει τον τρόπο επικοινωνίας στο διαδίκτυο μεταξύ των διακομιστών και των πελατών (servers - clients). Είναι ένα γενικό, αντικειμενοστραφές πρωτόκολλο που μεταβιβάζει πληροφορία μεταξύ των διακομιστών και των πελατών. Ξεκίνησε από την έκδοση HTTP/0.9 κατά τη διάρκεια της πρώιμης ανάπτυξης του διαδικτύου και ακολούθησε η έκδοση HTTP/1.0 το 1995. Η πιο πρόσφατη έκδοσή του, HTTP/1.1, παρέχει περισσότερη λειτουργικότητα και υποστηρίζει πολλαπλές συναλλαγές μεταξύ πελάτη και διακομιστή κατά τη διάρκεια της ίδιας αίτησης.

5.2.3. Apache HTTP Web server

Ο Apache HTTP γνωστός και απλά σαν Apache είναι ένας εξυπηρετητής του παγκόσμιου ιστού (web). Όποτε ένας χρήστης επισκέπτεται έναν ιστότοπο το πρόγραμμα πλοήγησης (browser) επικοινωνεί με ένα διακομιστή (server) μέσω του πρωτοκόλλου HTTP, ο οποίος παράγει τις ιστοσελίδες και τις αποστέλλει στο πρόγραμμα πλοήγησης. Ο Apache είναι ένας από τους δημοφιλέστερους, εν μέρει γιατί λειτουργεί σε διάφορες πλατφόρμες όπως τα Windows, το Linux, το Unix και το MacOSX. Συντηρείται τώρα από μια κοινότητα ανοικτού κώδικα με επιτήρηση από το Ίδρυμα Λογισμικού Apache (ApacheSoftwareFoundation).

Ο Apache χρησιμοποιείται και σε τοπικά δίκτυα σαν διακομιστής συνεργαζόμενος με συστήματα διαχείρισης Βάσης Δεδομένων π.χ. Oracle, MySQL.

Η πρώτη του έκδοση, γνωστή ως NCSAHTTPd, δημιουργήθηκε από τον RobertMcCool και κυκλοφόρησε το 1993. Θεωρείται ότι έπαιξε σημαντικό ρόλο στην αρχική επέκταση του παγκόσμιου ιστού. Ήταν η πρώτη βιώσιμη εναλλακτική επιλογή που παρουσιάστηκε απέναντι στον εξυπηρετητή http της εταιρείας Netscape και από τότε έχει εξελιχθεί στο σημείο να ανταγωνίζεται άλλους εξυπηρετητές βασισμένους στο Unix σε λειτουργικότητα και απόδοση. Από το 1996 ήταν από τους πιο δημοφιλείς όμως από τον Μάρτιο του 2006 έχει μειωθεί το ποσοστό της εγκατάστασής του κυρίως από τον MicrosoftInternetInformationServices και την πλατφόρμα .NET . Τον Οκτώβριο του 2007 το μερίδιο του ήταν 47.73% από όλους τους ιστοτόπους.

5.3. MySQL

5.3.1. Είδη βάσεων δεδομένων

Υπάρχουν δύο ειδών βάσεων δεδομένων:

- **Αντικειμενοστραφείς βάσεις δεδομένων**

Πολύπλοκες δομές δεδομένων, οι οποίες όταν αναπαρασταθούν γραφικά παρομοιάζονται σαν δίκτυα ή σαν οντολογίες.

- **Σχεσιακές βάσεις δεδομένων**

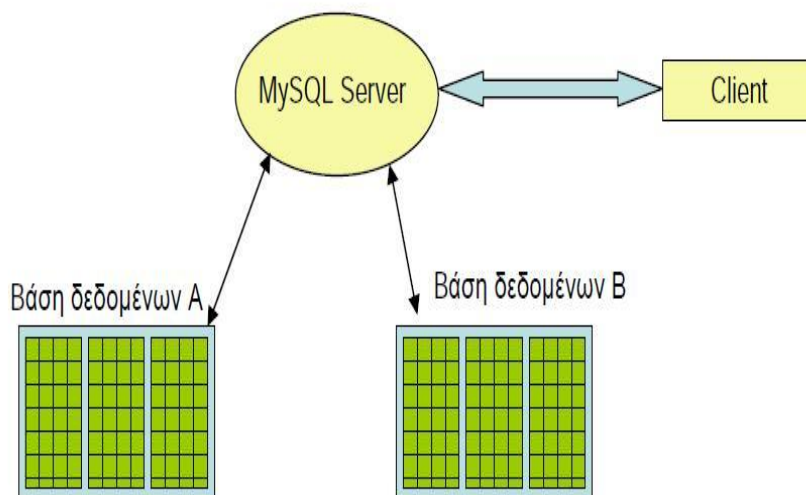
Οι δομές δεδομένων αναπαριστώνται με δισδιάστατους πίνακες. Δύο ή περισσότεροι πίνακες μπορούν να συσχετισθούν μεταξύ τους ώστε να προκύψει μια νέα δομή δεδομένων.

Η MySQL είναι ένα **σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων** (relational database management system) ή RDBMS. Με τη χρήση της MySQL είναι εύκολη η πρόσβαση σ' αυτές τις πληροφορίες χρησιμοποιώντας μια γλώσσα συγγραφής σεναρίων στην πλευρά του διακομιστή (server-side scripting languages), όπως είναι η PHP.

