



ΑΛΕΞΑΝΔΡΕΙΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΘΕΣΣΑΛΟΝΙΚΗΣ  
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ - ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Πτυχιακή Εργασία

# Cascading Style Sheets

## Εκπαιδευτική Εφαρμογή Ιστού



Του φοιτητή:

Λάυκα Γεωργίου

Αρ. Μητρώου: 05/2855

Επιβλέπων Καθηγητής

Φώτης Κόκκορας

ΘΕΣΣΑΛΟΝΙΚΗ 2010



# Πρόλογος

Ο Παγκόσμιος Ιστός έχει εξελιχθεί σε ένα από τα κυρίαρχα μέσα παροχής πληροφοριών παντός τύπου. Στις μέρες μας χρησιμοποιείται από τεράστιο αριθμό χρηστών ανα τον κόσμο για κάθε είδους σκοπό.

Το περιβάλλον αυτό έχει δημιουργηθεί με την χρήση κάποιων τεχνολογιών. Μία από αυτές, που αφορά κυρίως αυτό που ο τελικός χρήστης βλέπει στην οθόνη του υπολογιστή του (και όχι μόνο), είναι η γλώσσα των *Cascading Style Sheets*. Στην πτυχιακή αυτή εργασία το θέμα αφορά στην κατασκευή ενός ιστότοπου για την υποβοήθηση στην εκμάθηση αυτής της γλώσσας. Στις σελίδες αυτού του πονήματος βρίσκονται πληροφορίες που αφορούν τόσο στο τεχνικό κομμάτι της κατασκευής του ιστότοπου όσο και σε πιο θεωρητικά θέματα σχετικά με τον Ιστό.

Η πτυχιακή εργασία έλαβε χώρα στο Τμήμα Πληροφορικής της Σχολής Τεχνολογικών Εφαρμογών του Αλεξάνδρειου Τεχνολογικού Εκπαιδευτικού Ιδρύματος Θεσσαλονίκης, με επιβλέποντα καθηγητή τον κ. Φώτη Κόκκορα.

Γεώργιος Λαύκας

15/04/2010

# Περίληψη

Το θέμα της παρούσας εργασίας αφορά στην περιγραφή του πιο σημαντικού τμήματος της διαδικασίας που έλαβε χώρα για την ανάπτυξη ιστότοπου εκμάθησης της γλώσσας CSS (Cascading Style Sheets). Πιο συγκεκριμένα, ο κύριος όγκος της εργασίας αναφέρεται στα δύο βασικά μέρη της διαδικασίας: τον σχεδιασμό του ιστότοπου και τον προγραμματισμό της εφαρμογής που εκτελείται σε αυτόν.

Το κάθε ένα από τα παραπάνω τμήματα καταλαμβάνει ένα κεφάλαιο και χωρίζεται σε υποενότητες. Παρατίθενται τμήματα κώδικα και ακολουθεί μια σύντομη και περιεκτική εξήγηση της λειτουργικότητάς τους.

Τα υπόλοιπα κεφάλαια της εργασίας αναφέρονται σε θέματα που άπτονται του αντικειμένου, όπως τεχνολογίες Ιστού και προγράμματα φυλλομετρητών, ενώ παράλληλα μια υποενότητα είναι αφιερωμένη στο συνοδευτικό μάθημα που αναρτήθηκε στο σύστημα Blackboard. Επιπλέον, στο τελευταίο κεφάλαιο αναφέρονται τα συμπεράσματα που προέκυψαν κατά τη διάρκεια εκπόνησης της εργασίας, καθώς και παρατηρήσεις για πιθανή μελλοντική επέκταση.

Στις σελίδες αυτές γίνεται εστίαση στα πιο σημαντικά και κρίσιμα σημεία στην κατασκευή του ιστότοπου από θεωρητικής και πρακτικής πλευράς. Για τεχνικές πληροφορίες σχετικά με την ίδια την γλώσσα των CSS ο αναγνώστης παραπέμπεται στο προϊόν της εργασίας, τον ιστότοπο *CSS Tutor*. Γι' αυτό το λόγο το CD με το αποτέλεσμα της πτυχιακής αποτελεί το ουσιαστικότερο κομμάτι της εργασίας.

# Abstract

The subject of this paper is to describe the most important part of the procedure that took place for the development of a CSS (Cascading Style Sheets) teaching website. More specifically, the main part of the paper refers to the two basic aspects of the procedure: designing the website and programming the application that runs on it.

Both the above sections take up one chapter, each separated into subsections. Pieces of code are displayed with a short and brief explanation of their functionality.

The remaining chapters of the paper concern matters that are relative to the subject, such as Web technologies and browsers, while one short section is about the extra accompanying lesson on Blackboard Learning System. Moreover, in the last chapter the conclusions made during the time of development are documented, as well as remarks for possible future expansion.

The following pages focus more on the most important and critical points in the construction of the website from a designing and programming point of view. For technical information regarding the CSS language itself the reader is advised to refer to the actual product of the process, the *CSS Tutor* website. For this reason, the resulting CD of the paper is the most substantial part of the project.



# Περιεχόμενα

ΠΡΟΛΟΓΟΣ.....	I
ΠΕΡΙΛΗΨΗ .....	II
ABSTRACT .....	III
ΠΕΡΙΕΧΟΜΕΝΑ.....	V
ΕΥΡΕΤΗΡΙΟ ΕΙΚΟΝΩΝ.....	IX
<b>1 ΕΙΣΑΓΩΓΗ .....</b>	<b>1</b>
<b>2 ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ .....</b>	<b>3</b>
2.1 ΓΛΩΣΣΑ ΣΗΜΑΝΣΗΣ ΥΠΕΡΚΕΙΜΕΝΟΥ (HTML).....	3
2.2 DOCUMENT OBJECT MODEL - DOM .....	5
2.3 Η ΓΛΩΣΣΑ JAVASCRIPT .....	7
2.4 ΣΥΖΕΥΓΜΕΝΑ ΣΤΥΛ – CASCADING STYLE SHEETS (CSS).....	9
2.5 ΣΥΝΟΨΗ .....	12
<b>3 ΣΧΕΔΙΑΣΜΟΣ ΤΟΥ ΙΣΤΟΤΟΠΟΥ .....</b>	<b>13</b>
3.1 ΕΠΙΛΟΓΗ ΠΡΟΤΥΠΟΥ (TEMPLATE) .....	13
3.1.1 Κώδικας προτύπου .....	17
3.2 ΜΟΡΦΟΠΟΙΗΣΗ ΠΡΟΤΥΠΟΥ .....	25
3.2.1 Διάταξη.....	26
3.2.2 Χρώματα, γραμματοσειρές και μεγέθη .....	28
3.3 ΠΛΟΗΓΗΣΗ ΣΤΙΣ ΙΣΤΟΣΕΛΙΔΕΣ .....	28
3.3.1 Πλοήγηση με το sidebar .....	29
3.3.2 Πλοήγηση με τους συνδέσμους.....	29
3.3.3 Πλοήγηση με το breadcrumb .....	30
3.4 ΣΥΝΟΨΗ .....	30
<b>4 ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΤΟΥ ΙΣΤΟΤΟΠΟΥ .....</b>	<b>31</b>
4.1 Η ΒΙΒΛΙΟΘΗΚΗ JQUERY .....	31
4.1.1 Επιλογή framework.....	31
4.2 ΚΩΔΙΚΑΣ ΕΦΑΡΜΟΓΗΣ .....	33

4.2.1	Το αρχείο <i>jtutor.js</i> .....	33
4.3	ΚΩΔΙΚΑΣ ΕΦΦΕ.....	42
4.3.1	<i>Fade-in/fade-out</i> στις «Συχνές Ερωτήσεις».....	42
4.3.2	<i>Εμφέ</i> στην πλευρική μπάρα.....	44
4.3.3	<i>Εμφέ</i> στα <i>links</i> Προηγούμενο/Επόμενο.....	47
4.3.4	Προεπισκόπηση εικόνων.....	47
4.4	ΣΥΝΟΨΗ.....	48
<b>5</b>	<b>ΤΟ ΣΥΣΤΗΜΑ ΣΕ ΧΡΗΣΗ: ΦΥΛΛΟΜΕΤΡΗΤΕΣ ΚΑΙ BLACKBOARD .....</b>	<b>51</b>
5.1	ΤΟ ΠΡΟΒΛΗΜΑ ΤΩΝ ΦΥΛΛΟΜΕΤΡΗΤΩΝ.....	51
5.1.1	Ένα χαρακτηριστικό πρόβλημα .....	53
5.1.2	Παράκαμψη και αντιμετώπιση των ιδιομορφιών .....	53
5.2	CSS TUTOR ΚΑΙ ΦΥΛΛΟΜΕΤΡΗΤΕΣ .....	55
5.2.1	<i>Mozilla Firefox</i> .....	56
5.2.2	<i>Internet Explorer</i> .....	56
5.3	ΠΕΙΡΑΜΑΤΙΚΟ ΜΑΘΗΜΑ ΣΤΟ ΣΥΣΤΗΜΑ BLACKBOARD .....	59
5.3.1	Δημιουργία ειδικού μαθήματος.....	59
5.3.2	Δημιουργία αρχείων ερωτήσεων .....	59
5.3.3	Χρησιμότητα μαθήματος .....	60
5.4	ΣΥΝΟΨΗ.....	61
<b>6</b>	<b>ΕΞΕΛΙΞΕΙΣ ΣΤΟ ΠΕΔΙΟ.....</b>	<b>63</b>
6.1	ΠΑΓΚΟΣΜΙΟΣ ΙΣΤΟΣ .....	63
6.2	HTML5.....	66
6.3	CSS3 .....	68
<b>7</b>	<b>ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΕΠΕΚΤΑΣΕΙΣ .....</b>	<b>71</b>
7.1	ΣΥΜΠΕΡΑΣΜΑΤΑ .....	71
7.1.1	Σχεδιασμός Ιστοσελίδων.....	71
7.1.2	Διαδικτυακός Προγραμματισμός.....	72
7.1.3	Τεχνολογίες Παγκοσμίου Ιστού.....	72
7.1.4	Φυλλομετρητές.....	73
7.2	ΕΝΑΛΛΑΚΤΙΚΗ ΠΡΟΣΕΓΓΙΣΗ ΚΑΙ ΕΠΕΚΤΑΣΕΙΣ .....	74
7.2.1	<i>Server-side</i> προσέγγιση .....	74
7.2.2	Πιθανές επεκτάσεις.....	75



7.3 ΣΥΝΟΨΗ .....	76
<b>ΒΙΒΛΙΟΓΡΑΦΙΑ .....</b>	<b>77</b>
<b>ΠΑΡΑΡΤΗΜΑ Α – ΣΤΙΓΜΙΟΤΥΠΑ ΣΕΛΙΔΩΝ.....</b>	<b>79</b>
<b>ΠΑΡΑΡΤΗΜΑ Β – BLACKBOARD.....</b>	<b>83</b>



# Ευρετήριο Εικόνων

Εικόνα 1: Ιεραρχία DOM δέντρου (Sam’s Teach Yourself JavaScript in 24 hours).....	7
Εικόνα 2: jQuery UI και Blueprint CSS .....	15
Εικόνα 3: Κεφαλίδα (κόκκινο και σκούρο γκρι), μπάρα πλοήγησης (γαλάζιο) και περιεχόμενο (ανοιχτό γκρι).....	16
Εικόνα 4: Το κάτω μέρος της σελίδας, με σκούρο γκρι χρώμα το υποσέλιδο....	16
Εικόνα 5: Η μορφή της ιστοσελίδας με την εφαρμογή των νέων κανόνων CSS	25
Εικόνα 6: Μια από τις σελίδες, με θεωρία στο πάνω μέρος και την εφαρμογή στο κάτω.....	26
Εικόνα 7: «Result Area», η περιοχή όπου φαίνεται το αποτέλεσμα του κώδικα που βρίσκεται στον «Επεξεργαστή CSS».....	35
Εικόνα 8: Μετά το κλικ σε κάποια ερώτηση η απάντηση εμφανίζεται σταδιακά, όπως φαίνεται στην εικόνα. ....	43
Εικόνα 9: Το φόντο των συνδέσμων γίνεται πιο σκούρο όταν το ποντίκι βρίσκεται πάνω τους. ....	45
Εικόνα 10: Προεπισκόπηση των εικόνων στις σελίδες του CSS Tutor .....	48
Εικόνα 11: Στιγμιότυπο σελίδας του CSS Tutor όπως την προβάλλει ο Internet Explorer.....	58
Εικόνα 12: Τα νέα στοιχεία της HTML5 ορίζουν τη δομή των εγγράφων .....	67
Εικόνα 13: Αρχική σελίδα του CSS Tutor.....	79
Εικόνα 14: Οι ιδιότητες στην ενότητα μαθημάτων χωρίζονται σε κατηγορίες ....	79
Εικόνα 15: Σε κάθε σελίδα μαθήματος υπάρχει θεωρία που επεξηγεί κάποια ιδιότητα ή έννοια.....	80
Εικόνα 16: Στις σελίδες μαθημάτων υπάρχει το γραφικό περιβάλλον για την δοκιμή των ιδιοτήτων.....	80

Εικόνα 17: : Μεγέθυνση των εικόνων όταν το ποντίκι περνάει πάνω από αυτές .....	81
Εικόνα 18: Αλλαγή στην όψη του site με ένα κλικ .....	81
Εικόνα 19: Οι ερωτήσεις χωρίστηκαν σε κατηγορίες (question pools) .....	83
Εικόνα 20: Ερώτηση πολλαπλής επιλογής με feedback για τον χρήστη .....	83
Εικόνα 21: Ερώτηση ταξινόμησης: διατύπωση, απαντήσεις και feedback .....	84
Εικόνα 22: Ερώτηση αντιστοίχισης: διατύπωση, στοιχεία και feedback.....	84
Εικόνα 23: Στις ερωτήσεις αντιστοιχούν βαθμοί για την αξιολόγηση .....	85
Εικόνα 24: Ερώτηση τύπου Hot Spot, όπου ο χρήστης απαντά πάνω στην εικόνα.....	85
Εικόνα 25: Ερώτηση τύπου Hot Spot για την επίδειξη της ιδιότητας z-index .....	86
Εικόνα 26: Το CSS Tutor ανοίγει μέσω του Blackboard .....	86

# 1 Εισαγωγή

Στις μέρες μας ο Παγκόσμιος Ιστός (World Wide Web) αποτελεί ένα μέσο που χρησιμοποιείται σε μεγάλο βαθμό από διαφορετικού τύπου χρήστες. Πάσης φύσεως υπηρεσίες και εφαρμογές διατίθενται μέσω αυτού, όπως σχετικές με ενημέρωση, πληροφόρηση, εκπαίδευση και ψυχαγωγία.

Το σύστημα του Παγκοσμίου Ιστού αποτελείται από έγγραφα τα οποία συνδέονται μεταξύ τους με χρήση συνδέσμων. Για αυτό το λόγο αποκαλείται και «ιστός»: διότι οι πολλαπλές αυτές συνδέσεις θυμίζουν το σχηματισμό ενός ιστού. Αυτά τα έγγραφα μπορούν να ανακτηθούν από το χρήστη με την χρήση κατάλληλων προγραμμάτων, που λέγονται φυλλομετρητές, browsers ή user agents.

Τα έγγραφα είναι γραμμένα συνήθως στη γλώσσα *HTML* (Hypertext Markup Language). Η HTML, που εμφανίστηκε στις αρχές της δεκαετίας του 1990, βοηθά στην κατασκευή εγγράφων παρέχοντας έναν αριθμό από στοιχεία (elements). Τα στοιχεία αυτά χρησιμοποιούνται για να προσδιορίσουν κάποιο περιεχόμενο ή κάποια μορφή διάταξης. Έτσι υπάρχουν στοιχεία που αναπαριστούν παραγράφους, εικόνες, συνδέσμους, επικεφαλίδες, πίνακες και άλλο περιεχόμενο.

Τα έγγραφα αυτά ονομάζονται επίσης σελίδες ή *ιστοσελίδες* (webpages). Ένα σύνολο σελίδων που βρίσκονται στην ίδια διεύθυνση του Ιστού ονομάζεται ιστότοπος (website). Όπως αναφέρθηκε παραπάνω, ένας ιστότοπος μπορεί να έχει χαρακτήρα ενημέρωσης, εκπαίδευσης ή και ψυχαγωγίας.

Για να γίνει πιο αποδοτικό το σύστημα που περιγράφεται αναπτύσσονται διάφορες τεχνολογίες. Οι τεχνολογίες αυτές επιτρέπουν την ανάπτυξη ποικίλων εφαρμογών. Τέτοιες τεχνολογίες είναι για παράδειγμα η γλώσσα διαδικτυακού προγραμματισμού Javascript, το μοντέλο DOM, η έννοια AJAX, τα στοιχεία ActiveX, τα Java applets και πολλά άλλα.

Η τεχνολογία που ασχολείται με τη μορφοποίηση των σελίδων του Ιστού βασίζεται στη γλώσσα των Cascading Style Sheets ή CSS. Ο ρόλος της γλώσσας αυτής είναι να προσδιορίζει τον τρόπο με τον οποίο τα προγράμματα φυλλομετρητών που αναφέρθηκαν παραπάνω θα απεικονίζουν τα διάφορα στοιχεία μιας ιστοσελίδας. Έτσι η

γλώσσα αυτή χρησιμοποιείται κατα κόρον σήμερα για να επιτευχθούν διάφορα στυλ στις σελίδες.

Στα κεφάλαια αυτής της εργασίας θα παρουσιαστούν μερικές από τις τεχνολογίες που προαναφέρθηκαν, ενώ θα παρατεθούν και κάποιες λεπτομέρειες τεχνικής φύσεως σχετικά με τη χρήση τους.

Συγκεκριμένα, δίνεται έμφαση στις τεχνολογίες HTML, DOM, Javascript και CSS καθώς αυτές επηρέασαν σε πρακτικό επίπεδο το αποτέλεσμα της εργασίας: τον ιστότοπο *CSS Tutor*.

Ο ιστότοπος έχει σκοπό την εκμάθηση της γλώσσας των CSS. Στοχεύει στο να παρέχει τεχνικές γνώσεις περί της γλώσσας. Οι γνώσεις αυτές περιλαμβάνουν τον τρόπο χρήσης, τη σύνταξη και τις ιδιότητες. Παράλληλα προσφέρει και θεωρητικού επιπέδου γνώσεις ώστε να υπάρχει ένα βασικό υπόβαθρο για τον χρήστη.

Η υποστήριξη των παραπάνω επιτυγχάνεται με τη χρήση μιας αλληλεπιδραστικής εφαρμογής που διατίθεται στις σελίδες του ιστότοπου. Πιο αναλυτικά, στις σελίδες υπάρχει οργανωμένο το υλικό για τη μελέτη της θεωρίας καθώς και μια περιοχή όπου ο χρήστης μπορεί να τροποποιεί διάφορες ιδιότητες των CSS και να βλέπει το αποτέλεσμα σε ένα χώρο επίδειξης. Οι ιδιότητες χωρίζονται σε κατηγορίες ανάλογα με το ρόλο και τη φύση τους.

Η εργασία έχει σκοπό να παρουσιάσει τον τρόπο λειτουργίας του ιστότοπου, παραθέτοντας όπου κρίνεται απαραίτητο και αποσπάσματα κώδικα HTML και CSS. Ο κώδικας αυτός σχολιάζεται ώστε να γίνει πιο κατανοητή η λειτουργία του.

Εκτός αυτού καταγράφονται κάποιες δυσκολίες που συναντήθηκαν στην πορεία, και που αφορούν κυρίως τη συμπεριφορά του ιστότοπου σε δύο δημοφιλείς φυλλομετρητές: τον Internet Explorer και τον Mozilla Firefox.

Τέλος, γίνεται λόγος στην πρόοδο που σημειώνεται σε κάποιες από τις προαναφερθείσες τεχνολογίες ενώ παρουσιάζονται και τα συμπεράσματα που προέκυψαν μετά την ολοκλήρωση της εργασίας.

## 2 Θεωρητικό Υπόβαθρο

Στο παρόν κεφάλαιο θα λάβει χώρα μια περιεκτική αναφορά στις τεχνολογίες και έννοιες που αποτελούν το μεγαλύτερο μέρος της ανάπτυξης του ιστότοπου της εφαρμογής. Συγκεκριμένα, παρουσιάζονται οι έννοιες HTML, DOM, JavaScript, και CSS.

### 2.1 Γλώσσα Σήμανσης Υπερκειμένου (HTML)

Μια ιστοσελίδα μπορεί να περιέχει κείμενο, γραφικά όπως εικόνες και πολυμέσα, υπερσυνδέσμους δηλαδή παραπομπές σε άλλες ιστοσελίδες, και άλλα στοιχεία. Όλα αυτά τα στοιχεία είναι ευρέως γνωστά με τον όρο *υπερκείμενο*, δηλαδή αντικείμενα που δύναται να συνδεθούν μεταξύ τους με χρήση συνδέσμων (links). Ο τρόπος με τον οποίο συνδέονται και τοποθετούνται σε ένα ενιαίο οικοδόμημα είναι μέσω της γλώσσας σήμανσης υπερκειμένου ή HTML (HyperText Markup Language).

Η HTML είναι λοιπόν η γλώσσα που χρησιμοποιείται για την κατασκευή μιας ιστοσελίδας. Είναι αυτό που βρίσκεται «πίσω» από αυτό που απεικονίζεται στην οθόνη του υπολογιστή όταν ένας φυλλομετρητής ή πρόγραμμα πλοήγησης (browser ή user agent) ανοίγει ένα αρχείο που βρίσκεται στον Παγκόσμιο Ιστό, και το οποίο μπορεί να περιέχει οποιοδήποτε τύπο στοιχείου από αυτά που αναφέρθηκαν παραπάνω.

Η γλώσσα αυτή ανήκει σε μια ευρύτερη κατηγορία γλωσσών, τις γλώσσες σήμανσης (markup languages) στην οποία ανήκει και η XML (Extensible Markup Language). Μιλώντας πιο τυπικά θα λέγαμε πως αποτελεί εφαρμογή της SGML (Standardized General Markup Language) η οποία είναι μια γενικευμένη γλώσσα σήμανσης από την οποία προκύπτουν πιο ειδικές γλώσσες.

Λέγοντας *σήμανση* (tag) αναφερόμαστε σε μια δήλωση εντός του HTML εγγράφου που υποδηλώνει την ύπαρξη ενός *στοιχείου* (element). Στοιχείο μπορεί να είναι μια παράγραφος με κείμενο, μια λίστα με στοιχεία, ένα κουμπί μιας φόρμας, ένα πεδίο κειμένου, μια εικόνα, ένας υπερσύνδεσμος και γενικότερα οτιδήποτε άλλο συναντάται σε μια ιστοσελίδα. Εδώ να σημειωθεί πως μια τέτοια σελίδα δεν περιέχει μόνο ορατά

στοιχεία, αλλά και διάφορα είδη στοιχείων που έχουν ρόλο οργάνωσης ή ενσωμάτωσης άλλων στοιχείων (container elements), όπως το στοιχείο div.