Η MySQL αναπτύσσεται, διανέμεται, και υποστηρίζεται από την εταιρεία MySQL AB, η οποία είναι μια εμπορική επιχείρηση ανάπτυξης ανοικτού λογισμικού. Ανοικτό λογισμικό, σημαίνει ότι είναι δυνατό για οποιονδήποτε να χρησιμοποιήσει και να τροποποιήσει το λογισμικό, χωρίς καμία οικονομική επιβάρυνση. Ο διακομιστής MySQL ελέγχει την πρόσβαση στα δεδομένα μας για να διασφαλίσει ότι πολλοί χρήστες θα μπορούν να δουλεύουν ταυτόχρονα, για να παρέχει γρήγορη πρόσβαση και για να διασφαλίσει ότι μόνο οι πιστοποιημένοι χρήστες θα μπορούν να έχουν πρόσβαση.

5.3.2. Χαρακτηριστικά της MySQL

Η MySQL λειτουργεί με το μοντέλο Πελάτη/Εξυπηρετητή (Client/Server). Ο εξυπηρετητής δέχεται αιτήματα από τους πελάτες σχετικά με τη διαχείριση μίας ή περισσότερων βάσεων δεδομένων και προβαίνει στις απαιτούμενες ενέργειες, όπως φαίνεται παρακάτω.



Εικόνα 6 : Το μοντέλο Πελάτη/Εξυπηρετητή

Οι κύριες λειτουργίες διαχείρισης βάσεων δεδομένων τις οποίες υλοποιεί ο εξυπηρετητής μετά από αίτηση του πελάτη, δε διαφέρουν σημαντικά από τις αντίστοιχες λειτουργίες των άλλων εφαρμογών βάσεων δεδομένων και περιλαμβάνουν τη δημιουργία-διαγραφή βάσεων δεδομένων, την εισαγωγή τροποποίηση-διαγραφή πινάκων (tables) και πεδίων (fields), την εισαγωγή-τροποποίηση-διαγραφή εγγραφών (records) και τέλος την ανάκτηση δεδομένων από τη βάση με τη χρήση συγκεκριμένων κριτηρίων.

Για την υλοποίηση των παραπάνω λειτουργιών, η MySQL χρησιμοποιεί την SQL (Δομημένη Γλώσσα Ερωτήσεων – StructuredQueryLanguage). ΗSQL είναι η ευρύτερα χρησιμοποιούμενη τυποποιημένη γλώσσα πρόσβασης στις βάσεις δεδομένων και αποτελείται από εντολές οι οποίες επιτρέπουν την ανάκτηση και ενημέρωση δεδομένων σε μια βάση. Εκτός από τη MySQL, η γλώσσα SQL συνεργάζεται με άλλα προγράμματα βάσεων δεδομένων όπως είναι η Access, MicrosoftSQLServer, Oracle, Sybase και κ.α.

5.3.3. Οι βάσεις δεδομένων της MySQL

Μια βάση δεδομένων της MySQL αποτελείται από έναν ή περισσότερους πίνακες, ο καθένας από τους οποίους περιέχει μια λίστα από δεδομένα. Αν, για παράδειγμα, θελήσουμε να δημιουργήσουμε μια βάση δεδομένων για να αποθηκεύσουμε έναν τηλεφωνικό κατάλογο, θα μπορούσαμε να κατασκευάσουμε έναν πίνακα με το όνομα *ΕΠΑΦΕΣ*. Η μορφή που θα μπορούσε να έχει αυτός ο πίνακας φαίνεται παρακάτω:

Επαφές

id	Όνομα	Επώνυμο	Τηλέφωνο
1	Γιάννης	Παπαδόπουλος	555555
2	Βούλα	Ζαφειρίου	222222

Πίνακας 6: Μορφή πίνακα βάσης δεδομένων

5.3.4. Τύποι δεδομένων

Κάθε πίνακας αποτελείται από πεδία, τα οποία αντιπροσωπεύουν τα γνωρίσματα. Για κάθε πεδίο πρέπει να οριστεί ο τύπος τους. Θα πρέπει πάντα να χρησιμοποιείται ο τύπος δεδομένων που χρειάζεται για να αποθηκευτούν οι τιμές που μπορεί να πάρει το γνώρισμα. Επίσης πρέπει να ορίζεται και το μέγεθος του πεδίου, σύμφωνα με τις ανάγκες.

Αριθμητικοί

- **int** - ακέραιος μήκους 11 ψηφίων
- **tinyint** - ακέραιος μήκους 4 ψηφίων
- **smallint** - ακέραιος μήκους 5 ψηφίων
- **mediumint** - ακέραιος μήκους 9 ψηφίων
- **bigint** - ακέραιος μήκους 20 ψηφίων
- **float(m,d)** - κινητής υποδιαστολής ψηφίων και δεκαδικών (μέχρι 24), είναι πάντα προσημασμένος
- **double(m,d)** - διπλής ακρίβειας ψηφίων και δεκαδικών (μέχρι 53), είναι πάντα προσημασμένος

Ημερομηνίες & Ώρες

- **date** - Τιμή ημερομηνίας με μορφή:EEEE-MM-HH
- **datetime** - Ημερομηνία και ώρα με μορφή: EEEE-MM-HH ΩΩ:ΛΛ:
- **time** - Ώρα με μορφή:ΩΩ:ΛΛ:ΔΔ
- **year(m)** - Αποθηκεύει έτος σε διψήφια ή τετραψήφια μορφή. Στη διψήφια μορφή περιλαμβάνονται τα έτη από 1970-2069. Στην τετραψήφια από 1901-2155
- **timestamp** - Χρονική ένδειξη με μορφές:
14(EEEEMMHHΩΩΛΛΔΔ) 12(EEMMHHΩΩΛΛΔΔ) 8(EEEEMMHH) 6(EEMMHH)