Ένα HTML έγγραφο ξεκινά με τη σήμανση <HTML>. Τυπικά μέσα σε αυτό θα συναντηθεί η σήμανση <HEAD>, δηλαδή κεφαλή του εγγράφου, η οποία περιέχει πληροφορίες που αφορούν τη σελίδα όπως ο τίτλος που εσωκλείεται στη σήμανση <TITLE> ΤΙΤΛΟΣ </TITLE>. Τα περισσότερα tags τελειώνουν με τη χρήση της σήμανσης τέλους που είναι </> όπου μετά την κάθετο ακολουθεί το όνομα της σήμανσης. Η κεφαλή μπορεί επίσης να περιέχει scripts κάποιας γλώσσας, συνδέσμους για την φόρτωση αρχείων με scripts ή CSS (εκτενέστερα σε επόμενες ενότητες) και μεταδεδομένα που αφορούν τη σελίδα και περιέχονται στη σήμανση <META.../>. Το τμήμα κεφαλής τελειώνει με τη σήμανση </HEAD> και ακολουθεί το κυρίως σώμα της σελίδας <BODY>. Μέσα σε αυτό απαντώνται οι σημάνσεις που αντιστοιχούν σε περιεχόμενο και μπορεί να είναι ό,τι έχει αναφερθεί στην προηγούμενη παράγραφο. Το τμήμα σώματος λήγει με τη σήμανση </BODY> και μετά κλείνει η σήμανση που αφορά το HTML έγγραφο </HTML>. Σημειώνεται εδώ πως η HTML δεν είναι case-sensitive, δηλαδή δεν έχει διαφορά η χρήση κεφαλαίων ή πεζών χαρακτήρων.

Ένα σημαντικό χαρακτηριστικό της γλώσσας αυτής είναι πως δεν χρησιμοποιεί τύπους δεδομένων, και είναι ανεκτική στα σφάλματα (tolerant). Αυτό σημαίνει πως σε περίπτωση π.χ. κάποιου συντακτικού σφάλματος δε θα προκύψει κάποια διακοπή αλλά μπορεί να προκύψει πρόβλημα στην σωστή απεικόνιση της σελίδας. Αυτό εξαρτάται σε μεγάλο βαθμό από το πρόγραμμα πλοήγησης που χρησιμοποιείται, καθώς με την περίφημη «μάχη» που επικρατεί ανάμεσα στα διάφορα λογισμικά της κατηγορίας το τοπίο παραμένει θολό ως ένα βαθμό.

Εν συντομία, όταν ένας φυλλομετρητής συναντά κάποιο στοιχείο χρησιμοποιεί τις πληροφορίες που του παρέχει η μηχανή διάταξης (layout engine) στην οποία στηρίζεται καθώς και τις παραμέτρους που συνοδεύουν κάθε σήμανση (σχετικές με την τοποθέτησή του στη σελίδα, τη μορφοποίησή του, τη στοίχιση κλπ) ώστε να εμφανίσει το σύνολο των στοιχείων στην οθόνη. Η διαδικασία αυτή της σάρωσης των στοιχείων του εγγράφου HTML ονομάζεται parsing.

Ένα ακόμα σημαντικό στοιχείο της γλώσσας είναι πως δε διαχωρίζει το περιεχόμενο από τη μορφοποίηση. Αυτό σημαίνει πως οι σημάνσεις που αφορούν περιεχόμενο (όπως μια παράγραφος) και οι σημάνσεις που αφορούν τη μορφοποίηση (όπως το χρώμα ή το μέγεθος μιας γραμματοσειράς) συνυπάρχουν στο ίδιο έγγραφο.



Αυτό καθιστά δύσκολη πολλές φορές την ανάγνωση του κώδικα της σελίδας και κυρίως την τροποποίησή του. Περισσότερα για αυτό το θέμα θα αναλυθούν στην ενότητα του κεφαλαίου που αφορά τα συζευγμένα στυλ (Cascading Style Sheets – CSS).

Στα πρώτα της στάδια η HTML επέτρεπε την δημιουργία στατικού περιεχομένου σελίδων, δηλαδή σελίδων που δεν επηρεάζονταν από την επέμβαση του χρήστη, δεν υπήρχε δηλαδή ορατό αποτέλεσμα από την αλληλεπίδραση με αυτόν. Αργότερα όμως, τέθηκε στην διάθεση των προγραμματιστών μέσω της DHTML (Dynamic HTML) ένα σύνολο από ιδιότητες, μεθόδους και γεγονότα ώστε η σελίδα να αποκρίνεται στην δράση του χρήστη αλλάζοντας εν μέρει ή ολικά μορφή.

Η HTML είναι η γλώσσα του Διαδικτύου και συνεπώς πολλές τεχνολογίες που αφορούν τον διαδικτυακό προγραμματισμό αλληλεπιδρούν με αυτή. Στις επόμενες ενότητες θα παρουσιαστούν κάποιες από αυτές.

## 2.2 Document Object Model - DOM

Όπως ειπώθηκε παραπάνω, ήταν δυνατή αρχικά η κατασκευή στατικών μόνο σελίδων με τη χρήση της HTML. Για να προκύψει ένα αποτέλεσμα όπου η σελίδα θα άλλαζε μορφή ή περιεχόμενο δυναμικά, σε πραγματικό χρόνο, έπρεπε να γίνει δυνατή η τροποποίηση των σημάνσεων του εγγράφου μετά από ενέργεια του χρήστη. Μια έννοια που απλοποίησε αυτή τη διαδικασία είναι το Document Object Model.

Το DOM είναι ένα μοντέλο που αναπτύχθηκε για να παρέχει τη δυνατότητα στον προγραμματιστή να προσπελαίνει μέσω κώδικα γραμμένου σε μια γλώσσα διαδικτυακού προγραμματισμού (π.χ. JavaScript, VBscript κ.α.) τα διάφορα στοιχεία που περιέχει το HTML έγγραφο. Το μοντέλο αυτό αποτελεί κατα μια έννοια μια τεκμηρίωση ή μια αναφορά, στην οποία μπορεί να ανατρέχει ο προγραμματιστής εάν θέλει να αποκτήσει πληροφορίες για μια σήμανση και τις *ιδιότητες* της (properties), τις *μεθόδους* που μπορούν να την προσπελάσουν ή να την τροποποιήσουν (methods), καθώς και τα *γεγονότα* (events) που δύναται να προκαλέσουν μια αλλαγή σε αυτή. Το κάθε στοιχείο του εγγράφου αποτελεί ένα *αντικείμενο* του DOM. Για παράδειγμα, έστω ότι υπάρχει στο έγγραφο μια εικόνα (αντικείμενο) και ο προγραμματιστής θέλει να αντικαθίσταται από μια άλλη όταν ο χρήστης κάνει κλικ σε ένα κουμπί (γεγονός). Μέσω του γεγονότος καλείται η αντίστοιχη μέθοδος (γραμμένη σε μια γλώσσα διαδικτυακού προγραμματισμού) διαχείρισης του γεγονότος (event handler) και, στην προκειμένη περίπτωση, αλλάζει την ιδιότητα που αφορά το αρχείο εικόνας που

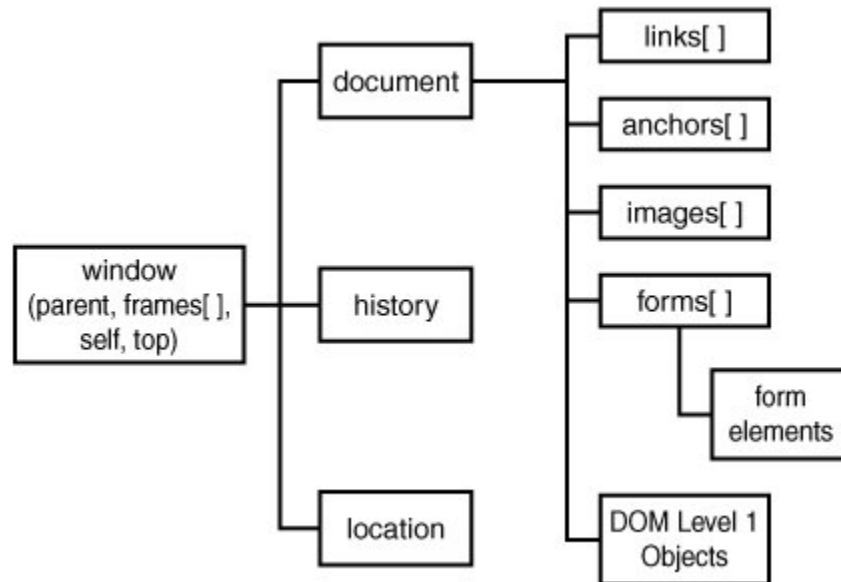
φορτώνεται. Έτσι είναι πλέον δυνατό να υπάρχει επιρροή πάνω σε οποιοδήποτε στοιχείο μιας σελίδας.

Το DOM πέρα από την παραπάνω λειτουργικότητά του αποτελεί και κάτι ακόμα σημαντικό. Ένα πρότυπο. Ο οργανισμός W3C (World Wide Web Consortium) ασχολείται με την δημιουργία προτάσεων προτυποποίησης (δεν αποτελεί επίσημο φορέα προτυποποίησης γι' αυτό συναντάται συνεχώς η έκφραση W3C recommendation – πρόταση) εννοιών που αφορούν τον Παγκόσμιο Ιστό ώστε να υπάρχει συμβατότητα στην λειτουργία των διαδικτυακών εφαρμογών ανεξάρτητα με την πλατφόρμα ανάπτυξης και το εκάστοτε πρόγραμμα φυλλομετρητή που χρησιμοποιείται. Με αυτή τη λογική ο οργανισμός αυτός προχώρησε στην τυποποίηση του DOM ώστε να αποτελεί ένα πρότυπο για όλους. Κάτι τέτοιο βέβαια στην πράξη δεν ισχύει, τουλάχιστον ολικά για την ώρα, διότι υπάρχουν διαφορετικές υλοποιήσεις του DOM από κάποιους browsers καθώς και λεπτομέρειες στη σύνταξη των εφαρμογών που καθιστούν δύσκολη την ύπαρξη της ίδιας λειτουργικότητας σε όλους τους φυλλομετρητές. Το πρόβλημα αυτό βεβαίως, πηγάζει από παλιά, και οι εκτάσεις του είναι μικρότερες σήμερα, παρόλα αυτά εξακολουθεί να εμφανίζεται.

Όπως ειπώθηκε παραπάνω, κάθε HTML στοιχείο αποτελεί ένα αντικείμενο του DOM. Έτσι μια σήμανση IMG που σημαίνει την ύπαρξη εικόνας είναι ένα αντικείμενο του DOM με ιδιότητες που αφορούν το μέγεθος, την θέση στην οθόνη κ.ά. Εκτός των HTML στοιχείων όμως υπάρχουν και άλλα αντικείμενα στο DOM που δεν αποτελούν σήμανσεις αλλά στοιχεία του περιβάλλοντος που χρησιμοποιείται για την απεικόνιση της ιστοσελίδας ή την εκτέλεση της διαδικτυακής εφαρμογής. Ένα τέτοιο αντικείμενο είναι για παράδειγμα το navigator που αναφέρεται στο ίδιο το πρόγραμμα φυλλομετρητή. Έτσι υπάρχει η δυνατότητα να προσπελούνται και ιδιότητες που αφορούν το ίδιο το περιβάλλον, παρέχοντας ευελιξία στον προγραμματισμό. Παραδείγματος χάριν, ένας απλός έλεγχος για το ποιος navigator χρησιμοποιείται μπορεί να οδηγήσει στην εκτέλεση διαφορετικού κώδικα ώστε να επιλύεται σε ένα βαθμό το πρόβλημα που αναφέρθηκε παραπάνω και αφορά τον προγραμματισμό για κάθε πλατφόρμα (cross-platform programming). Τα στοιχεία αυτά είναι γνωστά ως *core DOM* αντικείμενα.

Το DOM μιας ιστοσελίδας, δηλαδή το σύνολο των στοιχείων του HTML εγγράφου που απαρτίζουν την ιστοσελίδα, μπορεί να αναπαρασταθεί ως ένα δέντρο του οποίου η ρίζα είναι το ψηλότερα σε ιεραρχία στοιχείο, και εσωκλείει τα κατώτερα σε ιεραρχία

στοιχεία που αποτελούν κόμβους ή φύλλα του δέντρου. Η ιεραρχική αυτή οργάνωση εάν παρασταθεί οπτικά μας δίνει και μια εικόνα για το πως μπορεί να υπάρχει πρόσβαση σε ένα στοιχείο του εγγράφου μέσω των αντικειμένων που βρίσκονται πάνω απο αυτό. Η παρακάτω εικόνα δίνει μια ιδέα για την ιεραρχική μορφή που έχει το δέντρο DOM.



Εικόνα 1: Ιεραρχία DOM δέντρου (Sam's Teach Yourself JavaScript in 24 hours)

Το DOM βρίσκεται τη χρονική στιγμή που γράφεται αυτό το κείμενο στην έκδοση 3 από το W3C. Όπως ειπώθηκε παραπάνω δεν υλοποιείται πλήρως από κανένα browser καθώς οι εταιρείες αναπτύσσουν και δικές τους εκδόσεις. Παρα ταύτα αποτελεί πρότυπο και χρησιμοποιήθηκε ως τέτοιο για την εκπόνηση της παρούσας εργασίας.

## 2.3 Η γλώσσα JavaScript

Στις προηγούμενες ενότητες παρουσιάστηκαν οι έννοιες της HTML και του DOM, δύο θεμελιώδεις έννοιες όσο αφορά στον Παγκόσμιο Ιστό. Σε αυτή την ενότητα θα γίνει αναφορά στην γλώσσα JavaScript που αποτελεί ένα εργαλείο με το οποίο γίνεται η διαχείριση των στοιχείων του DOM. Με λίγα λόγια το μεγαλύτερο μέρος της λειτουργικότητας μιας ιστοσελίδας-εφαρμογής λαμβάνει χώρα με χρήση της JavaScript, σε πάρα πολλές περιπτώσεις.

Η JavaScript είναι μια διερμηνευμένη γλώσσα προγραμματισμού που έχει ιδιότητες αντικειμενοστραφούς γλώσσας. Συναντάται παντού στο Διαδίκτυο, συγκεκριμένα δε

στις εφαρμογές του Παγκόσμιου Ιστού. Τα προγράμματα γραμμένα σε αυτή τη γλώσσα λέγονται scripts και πολλές φορές βρίσκονται αποθηκευμένα στο ίδιο το HTML έγγραφο, μαζί με όλα τα στοιχεία της σελίδας. Για να τοποθετηθεί ένα script σε ένα HTML έγγραφο χρησιμοποιείται το tag `<script...>` και ακολουθεί το πρόγραμμα με τις δηλώσεις μεταβλητών, τις μεθόδους και ό,τι άλλο απαιτεί η εφαρμογή. Μπορεί φυσικά το script να βρίσκεται και σε εξωτερικό αρχείο με την κατάληξη .js και να συνδέεται με το HTML έγγραφο μέσω της παραπάνω σήμανσης ακολουθούμενης από την ιδιότητα src που λαμβάνει ως τιμή μια διαδρομή αρχείου (filepath) για την υπόδειξη του αρχείου που περιέχει το script.

Ως γλώσσα προγραμματισμού η JavaScript θυμίζει τις κλασσικές γλώσσες προγραμματισμού C/C++ και Java. Η δομή ενός script είναι όμοια με ενός προγράμματος μιας εκ των προαναφερθέντων γλωσσών ενώ και συντακτικά οι εντολές και οι δηλώσεις δε διαφέρουν σημαντικά.

Εκεί που διαφέρει η JavaScript από τις άλλες γλώσσες είναι στον τρόπο που αντιμετωπίζει τους τύπους δεδομένων. Συγκεκριμένα, ανήκει στις loosely typed γλώσσες προγραμματισμού, δηλαδή κάνει «χαλαρή» διαχείριση των τύπων εν αντιθέσει με την αυστηρή διαχείριση (strongly typed) των άλλων γλωσσών. Αυτό σημαίνει πως οι μεταβλητές της JavaScript δεν είναι απαραίτητο να έχουν κάποιο τύπο δεδομένων ενώ και ο τύπος τους μπορεί να αλλάζει στη διάρκεια εκτέλεσης του script. Επιπλέον η δέσμευση των τύπων στην JavaScript γίνεται στη διάρκεια εκτέλεσης (dynamic binding).

Όπως αναφέρθηκε παραπάνω τα scripts ενσωματώνονται ή συνδέονται στο αρχείο HTML και εκτελούνται ανάλογα με τις συνθήκες που τα προκαλούν (μετά από κάποιο γεγονός) τοπικά δηλαδή στον υπολογιστή του χρήστη. Αυτός ο τρόπος διαδικτυακού προγραμματισμού αναφέρεται ως client-side scripting διότι η εκτέλεση του προγράμματος γίνεται στην «πλευρά του πελάτη» που είναι ο υπολογιστής που κατεβάζει κάθε φορά το HTML έγγραφο που περιέχει το script. Η JavaScript μπορεί να χρησιμοποιηθεί και για server-side scripting αλλά στην παρούσα εργασία ακολουθήθηκε το μοντέλο του client-side προγραμματισμού.

Ως ένα θεωρητικό παράδειγμα χρήσης ενός script ας επανέλθουμε στο παράδειγμα με την αλλαγή εικόνας που αναφέρθηκε παραπάνω. Σε αυτή την περίπτωση υπάρχει ένα κουμπί, με κλικ στο οποίο αλλάζει η εικόνα που εμφανίζεται στην ιστοσελίδα. Εδώ υπάρχει το γεγονός «πάτημα του κουμπιού» που ενεργοποιεί τη μέθοδο που υπάρχει

στο script η οποία με τη σειρά της επενεργεί στην ιδιότητα src της εικόνας, αλλάζοντας το αρχείο εικόνας που φορτώνεται με ένα άλλο. Αυτός ο τύπος προγραμματισμού εμπεριέχει και την έννοια του προγραμματισμού οδηγούμενου από γεγονότα (event-driven programming) καθώς κάθε ενέργεια που επιτελεί το script είναι η απάντηση σε ένα γεγονός, και διαφέρει από την απλή σειριακή εκτέλεση εντολών των διαδικασιακών γλωσσών.

Για την ανάπτυξη του ιστότοπου της εργασίας χρησιμοποιήθηκε μια βιβλιοθήκη της JavaScript που ονομάζεται JQuery και έχει ως σκοπό την απλοποίηση της διαδικασίας ανάκτησης και τροποποίησης των HTML στοιχείων. Περισσότερα για την βιβλιοθήκη αυτή θα αναφερθούν στο κεφάλαιο που αφορά τον προγραμματισμό του ιστότοπου..

Η JavaScript εμφανίστηκε το 1995 την εποχή που σημειώθηκε η έκρηξη στην χρήση του Παγκόσμιου Ιστού . Η εταιρεία που την ανέπτυξε ήταν η Netscape Communications και αρχική πρόθεσή της ήταν να ενσωματωθεί στο δημοφιλές πρόγραμμα πλοήγησης της εποχής, τον Netscape Navigator. Να σημειωθεί πως η JavaScript δεν πρέπει να συγχέεται με την Java. Δεν έχουν καμία σχέση, η επιλογή ονόματος μπορεί να έγινε για λόγους καθαρά προώθησης, ενώ και η χρήση των εφαρμογών της Java για τον Ιστό (Java applets) έχει διαφορετική φιλοσοφία. Εκείνη την εποχή υπήρχε πληθώρα στατικών ιστοσελίδων. Με την πάροδο του χρόνου και μετά από διαμάχες για την προώθηση της μιας ή της άλλης γλώσσας τελικά επικράτησε σε μεγάλο βαθμό η JavaScript και πλέον η υποστήριξή της είναι ενσωματωμένη στον κάθε φυλλομετρητή.

## **2.4 Συζευγμένα Στυλ – Cascading Style Sheets (CSS)**

Ένα από τα μειονεκτήματα που παρουσιάζει η HTML είναι η συνύπαρξη σε ένα έγγραφο των σημάνσεων που αφορούν περιεχόμενο και αυτών που αφορούν μορφοποίηση. Με αυτόν τον τρόπο καθίσταται δύσκολη η επανεξέταση του κώδικα μιας σελίδας, για τις περιπτώσεις όπου απαιτείται μια αλλαγή σε κάποιο σημείο. Επιπλέον, στις περιπτώσεις όπου η ίδια μορφοποίηση απαιτείται για έναν αριθμό από σελίδες, όπως για το σύνολο των σελίδων ενός ιστότοπου, είναι χρονοβόρο και μη αποδοτικό να συντάσσεται για κάθε HTML έγγραφο ξεχωριστά. Αυτά αλλά και άλλου τύπου προβλήματα επιλύει η χρήση των Συζευγμένων Στυλ ή Cascading Style Sheets (CSS).

Τα CSS εμφανίστηκαν για πρώτη φορά ως πρόταση του W3C το 1996. Έκτοτε έχουν προστεθεί πολλές νέες ιδιότητες για την μορφοποίηση ιστοσελίδας. Ο όρος «cascading» (ελλ. διαδοχικός, σε ελεύθερη μετάφραση «συζευγμένα») αναφέρεται ακριβώς στο ότι δεν εφαρμόζεται ένα ενιαίο στυλ σε κάποια σήμανση αλλά οι μορφοποιήσεις εναλλάσσονται ανάλογα με τους κανόνες που έχουν συνταχθεί.

Ένα CSS αρχείο αποτελείται από κανόνες που αφορούν τη μορφοποίηση των στοιχείων του HTML εγγράφου. Οι κανόνες δεν είναι τίποτα άλλο, παρά δηλώσεις που προσδιορίζουν το πως θα εμφανίζεται ένα στοιχείο στην οθόνη. Για παράδειγμα, ίσως είναι επιθυμητό όλες οι παράγραφοι να εμφανίζονται με γραμματοσειρά Arial, μέγεθος 12 εικονοστοιχείων και χρώμα μπλε. Στο CSS αρχείο υπάρχει μια δήλωση που περικλείει ακριβώς αυτές τις εντολές. Έτσι όλες οι παράγραφοι της σελίδας θα έχουν αυτή την μορφοποίηση. Είναι φυσικά δυνατό να μορφοποιούνται συγκεκριμένες παράγραφοι, με χρήση κάποιου χαρακτηριστικού (της ιδιότητας ID, NAME ή CLASS). Γενικώς υπάρχουν τρόποι για να μορφοποιείται με διαφορετικό τρόπο κάθε στοιχείο της σελίδας.

Υπάρχουν τρεις τρόποι για να χρησιμοποιηθούν κανόνες των CSS σε μια ιστοσελίδα:

- Inline – Εντός της σήμανσης στο αρχείο HTML χρησιμοποιείται η ιδιότητα `style = "..."` και ακολουθεί ο κώδικας CSS.
- Embedded – Στο τμήμα `<HEAD>` της ιστοσελίδας τοποθετείται η σήμανση `<style type = "text/css"></style>` και εντός αυτής ορίζονται οι κανόνες εμφάνισης.
- External – Όλοι οι ορισμοί CSS βρίσκονται αποθηκευμένοι σε ξεχωριστό αρχείο το οποίο συνδέεται με το HTML έγγραφο μέσω της σήμανσης `<link href = "<css_filename.css>" rel = "stylesheet" type = "text/css">`. Το αρχείο είναι τύπου `.css`.

Η σειρά που παρουσιάστηκαν οι παραπάνω μορφές είναι και η σειρά προτεραιότητας στην εφαρμογή των κανόνων από τον browser. Δηλαδή σε περίπτωση ορισμού κανόνων για το ίδιο αντικείμενο μεγαλύτερη προτεραιότητα έχει ο κανόνας που βρίσκεται inline, ακολουθεί η μορφοποίηση από το ενσωματωμένους κανόνες (embedded) και τέλος οι κανόνες του εξωτερικού αρχείου.

Στη γλώσσα CSS, όπως ακριβώς σε κάθε γλώσσα προγραμματισμού, υπάρχει ένα συγκεκριμένο συντακτικό. Έτσι, όταν γίνεται αναφορά σε κανόνες, γίνεται αναφορά σε

δηλώσεις που έχουν μια συγκεκριμένη μορφή. Οι κανόνες λοιπόν, στη γενική τους μορφή, ακολουθούν την εξής σύνταξη: *selector {property1: value; property2: value; ...propertyN: value;}*.

Με τον όρο *selector* εννοείται οτιδήποτε χρησιμοποιείται για να προσδιορίσει το στοιχείο της σελίδας που θα μορφοποιηθεί. Ως *selector* μπορεί να χρησιμοποιηθεί ένα απλό στοιχείο, όπως για παράδειγμα το *p* για να μορφοποιηθούν οι παράγραφοι της σελίδας. Σε αυτή την περίπτωση μιλάμε για *type selector*, καθώς η επιλογή γίνεται ανάλογα με τον τύπο του στοιχείου. Οι δύο άλλοι συνηθισμένοι *selectors* είναι οι *class selectors* και οι *ID selectors*. Τα *class* και *ID*, όπως αναφέρθηκε παραπάνω, είναι χαρακτηριστικά (*attributes*) των *HTML* στοιχείων. Τοποθετούνται δηλαδή εντός του *HTML tag* στο αρχείο *HTML*. Σε αυτά αποδίδεται μια τιμή και αυτή η τιμή είναι το *class* ή το *ID* αντίστοιχα. Οι *class selectors* ξεκινούν με τελεία (.) και ακολουθεί το όνομα της κλάσης ενώ οι *ID* με δίεση (#) ακολουθούμενη από το όνομα, την τιμή που έχει αποδοθεί στο *ID*. Ακολουθεί παράδειγμα:

```
<p class = "par" id ="first">This is the first paragraph.</p>
```

Έστω η παραπάνω γραμμή κώδικα σε ένα έγγραφο *HTML*. Η παράγραφος αυτή έχει κλάση *par* και *ID* *first*. Ο καθένας από τους παρακάτω κανόνες επηρεάζει την απεικόνιση αυτής της παραγράφου:

```
p {color: blue; font-size: 12px;}  
.par {color: blue; font-size: 12px;}  
#first {color: blue; font-size: 12px;}
```

Η παράγραφος αποκτά μπλε χρώμα και μέγεθος γραμματοσειράς 12 εικονοστοιχεία.

Φυσικά είναι δυνατό να γίνουν διάφοροι συνδυασμοί *selectors* για να τροποποιηθούν διαφορετικά στοιχεία της σελίδας, ενώ υπάρχουν και άλλοι τρόποι να γίνει αναφορά σε στοιχεία. Τα παραπάνω αποτελούν μια εισαγωγή στους κανόνες. Πιο αναλυτικές πληροφορίες που αφορούν στη σύνταξη κανόνων, τους τύπους *selectors* και τις ιδιότητες και τιμές που μπορεί να συναντήσει κανείς στη δήλωση ενός κανόνα βρίσκονται στις σελίδες του *CSS Tutor*.

Στην ενότητα αυτή έγινε αναφορά αρκετές φορές στο πρόβλημα του *cross-platform* προγραμματισμού. Και η χρήση των *CSS* λοιπόν δεν εγγυάται την ορθή απεικόνιση της μορφοποίησης σε κάθε πρόγραμμα πλοήγησης. Αν και γενικά υπάρχει ασφάλεια στη χρήση της γλώσσας αυτής συναντώνται και πάλι τα προβλήματα της ασυμβατότητας μεταξύ των *browsers*. Κάτι το οποίο αναπαρίσταται όπως ακριβώς το θέλει ο

σχεδιαστής και ο προγραμματιστής στον φυλλομετρητή Firefox δεν θα αναπαρίσταται απαραίτητα έτσι και στον Internet Explorer (η επιλογή των συγκεκριμένων για αναφορά γίνεται διότι μαζί καλύπτουν τη συντριπτική πλειοψηφία των χρηστών). Μπορεί να χρειάζεται κάποια τροποποίηση ή κάποια περαιτέρω δήλωση κανόνα για να απεικονιστεί σωστά το στυλ σε όλους τους browsers. Οι τεχνικές που υπάρχουν για να παρακαμφθούν τα προβλήματα απεικόνισης ονομάζονται CSS hacks ή CSS filters. Κάποια ενδεικτικά hacks παρουσιάζονται στο CSS Tutor. Επιπλέον, ζητήματα που προέκυψαν και αφορούν στην απεικόνιση του ίδιου του ιστότοπου θα αναφερθούν στην ενότητα που γίνεται λόγος για τους διάφορους φυλλομετρητές και την υποστήριξη του site.

## 2.5 Σύνοψη

Συνοψίζοντας τα παραπάνω, θα λέγαμε πως τα συζευγμένα φύλλα στυλ χρησιμοποιούνται κυρίως για να διαχωριστεί η διακόσμηση από το περιεχόμενο και να διευκολυνθεί έτσι η διαδικασία μορφοποίησης των σημάνσεων. Αποτελούνται από κανόνες, που είναι ουσιαστικά δηλώσεις που αφορούν ιδιότητες των στοιχείων και συνδέονται με το περιεχόμενο με τρεις τρόπους: εντός της σήμανσης, εντός του εγγράφου ή σε εξωτερικό αρχείο. Χρησιμοποιούνται σε συνδυασμό με τις υπόλοιπες τεχνολογίες του διαδικτύου για να «χτιστεί» μια ιστοσελίδα. Παρότι διευκολύνουν σημαντικά τη διαδικασία μορφοποίησης, δεν αποφεύγουν τελείως το πρόβλημα της διαφορετικής υλοποίησης από τους φυλλομετρητές.



## 3 Σχεδιασμός του ιστότοπου

Σε αυτό το κεφάλαιο θα περιγραφεί η διαδικασία που ακολουθήθηκε για την κατασκευή του ιστότοπου, όσο αφορά στο σχεδιαστικό κομμάτι. Συγκεκριμένα θα λάβει χώρα μια αναφορά στους λόγους που οδήγησαν στην επιλογή της συγκεκριμένης μορφής της ιστοσελίδας, ενώ θα περιγραφούν συνοπτικά τα τμήματα κώδικα εκείνα που καθορίζουν σε μεγάλο βαθμό την τελική μορφή.

### 3.1 Επιλογή Προτύπου (Template)

Ο σχεδιασμός μιας ιστοσελίδας είναι μια διαδικασία ιδιαίτερα απαιτητική. Υπάρχει ένας αριθμός από ζητήματα που χρήζουν σωστής εκτίμησης αρχικά και αντιμετώπισης τελικά, ώστε να προκύψει ένα αποτέλεσμα τόσο λειτουργικό όσο και κομψό.

Ένα βασικό ζήτημα είναι η απεικόνιση σε διαφορετικές αναλύσεις οθόνης. Όπως είναι λογικό, μια σελίδα πρέπει να εμφανίζεται ακριβώς όπως τη θέλει ο σχεδιαστής ανεξάρτητα με την οθόνη στην οποία προβάλλεται. Οι οθόνες όμως δεν έχουν όλες την ίδια ανάλυση. Υπάρχουν άλλες με μεγαλύτερες διαστάσεις και υψηλότερες αναλύσεις και άλλες με μικρότερες. Έτσι είναι δυνατό μια σελίδα να εμφανίζεται διαφορετικά στην πρώτη περίπτωση, ενδεχομένως με μεγαλύτερες κενές περιοχές, απ' ό,τι στη δεύτερη όπου τα στοιχεία μπορεί να εμφανίζονται σε διαφορετικές θέσεις από τις προβλεπόμενες και να επικαλύπτονται. Κάτι τέτοιο θα καθιστούσε την πλοήγηση στη σελίδα πολύ δύσκολη, ίσως και αδύνατη. Εκτός αυτού θα απωθούσε το χρήστη απ' το να ασχοληθεί με το περιεχόμενο της ιστοσελίδας.

Ένα άλλο ζήτημα σχετικό με τον σχεδιασμό της ιστοσελίδας αφορά στα χρώματα που θα χρησιμοποιηθούν, τόσο για το φόντο όσο και για τις γραμματοσειρές. Τα χρώματα θα πρέπει να επιλεγούν με τέτοιο τρόπο ώστε να μην κουράζουν το χρήστη αλλά και να τον διευκολύνουν στην ανάγνωση του κειμένου. Θα πρέπει δηλαδή να υπάρχει ένας συνδυασμός με κάποιες ταιριαστές αντιθέσεις.

Κάτι ακόμα που θα πρέπει να έχει στο μυαλό του ο σχεδιαστής είναι η διάταξη των στοιχείων στη σελίδα. Η διαρρύθμιση γενικώς ακολουθεί κάποιους κανόνες, όπως στην πραγματική ζωή ακολουθούνται κανόνες για την διακόσμηση ενός σπιτιού. Αυτό είναι

πολύ βολικό, καθώς έτσι αναπτύσσεται μια σχετική ομοιομορφία στις σελίδες του ιστού. Η ομοιομορφία αυτή είναι θεμιτή από πολλές απόψεις, καθώς ο χρήστης «εκπαιδεύεται» στο να πλοηγείται στους διάφορους ιστότοπους. Επιπλέον, οι κανόνες διάταξης που αναφέρθηκαν προκύπτουν από μελέτη συμπεριφορών και αντιδράσεων όταν διάφοροι χρήστες (ανεξάρτητα της εξοικείωσής τους με την τεχνολογία ή το διαδίκτυο) τίθενται να απαντήσουν στο κατά πόσο τους ωφέλησε μια συγκεκριμένη τοποθέτηση των στοιχείων. Υπάρχει λοιπόν ένα υπόβαθρο επιστημονικό, που αφορά τις διεπαφές χρήστη-υπολογιστή και τις λεγόμενες *γνωστικές επιστήμες* που οδηγεί στην παραγωγή προτάσεων προς τους σχεδιαστές για την κατασκευή ενός ιστότοπου.

Όπως αναφέρθηκε στο προηγούμενο κεφάλαιο, η αναπαράσταση μιας σελίδας δεν είναι η ίδια σε κάθε φυλλομετρητή. Έτσι κάτι ακόμα που πρέπει να αντιμετωπιστεί είναι αυτή η ασυμβατότητα μεταξύ προγραμμάτων περιηγητών και ιστοσελίδων.

Τα παραπάνω αποτελούν ένα ενδεικτικό κομμάτι των προβλημάτων που τίθεται να αντιμετωπίσει ο σχεδιαστής μιας σελίδας, και μάλιστα είναι τα πιο βασικά. Ένας βασικός τρόπος να αποφευχθούν οι επιπτώσεις τους είναι η χρήση *frameworks* ή *templates*.

Στην πρώτη περίπτωση, γίνεται χρήση ενός προτύπου το οποίο έχει δοκιμαστεί και επιτύχει σε διάφορες συνθήκες όπως αυτές που αναφέρθηκαν. Τα *frameworks* αποτελούν θεμέλιο για την ανάπτυξη ιστοσελίδων. Πάνω σε αυτό το θεμέλιο ο σχεδιαστής μπορεί να χτίσει, κάνοντας τις απαραίτητες αλλαγές για να το προσαρμόσει στις ανάγκες του και στις ανάγκες του κοινού το οποίο καλείται να εξυπηρετήσει. Τα *frameworks* είναι διαφόρων ειδών. Υπάρχουν Javascript *frameworks* που ορίζουν βιβλιοθήκες μεθόδων για τον σχεδιασμό και χειρισμό *interfaces* αλλά και CSS *frameworks* που ορίζουν ένα σύνολο από κανόνες CSS για την διάταξη και εμφάνιση των στοιχείων στο χώρο της σελίδας. Αυτά μπορούν να χρησιμοποιηθούν συμπληρωματικά. Στην εικόνα 2 φαίνεται ένα στιγμιότυπο ενός Javascript *framework*, του **jQuery UI**, παράγωγου της βιβλιοθήκης jQuery η οποία θα παρουσιαστεί σε επόμενο κεφάλαιο και ενός CSS *framework*, του δημοφιλούς *Blueprint CSS*.



Εικόνα 2: jQuery UI και Blueprint CSS

Τα templates δε διαφέρουν και πολύ από τα frameworks. Πρόκειται για «καλούπια», έτοιμες δηλαδή σελίδες όπου απλώς ο σχεδιαστής εισάγει το περιεχόμενο. Αυτό δε σημαίνει βέβαια πως δε μπορούν να τροποποιηθούν. Κάτι τέτοιο συνέβη μάλιστα και στην περίπτωση της σελίδας του CSS Tutor, όπως θα περιγραφεί παρακάτω. Τα περισσότερα templates αποτελούνται από ένα αρχείο html με κάποιο περιεχόμενο για επίδειξη και ένα αρχείο css με τους κανόνες που ορίζουν την απεικόνιση των στοιχείων. Υπάρχουν επιπλέον και συνοδευτικά αρχεία, όπως εικόνες. Τα templates είναι ένας εύκολος και γρήγορος τρόπος να δομηθεί μια σελίδα, ενώ επιπλέον αποτελούν δοκιμασμένες λύσεις. Στα περισσότερα sites αναζήτησης τέτοιων προτύπων, υπάρχουν κατηγορίες ανάλογα με τη φύση της ιστοσελίδας (π.χ. εκπαιδευτική, εφαρμογής, ψυχαγωγίας κ.ά.).

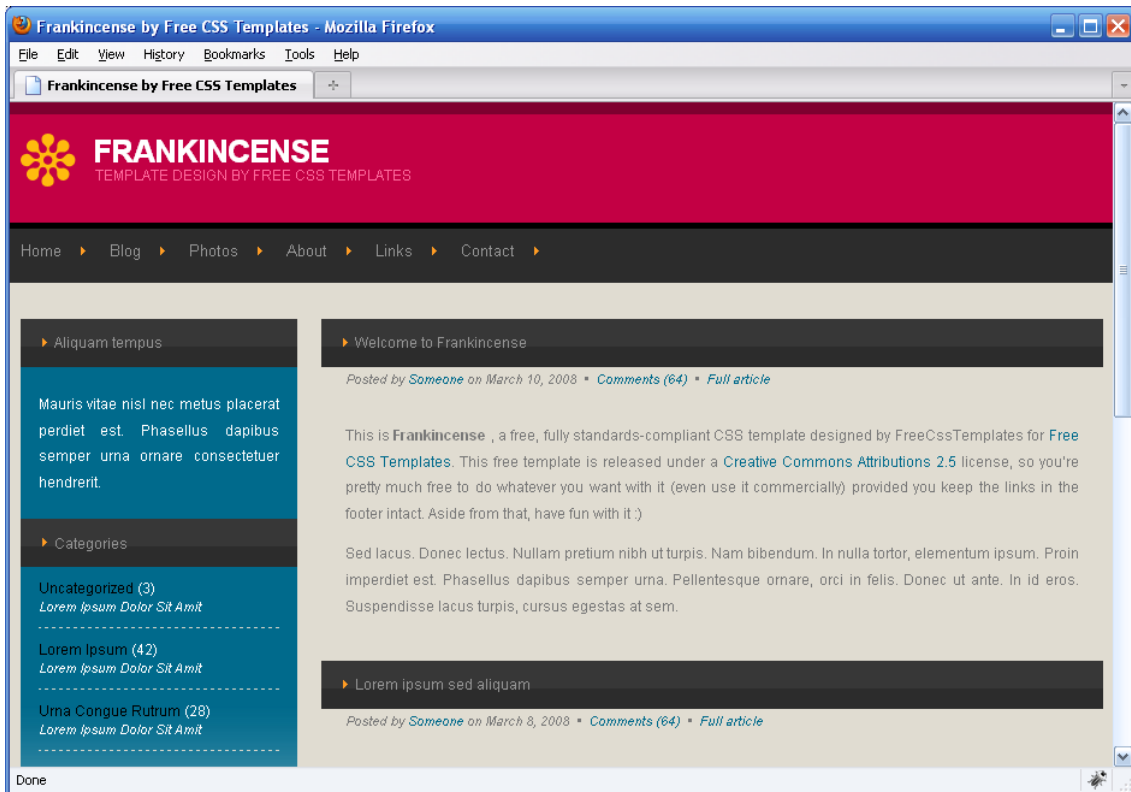
Για την ιστοσελίδα του CSS Tutor χρησιμοποιήθηκε το template **Frankinsence** το οποίο βρέθηκε στην ιστοσελίδα [www.freecsstemplates.org](http://www.freecsstemplates.org). Το συγκεκριμένο, όπως και άλλα σε αυτή τη σελίδα, διατίθεται για ελεύθερη χρήση υπό την **Creative Commons License**.

Συγκεκριμένα, μετά από αναζήτηση σε διάφορες ιστοσελίδες παροχής templates, το συγκεκριμένο κρίθηκε καταλληλότερο για την ιστοσελίδα, λόγω της απλής δομής του. Η δομή αυτή ευνοεί την ανάπτυξη περιεχομένου χωρίς να δυσκολεύεται ο χρήστης να το παρακολουθήσει. Επιπλέον, η διάταξή του είναι μια τυπική διάταξη που συναντάται πολύ συχνά.

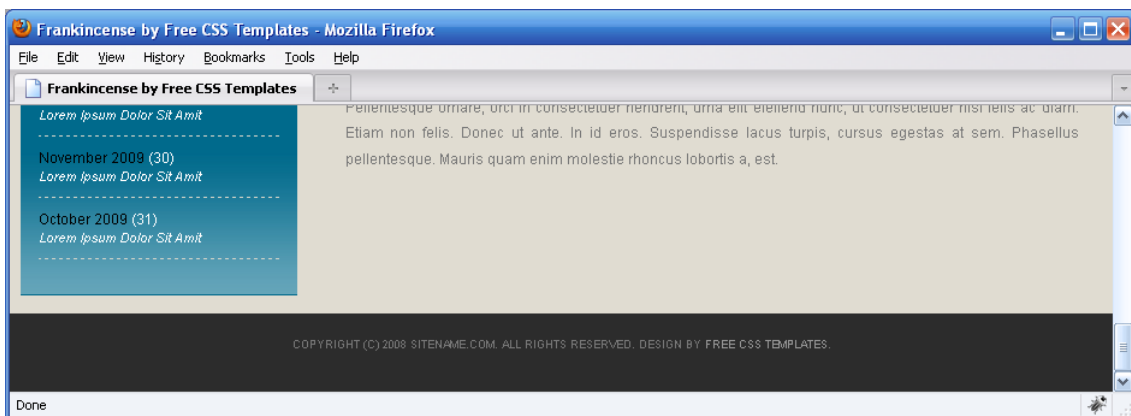
Όπως είναι εύκολο να παρατηρηθεί, αποτελείται από τέσσερα μέρη: την κεφαλίδα (header) που φιλοξενεί ένα λογότυπο και ένα απλό οριζόντιο μενού, την πλευρική μπάρα πλοήγησης στα αριστερά που περιέχει συνδέσμους για τις σελίδες του

ιστότοπου, την περιοχή περιεχομένου όπου τοποθετούνται οι καταχωρήσεις και τέλος ένα υποσέλιδο.

Στις εικόνες 3 και 4 φαίνεται η δομή του template. Αυτή είναι η αρχική του μορφή. Στη συνέχεια θα παρατεθεί το σύνολο των αλλαγών που έγιναν στους κανόνες ώστε να προκύψει η τελική του μορφή, όπως φαίνεται στο CSS Tutor.



Εικόνα 3: Κεφαλίδα (κόκκινο και σκούρο γκρι), μπάρα πλοήγησης (γαλάζιο) και περιεχόμενο (ανοιχτό γκρι)



Εικόνα 4: Το κάτω μέρος της σελίδας, με σκούρο γκρι χρώμα το υποσέλιδο

### 3.1.1 Κώδικας προτύπου

Εδώ θα παρατεθούν ενδεικτικά κάποια τμήματα του HTML και CSS κώδικα που βρίσκεται πίσω από το template, για να γίνει πιο κατανοητή η δομή του. Σημειώνεται πως οι κανόνες CSS βρίσκονται σε εξωτερικό αρχείο με όνομα *styleTutor.css* το οποίο και συνδέεται με το HTML έγγραφο.

Όπως αναφέρθηκε παραπάνω και φαίνεται από τις εικόνες, οι σελίδες χωρίζονται σε 4 τμήματα. Το header, το sidebar, το content και το footer. Ο κώδικας και μια σύντομη περιγραφή για το καθένα φαίνεται παρακάτω.

#### To header

Το header είναι η περιοχή στο πάνω μέρος της σελίδας. Όπως φαίνεται στην εικόνα 3 περιέχει ένα λογότυπο καθώς και ένα τίτλο, που ταυτόχρονα είναι και σύνδεσμος (link) στην αρχική σελίδα. Οι παρακάτω δηλώσεις HTML χρησιμοποιούνται για την κατασκευή αυτού του τμήματος της σελίδας.

```
<div id="header">
  <div id="logo">
    <h1><a href="#">Τίτλος header</a></h1>
    <p>Κείμενο κάτω από τον τίτλο</p>
  </div>
</div>
```

Ο κώδικας CSS για να εμφανίζεται το header στη συγκεκριμένη θέση, με αυτό το χρώμα φόντου, γραμματοσειρά και γενικότερα τη μορφοποίηση που φαίνεται στην εικόνα, εμφανίζεται παρακάτω:

```
#header {
  width: 900px;
  height: 100px;
  margin: 0 auto;
}
```

Αυτός ο κανόνας ορίζει το ύψος και το πλάτος του header ενώ δίνει και μια τιμή στην ιδιότητα margin (περιθώριο). Συγκεκριμένα δίνει μια αρχική τιμή 0 και στη συνέχεια το auto κατευθύνει τον browser να υπολογίζει αυτόματα τα περιθώρια, ανάλογα με την υπόλοιπη δομή της σελίδας.

```
/* Logo */
```

```
#logo {
    float: left;
    width: 900px;
    height: 100px;
    margin: 0;
    color: #FFFFFF;
    background: url(images/img03.gif) no-repeat left 25px;
}
```

Ο παραπάνω κανόνας ορίζει το ύψος, πλάτος, περιθώριο του div με id logo καθώς και το χρώμα των περιεχομένων του να είναι λευκό (FFFFFF) και τοποθετεί μια εικόνα (img03.gif) σε μια συγκεκριμένη θέση.

```
#logo h1, #logo p {
    margin: 0;
    padding: 0;
    text-transform: uppercase;}
#logo h1 {
    padding-top: 25px;
    padding-left: 60px;
    letter-spacing: -1px;
    font-size: 2.2em;
    font-weight: bold;
}
#logo p {
    margin-top: -6px;
    padding-left: 62px;
    color: #FF6B9D;
}
#logo p a {
    color: #FF6B9D;
}
#logo a {
    border: none;
```

```
background: none;
text-decoration: none;
color: #FFFFFF;
}
```

Οι κανόνες που προηγήθηκαν προσδιορίζουν την απεικόνιση της παραγράφου και του τίτλου μέσα στο `div` με `id` `logo`. Ιδιότητες που αφορούν τη θέση, το μέγεθος και το χρώμα λαμβάνουν τιμές ώστε να προκύψει το αποτέλεσμα που φαίνεται. Δε θα γίνει αναλυτική επεξήγηση των ιδιοτήτων, καθώς κάτι τέτοιο δε κρίνεται σκόπιμο, αλλά οι παραπάνω ιδιότητες παρουσιάζονται στην ενότητα μαθημάτων του CSS Tutor όπου και επεξηγείται η σημασία τους.