Αλφαριθμητικά

- **char(m)** - Αλφαριθμητικό σταθερού μεγέθους μεταξύ 1-255 χαρακτήρες. Αν η τιμή δεν φτάνει το μέγεθος συμπληρώνεται με κενά ως το προσδιοριζόμενο μήκος.
- **varchar(ma)** - Αλφαριθμητικό μεταβλητού μεγέθους, από 1-255 χαρακτήρες.
- **text** - Πεδίο κειμένου με μέγιστο μήκος 65535
- **tinytext** - Πεδίο κειμένου με μέγιστο μήκος 255
- **mediumtext** - Πεδίο κειμένου με μέγιστο μήκος 16777215
- **longtext** - Πεδίο κειμένου με μέγιστο μήκος 4294967295

5.4 PHP

Η PHP είναι μια γλώσσα προγραμματισμού, που σχεδιάστηκε με σκοπό τη δημιουργία δυναμικών σελίδων και είναι επισήμως γνωστή ως: HyperText preprocessor. Η PHP είναι μια γλώσσα προγραμματισμού ειδικά για την κατασκευή δυναμικών ιστοσελίδων. Με τον όρο δυναμική εννοείται μια ιστοσελίδα, που αλλάζει αυτόματα, ανάλογα με τα στοιχεία του χρήστη όπως το λειτουργικό του σύστημα, η διεύθυνση ip του κ.ά. Ανταγωνιστικές ως προς την PHP γλώσσες προγραμματισμού είναι οι εξής: ASP (Active Server Pages) της εταιρείας Microsoft, CFML (ColdFusion Markup Language) της εταιρείας Allaire και JSP (JavaServer Pages) της εταιρείας Sun.

5.4.1. Εξέλιξη της PHP μέχρι σήμερα

Η PHP δημιουργήθηκε το 1994 από έναν φοιτητή, τον Rasmus Lerdorf ο οποίος δημιούργησε χρησιμοποιώντας τη γλώσσα προγραμματισμού Perl ένα απλό script με όνομα PHP.cgi και το ενσωμάτωσε στην προσωπική του ιστοσελίδα. Το script αυτό είχε σαν σκοπό να διατηρεί μια λίστα στατιστικών για τα άτομα που έβλεπαν το online βιογραφικό του σημείωμα. Αργότερα αυτό το script το διέθεσε και σε φίλους του, οι οποίοι άρχισαν να του ζητούν να προσθέσει περισσότερες δυνατότητες. Η γλώσσα τότε ονομαζόταν PHP/FI από τα αρχικά Personal Home Page/Form Interpreter και το κύριο χαρακτηριστικό αυτής της έκδοσης ήταν ότι ο PHP/FI parser ήταν γραμμένος κυρίως με το χέρι και έτσι προέκυπταν συχνά λάθη. Ο όρος parser (αναλυτής) αναφέρεται στον μηχανισμό ο οποίος δέχεται ένα script και το μετατρέπει σε κάτι που μπορεί να κατανοήσει ο υπολογιστής.

Το 1997 η PHP/FI έφθασε στην έκδοση 2.0, βασιζόμενη αυτή τη φορά στη γλώσσα C και αριθμώντας περισσότερους από 50.000 ιστότοπους που τη χρησιμοποιούσαν, ενώ αργότερα την ίδια χρονιά οι Andi Gutmans και Zeev Suraski ξαναέγραψαν τη γλώσσα από την αρχή, βασιζόμενοι όμως αρκετά στην PHP/FI 2.0. Έτσι η PHP έφθασε στην έκδοση 3.0 η οποία θύμιζε περισσότερο τη σημερινή μορφή της. Στη συνέχεια, οι Zeev και Andi δημιούργησαν την εταιρεία Zend (από τα αρχικά των ονομάτων τους), η οποία συνεχίζει μέχρι και σήμερα την ανάπτυξη και εξέλιξη της γλώσσας PHP. Ακολούθησε το 1998 η έκδοση 4 της PHP, τον Ιούλιο του 2004 διατέθηκε η έκδοση 5, ενώ αυτή τη στιγμή έχουν ήδη διατεθεί και οι πρώτες δοκιμαστικές εκδόσεις της επερχόμενης PHP 6, για οποιονδήποτε προγραμματιστή θέλει να τη χρησιμοποιήσει. Οι περισσότεροι ιστότοποι επί του παρόντος χρησιμοποιούν κυρίως τις εκδόσεις 4 και 5 της PHP.

5.4.2. Ενσωμάτωση της PHP

Οι περισσότερες PHP σελίδες έχουν PHP και HTML κώδικα μαζί. Όταν ο server διαβάζει μια PHP σελίδα ψάχνει να δει από πού θα αρχίσει να διαβάζει την σελίδα ως PHP και από πού ως HTML. Η σελίδα διαβάζεται ως PHP όταν συναντήσει τις σημάνσεις `<? PHP και ?>`. Το μεγαλύτερο μέρος της σύνταξής της, η PHP το έχει δανειστεί από την C, την Java και την Perl και διαθέτει και μερικά δικά της μοναδικά χαρακτηριστικά. Ο σκοπός της γλώσσας είναι να δώσει τη δυνατότητα στους web developers να δημιουργούν δυναμικά παραγόμενες ιστοσελίδες.