## To sidebar

Η κάθετη μπάρα που φαίνεται στα αριστερά της εικόνας 3 αποτελεί τον κύριο τρόπο πλοήγησης στις σελίδες του ιστότοπου. Χρησιμοποιείται για την κατηγοριοποίηση των σελίδων ανάλογα με το περιεχόμενο που έχουν. Οι σύνδεσμοι μέσα σε κάθε κατηγορία οδηγούν στην αντίστοιχη σελίδα. Ο κώδικας για την κατασκευή της πλευρικής μπάρας παρατίθεται εδώ:

```
<div id="sidebar">
  <ul>
    <li>
      <h2>Τίτλος κατηγορίας</h2>
      <p>Κείμενο κατηγορίας.</p>
    </li>
    <li>
      <h2>Τίτλος 2ης κατηγορίας</h2>
      <ul>
        <li><a href="#">Σύνδεσμος</a></li>
        <li><a href="#">Σύνδεσμος</a></li>
        <li><a href="#">Σύνδεσμος</a></li>
        <li><a href="#">Σύνδεσμος</a></li>
      </ul>
    </li>
  </ul>
```

</div>

Η δομή του sidebar είναι σχετικά απλή. Ένα div με id sidebar φιλοξενεί μια μη-ταξινομημένη λίστα (ul). Εντός αυτής διάφορες λίστες (li) χωρίζουν την περιοχή σε κατηγορίες, όπου αποδίδεται και ένας τίτλος. Μέσα σε κάθε λίστα δεν είναι απαραίτητο να βρίσκεται κείμενο, μπορεί να υπάρχει οποιοδήποτε στοιχείο της HTML. Έτσι εμφανίζονται και εμφωλεύσεις λίστα μέσα σε λίστα. Το παραπάνω δεν είναι όλος ο κώδικας που σχηματίζει το sidebar, αλλά ένα τμήμα αυτού χάριν συντομίας.

Για να προκύψει το αποτέλεσμα της εικόνας χρησιμοποιούνται αρκετοί κανόνες CSS, οι πιο χαρακτηριστικοί από τους οποίους φαίνονται παρακάτω:

```
#sidebar {  
    float: left;  
    width: 230px;  
    background: #006A8C;  
    color: #FFFFFF;  
}
```

Ο πάνω κανόνας τοποθετεί το div στα αριστερά του περιεχομένου, ορίζει το πλάτος του στα 230px, το χρώμα στο φόντο και το χρώμα των περιεχομένων.

```
#sidebar ul {  
    margin: 0;  
    padding: 0;  
    list-style: none;  
}  
#sidebar li {  
    margin: 0;  
    padding: 0;  
}  
#sidebar li ul {  
    background: url(images/img05.jpg) no-repeat left bottom;  
    padding-bottom: 30px;  
}  
#sidebar li li {
```



```
margin-left: 15px;
margin-right: 15px;
line-height: 35px;
border-bottom: 1px dashed #D2D4C9;}
```

Οι κανόνες που αφορούν τις λίστες προσδιορίζουν σημαντικές ιδιότητες για τη διάταξη και την ευκρίνεια των στοιχείων. Ορίζονται τα περιθώρια (margins), τα paddings ή παραγεμίσματα, καθώς και κάποια borders, το ύψος γραμμών και το φόντο.

```
#sidebar h2 {
    background: #2C2C2C url(images/img04.jpg) no-repeat left top;
    height: 31px;
    padding: 9px 0 0 28px;
    font-size: 12px;
    color: #969696;
}
#sidebar p {
    margin: 0 15px;
    padding: 20px 0px;
    text-align: justify;
}
#sidebar a {
    color: #000000;
    border: none;
}
```

Οι παραπάνω κανόνες αφορούν τα περιεχόμενα των καταχωρήσεων κάθε λίστας και την απεικόνισή τους. Βλέπετε ότι δίνονται τιμές σε ιδιότητες που αφορούν τη γραμματοσειρά, τα χρώματα, κάποια περιθώρια, το φόντο και μερικές άλλες.

### **Το περιεχόμενο (content)**

Η περιοχή περιεχομένου βρίσκεται στο κέντρο της σελίδας και σε αυτή προσαρτάται το υλικό που διαθέτει το site. Αυτό το υλικό μπορεί να έχει τη μορφή κειμένου, εικόνων η οποιουδήποτε άλλου στοιχείου. Ουσιαστικά πρόκειται για κάποια div στοιχεία, το ένα μέσα στο άλλο για λόγους οργάνωσης, και στοιχεία παραγράφων μέσα σε αυτά. Ακολουθούν τα πιο σημαντικά σημεία κώδικα HTML:

```

<div id="content">
  <div class="post">
    <div class="post-bgtop">
      <div class="post-bgbtm">
        <h2 class="title"><a href="#">Τίτλος περιεχομένου</a></h2>
        <p class="meta">Κείμενο κάτω από τον τίτλο</p>
        <div class="entry">
          <p>...περιεχόμενο...</p>
          <p>...περιεχόμενο...</p>
        </div>
      </div>
    </div>
  </div>
</div>

```

Όπως είναι ορατό, υπάρχουν διάφορα στοιχεία div που έχουν το ρόλο container, δηλαδή περιέχουν άλλα div για την καλύτερη και πιο «στιβαρή» οργάνωση του template. Εντός του ενδότερου div βρίσκεται αυτό καθαυτό το περιεχόμενο, μέσα σε διάφορες παραγράφους <p>. Ο τίτλος που αναφέρεται στην αρχή είναι αυτός που εμφανίζεται επάνω από το περιεχόμενο, μέσα στην γκρι περιοχή. Και πάλι έχουν παραλειφθεί κομμάτια κώδικα για λόγους απλότητας.

Ο ενδεικτικός κώδικας CSS για αυτή την περιοχή φαίνεται παρακάτω:

```

#content {
  float: right;
  width: 650px;
}

```

Ο πρώτος κανόνας που αφορά στο div με id content, το τοποθετεί στα δεξιά της σελίδας (πιο πάνω θα είδατε πως το sidebar τοποθετείται αριστερά του υπόλοιπου περιεχομένου με χρήση της float: left) και ορίζει το πλάτος του ίσο με 650px.

```

.post {
  margin-bottom: 8px;
}

```

Ο παραπάνω κανόνας θέτει το κάτω περιθώριο του div με class post ίσο με 8px.

```

.post .title {
    background: #2C2C2C url(images/img06.gif) no-repeat left top;
    height: 28px;
    padding: 12px 0 0 28px;
    font-size: 12px;
    color: #969696;
}

.post .title a {
    color: #969696;
}

.post .meta {
    padding-left: 20px;
    padding-bottom: 5px;
    text-align: left;
    font-family: Arial, Helvetica, sans-serif;
    font-size: 11px;
    font-style: italic;
}

.post .entry {
    padding: 10px 20px 15px 20px;
    text-align: justify;
}

```

Οι τελευταίοι κανόνες αφορούν το div με class post και τα p και div μέσα σε αυτό (με class meta και entry αντίστοιχα). Είναι προφανείς πως οι περισσότερες δηλώσεις αφορούν την απεικόνιση του περιεχομένου. Παρόλα αυτά οι ιδιότητες που χρησιμοποιούνται δεν είναι διαφορετικές από αυτές που συναντώνται στα προηγούμενα σημεία της σελίδας. Πέραν των margin, padding, color, background, height εμφανίζονται ιδιότητες που αφορούν τη γραμματοσειρά όπως οι font-family, font-style για επιλογή γραμματοσειράς/στυλ, και το κείμενο όπως η text-align που θέτει αριστερή στοίχιση.

## To footer

Το υποσέλιδο είναι μια περιοχή στην οποία συνήθως εμφανίζεται κάποιος υπότιτλος που αφορά το σχεδιασμό του site και κάποια στοιχεία επικοινωνίας. Και σε αυτή την περίπτωση χρησιμοποιείται ένα div, όπως φαίνεται παρακάτω:

```
<div id="footer">
  <p>Περιεχόμενο <a href="#">σύνδεσμος</a></p>
</div>
```

Ο κώδικας CSS για το υποσέλιδο:

```
#footer {
  width: 100%;
  background: #2C2C2C;
  height: 50px;
  margin: 0 auto;
  padding: 0px 0 15px 0;
  font-family: Arial, Helvetica, sans-serif;
}
```

Και σε αυτή τη δήλωση αποδίδεται μια τιμή για το πλάτος καθώς και το φόντο, το ύψος, τα περιθώρια και τη γραμματοσειρά.

```
#footer p {
  margin: 0;
  padding-top: 20px;
  line-height: normal;
  font-size: 9px;
  text-transform: uppercase;
  text-align: center;
}

#footer a {
  color: #969696;
}
```

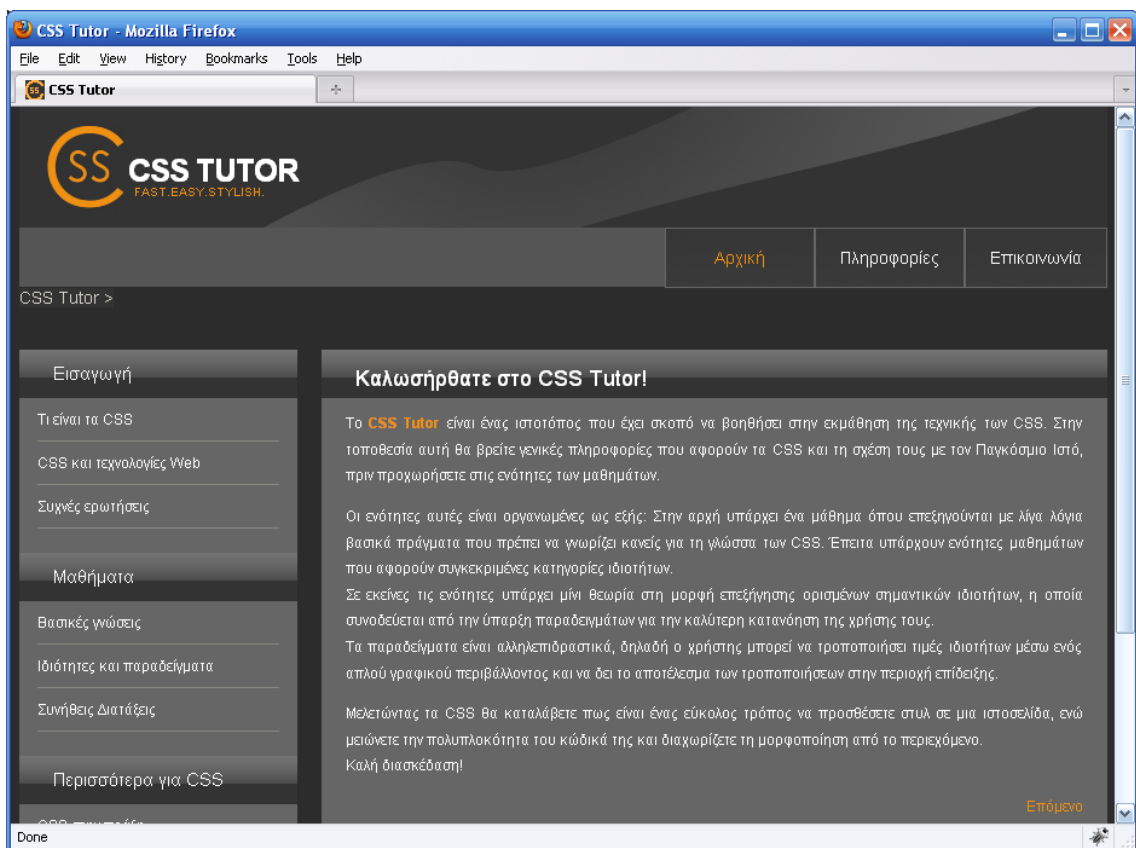
Τέλος, ορίζεται η μορφή του περιεχομένου, δηλαδή της παραγράφου και του συνδέσμου στο υποσέλιδο.

Στην ενότητα αυτή παρουσιάστηκαν τα πιο χαρακτηριστικά κομμάτια κώδικα HTML και CSS που ορίζουν τη διάταξη του CSS Tutor, με σκοπό να αποκτηθεί μια εικόνα για αυτό που βρίσκεται «πίσω» από το αποτέλεσμα.

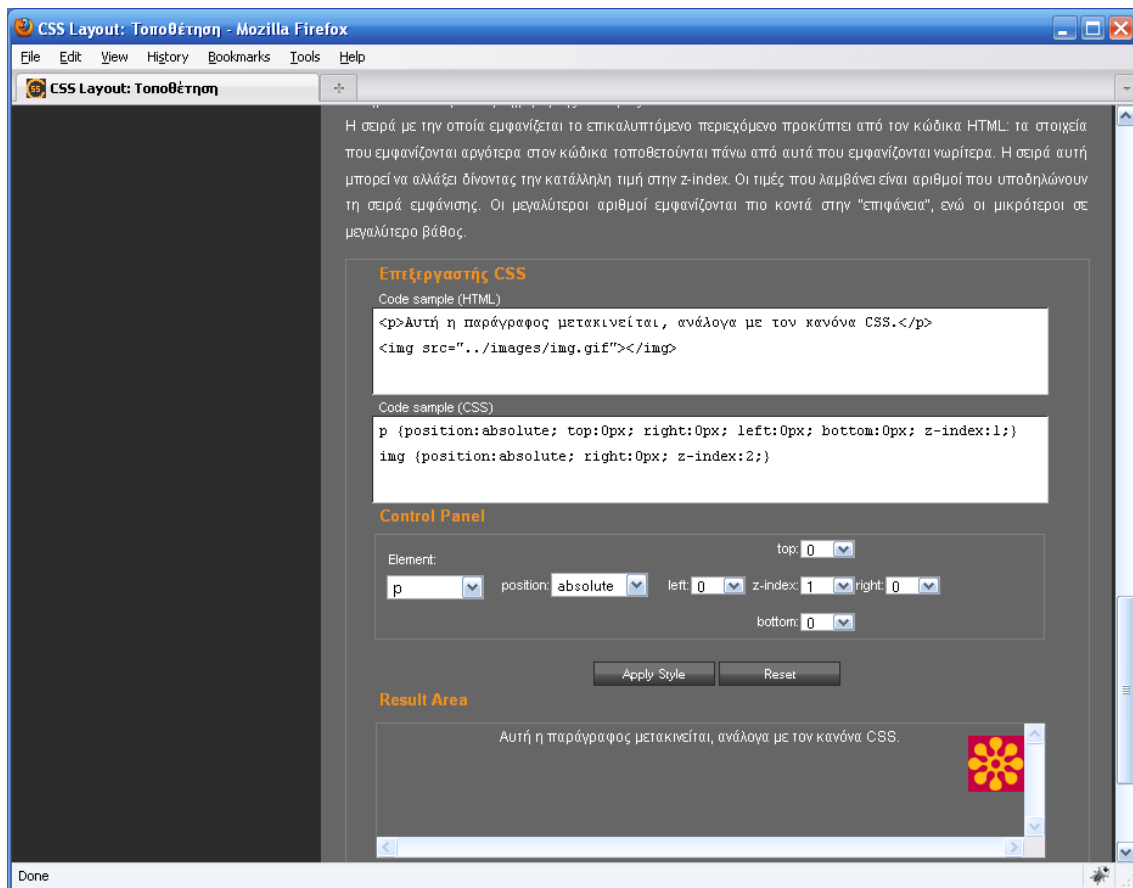
## 3.2 Μορφοποίηση Προτύπου

Αφού επελέγη το template, εν συνεχεία έγιναν αλλαγές ώστε να ταιριάζει καλύτερα στο ύφος και το περιεχόμενο του θέματος της εργασίας, σύμφωνα με την προσωπική εκτίμηση του σχεδιαστή αλλά και το στυλ ιστοσελίδων που κυκλοφορούν στο web. Στην παρούσα ενότητα γίνεται μια σύντομη επισκόπηση της νέας όψης της ιστοσελίδας καθώς και των τροποποιήσεων στον κώδικα CSS που έφεραν αυτό το αποτέλεσμα.

Οι εικόνες που ακολουθούν είναι στιγμιότυπα από τη χρήση του site. Συγκεκριμένα στην εικόνα 5 βλέπετε την αρχική σελίδα ενώ στην εικόνα 6 βλέπετε μια από τις σελίδες που περιέχουν υλικό καθώς και την αλληλεπιδραστική περιοχή παραδείγματος (στο κάτω μέρος).



Εικόνα 5: Η μορφή της ιστοσελίδας με την εφαρμογή των νέων κανόνων CSS



Εικόνα 6: Μια από τις σελίδες, με θεωρία στο πάνω μέρος και την εφαρμογή στο κάτω

Η πιο μεγάλη αλλαγή αφορά στα χρώματα της ιστοσελίδας, όπου έχουν επιλεγεί αποχρώσεις του γκρι για το φόντο ενώ το κείμενο έχει χρώμα είτε λευκό είτε πορτοκαλί. Στις ακόλουθες υποενότητες γίνεται αναφορά στις αλλαγές που έλαβαν χώρα.

### 3.2.1 Διάταξη

Είναι φανερό από τις εικόνες ότι δεν υπάρχουν αλλαγές στη διάταξη της ιστοσελίδας αυτό καθαυτό. Τα τέσσερα τμήματα της σελίδας διατηρήθηκαν ως είχαν.

Παρόλα αυτά υπήρξαν κάποιες μικρές αλλαγές στον κώδικα HTML. Αυτές αφορούν κυρίως κώδικα που αφαιρέθηκε, όπως κάποια div στοιχεία, που κρίθηκε πως δεν είναι χρήσιμα για τις συγκεκριμένες σελίδες.

Έγιναν όμως και προσθήκες κώδικα. Συγκεκριμένα, προστέθηκε στις σελίδες με την ύλη και τα παραδείγματα κώδικας που διευκολύνει την ανάρτηση του περιεχομένου. Χαρακτηριστικό παράδειγμα αποτελεί το παρακάτω απόσπασμα κώδικα:

```
<div class="entry">
```

```

<div class = "theory">
<p>...περιεχόμενο...</p>
<p>...περιεχόμενο...</p>
</div>

  <div class = "example">
    <div class = "codeHTML">
      HTML κώδικας παραδείγματος
    </div>
    <div class = "code">
      CSS κώδικας παραδείγματος
    </div>
    <div class = "gui">
      Γραφικό περιβάλλον (controls) παραδείγματος
    </div>
    <div class = "result">
      Περιοχή επίδειξης αποτελέσματος
    </div>
  </div>
</div>

```

Όλα τα παραπάνω βρίσκονται εντός ενός div με id entry, όπως θα θυμάστε από την προηγούμενη ενότητα. Αυτό που καταφέρνει ο κώδικας αυτός είναι να χωρίζει την σελίδα όπου θα βρίσκεται το υλικό σε τμήματα ανάλογα με το είδος του υλικού που αναρτάται. Το div με id theory περιέχει όλη τη θεωρία μιας σελίδας, επεξηγήσεις ιδιοτήτων, εικόνες και άλλα στοιχεία. Το div με id example είναι η περιοχή όπου ο χρήστης μπορεί να αλληλεπιδράσει με τη σελίδα, μέσω του GUI που έχει κατασκευαστεί. Μέσα σε αυτό το div βρίσκονται άλλα, που φιλοξενούν τον HTML και CSS κώδικα του εκάστοτε παραδείγματος.

Αυτή είναι η δομή όλων σχεδόν των σελίδων. Στις περιπτώσεις εκείνες που δε γίνεται επίδειξη κάποιας ιδιότητας, όπως στις ενότητες όπου υπάρχει μόνο θεωρία, απλώς λείπει το αντίστοιχο τμήμα. Έτσι ήταν πολύ απλό να αναπτυχθεί ο ιστότοπος καθώς ήταν εύκολο να προσαρμοστεί η κάθε σελίδα ανάλογα με το περιεχόμενό της.

### 3.2.2 Χρώματα, γραμματοσειρές και μεγέθη

Όπως αναφέρθηκε και στην αρχή, η πιο εμφανής διαφορά μεταξύ του προτύπου και του τελικού αποτελέσματος έγκειται στη χρήση των χρωμάτων.

Τα χρώματα που επελέγησαν για το φόντο των διαφόρων στοιχείων κυμαίνονται στις αποχρώσεις του γκρι. Άλλοτε πιο σκούρες και άλλοτε πιο ανοιχτές, αυτές οι αποχρώσεις ωφελούν στο ότι δεν αποσπάνε τον χρήστη από το αντικείμενο, ενώ παράλληλα η λευκή γραμματοσειρά προσφέρει την απαιτούμενη αντίθεση. Για τους τίτλους και κάποιους συνδέσμους μέσα στο κείμενο επελέγη το πορτοκαλί χρώμα. Ο συγκεκριμένος συνδυασμός χρωμάτων γκρι-λευκό-πορτοκαλί συναντάται στο Web και σε επαγγελματικού περιεχομένου ιστοσελίδες η ιστολόγια.

Όσο αφορά στον τύπο γραμματοσειράς, η παρακάτω δήλωση CSS εφαρμόζεται για την επιλογή του:

```
font-family: Arial, Helvetica, sans-serif;
```

Η δήλωση βρίσκεται εντός ενός κανόνα που έχει για selector το body του εγγράφου, ώστε όλα τα στοιχεία να «αρχικοποιούνται» με αυτή τη γραμματοσειρά. Ο λόγος που υπάρχουν τρεις *οικογένειες γραμματοσειρών* είναι για να αντιμετωπιστεί το ενδεχόμενο να λείπει μία από το σύστημα του χρήστη. Έτσι φορτώνεται η δεύτερη, ενώ αν δεν υπάρχει επιλέγεται η πιο κατάλληλη *sans-serif* γραμματοσειρά του συστήματος που τρέχει το site.

Τα μεγέθη των γραμματοσειρών είναι κατα βάση 12px, ένα φυσιολογικό μέγεθος για ανάγνωση. Σε κάποια σημεία στη σελίδα το μέγεθος μεγαλώνει ή μικραίνει ανάλογα με το σκοπό που εξυπηρετεί το κείμενο (π.χ. τίτλος, ετικέτα).

Όσο αφορά στα πλάτη των στοιχείων, αυτά έχουν κατα κανόνα μείνει ίδια με του προτύπου.

## 3.3 Πλοήγηση στις ιστοσελίδες

Η ευκολία και η άνεση στην πλοήγηση σε ένα website είναι κρίσιμες για την επιτυχία του, τουλάχιστον στο σχεδιαστικό κομμάτι. Σε πολλές περιπτώσεις μάλιστα, όπως προκύπτει από έρευνες, μια ιστοσελίδα μπορεί να χάσει μεγάλο ποσοστό επισκεπτών ή αντιστοίχως να κερδίσει, εάν δεν παρέχει έναν απλό τρόπο στη μεταφορά από ένα σημείο σε ένα άλλο.



Στο CSS Tutor υπάρχουν κατα βάση τρεις τρόποι για να ανοίξει ο χρήστης κάποια σελίδα. Συγκεκριμένα, πέρα από τα διάφορα links μέσα στο περιεχόμενο που οδηγούν σε άλλες σελίδες του site, υπάρχουν το sidebar, οι σύνδεσμοι «Επόμενο» και «Προηγούμενο» στο τέλος κάθε σελίδας καθώς και το «μονοπάτι» που έχει διασχίσει ο χρήστης, γνωστό και ως breadcrumb, στο πάνω και αριστερό μέρος της σελίδας. Μια σύντομη περιγραφή κάθε μέρους λαμβάνει χώρα παρακάτω.

### **3.3.1 Πλοήγηση με το sidebar**

Η μπάρα στο αριστερό μέρος της σελίδας είναι ο πρωταρχικός τρόπος πλοήγησης στο CSS Tutor. Από αυτό το σημείο ο χρήστης μπορεί να μεταφερθεί στην ενότητα που φιλοξενεί το περιεχόμενο που τον ενδιαφέρει.

Το sidebar χωρίζεται σε τρία τμήματα που υποδεικνύουν την οργάνωση του υλικού. Η σειρά που εμφανίζονται τα τμήματα, τα οποία ξεχωρίζουν από τον τίτλο που έχει το καθένα, είναι μια λογική σειρά παρόμοια με αυτή που ακολουθεί κάποιος που βρίσκεται σε μια διαδικασία εκπαίδευσης πάνω σε κάποιο αντικείμενο. Έτσι η πρώτη ενότητα αφορά στην εισαγωγή στα CSS ενώ ακολουθεί η ενότητα των μαθημάτων. Η τρίτη ενότητα αφορά κάποια επιπλέον πράγματα σχετικά με τα CSS.

Κάθε ενότητα περιέχει τα links για το περιεχόμενο. Κάνοντας κλικ σε κάποιο από τα links ο χρήστης μπορεί να βρεθεί μπροστά είτε σε περιεχόμενο είτε σε άλλα links που θα τον οδηγήσουν σε αυτό, όπως στην περίπτωση που επιλέγει να ακολουθήσει τον σύνδεσμο «Ιδιότητες και παραδείγματα» στην ενότητα «Μαθήματα». Ως επι των πλείστων συμβαίνει το δεύτερο, δηλαδή το περιεχόμενο χωρίζεται σε αρκετές σελίδες εννοιολογικά. Έτσι οι σελίδες διατηρούνται σχετικά μικρές και δεν περιέχουν πολύ κείμενο, που να αποτρέπει τον χρήστη από το να το διαβάσει.

### **3.3.2 Πλοήγηση με τους συνδέσμους**

Στο κάτω μέρος κάθε σελίδας, μετά το τέλος του περιεχομένου, υπάρχουν δύο σύνδεσμοι: «Επόμενο» και «Προηγούμενο».

Αυτοί οι σύνδεσμοι παρέχουν τη δυνατότητα σειριακής «προσπέλασης» των σελίδων. Αυτό σημαίνει, όπως είναι προφανές, πως το «Προηγούμενο» οδηγεί στην αμέσως προηγούμενη σελίδα της ενότητας (όχι πάντα αυτή που είχε μπροστά του ο χρήστης πριν ανοίξει την τρέχουσα) ενώ αντίστοιχα ο σύνδεσμος «Επόμενο» οδηγεί στην αμέσως επόμενη σελίδα της ενότητας.

Με αυτόν τον τρόπο είναι εύκολο να ακολουθήσει κανείς τη λογική με την οποία φτιάχτηκε το site.

### 3.3.3 Πλοήγηση με το breadcrumb

Το breadcrumb είναι ένα μονοπάτι που σχηματίζεται στο πάνω μέρος της σελίδας (πάνω και αριστερά) και δείχνει τις σελίδες από τις οποίες πέρασε ο χρήστης μέχρι να φτάσει στην τρέχουσα. Συναντάται πολύ συχνά πλέον σε διάφορες ιστοσελίδες και forums.

Το breadcrumb βοηθάει στο να αντιλαμβάνεται ο χρήστης σε τι «βάθος» βρίσκεται μέσα στις διάφορες ενότητες. Κάνοντας κλικ σε κάθε σύνδεσμο, ο χρήστης μπορεί να θεωρηθεί πως «κατεβαίνει» σε μεγαλύτερο βάθος μέσα στη σελίδα. Έτσι, βλέποντας το μονοπάτι μπορεί να ξέρει ανα πάσα στιγμή το επίπεδο, σε ποιο σημείο και σε ποια ενότητα βρίσκεται.

Στην περίπτωση του CSS Tutor τα breadcrumbs είναι hardcoded, δηλαδή δεν δημιουργούνται δυναμικά. Κάθε σελίδα έχει το δικό της μέσα σε ένα στοιχείο span, που με κανόνες CSS τοποθετείται στο συγκεκριμένο σημείο της σελίδας.

## 3.4 Σύνοψη

Σε αυτή την ενότητα έγινε μια περιγραφή του σχεδιαστικού κομματιού στην διαδικασία κατασκευής της σελίδας. Παρουσιάστηκε το πρότυπο που λειτούργησε ως «καλούπι» για όλες τις σελίδες του ιστότοπου. Συγκεκριμένα, παρατέθηκαν κάποια τμήματα HTML και CSS κώδικα, αυτά που σχηματίζουν τον σκελετό της σελίδας. Ακολούθησε μια σύντομη περιγραφή αυτών των τμημάτων.

Η δεύτερη ενότητα του κεφαλαίου αφορά τις μορφοποιήσεις που έγιναν στο πρότυπο ώστε να έρθει στην τελική του μορφή. Και πάλι παρουσιάστηκαν συγκεκριμένα αποσπάσματα κώδικα και κάποιοι κανόνες CSS.

Τέλος, έγινε αναφορά στους τρόπους με τους οποίους μπορεί ο χρήστης να πλοηγηθεί στον ιστότοπο.

# 4 Προγραμματισμός του ιστότοπου

Στην προηγούμενη ενότητα παρουσιάστηκε κώδικας για τη διάταξη και τη διακόσμηση των σελίδων του ιστότοπου. Στο παρόν κεφάλαιο θα λάβει χώρα με παρόμοιο τρόπο μια περιγραφή του κώδικα που παρέχει τη λειτουργικότητα του site.

## 4.1 Η βιβλιοθήκη JQuery

Αρχικά είναι απαραίτητο να παρατεθούν κάποιες πληροφορίες σχετικά με το framework *JQuery*. Αυτή η βιβλιοθήκη, παρέχει ένα σημαντικό αριθμό από μεθόδους, αρκετές εκ των οποίων χρησιμοποιούνται στο script που εκτελείται στο site.

### 4.1.1 Επιλογή framework

Το συγκεκριμένο framework είναι πλέον πολύ δημοφιλές στο διαδίκτυο. Γενικότερα τα frameworks της Javascript απλοποιούν σε μεγάλο βαθμό τη διαδικασία συγγραφής κώδικα, καθώς δεν απαιτείται από τον προγραμματιστή να παράγει λειτουργικότητα από το μηδέν. Αντιθέτως αυτά του παρέχουν έτοιμες μεθόδους για πολλές και διαφορετικές χρήσεις.

Το API (Application Programming Interface) του JQuery χωρίζει τις μεθόδους στις εξής κατηγορίες: AJAX, Core, CSS, DOM, Events, Effects και Javascript. Σε κάθε κατηγορία ανήκουν μέθοδοι που αφορούν στην συγκεκριμένη τεχνολογία (π.χ. AJAX) ή που προσδίδουν συγκεκριμένη λειτουργικότητα. (π.χ. Effects).

Αυτό που πετυχαίνει σε μεγάλο βαθμό το συγκεκριμένο framework είναι να παρέχει τεράστια ευκολία στην προσπέλαση του DOM μιας ιστοσελίδας. Οι μέθοδοι προσπέλασης (traversal methods) του JQuery είναι αρκετές και ποικίλουν, όπως μπορεί να ποικίλει και η δομή των ιστοσελίδων. Έτσι μπορεί να ανακτηθεί κάποιο στοιχείο είτε με αναφορά στα «συγγενικά» του στοιχεία με βάση το δέντρο DOM είτε με αναφορά σε κάποιο χαρακτηριστικό του (attribute), είτε με την εφαρμογή κάποιων φίλτρων κ.ο.κ. Α

Υπάρχουν επίσης αρκετοί τρόποι με χρήση του JQuery API να τροποποιηθεί η εμφάνιση μιας σελίδας. Είτε εφαρμόζοντας κανόνες CSS από έτοιμες κλάσεις που

φορτώνουν με ένα κλικ, είτε ορίζοντας τις ιδιότητες απευθείας μέσω των ίδιων των μεθόδων που παρέχονται.

Το API ορίζει επιπλέον μεθόδους για την τροποποίηση περιεχομένου. Στοιχεία που διαθέτουν περιεχόμενο μπορούν να προστεθούν, να διαγραφούν, να τροποποιηθούν ανάλογα με τις ανάγκες του ιστότοπου, πολύ εύκολα.

Ένα ακόμα δυνατό σημείο της βιβλιοθήκης αυτής αποτελεί η δυνατότητα ανταπόκρισης στα γεγονότα, καθώς διαθέτει σημαντικό αριθμό μεθόδων για την «σύλληψη» ενός γεγονότος. Με αυτό τον τρόπο είναι εύκολο να κατασκευασθούν *event handlers* για την αντιμετώπιση οποιουδήποτε γεγονότος.

Όπως αναφέρεται παραπάνω, το API διαθέτει και μεθόδους για την παραγωγή εφφέ. Τα εφφέ αυτά μπορεί να είναι από ένα απλό *fade* μέχρι μια πιο πολύπλοκη κίνηση (*animation*) κάποιου στοιχείου.

Επιπροσθέτως, με τη χρήση ορισμένων μεθόδων της βιβλιοθήκης βελτιώνονται και κάποιες βασικές λειτουργίες που επιτελεί γενικότερα η Javascript όπως οι επαναλήψεις, καθώς οι μέθοδοι αυτές επενεργούν πάνω σε *σύνολα* (*sets*). Η τεχνική αυτή ονομάζεται *implicit iteration* και δεν απαιτεί *loops* για να ανακτηθεί κάθε ένα στοιχείο ξεχωριστά. Έτσι το script αποκτά ένα πιο στιβαρό χαρακτήρα καθώς δε βασίζεται απαραίτητα σε προκαθορισμένους δείκτες (*indexes*), ενώ παράλληλα μειώνεται και ο όγκος του κώδικα.

Πέραν των ευκολιών που παρέχει και που αναφέρθηκαν παραπάνω, η βιβλιοθήκη είναι και απλή στην εκμάθηση και χρήση. Αυτό συμβαίνει για διάφορους λόγους, βασικότερος εκ των οποίων το ότι οι JQuery selectors χρησιμοποιούν τους CSS selectors για να γίνει ανάκτηση κάποιου στοιχείου. Αυτό σημαίνει πως κάποιος εξοικειωμένος με τη γλώσσα των CSS μπορεί εύκολα να εξοικιωθεί και με την σύνταξη στο JQuery.

Άλλος ένας λόγος που την καθιστά τόσο χρήσιμη είναι η δυνατότητα επέκτασής της. Είναι αρκετά απλό να κατασκευαστούν plugins και extensions σε JQuery που θα προσδίδουν νέα λειτουργικότητα.

Όσο αφορά στον ίδιο τον κώδικα, ο όγκος του σίγουρα μειώνεται καθώς επιτρέπεται το λεγόμενο *chaining* των μεθόδων. Αυτό σημαίνει πως πολλές μέθοδοι μπορεί να χρησιμοποιηθούν στην ίδια γραμμή και η μία να χρησιμοποιεί το αποτέλεσμα της προηγούμενης. Έτσι απαλείφεται η ανάγκη για χρήση πολλών προσωρινών μεταβλητών για τη διατήρηση κάποιου ενδιάμεσου αποτελέσματος.

Για να χρησιμοποιηθεί η βιβλιοθήκη, πρέπει να συνδεθεί το αρχείο που την περιέχει (το οποίο βρίσκεται στην ιστοσελίδα του framework) με το εκάστοτε έγγραφο της html στο οποίο θα τρέξει ο κώδικας. Η σύνδεση επιτυγχάνεται μέσω του tag `script`.

Συνοψίζοντας τα παραπάνω, το πλαίσιο JQuery παρέχει έναν αριθμό από μεθόδους διαφορετικού τύπου που επιτελούν διάφορες λειτουργίες και που απλοποιούν κατα πολύ τη συγγραφή κώδικα για οποιαδήποτε περίπτωση scripting. Βασικά στοιχεία η ευκολία στην ανάκτηση και τροποποίηση στοιχείων της σελίδας και η μείωση του όγκου του παραγόμενου κώδικα.

## 4.2 Κώδικας εφαρμογής

Σε αυτό το σημείο θα παρουσιαστεί ο κώδικας JQuery που είναι υπεύθυνος για τη λειτουργία της εφαρμογής. Πιο συγκεκριμένα ως εφαρμογή εννοείται η εφαρμογή των ιδιοτήτων στον «Επεξεργαστή CSS», που βρίσκεται στο τέλος κάθε σελίδας η οποία αφορά σε κάποιες ιδιότητες CSS. Όπως και στα προηγούμενα, έτσι και τώρα η περιγραφή θα είναι σύντομη και περιεκτική ενώ χωρίζεται σε υποενότητες.

### 4.2.1 Το αρχείο `jtutor.js`

Όπως αναφέρθηκε στο κεφάλαιο 3, το αρχείο CSS περιέχει τους κανόνες για όλα τα στοιχεία της σελίδας και συνδέεται με την κάθε σελίδα μέσω του tag `link`. Με παρόμοιο τρόπο, ο κώδικας JQuery βρίσκεται σε ένα εξωτερικό αρχείο με όνομα `jtutor.js` το οποίο συνδέεται με την κάθε σελίδα μέσω του tag `script`.

Μέσα στο αρχείο βρίσκεται όλος ο κώδικας που αφορά τον ιστότοπο. Συγκεκριμένα στο αρχείο υπάρχουν κάποια plugins που χρησιμοποιούνται στη σελίδα και ο κώδικας που επιτελεί τις λειτουργίες του site.

Οι λειτουργίες αυτές αφορούν ανταπόκριση σε γεγονότα. Το βασικό γεγονός που πρέπει να λάβει χώρα για να εκτελεστεί **οποιοσδήποτε** κώδικας είναι η φόρτωση της σελίδας. Άλλα γεγονότα αποτελούν το κλικ στα κουμπιά «Apply» και «Reset». Αυτά είναι τα τρία βασικά γεγονότα που προκαλούν την εκτέλεση κώδικα όσο αφορά στο κομμάτι της εφαρμογής. Ο κώδικας που τρέχει για τα διάφορα εφέ στη σελίδα θα παρουσιαστεί στην επόμενη ενότητα.

## Το script κατά τη φόρτωση

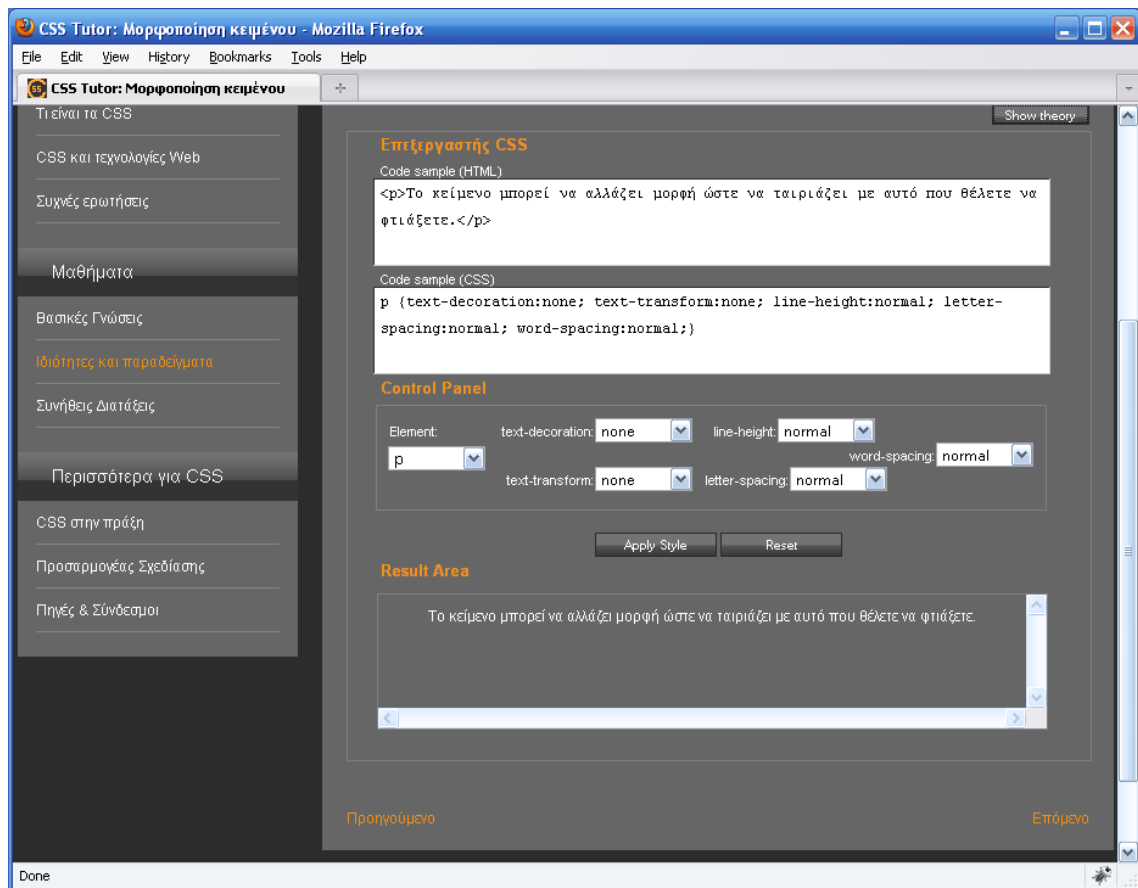
Λέγοντας φόρτωση της σελίδας εννοείται η διαδικασία του να απεικονιστούν όλα τα στοιχεία της σελίδας από τον browser. Αυτή η διαδικασία του *rendering* παίρνει κάποιο χρόνο.

Σε αυτό το χρονικό διάστημα της φόρτωσης δεν μπορεί να εκτελεστεί κάποιος κώδικας. Αυτό συμβαίνει και διότι δεν υπάρχει λόγος να υπάρχει script προς εκτέλεση, καθώς έστω για λίγο κάποια τμήματα της σελίδας θα λείπουν, και για λόγους συνέπειας εφόσον δεν υπάρχει περίπτωση να γίνεται αναφορά σε αντικείμενο που δεν υπάρχει στη σελίδα.

Υπάρχει λοιπόν μια δήλωση που επιτρέπει την εκτέλεση κώδικα μόνο αφού το έγγραφο HTML έχει φορτωθεί πλήρως από τον browser. Αφού γίνει η απεικόνιση και ο χρήστης μπορεί πλέον να δει τη σελίδα στην οθόνη, τότε έχει τη δυνατότητα να χρησιμοποιήσει το γραφικό περιβάλλον των παραδειγμάτων και να εφαρμόσει τους κανόνες του παραδείγματος.

Υπάρχει ένα κομμάτι κώδικα που εκτελείται αμέσως μόλις φορτώσει η σελίδα και δεν περιμένει τον χρήστη να κάνει κλικ σε κάποιο κουμπί. Αυτός ο κώδικας χρησιμοποιείται για να γίνει αρχικοποίηση της περιοχής επίδειξης του αποτελέσματος. Στην εικόνα 7 φαίνεται ο επεξεργαστής CSS με την περιοχή επίδειξης. Θα παρατεθεί σε αυτό το σημείο ένα τμήμα του script που εκτελείται αμέσως μόλις φορτωθεί η σελίδα.

```
$(document).ready(function()  
{  
    var defaultHTML = $('.codeHTML').text();  
    $('select').attr({selectedIndex: "0"});  
    $('.result').html(defaultHTML);  
    $('.code .element').each(function(i)  
    {  
        var elem = $('.element:eq('+i+')').attr("id");  
        var rule = $('.code #' + elem + ' .ruleBody').text();  
        $('.result ' + elem).attr({style: rule});  
    });  
}
```



Εικόνα 7: «Result Area», η περιοχή όπου φαίνεται το αποτέλεσμα του κώδικα που βρίσκεται στον «Επεξεργαστή CSS».

Το πρώτο πράγμα που παρατηρείται είναι πως όλος ο κώδικας βρίσκεται μέσα στη δήλωση  $\$(document).ready$  που εξασφαλίζει πως ό,τι εκτελεστεί θα λάβει χώρα μετά την πλήρη φόρτωση της σελίδας, δηλαδή του αρχείου HTML. Αυτό ισχύει ακόμα και για τις μεθόδους που έχουν οριστεί να είναι event handlers για το κλικ σε κάποιο κουμπί.

Κάτι ακόμα που παρατηρεί κανείς είναι η χρήση των CSS selectors σε κώδικα JQuery. Είναι προφανές για παράδειγμα, πως η δήλωση  $\$('.result')$  αφορά το div με class result, όπως παρουσιάστηκε στον κώδικα της ενότητας 3.2.1 «Διάταξη».

Στο συγκεκριμένο απόσπασμα κώδικα γίνεται αρχικοποίηση, όπως αναφέρθηκε παραπάνω, της περιοχής επίδειξης με τον κώδικα CSS και HTML που βρίσκεται μέσα στα αντίστοιχα div στοιχεία της περιοχής παραδείγματος.

Πιο αναλυτικά, πρώτα αποθηκεύεται σε μια μεταβλητή ο αρχικός κώδικας HTML που βρίσκεται στο div με class *codeHTML*. Γι' αυτό χρησιμοποιείται η μέθοδος *text()* του JQuery που επιστρέφει το κείμενο που περιέχει ένα στοιχείο. Στη συνέχεια όλα τα list boxes του γραφικού περιβάλλοντος αρχικοποιούνται στην πρώτη τιμή του δείκτη

τους *selectedIndex*. Η μεταβλητή *defaultHTML* που κρατά τον κώδικα που προαναφέρθηκε χρησιμοποιείται εν συνεχεία για να αποδώσει περιεχόμενο στην περιοχή επίδειξης που έχει κλάση *result* (να εμφανίσει δηλαδή τα στοιχεία). Αυτό επιτυγχάνεται επειδή δίνεται ως όρισμα στην μέθοδο *html()*, η οποία εφαρμόζει τον κώδικα HTML που τις δίνεται παραμετρικά.

Μέχρι εδώ δεν έχει αποδοθεί ακόμα το στυλ που ορίζουν οι κανόνες CSS εντός του αντίστοιχου *div* στην περιοχή παραδείγματος. Αυτό συμβαίνει μέσω μιας επαναληπτικής διαδικασίας. Αυτή η διαδικασία επιτελείται με τη μέθοδο *each()*. Η μέθοδος αυτή επιτρέπει την εκτέλεση μιας εντολής ή ομάδας από εντολές πάνω σε έναν αριθμό από στοιχεία. Στην προκειμένη περίπτωση τα στοιχεία αυτά είναι όλα τα στοιχεία με κλάση *element* που βρίσκονται μέσα στο *div* με κλάση *code*.

Μέσα στην ίδια την *each()* λαμβάνουν χώρα δύο ενέργειες: Πρώτα γίνεται το ταίριασμα μεταξύ του κανόνα CSS και του στοιχείου στο οποίο απευθύνεται. Έτσι το στυλ αποδίδεται σε αυτό *μονάχα* το στοιχείο. Έπειτα το στυλ αυτό εφαρμόζεται. Η πρώτη ενέργεια γίνεται στις πρώτες δύο γραμμές μέσα στην *each()*. Στην πρώτη γραμμή η μεταβλητή *elem* κρατάει το *id* του στοιχείου που προσπελαύνεται εκείνη τη στιγμή. Στην δεύτερη γίνεται η αντιστοίχιση που προαναφέρθηκε, το ταίριασμα δηλαδή με τον κανόνα CSS, ο οποίος και αποθηκεύεται στην μεταβλητή *rule*.