Ένα απλό παράδειγμα που δείχνει την εισαγωγή της PHP μέσα στην HTML είναι το παρακάτω:

```
<html>

<head>
  <title>Παράδειγμα</title>
</head>
<body>
  <?PHP echo "Καλημέρα!"; ?>
</body>
</html>
```

Γράφουμε ένα HTML script με κάποιον ενσωματωμένο κώδικα για να κάνει κάτι, όπως στην συγκεκριμένη περίπτωση να εμφανίσει κάποιο κείμενο. Ο κώδικας της PHP περικλείεται με ειδικές σημάνσεις αρχής και τέλους. Αυτό που ξεχωρίζει την PHP από μια γλώσσα όπως η JavaScript, η οποία εκτελείται στην πλευρά του χρήστη (client-side), είναι ότι ο κώδικας της εκτελείται στον server. Αυτό σημαίνει με απλά λόγια ότι οι τελικοί χρήστες δεν μπορούν ποτέ να δουν τον πηγαίο κώδικα (source code) της PHP.

Τα scripts της PHP αποθηκεύονται με την κατάληξη .PHP και κάθε φορά που ο Web server πρέπει να στείλει ένα αρχείο που τελειώνει σε .PHP, πρώτα το στέλνει στον διερμηνευτή (interpreter) της PHP, ο οποίος εκτελεί τον κώδικα της PHP που υπάρχει στο script πριν επιστρέψει το παραγόμενο αρχείο στον τελικό χρήστη.

5.4.3. PHPMyAdmin

Η PHPMyAdmin είναι ένα εργαλείο με γραφικό περιβάλλον που αποτελείται από ένα σύνολο PHP scripts με το οποίο διαχειριζόμαστε τις βάσεις δεδομένων που έχουμε μέσω web. Η PHPMyAdmin μπορεί να διαχειριστεί ένα ολόκληρο mysql server ή ακόμα και απλές βάσεις δεδομένων όπου ο κάθε χρήστης έχει ένα λογαριασμό και μπορεί να δημιουργήσει και να διαχειριστεί τις δικές του βάσεις

δεδομένων.

Ότι δηλαδή μπορεί να γίνει από την γραμμή εντολών της MySQL, μπορεί να γίνει πολύ πιο εύκολα από την PHPMyAdmin.

Συνοπτικά, με την PHPMyAdmin μπορούμε:

- να δημιουργήσουμε και να διαγράψουμε βάσεις δεδομένων
- να δημιουργήσουμε, να διαγράψουμε και να χειριστούμε πίνακες
- να διαγράψουμε, να προσθέσουμε και να επεξεργαστούμε πεδία πινάκων
- να εκτελέσουμε ερωτήματα και να εξάγουμε τα αποτελέσματα σε οποιαδήποτε μορφή (text, sql, κτλ)
- να εξαγάγουμε (back-up) πεδία, πίνακες αλλά και ολόκληρες βάσεις δεδομένων

6. ΤΕΧΝΟΛΟΓΙΕΣ ΑΝΑΠΤΥΞΗΣ

ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Οι τεχνολογίες που χρησιμοποιήθηκαν για το σκοπό της παρούσας πτυχιακής εργασίας είναι κυρίως το λογισμικό ανάπτυξης **xampp**, το **DreamweaverCS3** της Adobe για εφαρμογή επεξεργασίας γραφικών για την σχεδίαση του template και το πρόγραμμα **Notepad++** για τη συγγραφή του κώδικα της ιστοσελίδας μας. Παρακάτω παρουσιάζονται οι περιγραφές των τεχνολογιών αυτών .

6.1 XAMPP

Το λογισμικό xampp έχει πάρει την ονομασία του από x (σημαίνει λογισμικό ανεξάρτητο από πλατφόρμα (cross-platform)), a (apache http server), m (mysql), p (php), p (perl). Το xampp είναι ένα ελεύθερο λογισμικό. Η δυσκολία εγκατάστασης του διακομιστή web apache μαζί με την mysql και php κάνει το xampp προτιμότερο λόγω της απλούστευσης της διαδικασίας. Το xampp είναι πολύ εύκολο στην εγκατάσταση και στη χρήση του. Το xampp μπορεί να λειτουργήσει στα λειτουργικά windows, mac και linux, ενώ πολλοί το χρησιμοποιούν μαζί με τα εξής προγράμματα wordpress, joomla και drupal. Ένα συστατικό του xampp είναι το phpmyadmin που επιτρέπει τη διαχείριση της βάσης δεδομένων από το πρόγραμμα περιήγησης (browser). Το phpmyadmin μας δίνει πλήρη πρόσβαση και την δυνατότητα τροποποίησης της βάσης.

Το xampp περιέχει πολλούς φακέλους. Ο φάκελος htdocs περιέχει όλες τις αποθηκευμένες ιστοσελίδες που θέλουμε να εμφανίζονται στον ιστοτοπό μας. Ο φάκελος mysql περιέχει όλα τα δεδομένα που έχουμε δημιουργήσει με την βοήθεια του phpmyadmin. Μπορούμε να αποθηκεύσουμε δεδομένα και να διαχειριστούμε τη βάση δεδομένων και με άλλους τρόπους όπως με το mysql workbench. Εάν θέλουμε να τρέξουμε ένα αρχείο τότε πληκτρολογούμε localhost ή 127.0.0.1 και την ανάλογη διαδρομή φακέλων στον φάκελο της εφαρμογής που βρίσκεται το αρχείο καθώς και το όνομα του αρχείου.