Εφόσον έχουν συμβεί τα παραπάνω, ο κανόνας πλέον εφαρμόζεται στο κατάλληλο στοιχείο (που εξασφαλίζεται με χρήση της *elem* μέσα στον *selector*) με την μέθοδο *attr()*. Ουσιαστικά στο χαρακτηριστικό (*attribute*) *style* που διαθέτει κάθε στοιχείο της HTML τίθεται ως περιεχόμενο η συμβολοσειρά που υποδηλώνει η μεταβλητή *rule*. Αυτή η συμβολοσειρά είναι ο κανόνας CSS που σχηματίστηκε προηγουμένως.

Τα παραπάνω εκτελούνται κάθε φορά που φορτώνει κάποια από τις σελίδες, που παρουσιάζουν ιδιότητες στην ενότητα «Μαθήματα».

Πρέπει να σημειωθεί πως για να λειτουργήσει το παραπάνω script (όπως και αυτά που θα αναφερθούν παρακάτω) πρέπει η περιοχή παραδείγματος να έχει μια **συγκεκριμένη** δομή. Αυτό σημαίνει πως οι κλάσεις *.code*, *.element* καθώς και άλλες που περιέχονται μέσα στη σελίδα είναι αναγκαίο να βρίσκονται σε αυτή τη θέση για να δουλέψει σωστά η προσπέλαση (*traversing*) των στοιχείων. Βασίζεται λοιπόν στην ανάκτηση στοιχείων του DOM της σελίδας. Η δομή που πρέπει να υπάρχει επεξηγείται στα σχόλια του κώδικα, καθώς και στα ίδια τα HTML έγγραφα.



## Το κουμπί Apply

Κάνοντας κλικ στο κουμπί «Apply» πρέπει άμεσα να εφαρμόζονται οι κανόνες CSS που βρίσκονται μέσα στο `div` με κλάση `.code` πάνω στον HTML κώδικα του παραδείγματος.

Ο κώδικας JQuery που επιτελεί αυτή τη λειτουργία δε διαφέρει στην πραγματικότητα από αυτόν που εκτελείται κατά τη φόρτωση της σελίδας, καθώς έχουν τον ίδιο ρόλο. Η μόνη διαφορά όπως είναι προφανές έγκειται στο γεγονός (event) που ενεργοποιεί την εκτέλεση του κώδικα, όπου σε αυτή την περίπτωση είναι το κλικ στο κουμπί.

Ο κώδικας είναι ο παρακάτω:

```
$('.apply').bind('click', function()
{
    $('.code .element').each(function(i)
    {
        var elem = $('.element:eq('+i+')').attr("id");
        var rule = $('.code #' +elem+ ' .ruleBody').text();
        $('.result '+elem).attr({style: rule});
    })
});
```

Στην πρώτη γραμμή γίνεται η σύνδεση του script με το γεγονός που το προκαλεί. Το κομμάτι `$('.apply')` χρησιμοποιεί τον class selector που αντιστοιχεί στο κουμπί «Apply» για να γίνει αναφορά σε αυτό. Στη συνέχεια με τη μέθοδο `bind()` γίνεται η αναφορά στο γεγονός που πρέπει να λάβει χώρα για να εκτελεστεί η μέθοδος. Το γεγονός τίθεται πρώτο όρισμα αυτής της μεθόδου και είναι το `click`. Το δεύτερο όρισμα είναι η μέθοδος που θα εκτελεστεί. Ο κώδικας της μεθόδου είναι ο ίδιος με αυτόν της προηγούμενης υποενότητας, για την εφαρμογή του στυλ στην περιοχή επίδειξης.

Σημειώνεται εδώ πως η σύνταξη για τη σύνδεση του γεγονότος που αφορά ένα στοιχείο με την μέθοδο αντιμετώπισής του γίνεται και με άλλους τρόπους.

## Το κουμπί Reset

Το κουμπί Reset βρίσκεται δίπλα στο Apply στον επεξεργαστή CSS και όπως υποδηλώνει το όνομά του επαναφέρει τα πεδία στις αρχικές τους τιμές. Λέγοντας πεδία εννοείται το κάθε control αλλά και περιοχή που ενδέχεται να τροποποιηθεί. Έτσι το Re-

set επενεργεί στο div που περιέχει τους κανόνες CSS (μπορεί να τροποποιηθεί με αλλαγή τιμών στις ιδιότητες), στα list-boxes του γραφικού περιβάλλοντος καθώς και στο αποτέλεσμα εκτέλεσης του κώδικα μέσα στην περιοχή επίδειξης.

Το γεγονός που ενεργοποιεί τον κώδικα είναι προφανώς και πάλι το κλικ του ποντικιού. Έτσι το script έχει ως εξής:

```
$('.reset').bind('click', function()
{
    $('.select').attr({selectedIndex: "0"});
    $('.color').val("FFFFFF");
    $('.code .element').each(function(i)
    {
        $('.code #' + elements[i] + ' .ruleBody').html(defaultCSS[i]);
        $('.result ' + elements[i]).attr({style: defaultResult[i]});
    });
});
```

Αρχικά γίνεται η σύνδεση με το γεγονός. Όπως προηγουμένως χρησιμοποιείται ο class selector και το γεγονός *click*.

Εντός της μεθόδου, οι πρώτες δύο γραμμές θέτουν τις αρχικές τιμές στα στοιχεία του γραφικού περιβάλλοντος. Συγκεκριμένα η πρώτη στα list-boxes (στοιχεία τύπου select) ενώ η δεύτερη στο color-picker στοιχείο που υπάρχει σε μερικές σελίδες.

Η *each()* χρησιμοποιείται ξανά σε αυτό το script για τον ίδιο λόγο που χρησιμοποιείται και στα προηγούμενα: να προσπελάσει όλα τα στοιχεία στην περιοχή παραδείγματος. Έτσι η πρώτη γραμμή μέσα στην *each()* αποκαθιστά τον αρχικό κώδικα του κανόνα CSS (όπως αυτός υπάρχει γραμμένος μέσα στο κάθε HTML έγγραφο πριν τον αλλάξει ο χρήστης). Η δεύτερη γραμμή αποκαθιστά το αρχικό αποτέλεσμα στην περιοχή επίδειξης, εφαρμόζοντας τον αρχικό κανόνα στα στοιχεία του παραδείγματος.

Διευκρινίζεται πως ο πίνακας *elements* αποθηκεύει τους δείκτες για κάθε στοιχείο που βρίσκεται στην περιοχή παραδείγματος (πρακτικά κάθε στοιχείο που υπάρχει στην περιοχή επίδειξης, καθώς αυτά τροποποιούνται από τους κανόνες CSS). Ο δείκτης *i* προκύπτει από την *each()* που τον απαιτεί ως όρισμα στη μέθοδο που εκτελείται.

## Ο κώδικας για τα στοιχεία του γραφικού περιβάλλοντος (GUI controls)

Το γραφικό περιβάλλον στις σελίδες όπου επεξηγούνται οι διάφορες ιδιότητες CSS απαρτίζεται από διάφορα list boxes. Αυτά περιέχουν κάποιες τιμές που μπορεί να αλλάξει ο χρήστης. Με κάθε αλλαγή στην τιμή κάποιου control θα πρέπει να αλλάξει και ο κανόνας CSS του παραδείγματος, έχοντας πλέον τη νέα τιμή του αντίστοιχου control.

Επιπλέον, στο αριστερό τμήμα του γραφικού περιβάλλοντος υπάρχει ένα συγκεκριμένο list box που χρησιμοποιείται για να επιλεγεί το στοιχείο που θέλει να τροποποιήσει ο χρήστης, από το σύνολο των στοιχείων του παραδείγματος.

Για να διατηρηθεί η συνέπεια μεταξύ των τιμών των controls και των κανόνων CSS του παραδείγματος λειτουργεί ένας κώδικας. Αυτό το τμήμα κώδικα αναλαμβάνει να διαβάζει τις τιμές από τα controls και να δίνει την τρέχουσα τιμή τους στον κανόνα CSS που αφορά το στοιχείο που είναι επιλεγμένο εκείνη τη στιγμή. Ο κώδικας αυτός εξηγείται παρακάτω:

```
$( 'select, input' ).bind( 'change', function()
{
    var element = $( '#elementBox' ).val();
    var selName = $( this ).attr( "name" );
    var selVal = $( this ).val();
    $( '.code #' + element + ' .ruleBody #' + selName ).text( selVal );
});
```

Το παραπάνω script αφορά όλα τα στοιχεία *select* του γραφικού περιβάλλοντος. Στην πράξη πρόκειται για όλα τα στοιχεία καθώς χρησιμοποιήθηκε αποκλειστικά αυτό το HTML tag για την κατασκευή τους. Επιπλέον αφορά και το color-picker που είναι τύπου *input*. Σχετικά με τη σύνταξη στην πρώτη γραμμή του script, εδώ χρησιμοποιείται ένας πολλαπλός selector για να επιλεγούν και τα δύο στοιχεία σε μία γραμμή.

Το γεγονός που πρέπει να λάβει χώρα στα list boxes ή το color-picker είναι η *αλλαγή* στην τρέχουσα τιμή τους (γεγονός *change*). Η μέθοδος που εκτελείται λειτουργεί με τον ακόλουθο τρόπο: Αρχικά σε μια μεταβλητή *element* αντιστοιχίζεται το στοιχείο που είναι εκείνη τη στιγμή επιλεγμένο στο αντίστοιχο control (με id *elementBox*). Εν συνεχεία στις μεταβλητές *selName* και *selVal* αντίστοιχα αποθηκεύονται το όνομα και η νέα τιμή της ιδιότητας που έχει τροποποιηθεί από το χρήστη. Με τη

χρήση του δείκτη *this* γίνεται αναφορά κάθε φορά στο control που αλλάζει τιμή. Τέλος, η νέα τιμή που έχει το control τίθεται στο σωστό σημείο στον κανόνα CSS. Για να επιτευχθεί αυτό χρησιμοποιείται η συγκεκριμένη δομή της περιοχής παραδείγματος για την οποία έγινε λόγος προηγουμένως. Ο selector της τελευταίας εντολής του script εξασφαλίζει ότι θα αλλάξει το κείμενο του κανόνα στο σημείο που βρίσκεται η ιδιότητα που έχει λάβει καινούρια τιμή. Στη συνέχεια με τη μέθοδο *text()* αυτή η τιμή τίθεται στον κανόνα.

Συγκεκριμένα για το list-box από όπου επιλέγεται το στοιχείο του οποίου η απεικόνιση θα τροποποιηθεί υπάρχει ένα επιπλέον script:

```
$('#elementBox').bind('change', function()
{
    var element = $(this).val();
    $('.numeric, .text, .color').each(function(i)
    {
        var property = $(this).attr("name");
        var value = $('.code #' + element + ' #' + property).text();
        $(this).val(value);
    });
});
```

Σε αυτό το τμήμα κώδικα χρησιμοποιείται και πάλι το γεγονός *change*. Και αυτή τη φορά η μέθοδος διατηρεί στη μεταβλητή *element* το επιλεγμένο στοιχείο. Ύστερα χρησιμοποιείται ένας πολλαπλός selector για να επιλέξει όλα τα υπόλοιπα list-boxes του GUI του παραδείγματος. Αυτά τα list-boxes έχουν κάποια κλάση, ανάλογα με το είδος των τιμών που περιέχουν (*numeric* για τα αριθμητικά, *text* για τα κειμένου, *color* για το color-picker). Για κάθε ένα από αυτά, όπως υποδηλώνει η χρήση της *each()*, οι μεταβλητές *property*, *value* διατηρούν το όνομα της ιδιότητας στην οποία αντιστοιχούν και την τιμή που αυτή έχει **μέσα στον κανόνα CSS** του παραδείγματος. Η τελευταία εντολή του script θέτει την παραπάνω τιμή ως τιμή του list-box.

Ουσιαστικά αυτό που πετυχαίνει ο παραπάνω κώδικας, είναι να υπάρχει συνέπεια και αντιστόχιση ανάμεσα σε αυτό που υπάρχει στον συντεταγμένο κανόνα CSS του παραδείγματος και στα list-boxes του γραφικού περιβάλλοντος. Για παράδειγμα, αν το στοιχείο *div* του παραδείγματος έχει font-size 12px στον κανόνα CSS τότε αυτή η τιμή θα πρέπει να είναι η επιλεγμένη στο list-box με τις τιμές για την ιδιότητα font-size. Με

χρήση του παραπάνω κώδικα, κάθε φορά που ο χρήστης επιλέγει ένα στοιχείο προς τροποποίηση, οι τιμές των ιδιοτήτων του διαβάζονται από τον κανόνα CSS και τοποθετούνται στα controls.

## Επιλογή χρώματος από το GUI (color-picker)

Μερικές από τις ιδιότητες που αφορούν στοιχεία της σελίδας σχετίζονται με την επιλογή χρώματος. Τέτοιες ιδιότητες είναι παραδείγματος χάριν, η *color* για το χρώμα γραμματοσειράς, η *background-color* για το χρώμα φόντου και άλλες. Σε αυτές τις περιπτώσεις η επιλογή χρώματος γίνεται με τη βοήθεια ενός control που ονομάζεται *color-picker*.

Το *color-picker* που χρησιμοποιείται στις σελίδες του CSS Tutor είναι ένα plugin γραμμένο σε Javascript. Το όνομα του αρχείου είναι *jscolor.js* και βρίσκεται σε ένα κατάλογο εντός του καταλόγου που περιέχει όλα τα έγγραφα HTML της σελίδας. Η σύνδεσή του με το κάθε αρχείο που το χρησιμοποιεί γίνεται με το tag *script* στην αρχή του εγγράφου, μέσα στο τμήμα *HEAD*. Όπως ακριβώς συμβαίνει και με κάθε άλλο script δηλαδή. Η χρήση του είναι ελεύθερη και ορίζεται από την GNU Lesser General Public License. Ο δημιουργός του καθώς και τα απαιτούμενα links αναφέρονται ως σχόλια στο ίδιο το αρχείο.

Η χρήση του *color-picker* είναι πολύ απλή. Για να τοποθετηθεί στη σελίδα αρκεί να δημιουργηθεί ένα στοιχείο *input* με το χαρακτηριστικό class να έχει την τιμή *color*.

Όταν γίνεται κλικ στο *color-picker* μια παλλέτα εμφανίζεται και περιέχει όλο το φάσμα των χρωμάτων. Απο εκεί ο χρήστης επιλέγει αυτό που θέλει. Με το που επιλέγεται το χρώμα, ο κανόνας CSS του παραδείγματος πρέπει να αλλάξει για να έχει πλέον στην αντίστοιχη ιδιότητα την καινούρια τιμή για το χρώμα.

Ο παρακάτω κώδικας επιτελεί αυτή τη λειτουργία:

```
$('.color').bind('change', function()
{
    var col = "#" + $('.color').val();
    var newCode = $('.code').val().replace(/#...../, col);
    $('.code').val(newCode);
});
```

Η λογική είναι ίδια με αυτή που ακολουθείται για τις υπόλοιπες ιδιότητες. Η μεταβλητή *col* υπάρχει για να κρατάει την τιμή του χρώματος που υπάρχει στο *color-*

picker. Η *newCode* είναι όλος ο κανόνας CSS αλλά με την νέα τιμή χρώματος. Εδώ γίνεται χρήση της *replace()* και των regular expressions για να βρεθεί ακριβώς το σημείο όπου πρέπει να γίνει αντικατάσταση της παλιάς με τη νέα τιμή. Πιο αναλυτικά, εντοπίζεται το σύμβολο # μέσα στον κανόνα (καθώς σημαίνει την ύπαρξη τιμής για χρώμα) και αντικαθίστανται οι επόμενοι έξι χαρακτήρες με τους νέους έξι που έχουν επιλεγεί. Έτσι σχηματίζεται ξανά ολόκληρος ο κανόνας CSS με το καινούριο χρώμα. Η *newCode* τέλος, τίθεται ως τιμή στο div με class *code*, για να εμφανίζεται στην οθόνη ο νεοσύστατος κανόνας.

## 4.3 Κώδικας εφφέ

Τα εφφέ σε μια σελίδα αναφέρονται συνήθως σε κάποια μορφή κίνησης (animation) που προσδίδεται σε στοιχεία, ή κάποιου είδους δυναμική τροποποίηση της εμφάνισής τους. Τα εφφέ δίνουν ένα πιο ενδιαφέρον τόνο στη σελίδα και προκαλούν εντύπωση στον χρήστη. Στον σημερινό Ιστό είναι μάλλον απίθανο να συναντηθεί σελίδα που δεν διαθέτει κάποιου είδους εφφέ.

Τα εφφέ στις σελίδες του CSS Tutor είναι απλά και συμμορφώνονται με την γενικότερη εικόνα της σελίδας. Δεν υπάρχει δηλαδή κάτι το ιδιαίτερα «φантаχτερό», καθώς και το θέμα δεν ευνοεί την ύπαρξη τέτοιου είδους διακόσμησης.

Συγκεκριμένα, υπάρχουν τέσσερα είδη εφφέ: fade-in και fade-out στη σελίδα «Συχνές Ερωτήσεις» όταν γίνεται κλικ σε μια ερώτηση, fade όταν ο χρήστης περνάει τον δείκτη του ποντικιού πάνω από κάποιο link στο sidebar ή στο μενού που βρίσκεται κάτω από το header, μεγέθυνση των links όταν ο χρήστης περνάει τον δείκτη πάνω από τους συνδέσμους «Προηγούμενο» και «Επόμενο» καθώς και ένα εφφέ προεπισκόπησης των εικόνων που βρίσκονται στο site.

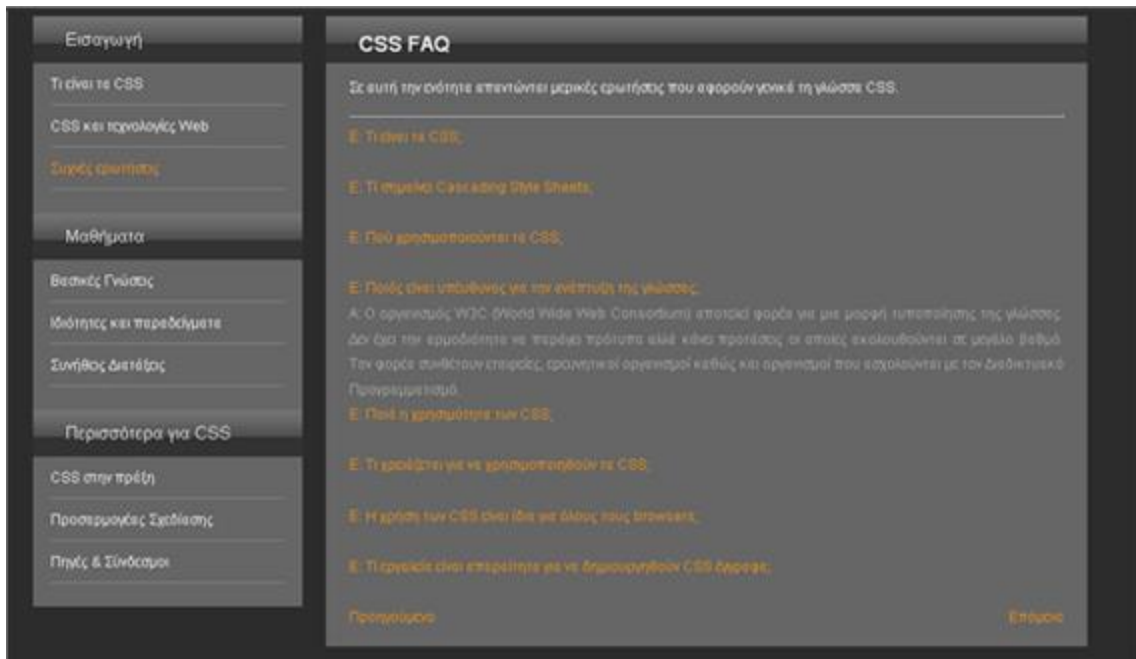
Ο κώδικας που εκτελεί όλα τα παραπάνω βρίσκεται μαζί με τον υπόλοιπο κώδικα στο αρχείο *jtutor.js*. Στις παρακάτω υποενότητες θα παρουσιαστεί αυτός ο κώδικας με τον τρόπο που παρουσιάστηκαν τα scripts μέχρι τώρα.

### 4.3.1 Fade-in/fade-out στις «Συχνές Ερωτήσεις»

Η ενότητα «Συχνές Ερωτήσεις» περιέχει σύντομες απαντήσεις σε συνήθη ερωτήματα που μπορεί να έχει κάποιος σχετικά με τη γλώσσα των CSS.

Αυτές οι ερωτήσεις είναι ορατές με το που ανοίγει η σελίδα, ενώ οι απαντήσεις δεν φαίνονται. Για να εμφανιστεί η απάντηση σε κάποια ή κάποιες από τις ερωτήσεις

πρέπει να γίνει κλικ στην αντίστοιχη ερώτηση. Στη συνέχεια η απάντηση εμφανίζεται κάτω ακριβώς από την ερώτηση με ένα εφφέ «ξεθωριάσματος» (fade-in). Στην εικόνα 8 φαίνεται ένα στιγμιότυπο όπου είναι ορατό το εφφέ.



Εικόνα 8: Μετά το κλικ σε κάποια ερώτηση η απάντηση εμφανίζεται σταδιακά, όπως φαίνεται στην εικόνα.

Ο κώδικας που είναι υπεύθυνος για αυτό είναι ο παρακάτω:

```

$ ('span.a').hide ();
$ ('span.q').bind ('click', function ()
{
    i = $ ('span.q').index ($ (this));
    var isVisible = $ ('span.a:eq ('+i+')').is (':visible');
    if (isVisible == true)
    {
        $ ('span.a:eq ('+i+')').fadeOut ('slow');
    }
    else if (isVisible == false)
    {
        $ ('span.a:eq ('+i+')').fadeIn ('slow');
    }
});

```

Τόσο οι ερωτήσεις όσο και οι απαντήσεις αποτελούνται από στοιχεία `span`. Οι ερωτήσεις έχουν στο χαρακτηριστικό `class` την τιμή `q` από το `question` ενώ οι απαντήσεις την τιμή `a` από το `answer`. Αρχικά, με τη χρήση του selector `$('.span.a')` επιλέγονται όλα τα `span` που περιέχουν απαντήσεις, και με τη μέθοδο `hide()` κρύβονται. Έτσι όταν φορτώνει η σελίδα φαίνονται μόνο οι ερωτήσεις. Στη συνέχεια γίνεται η σύνδεση του γεγονότος που προκαλεί την εμφάνιση των απαντήσεων με την μέθοδο που την πραγματοποιεί.

Περιγραφικά συμβαίνει το εξής: Χρησιμοποιείται ένας δείκτης για να υποδεικνύει σε ποιά ερώτηση έχει γίνει κλικ. Στη συνέχεια ο δείκτης αυτός χρησιμοποιείται για να εντοπιστεί η απάντηση που αντιστοιχεί στην συγκεκριμένη ερώτηση. Ταυτόχρονα λαμβάνει χώρα ένας έλεγχος για το αν εκείνη τη στιγμή είναι ορατή η απάντηση. Το αποτέλεσμα του ελέγχου, που είναι `true` η `false`, αποθηκεύεται στη μεταβλητή `isVisible`. Αν η μεταβλητή έχει τιμή `true`, σημαίνει πως η απάντηση είναι ορατή άρα αυτό που πρέπει να γίνει είναι απόκρυψή της. Αντιθέτως, εάν είναι `false` πρέπει να εμφανιστεί, καθώς σημαίνει πως είναι κρυμμένη.