Το xampp μπορεί να χρησιμοποιηθεί σε οποιοδήποτε λειτουργικό σύστημα εμείς επιθυμούμε. Οι ιστοσελίδες του xampp μπορούν να τρέξουν σε όλους τους φυλλομετρητές (browser). Μπορεί να δουλέψει επίσης με cms (content management system) όπως τα joomla, drupal, wordpress.

6.2 DREAMWEAVER

Το πρόγραμμα *DreamWeaver* της εταιρείας *Macromedia* είναι ένα κορυφαίο πρόγραμμα δημιουργίας και επεξεργασίας ιστοσελίδων, δηλαδή κώδικα HTML, που είναι ιδιαίτερα εύκολο και φιλικό στη χρήση του. Το όνομα *DreamWeaver* προέρχεται από ένα παλιό ρομαντικό τραγούδι. Το *DreamWeaver* είναι εξαιρετικό για να μπορούμε να δημιουργήσουμε στα γρήγορα φόρμες (forms), πλαίσια (frames), πίνακες (tables) και άλλα αντικείμενα της HTML. Είναι, όμως, ιδιαίτερα καλό όταν θέλουμε να δώσουμε σε μια ιστοσελίδα τη δυνατότητα να κάνει κάτι. Πρέπει να έχουμε υπόψη μας ότι το *DreamWeaver* μπορεί να χρησιμοποιηθεί και για τη δημιουργία εφαρμογών πολυμέσων.

Το *DreamWeaver* έχει δυνατότητες για δημιουργία δυναμικής HTML (DHMTL) και επιτρέπει κίνηση γραμμής χρόνου, απόλυτη τοποθέτηση περιεχομένων, δημιουργία επιπέδων (layers) και συγγραφή σεναρίων (scripts). Το *DreamWeaver* περιέχει δικές του συμπεριφορές (behaviors), που είναι έτοιμα scripts τα οποία μπορούμε να προσθέσουμε πολύ εύκολα σ' ένα αντικείμενο.

Το *DreamWeaver* μάς παρέχει την ελευθερία να σχεδιάσουμε οπτικά την εμφάνιση μιας ιστοσελίδας και τη δύναμη να την κάνουμε να λειτουργεί όπως ακριβώς θέλουμε. Μπορούμε να δημιουργήσουμε τη δική μας προσωπική ιστοσελίδα (personal web page) ή μια ολόκληρη περιοχή (web site) σ' ένα εταιρικό δίκτυο (intranet).

6.3 NOTEPAD++

Πρόκειται για τον πιο γνωστό επεξεργαστή κειμένου (text editor). Είναι το αγαπημένο εργαλείο συγγραφής κώδικα των προγραμματιστών, καθώς αποτελεί ένα ελαφρύ πρόγραμμα το οποίο όμως έχει πάρα πολλές δυνατότητες.

Περιβάλλον εργασίας:

Το περιβάλλον εργασίας είναι εξαιρετικά απλό και λειτουργικό. Στο επάνω μέρος του υπάρχει μία εργαλειοθήκη με τις κυριότερες λειτουργίες. Όλο το υπόλοιπο μέρος του καταλαμβάνεται από τον χώρο συγγραφής του κώδικα. Γύρω από τον χώρο αυτόν, μπορούν να εμφανιστούν διάφορα πλαίσια τα οποία παρουσιάζουν διάφορες πληροφορίες ανάλογα με τα plugins από τα οποία προέρχονται.

Χαρακτηριστικά και λειτουργίες:

Το σημαντικό του χαρακτηριστικό είναι το λεγόμενο syntax highlighting, το οποίο χρωματίζει τον κώδικα ανάλογα με την δομή του και διευκολύνει την ανάπτυξη του. Για κάθε γλώσσα προγραμματισμού υπάρχει διαφορετικό χρωματικό στυλ. Το πρόγραμμα έχει έτοιμα στυλ για πολλές γλώσσες και δίνει την δυνατότητα να τα ρυθμίσετε σύμφωνα με τις ανάγκες σας ή να φτιάξετε καινούργια.

Μπορείτε να επεξεργαστείτε πολλά έγγραφα συγχρόνως τα οποία ανοίγουν σε ξεχωριστές καρτέλες έτσι ώστε να επεξεργάζεστε τα αρχεία ενός ολόκληρου project. Επιπλέον μπορείτε να βλέπετε και να επεξεργάζεστε δύο έγγραφα συγχρόνως χάρη στο χαρακτηριστικό multi view.

Πολύ δυνατό χαρακτηριστικό του προγράμματος είναι οι εκτεταμένες δυνατότητες εύρεσης κειμένου στο αρχείο που δουλεύετε ή στα αρχεία ενός ολόκληρου φακέλου. Εκτός από τον κλασικό τρόπο έρευνας μπορεί να χρησιμοποιήσει και regular expressions για ακόμα πιο ακριβή αποτελέσματα.

Το πρόγραμμα υποστηρίζει την χρήση plugins, τα οποία επεκτείνουν τις δυνατότητες του. Υπάρχουν πολλά τέτοια plugins για να κατεβάσετε και να στήσετε το πρόγραμμα όπως σας βολεύει.

Επιπλέον προσφέρει την δυνατότητα καταγραφής και εκτέλεσης μακροεντολών, ώστε να αυτοματοποιήσετε τις εργασίες που εκτελείτε συχνά.