Στην περίπτωση της εμφάνισης χρησιμοποιείται η `fadeIn()` με τιμή `slow` ως όρισμα. Έτσι γίνεται ορατή με το εφέ της σταδιακής εμφάνισης με σχετικά χαμηλή ταχύτητα. Στην αντίθετη περίπτωση καλείται η `fadeOut()` με το ίδιο όρισμα και η απάντηση κρύβεται σταδιακά.

### 4.3.2 Εμφέ στην πλευρική μπάρα

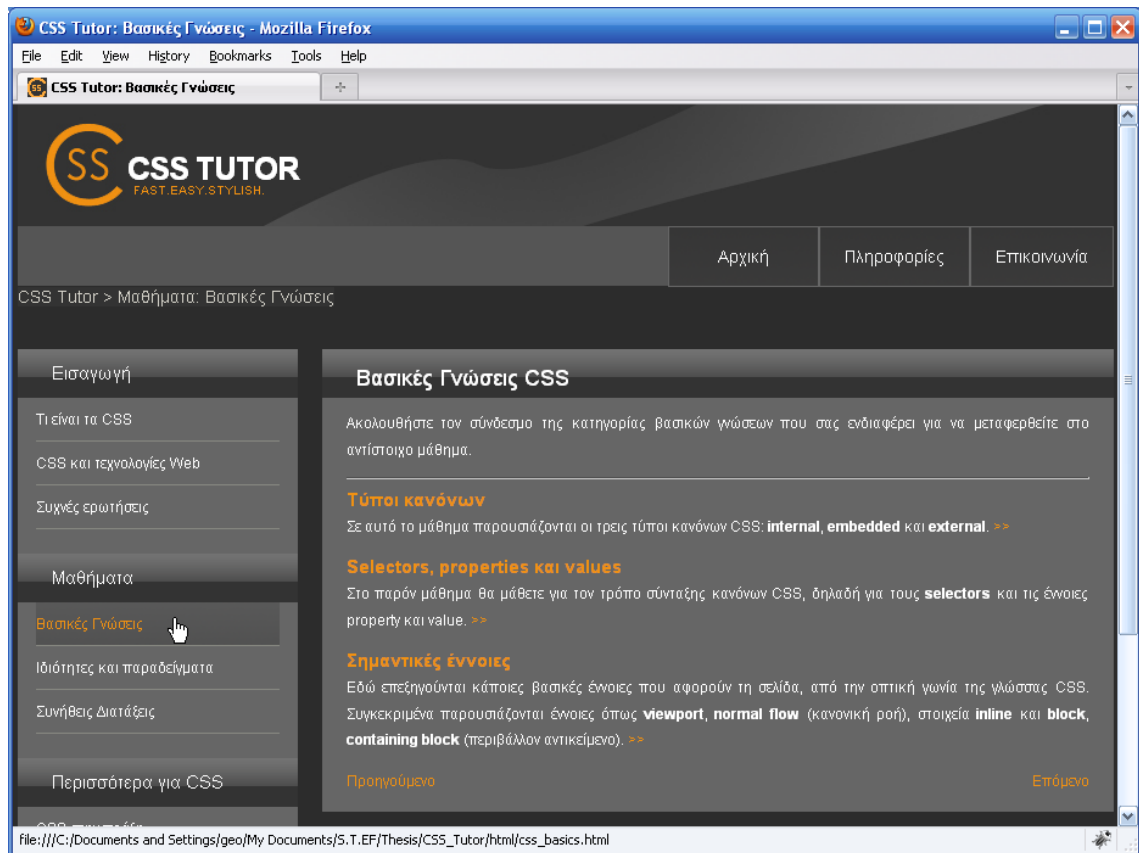
Η πλευρική μπάρα είναι ο βασικός τρόπος πλοήγησης, όπως αναφέρθηκε. Αυτό σημαίνει πως ο χρήστης του ιστότοπου θα το χρησιμοποιήσει αρκετές φορές. Συχνά σε τέτοιου είδους μενού επιλέγεται ένα είδος επισήμανσης (`highlight`) ως εμφέ, ώστε να είναι πιο εύκολο για το χρήστη να διακρίνει τις επιλογές του μενού.

Στο μενού του CSS Tutor έχει επιλεγεί ένα εμφέ επισήμανσης όπου όταν ο χρήστης σύρει το ποντίκι πάνω από ένα σύνδεσμο, το φόντο του γίνεται σταδιακά πιο σκούρο. Γίνεται χρήση του «ξεθωριάσματος» (`fade-in/fade-out`) για άλλη μια φορά. Στην εικόνα 9 φαίνεται ένα στιγμιότυπο της παραπάνω κατάστασης.

Για να επιτευχθεί το παραπάνω εμφέ χρησιμοποιείται μια εικόνα. Η εικόνα αυτή έχει χρώμα ίδιο με το χρώμα φόντου του sidebar στο πάνω μέρος, χρώμα πιο σκούρο στο κάτω και χρησιμοποιείται ως φόντο των `links`. Όταν το ποντίκι βρίσκεται πάνω από κάποιο `link`, η εικόνα στο φόντο μετατοπίζεται προς τα κάτω, κι έτσι εμφανίζεται το



σκούρο μέρος της. Αυτή η αλλαγή στην εικόνα δίνει την αίσθηση ότι το link «φωτίζεται».



Εικόνα 9: Το φόντο των συνδέσμων γίνεται πιο σκούρο όταν το ποντίκι βρίσκεται πάνω τους.

Για να επιτευχθεί η μετατόπιση της εικόνας, εκτός από το script που την επιτελεί, χρησιμοποιείται και ένα plugin που επιτρέπει την τροποποίηση της θέσης της μέσω μιας ιδιότητας. Ο κώδικας του plugin δε θα αναλυθεί, απλώς αναφέρεται πως επιτρέπει την χρήση της ιδιότητας *backgroundPosition* για να οριστούν οι συντεταγμένες μιας εικόνας ή ενός στοιχείου.

Το τμήμα κώδικα που σχετίζεται με τα παραπάνω εμφανίζεται σε αυτό το σημείο:

```
$('#sidebar li a, #menu a')  
  .css({backgroundPosition: "0 0"})  
  .hover(function()  
  {  
    $(this).stop().animate({backgroundPosition: "(0 90%)"},  
    {duration: "50"})  
  })  
  .mouseout(function())
```

```

{
    $(this).stop().animate({backgroundPosition: "(0 1%)"},
    {duration: "50"})
})

```

Αρχικά σημειώνεται πως σε αυτή την περίπτωση η σύνταξη διαφέρει από τις προηγούμενες για λόγους συντομίας. Αυτό που συμβαίνει είναι ότι χρησιμοποιείται ένας selector στην πρώτη γραμμή και σε κάθε νέα γραμμή γίνεται η σύνδεση με την εκάστοτε μέθοδο.

Με τον selector στην πρώτη γραμμή επιλέγονται όλα τα links (στοιχεία a) που βρίσκονται στο sidebar και στο menu, που είναι όπως έχει αναφερθεί το μενού με τα τρία links κάτω από το header της σελίδας.

Στην δεύτερη γραμμή γίνεται χρήση της μεθόδου *css()* που δέχεται ένα στυλ και το εφαρμόζει πάνω στο στοιχείο του selector. Εκεί γίνεται μια αρχικοποίηση της θέσης της εικόνας. Η θέση (0,0) σημαίνει πως δεν υπάρχει μετατόπιση, η εικόνα μένει στην θέση που έχει καταλάβει. Ύστερα καλείται η *hover()* που υποδεικνύει το γεγονός που προκαλεί το εφέ: το πέρασμα του δείκτη του ποντικιού πάνω από το link. Η εντολή μέσα στη *hover()* επιτυγχάνει ουσιαστικά τη μετατόπιση της εικόνας. Η συγκεκριμένη σύνταξη παρουσιάζει και το λεγόμενο *chaining* των μεθόδων, δηλαδή την «αλυσιδωτή» σύνδεσή τους. Η χρήση του *\$(this).stop()* επιβάλλεται για να εξασφαλισθεί πως το εφέ θα εκτελεστεί σωστά, χωρίς «σπαστή» κίνηση. Η *animate()* δέχεται ως όρισμα τη νέα θέση της εικόνας (*backgroundPosition* εκφρασμένη σε ποσοστά, το πρώτο για οριζόντια μετατόπιση και το δεύτερο για κάθετη) και τη διάρκεια (χρήση της *duration*) την οποία θα καταλάβει η όλη κίνηση σε χιλιοστά του δευτερολέπτου (ms). Με αυτό τον τρόπο η εικόνα μετατοπίζεται σε σημείο ώστε πλέον να είναι ορατό το σκούρο τμήμα της.

Αντίστοιχος κώδικας εκτελείται για να επανέλθει η εικόνα στην αρχική της θέση. Η διαφορά είναι πως το γεγονός αυτή τη φορά αφορά την απομάκρυνση του δείκτη του ποντικιού από το link, δηλαδή το *mouseout*.

Ο παραπάνω κώδικας αφορά, όπως προαναφέρθηκε, στα links του sidebar και του menu κάτω από το header.

### 4.3.3 Εμφάνιση στα links Προηγούμενο/Επόμενο

Όταν ο χρήστης μεταφέρει τον δείκτη του ποντικιού πάνω από έναν σύνδεσμο «Προηγούμενο» ή «Επόμενο», ο σύνδεσμος μεγενθύνεται με ένα εφέ κίνησης. Γίνεται κατα κάποιο τρόπο ένα zoom.

Ο κώδικας που επιτελεί την παραπάνω λειτουργία είναι ο παρακάτω:

```
$('.entry .next, .entry .prev').mouseover(function()
{
    $(this).stop().animate({fontSize: "15px"}, "fast");
});
$('.entry .next, .entry .prev').mouseout(function()
{
    $(this).stop().animate({fontSize: "12px"}, "fast");
});
```

Όπως είναι εύκολα αντιληπτό το script αυτό δε διαφέρει ιδιαίτερα από το παραπάνω. Χρησιμοποιούνται πολλαπλοί selectors για να επιλεγούν οι σύνδεσμοι (που έχουν κλάση *next* και *prev*) και στη συνέχεια γίνεται σύνδεση με το γεγονός *mouseover*. Μέσα στη μέθοδο χρησιμοποιείται και πάλι η *animate()* αλλά αυτή τη φορά με διαφορετικά ορίσματα. Η μεγέθυνση του link επιτυγχάνεται αυξάνοντας το μέγεθος της γραμματοσειράς. Η ιδιότητα *font-size* λαμβάνει την τιμή 15px από 12px που είναι η αρχική. Στο δεύτερο όρισμα ορίζεται η ταχύτητα με την οποία θα λάβει χώρα η μεγέθυνση. Ορίζεται εδώ να είναι γρήγορη με την τιμή *fast*.

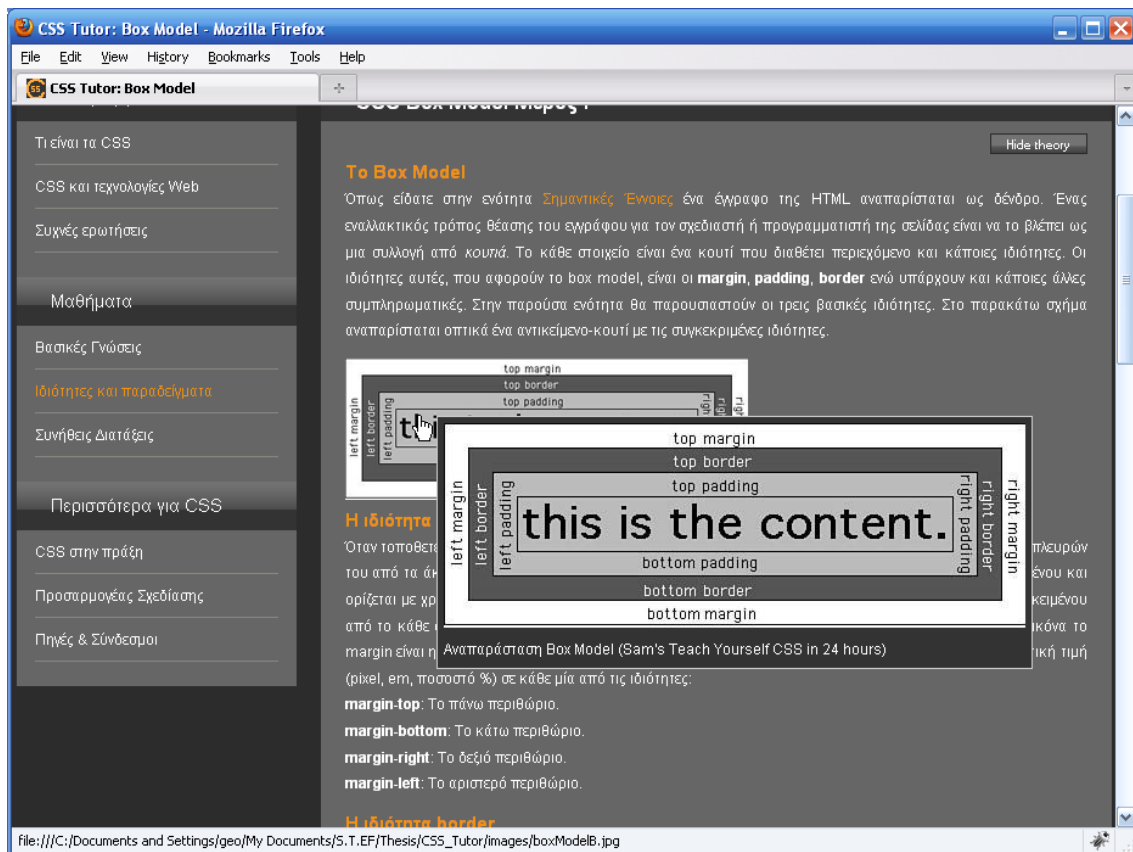
Το αντίστροφο από τα παραπάνω συμβαίνει κατά την απομάκρυνση του δείκτη από ένα link. Το μέγεθος επανέρχεται στα 12px ξανά με μεγάλη ταχύτητα.

### 4.3.4 Προεπισκόπηση εικόνων

Στις σελίδες που περιέχουν εικόνες, όταν ο χρήστης σύρει το ποντίκι πάνω από μία εξ αυτών αυτή μεγενθύνεται και διακρίνονται οι λεπτομέρειές της. Στην εικόνα 10 επιδεικνύεται αυτό το εφέ.

Για την επίτευξη του παραπάνω αποτελέσματος χρησιμοποιείται ένα plugin γραμμένο σε javascript. Ο κώδικας του plugin βρίσκεται γραμμένος στο αρχείο *jtutor.js*, πριν από τη δήλωση *\$(document).ready()*, μαζί με τα υπόλοιπα plugins. Για να χρησιμοποιηθεί όμως πρέπει να τοποθετηθεί η δήλωση *imagePreview()* στο έγγραφο, μετά το *\$(document).ready()*. Αυτή η δήλωση επιτρέπει την κλήση της μεθόδου μέσα

στο plugin όποτε το ποντίκι περνάει πάνω από μια εικόνα. Επιπλέον, για να λειτουργήσει ο παραπάνω κώδικας υπάρχουν κάποιες προϋποθέσεις, όπως το να τεθεί η τιμή *preview* στο χαρακτηριστικό class. Αυτό είναι απαίτηση του συγκεκριμένου plugin.



Εικόνα 10: Προεπισκόπηση των εικόνων στις σελίδες του CSS Tutor

Ο κώδικας του plugin είναι γραμμένος σε JQuery. Περισσότερα για αυτό μπορούν να βρεθούν στο ίδιο το αρχείο. Εδώ κρίθηκε σκόπιμο να γίνει απλώς μια αναφορά.

## 4.4 Σύνοψη

Το παρόν κεφάλαιο αφιερώθηκε στην παρουσίαση του κώδικα που εκτελείται στα διάφορα σημεία του CSS Tutor.

Αρχικά έγινε μια θεωρητική εισαγωγή στην βιβλιοθήκη JQuery που χρησιμοποιήθηκε εκτενώς. Αναφέρθηκαν βασικά πλεονεκτήματα που προκύπτουν από τη χρήση της, όπως η μείωση του όγκου του κώδικα και η ταυτόχρονη ευκολία στην

προσπέλαση του DOM μιας σελίδας. Αυτά τα πλεονεκτήματα αποτελούν και τα κριτήρια επιλογής της.

Οι επόμενες ενότητες αφορούν στην προβολή και περιγραφή του κώδικα στο script της σελίδας. Για να γίνει αυτό ο κώδικας χωρίστηκε σε δύο κατηγορίες: αυτόν που στηρίζει την εφαρμογή στις σελίδες που διαθέτουν παραδείγματα ιδιοτήτων και αυτόν που έχει ρόλο δημιουργίας εφέ. Και στις δύο περιπτώσεις υπάρχουν υποενότητες που αφορούν στα συγκεκριμένα σημεία όπου τρέχει κάποιο script.



# 5 Το σύστημα σε χρήση: Φυλλομετρητές και Blackboard

Τα περιεχόμενα αυτής της ενότητας αφορούν τις δοκιμές του ιστότοπου σε φυλλομετρητές, αφού πρώτα γίνει μια αναφορά σε αυτούς και στις μηχανές διάταξης (layout engines) που χρησιμοποιούν. Επιπλέον παρουσιάζονται οι διαφορές στην απεικόνιση των σελίδων από το εκάστοτε πρόγραμμα. Τέλος, ένα τμήμα του κεφαλαίου αναφέρεται στο μάθημα που δημιουργήθηκε στο εκπαιδευτικό σύστημα Blackboard για την περαιτέρω υποστήριξη του site.

## 5.1 Το πρόβλημα των φυλλομετρητών

Έχει γίνει λόγος αρκετές φορές μέχρι αυτό το σημείο στο ζήτημα της διαφορετικής απεικόνισης μιας σελίδας μεταξύ των φυλλομετρητών. Εδώ θα γίνει μια πιο αναλυτική αναφορά σε αυτό το θέμα.

Ένας φυλλομετρητής είναι ως γνωστόν, ένα πρόγραμμα που χρησιμοποιείται για να προβάλλονται οι σελίδες του ιστού. Γνωστά τέτοια προγράμματα είναι ο Internet Explorer της Microsoft, ο Firefox του Mozilla Foundation, ο Opera της Opera Software, ο Safari της Apple, ο Chrome της Google, ο Konqueror (για KDE Linux) που αναπτύσσεται από εθελοντική βάση, και άλλα λιγότερο δημοφιλή.

Ο φυλλομετρητής ερμηνεύει τις εντολές HTML που συναντά σε κάποιο έγγραφο HTML ώστε να συλλέξει, οργανώσει και εμφανίσει τα τμήματα της σελίδας. Το «μυαλό» ενός τέτοιου προγράμματος είναι η λεγόμενη *layout engine* (θα μπορούσε να αποδοθεί ως «μηχανή διάταξης»). Αυτό το «υποσύστημα» του browser διαβάζει τη σελίδα, δηλαδή το HTML έγγραφο και εφαρμόζει σε αυτή τους προεπιλεγμένους κανόνες μορφοποίησης που ορίζει ο browser, τους κανόνες που έχει ρυθμίσει ο χρήστης για τα διάφορα στοιχεία (μέσα από τις ρυθμίσεις του browser) καθώς και τα διάφορα στυλ που ορίζουν οι κανόνες CSS.

Όπως ο φυλλομετρητής κάθε εταιρείας είναι διαφορετικός από τους υπόλοιπους, έτσι και οι μηχανές διάταξης διαφέρουν. Πιο σωστά, σχεδόν κάθε browser στηρίζεται

σε διαφορετική μηχανή. Στον παρακάτω πίνακα φαίνονται οι δημοφιλείς browsers και οι αντίστοιχες layout engines.

<b><u>Browser</u></b>	<b><u>Εταιρεία/Ανάπτυξη</u></b>	<b><u>Layout Engine</u></b>
Internet Explorer	Microsoft	Trident
Firefox	Mozilla Foundation	Gecko
Opera	Opera Software	Presto
Safari	Apple	WebKit
Chrome	Google	WebKit
Konqueror	Κοινότητα εθελοντών για συστήματα τύπου Unix	KHTML

Πίνακας 1: Οι σύγχρονοι browsers, οι υπεύθυνοι για την ανάπτυξή τους και οι αντίστοιχες layout engines

Η κάθε layout engine έχει τις δικές της ιδιαιτερότητες. Το γεγονός αυτό μπορεί να επιφέρει την απεικόνιση μιας σελίδας με διαφορετικό τρόπο απο κάθε browser. Περισσότερα γι' αυτό θα αναφερθούν στην ενότητα που γίνεται λόγος για CSS hacks και CSS filters.

Το πρόβλημα των φυλλομετρητών έγκεται ακριβώς σε αυτή την διαπίστωση: ότι μια σελίδα μπορεί να μην έχει την ίδια εμφάνιση σε όλους τους browsers. Όπως είναι εύκολα αντιληπτό, είναι εξαιρετικά δύσκολο να δημιουργηθούν σελίδες που θα αντιμετωπίζουν την ιδιομορφία κάθε browser, εάν οι εταιρείες δεν ακολουθούν κάποια κοινή γραμμή κατα τον προγραμματισμό τους.

Από την πλευρά του σχεδιαστή της ιστοσελίδας, η κατάσταση μπορεί να αντιμετωπιστεί εάν ακολουθηθούν τα λεγόμενα *web standards*. Αυτά είναι προτάσεις που διατυπώνονται και προωθούνται απο τον οργανισμό W3C ώστε να εξασφαλίζεται η κατα το δυνατόν ομαλή και προσδοκώμενη εμφάνιση των σελίδων. Το πρόβλημα είναι πως οι εταιρείες ανάπτυξης των browsers δεν τηρούν πάντα τις προτάσεις αυτές.

Τα τελευταία χρόνια έχει παρατηρηθεί τόσο βελτίωση των προγραμμάτων φυλλομετρητών (χαρακτηριστικό παράδειγμα ο Internet Explorer 8) όσο και πιο πιστή τήρηση των προτάσεων. Εξακολουθούν να υπάρχουν προβλήματα λόγω του ανταγωνισμού αλλά είναι σίγουρο πως με τις νέες προτάσεις και την έλευση της CSS3



θα υπάρξει βελτίωση σε αυτόν τον τομέα. Τα web standards άλλωστε υπάρχουν για να καθοδηγούν τόσο σχεδιαστές όσο και εταιρείες, για να επιτευχθεί η μέγιστη δυνατή συμβατότητα.

### 5.1.1 Ένα χαρακτηριστικό πρόβλημα

Οι ιδιομορφίες που διαθέτει κάθε φυλλομετρητής, καθώς και τα προβλήματα που μπορεί να παρουσιάζει είναι γνωστά με τους όρους *browser quirks* και *browser bugs* αντίστοιχα.

Τα browser bugs, δηλαδή τα προβλήματα που αντιμετωπίζει κάποιος browser, ανήκουν σε δύο κατηγορίες: αυτά που προκύπτουν από τον τρόπο που διαβάζεται ένας κανόνας CSS από την layout engine και αυτά που είναι αποτέλεσμα του τρόπου τοποθέτησης περιεχομένου στη σελίδα.