Πληροφορίες εγκατάστασης:

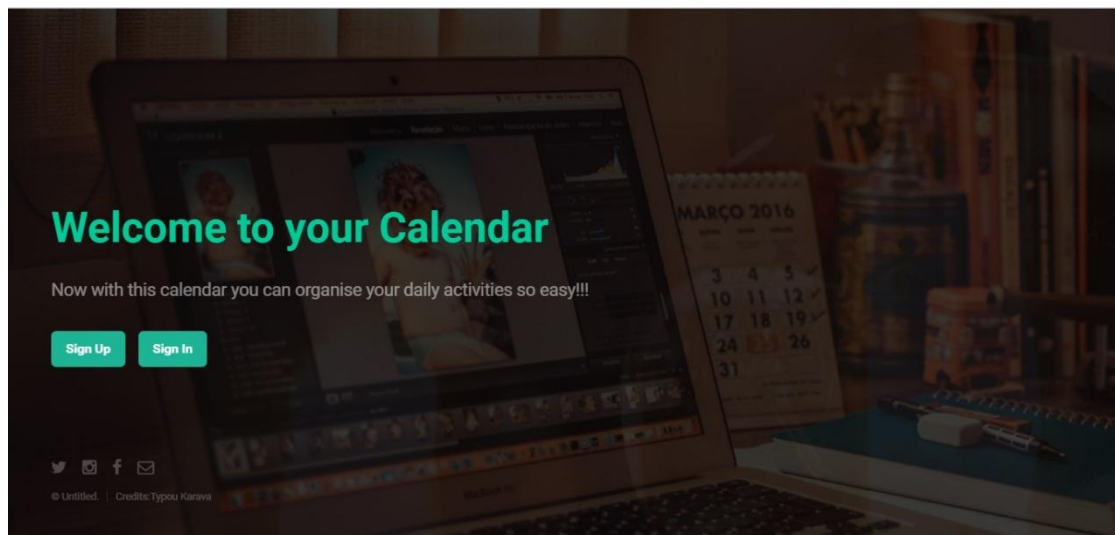
Το μέγεθός του είναι εξαιρετικά μικρό και οι απαιτήσεις του ελάχιστες.

7.ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΧΡΗΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

7.1 Εισαγωγή

Παρακάτω δίνονται οδηγίες σχετικά με την περιήγηση του χρήστη στην εφαρμογή που κατασκευάσαμε και παρουσιάζονται κάποια χαρακτηριστικά στιγμιότυπα αυτής.

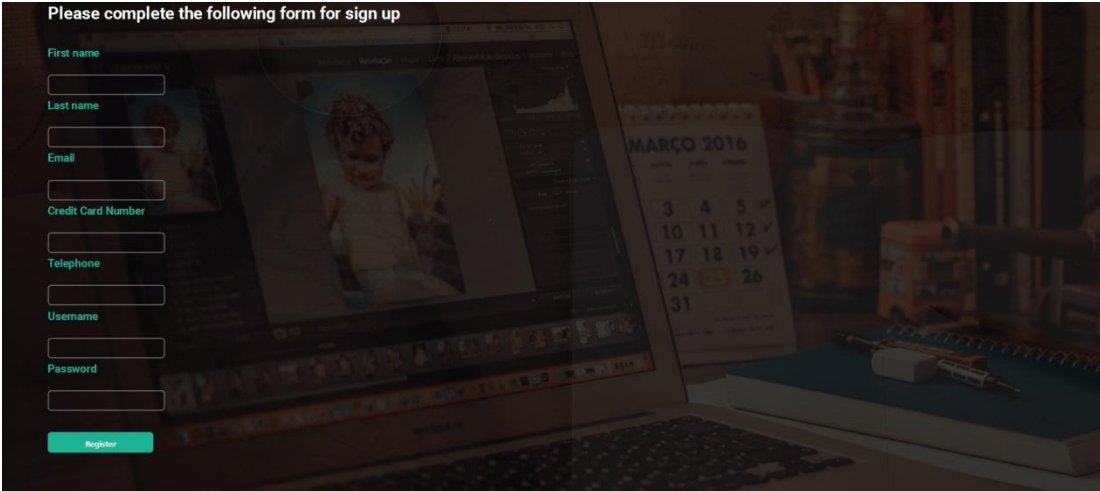
7.2 Αρχική σελίδα



Εικόνα 7.2.1 Η πρώτη σελίδα της εφαρμογής μας

Αυτή είναι η πρώτη σελίδα της εφαρμογής μας. Στην σελίδα αυτή, αφού καλωσορίζουμε τον χρήστη του δίνουμε την δυνατότητα να κάνει εγγραφή(sign_up) ή σύνδεση(sign_in) αναλόγως με το εάν επισκέπτεται για πρώτη φορά το προσωπικό του ημερολόγιο...

7.3 Εγγραφή στο Calendar



Please complete the following form for sign up

First name

Last name

Email

Credit Card Number

Telephone

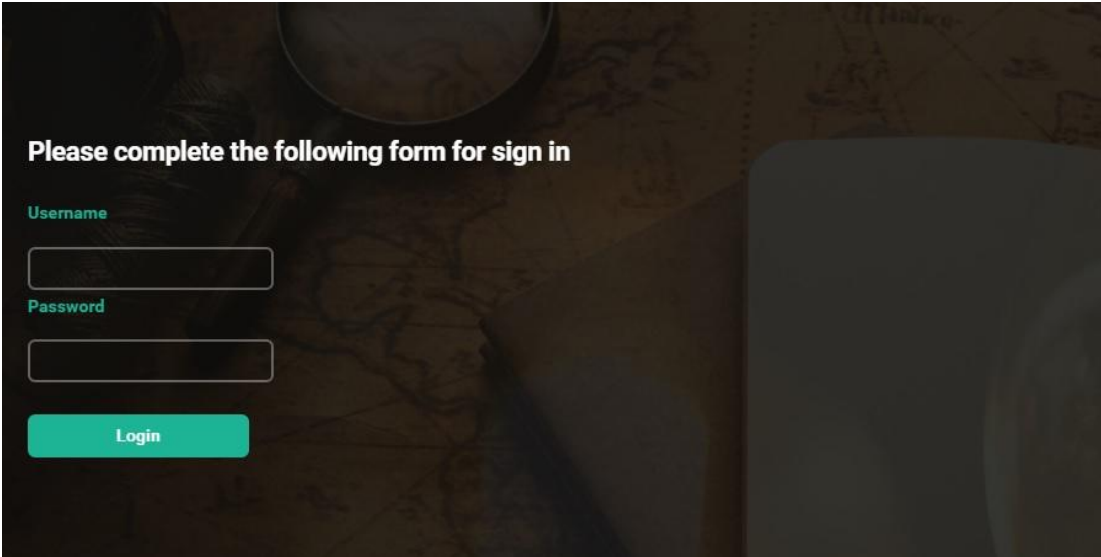
Username

Password

Εικόνα 7.3.1 Εγγραφή στο σύστημα

Στην σελίδα αυτή μπορεί ο χρήστης να κάνει εγγραφή στο προσωπικό του ημερολόγιο συμπληρώνοντας την παραπάνω φόρμα και έπειτα πατώντας το κουμπί **Register**

7.4 Σύνδεση στο Calendar



Please complete the following form for sign in

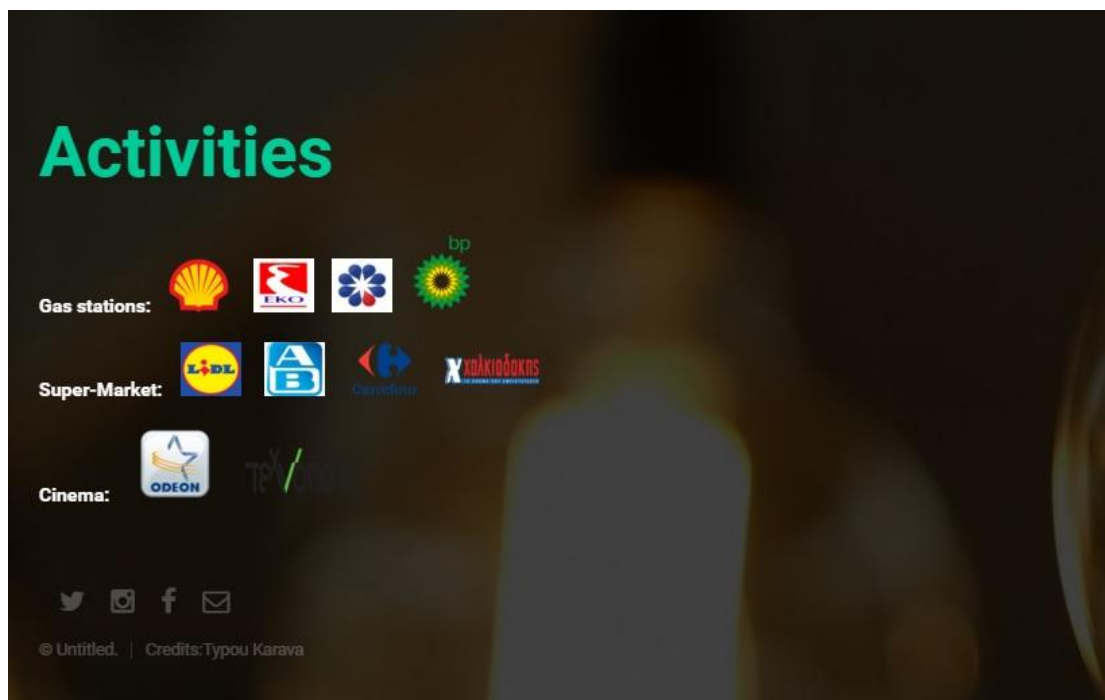
Username

Password

Εικόνα 7.4.1 Σύνδεση στο σύστημα

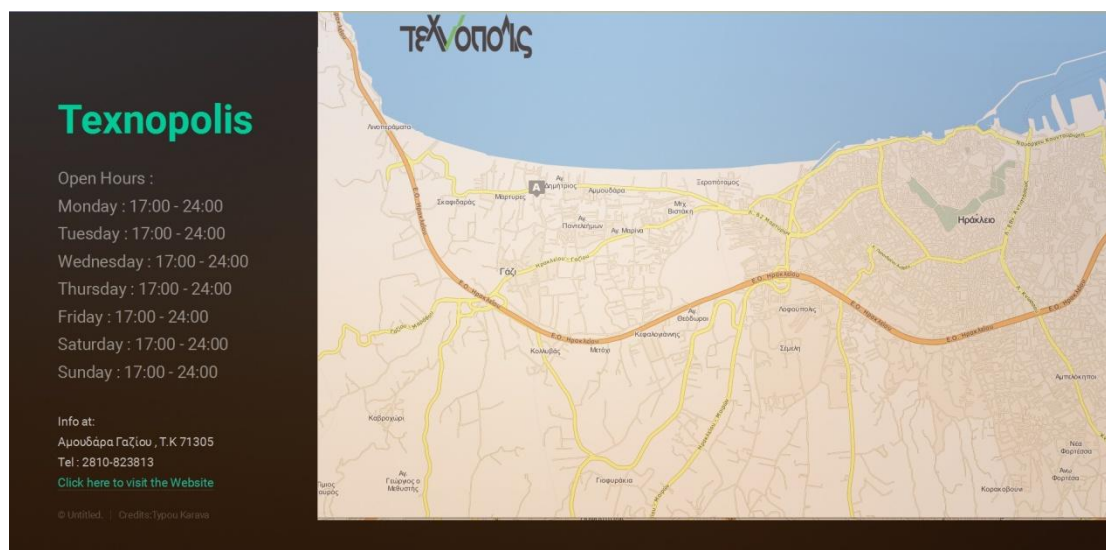
Στην παραπάνω σελίδα, ο χρήστης έχει τη δυνατότητα να εισαχθεί στο προσωπικό του ημερολόγιο, συμπληρώνοντας το username και το password που διάλεξε κατά την εγγραφή του στο σύστημα.

7.5 Μενού δραστηριοτήτων



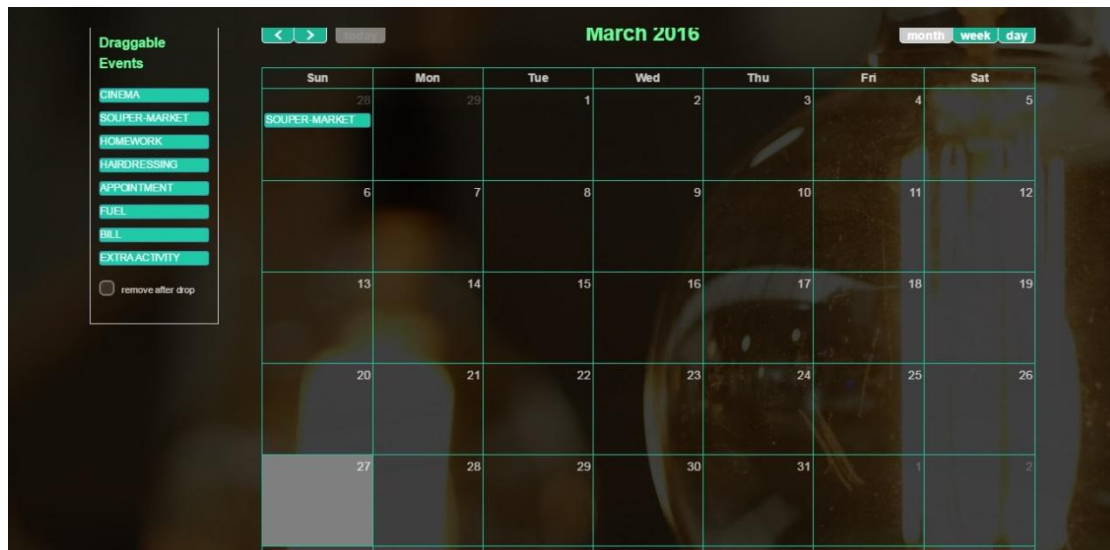
Εικόνα 7.5.1 Μενού Δραστηριοτήτων

Στην παραπάνω σελίδα ο χρήστης μπορεί να διακρίνει τις 3 κατηγορίες προσωπικών δραστηριοτήτων του (βενζινάδικα , σουπερμάρκετ , κινηματογράφους) καθώς και να περιηγηθεί σε καθένα από αυτά πατώντας απλώς πάνω στο αντίστοιχο εικονίδιο του καταστήματος που τον ενδιαφέρει και αυτόματα μεταφέρεται σε άλλη μια σελίδα στην οποία μπορεί να βρει πληροφορίες για τη τοποθεσία και το ωράριο λειτουργία του καταστήματος. Παρακάτω, αυτό φαίνεται ενδεικτικά :



Εικόνα 7.5.2 Πληροφορίες καταστήματος

7.6 Σημειωματάριο



Εικόνα 7.6.1 Σημειωματάριο δραστηριοτήτων

Αφού ο χρήστης επιλέξει τις καθημερινές του δραστηριότητες και συλλέξει της απαραίτητες πληροφορίες ως προς την τοποθεσία και το ωράριο λειτουργίας του κάθε καταστήματος μπορεί με ένα απλό draganddrop να σημειώσει την ημερομηνία και την ώρα στην οποία επιθυμεί να επισκεφθεί το συγκεκριμένο κατάστημα. Ο τρόπος σημείωσης φαίνεται στην παραπάνω εικόνα.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- ❖ Gehrke, Johannes, Ramakrishnan, και Raghu (2002), Συστήματα διαχείρισης βάσεων δεδομένων, Εκδόσεις Τζιόλα
- ❖ Meloni, J. (2006), Μάθετε PHP, Apache και MySQL σε 24 ώρες, Εκδόσεις Μ.Γκιούρδας.
- ❖ Σαλαμπάσης, Μ. (2008), Εισαγωγή στον προγραμματισμό διαδικτυακών εφαρμογών.
- ❖ Willing, L. και Thomson, L. (2005), Ανάπτυξη Web Εφαρμογών με Php και Mysql.
- ❖ Hall, Marty(2007), Servlets και σελίδες διακομιστή Java : Τεχνολογίες πυρήνα.

Πηγές Διαδικτύου

- ❖ <http://myphp.gr/manual/el>
- ❖ <http://techteam.gr/>
- ❖ <http://www.freestuff.gr/forums/viewtopic.php?t=18943>
- ❖ <https://el.wikipedia.org/>
- ❖ <http://www.w3schools.com>
- ❖ <http://www.php.net>