#### Internet Explorer Box Model

Μια χαρακτηριστική ιδιομορφία του Internet Explorer μέχρι και τις εκδόσεις 5.x αφορά στην αντίληψη περί του box model. Κατα κανόνα, οι τιμές των ιδιοτήτων padding και border δεν συνυπολογίζονται στο ύψος και πλάτος ενός στοιχείου. Αυτό σημαίνει πως οι τιμές αυτών των ιδιοτήτων **προστίθενται** στην τιμή της ιδιότητας width όταν υπολογίζεται το ολικό πλάτος ή ύψος ενός στοιχείου.

Στον Internet Explorer των εκδόσεων 5.x το παραπάνω δεν ισχύει. Σε αυτές τις εκδόσεις οι τιμές των ιδιοτήτων του box model που προαναφέρθηκαν συνυπολογίζονται στο πλάτος ή ύψος των στοιχείων. Αν δηλαδή οριστεί κάποιο πλάτος για το στοιχείο, αυτό συμπεριλαμβάνει και τις τιμές των padding και border. Αποτέλεσμα αυτού είναι τα στοιχεία να εμφανίζονται πιο στενά ή πιο κοντά στον Internet Explorer 5.x απ'ότι στους άλλους browsers.

Αυτό βέβαια εάν δεν ληφθεί κάποιο μέτρο για τη σωστή αναπαράσταση του στοιχείου. Περισσότερα για το ίδιο το πρόβλημα και τρόπους αντιμετώπισής του μπορούν να βρεθούν στην αντίστοιχη σελίδα του CSS Tutor που αφορά τα προβλήματα.

### 5.1.2 Παράκαμψη και αντιμετώπιση των ιδιομορφιών

Υπάρχουν δύο τρόποι να αντιμετωπιστούν τα προβλήματα ιδιομορφιών των φυλλομετρητών. Ο πρώτος είναι τα CSS hacks ή CSS filters και ο δεύτερος τα conditional comments. Στις επόμενες παραγράφους γίνεται αναφορά σε αυτές τις έννοιες.

## CSS hacks

Τα CSS hacks είναι μια τεχνική αντιμετώπισης των browser bugs. Κάθε hack στοχεύει στην αντιμετώπιση ενός προβλήματος που αφορά έναν συγκεκριμένο browser. Αυτό σημαίνει πως ένα hack που αντιμετωπίζει κάποια ιδιομορφία στον Explorer δε θα καλύπτει παραδείγματος χάριν και τον Opera.

Επιπλέον τα hacks είναι σχεδόν πάντοτε έγκυροι κανόνες CSS. Με αυτό εννοείται πως δεν υιοθετείται μια λογική καταπάτησης της εγκυρότητας των CSS για να καταπολεμηθεί ένα πρόβλημα. Απλώς συντάσσονται κάποιοι εξειδικευμένοι κανόνες, ίσως και επιπλέον των απολύτως απαραίτητων, για να λυθεί ένα ζήτημα.

Τα CSS hacks εκμεταλλεύονται ουσιαστικά κάποια ανεπάρκεια των browsers που έχει να κάνει με τον τρόπο που αυτοί χειρίζονται τους έγκυρους κανόνες CSS. Εκεί δηλαδή που ένας έγκυρος κανόνας δεν δίνει το επιθυμητό αποτέλεσμα εφαρμόζεται ένα CSS hack.

Μια υποκατηγορία των CSS hacks είναι τα CSS *filters*. Τα φίλτρα αυτά είναι και πάλι κανόνες οι οποίοι όμως περιέχουν κάποια δήλωση που δεν αναγνωρίζεται από κάποιον browser. Έτσι ο συγκεκριμένος browser δεν ασχολείται με τον συγκεκριμένο κανόνα. Αυτό πρακτικά σημαίνει πως γράφεται ένας κανόνας για όλους τους browsers που αναγνωρίζουν την δήλωση και ένας δεύτερος που δεν την περιέχει, για τον browser που δεν την αναγνωρίζει.

Σε κάθε περίπτωση επιζητάται το αποτέλεσμα στην απεικόνιση να είναι το ίδιο. Οπότε και οι δύο κανόνες θα δίνουν το ίδιο αποτέλεσμα. Απλώς για να επιτευχθεί αυτό χρειάζονται διαφορετικές δηλώσεις μέσα στον κανόνα.

Κλασικό παράδειγμα CSS filter είναι η χρήση σε κανόνες του child selector που συμβολίζεται με το >. Αυτός ο τύπος selector, που υποδηλώνει πως ένα στοιχείο είναι «παιδί» ενός άλλου, δεν αναγνωρίζεται από τον Internet Explorer τουλάχιστον μέχρι και την έκδοση 6. Γι' αυτό χρησιμοποιείται αρκετά συχνά ώστε να ορίζονται κανόνες για όλους τους browsers εκτός του Explorer. Αφού έχουν οριστεί αυτοί οι κανόνες στη συνέχεια ορίζεται και ένας ακόμα που θα αφορά τον Explorer, τον οποίο θα αγνοούν οι υπόλοιποι.

Με τον παραπάνω τρόπο επιτυγχάνεται σε κάποιο βαθμό και το λεγόμενο *browser detection* δηλαδή η ανίχνευση φυλλομετρητή. Με χρήση του φίλτρου δηλαδή εντοπίζεται κατα κάποιο τρόπο, ποιος browser χρησιμοποιείται. Πιο κατάλληλη μέθοδος για αυτό όμως είναι τα *conditional comments* που περιγράφονται παρακάτω.

## Conditional Comments

Τα conditional comments είναι εντολές HTML που χρησιμοποιούνται για να ανιχνευθεί ποιος browser χρησιμοποιείται και ανάλογα με το αποτέλεσμα να φορτωθεί διαφορετικό αρχείο CSS.

Λέγονται έτσι διότι αφενός αποτελούν *συνθήκες* αφετέρου συντάσσονται με χρήση σχολίων της HTML. Δηλαδή στην αρχή του αρχείου, μέσα σε tags σχολίων συντάσσονται με συγκεκριμένο τρόπο συνθήκες IF...THEN.

Ακολουθεί ένα απόσπασμα τέτοιου κώδικα:

```
<head>
<title>Test</title>
<link href="all_browsers.css" rel="stylesheet" type="text/css">

<!--[if IE]> <link href="ie_only.css" rel="stylesheet"
type="text/css"> <![endif]-->

<!--[if !IE]>--> <link href="not_ie.css" rel="stylesheet"
type="text/css"> <!--<![endif]-->
</head>
```

Υπάρχουν δύο conditional comments στο παραπάνω απόσπασμα. Η μία συνθήκη ελέγχει εάν ο browser είναι ο Internet Explorer με τη δήλωση *[if IE]*, ώστε να φορτωθεί αρχείο με κανόνες προσαρμοσμένους στις ιδιαιτερότητες αυτού του browser.

Η δεύτερη συνθήκη ελέγχει εάν χρησιμοποιείται οποιοσδήποτε άλλος browser με τη δήλωση *[if !IE]*. Αν ισχύει αυτό τότε χρησιμοποιείται το αρχείο *not\_ie.css* για τη μορφοποίηση.

Σε κάθε άλλη περίπτωση φορτώνεται το αρχείο *all\_browsers.css* μέσω της δήλωσης στην πρώτη γραμμή.

## 5.2 CSS Tutor και φυλλομετρητές

Κατα την διαδικασία σχεδιασμού και προγραμματισμού του ιστότοπου έγιναν δοκιμές στους δύο πιο δημοφιλείς browsers: τον Mozilla Firefox και τον Internet Explorer. Σε αυτή την ενότητα παρουσιάζεται η αντιμετώπιση του CSS Tutor από αυτούς τους φυλλομετρητές.

### 5.2.1 Mozilla Firefox

Κατα τη διάρκεια ανάπτυξης του ιστότοπου οι δοκιμές λάβαιναν χώρα κατα κύριο λόγο στον browser του Mozilla Foundation.

Ο Firefox μεταξύ των δύο πρώτων σε δημοφιλία φυλλομετρητής μαζί με τον Explorer, με τη διαφορά τους να είναι πολύ μικρή. Για την ακρίβεια ο Firefox φαίνεται να είναι πλέον ο πιο δημοφιλής, με βάση στατιστικές μετρήσεις. Τα χαρακτηριστικά που τον έκαναν γνωστό αρχικά είναι η ταχύτητα και η ασφάλεια.

Ο Firefox είναι ένα πρόγραμμα ανοιχτού κώδικα (open source) και διατίθεται για διάφορα λειτουργικά συστήματα, συμπεριλαμβανομένων Windows, MAC OS X, Linux και διάφορα άλλα συστήματα τύπου Unix.

Ο browser βασίζεται στην Gecko μηχανή διάταξης. Ο συγκεκριμένος μηχανισμός έχει διαπιστωθεί ότι προσφέρει αξιόπιστη και στιβαρή υποστήριξη της γλώσσας CSS.

Τη στιγμή που γράφεται αυτό το κείμενο ο Firefox βρίσκεται στην έκδοση 3.6. Όλες οι δοκιμές λοιπόν έγιναν σε εκδόσεις 3.x του φυλλομετρητή αυτού.

Από πλευράς σχεδιασμού τα αποτελέσματα στην απεικόνιση των στοιχείων της σελίδας είναι τα αναμενόμενα σε αυτόν τον browser. Λόγω της σωστής υποστήριξης CSS δεν χρειάστηκε να γίνει κάποια αλλαγή ή να εφαρμοστεί κάποιο CSS hack.

Ο browser εμφανίζει τα χρώματα όπως αυτά φαίνονται στην παλέτα που χρησιμοποιήθηκε (ένας πίνακας με τις τιμές των χρωμάτων σε δεκαεξαδική μορφή).

Παρόμοια είναι και η εντύπωση που σχηματίζεται από προγραμματιστικής άποψης. Το script δουλεύει όπως σχεδιάστηκε, χωρίς να υπάρχει ανάγκη για κάποιου τύπου browser detection και εκτέλεση άλλου script. Χαρακτηριστικό παράδειγμα είναι η ομαλή εκτέλεση του fade-in/fade-out εφέ στην κατηγορία «Συχνές Ερωτήσεις».

Από πλευράς ταχύτητας εκτέλεσης του script, αυτή είναι ικανοποιητική. Σε ορισμένες περιπτώσεις ίσως υπάρχει ελάχιστη καθυστέρηση στην απόκριση των controls του GUI, χωρίς αυτό να το καθιστά δύσκολο στη χρήση ή αργό.

### 5.2.2 Internet Explorer

Ο Internet Explorer ήταν για πολλά χρόνια ο πιο δημοφιλής φυλλομετρητής που κυκλοφορεί, κυρίως λόγω της κυριαρχίας της Microsoft στον χώρο. Αυτή την περίοδο φαίνεται να έπεται του Firefox. Κάτι που πρέπει να παρατηρηθεί και σημειωθεί σχετικά με τον συγκεκριμένο browser είναι πως ακόμα και μετά την έλευση των νέων εκδόσεων, εξακολουθούν να χρησιμοποιούνται σε σημαντικό ποσοστό παλαιότερες

κυκλοφορίες. Έτσι για να υπολογιστεί το συνολικό ποσοστό χρήσης του πρέπει να αθροιστούν τα ποσοστά που αντιστοιχούν σε κάθε έκδοση.

Το πιο σημαντικό βέβαια δεν είναι ο υπολογισμός του μεριδίου της αγοράς, αλλά το ότι οι παλιές εκδόσεις του έχουν σημαντικά προβλήματα. Ζητήματα που απασχολούν εν πολλοίς τους επαγγελματίες του διαδικτύου και συγκεκριμένα του παγκοσμίου ιστού.

Μολονότι είναι πολύ διαδεδομένη ή χρήση του λοιπόν, είναι πιθανότα ο λιγότερο δημοφιλής ανάμεσα στους προγραμματιστές και σχεδιαστές ιστοσελίδων. Αυτό συμβαίνει διότι παρουσιάζει τις περισσότερες ιδιοτροπίες στην υλοποίηση και εφαρμογή των CSS. Η layout engine που χρησιμοποιεί είναι η Trident, οπότε σε αυτήν οφείλονται τα όποια ζητήματα ασυμφωνίας στην απεικόνιση σε σχέση με τους άλλους browsers. Ιδίως μέχρι και την έκδοση 6 ο Explorer προκαλούσε αρκετά προβλήματα στους προγραμματιστές (όπως το box model bug αλλά και άλλα).

Στις καινούριες του εκδόσεις, και δη στον Internet Explorer 8, αρκετά από τα προβλήματα αυτά έχουν εξαιρεθεί.

Σχετικά με το CSS Tutor, ο Explorer παρουσιάζει κάποιες μικρές διαφορές στην εμφάνιση σε σχέση με το αποτέλεσμα που προβάλλει ο Firefox.

Πιο συγκεκριμένα οι διακριτές διαφορές είναι δύο: το border που υπάρχει στο sidebar κάτω από κάθε link είναι πιο σκούρο και δεν διακρίνεται τόσο εύκολα, ενώ η γραμματοσειρά Arial εμφανίζεται ελαφρώς διαφορετικά.

Μια άλλη μικρή διαφορά αφορά τα κουμπιά στην σελίδα. Αυτά εμφανίζονται να έχουν περισσότερο ύψος απ' ό,τι στον Firefox.

Αν και θα υπήρχε τρόπος να συνταχθούν κανόνες τύπου CSS filter για να αντιμετωπιστεί η κατάσταση τουλάχιστον για τα δύο από τα τρία φαινόμενα, επιλέχθηκε να μη γίνει κάτι τέτοιο. Αυτό διότι δεν αποτελούν σοβαρά ζητήματα απεικόνισης.

Από πλευράς εκτέλεσης του κώδικα που υπάρχει στο site ο Explorer παρουσιάζει μικρές διαφορές σε σχέση με τον Firefox.

Πιο συγκεκριμένα, υπάρχει μια δυσκολία την αλλαγή της μορφοποίησης σε ένα συγκεκριμένο στοιχείο. Πρόκειται για την αλλαγή τιμής στην ιδιότητα *text-transform* που βρίσκεται στην ενότητα «Ιδιότητες και παραδείγματα» στην κατηγορία «Μορφοποίηση κειμένου». Αυτό που συμβαίνει είναι πως αν αλλάξει μια φορά τιμή η

συγκεκριμένη ιδιότητα, για να αλλάξει δεύτερη φορά πρέπει να γίνει reset του κανόνα μέσω του κουμπιού Reset ή να γίνει ανανέωση της σελίδας.

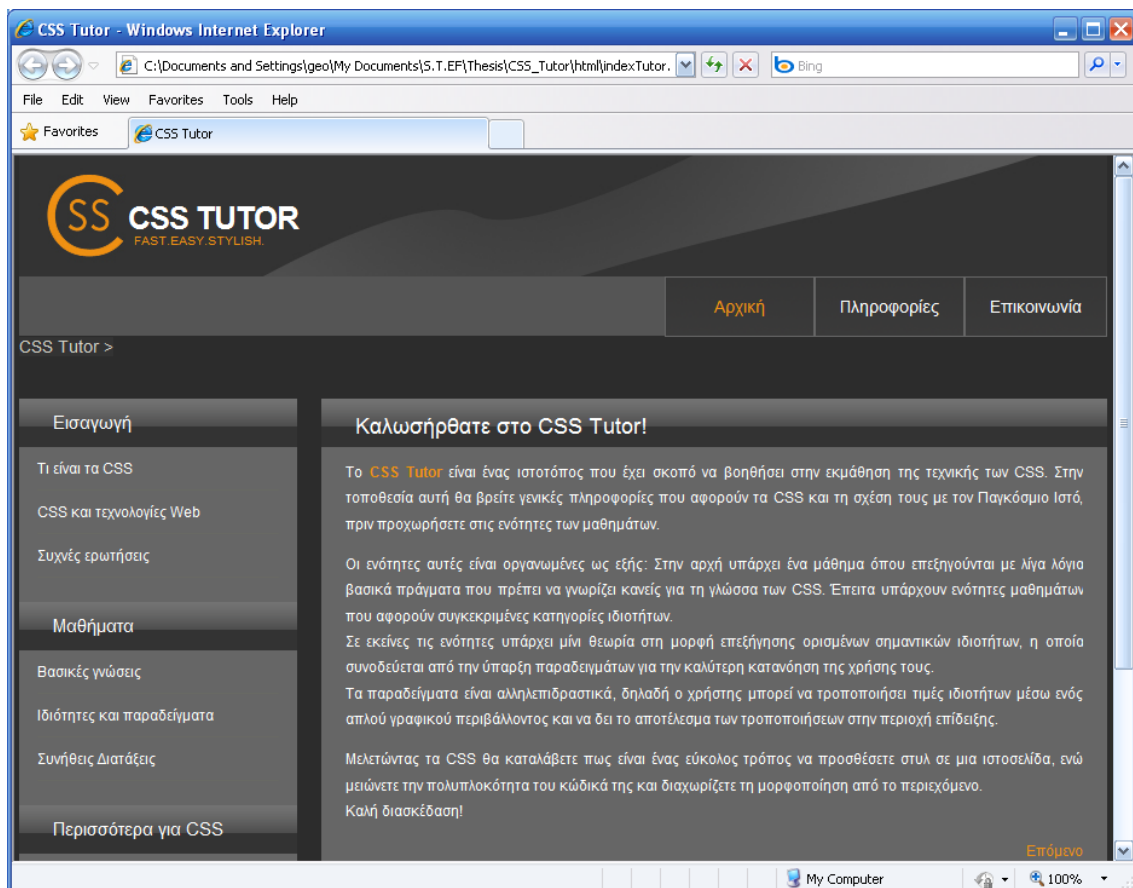
Το παραπάνω παρατηρήθηκε μόνο στον Explorer. Στον Firefox η ιδιότητα αλλάζει κανονικά.

Κάτι ακόμα που παρατηρήθηκε είναι πως δεν λειτουργεί το fade-in/fade-out της ενότητας «Συχνές Ερωτήσεις». Οι απαντήσεις εμφανίζονται και εξαφανίζονται αλλά όχι με το εφέ του «ξεθωριάσματος».

Τα παραπάνω είναι δύο περιπτώσεις που έγιναν αντιληπτές. Σε γενικές γραμμές δεν υπάρχουν μεγάλες διαφορές και είναι πιθανό να μην υπάρχουν άλλες.

Από πλευράς ταχύτητας εκτέλεσης, δεν παρατηρήθηκε κάποια καθυστέρηση. Η απόκριση είναι το ίδιο καλή με αυτή του Firefox.

Στην παρακάτω εικόνα παρατίθεται στιγμιότυπο του ιστοτόπου από τον Explorer. Όπως φαίνεται τα borders στο κάτω μέρος των συνδέσμων που βρίσκονται στην μπάρα πλοήγησης στα αριστερά δεν είναι ευδιάκριτα.



Εικόνα 11: Στιγμιότυπο σελίδας του CSS Tutor όπως την προβάλλει ο Internet Explorer

## 5.3 Πειραματικό μάθημα στο σύστημα Blackboard

Το σύστημα blackboard προσφέρει τη δυνατότητα να οργανωθεί διαδικτυακά μια εκπαιδευτική διαδικασία. Είναι ιδανικό για να φιλοξενεί υλικό, να παρέχει πηγές και συνδέσμους σχετικά με το αντικείμενο αλλά και να χρησιμοποιείται για την αξιολόγηση του χρήστη/φοιτητή. Γενικότερα πρόκειται για ένα χρήσιμο εργαλείο που διευκολύνει την ενασχόληση και αλληλεπίδραση με το εκάστοτε αντικείμενο.

### 5.3.1 Δημιουργία ειδικού μαθήματος

Στα πλαίσια της πτυχιακής εργασίας με θέμα την ανάπτυξη διαδικτυακής εφαρμογής για την εκμάθηση CSS δημιουργήθηκε ένα ειδικό μάθημα στο blackboard.

Στα έγγραφα αυτού του μαθήματος μπορεί να βρεθεί ένα link που οδηγεί στις σελίδες του CSS Tutor.

Το blackboard όπως αναφέρθηκε ευνοεί την οργάνωση ενός μαθήματος διαδικτυακά. Έτσι το ειδικό μάθημα που δημιουργήθηκε και αφορά στα CSS μπορεί να εμπλουτίζεται συνεχώς με νέο υλικό και παραπομπές. Όλα αυτά μπορούν να βρίσκονται συγκεντρωμένα σε ένα σημείο.

Το μάθημα μπορεί να καταστεί διαθέσιμο μόνιμα, δηλαδή δεν δεσμεύεται από περιορισμούς σχετικά με τον χώρο που διαθέτει ο κάθε φοιτητής στους servers της σχολής.

### 5.3.2 Δημιουργία αρχείων ερωτήσεων

Μέσω της λειτουργίας «Διαμόρφωση Αρχείου Ερωτήσεων» που παρέχει ο πίνακας ελέγχου του μαθήματος δημιουργήθηκαν κάποια question pools. Συλλογές ερωτήσεων δηλαδή, για μελλοντική χρήση σε τεστ αξιολόγησης.

Οι συλλογές αυτές περιέχουν ερωτήσεις από όλο το φάσμα της ύλης που παρουσιάζεται και υπάρχει στον ιστότοπο του CSS Tutor.

Πιο αναλυτικά, υπάρχουν ερωτήσεις που αφορούν σε βασικές γνώσεις περι της γλώσσας, τεχνικές γνώσεις γύρω από την τοποθέτηση στοιχείων, το box model και τη μορφοποίηση. Υπάρχουν επίσης ερωτήσεις γύρω από τα CSS και τις υπόλοιπες τεχνολογίες του ιστού.

Οι ερωτήσεις όπως αναφέρθηκε είναι ανοιχτές προς χρήση σε μελλοντικά τεστ αξιολόγησης. Δίνεται η δυνατότητα μάλιστα να οργανωθούν ανάλογα με κάποιο επίπεδο δυσκολίας, το οποίο ορίζει για κάθε ερώτηση ο δημιουργός του test.

Τα test που υπάρχουν είναι συνοδευτικά του site οπότε το επίπεδο των ερωτήσεων συμβαδίζει με αυτό των τεχνικών γνώσεων που παρέχονται στο site.

Οι ερωτήσεις είναι διαφόρων ειδών, από σύντομης απάντησης μέχρι αντιστοίχισης και πολλαπλής επιλογής. Κυρίως επιλέχθηκαν οι τελευταίοι τύποι ερωτήσεων.

Αρκετές ερωτήσεις είναι θεωρητικής φύσεως. Εξετάζονται δηλαδή οι γνώσεις που αποκόμισε ο ενδιαφερόμενος σχετικά με την λειτουργία των CSS, τους τρόπους χρήσης των, την χρησιμότητά τους και τη σχέση τους με τον Παγκόσμιο Ιστό.

Από την άλλη υπάρχουν ερωτήσεις σε πρακτικό επίπεδο. Αυτό σημαίνει πως παρατίθεται κάποιο τμήμα κώδικα και γίνονται ερωτήσεις επ'αυτού. Οι ερωτήσεις μπορεί να αφορούν στο αποτέλεσμα που θα επιφέρει η εκτέλεση του HTML/CSS κώδικα, στην εγκυρότητα του κώδικα ή στην επιλογή της πιο κατάλληλης σύνταξης κάποιου κανόνα.

Γενικότερα έγινε προσπάθεια να καλυφθεί όσο το δυνατόν πληρέστερα όλο το θέμα, στο βαθμό που παρουσιάζεται και στον ιστότοπο. Ταυτόχρονα επιδιώχθηκε να δοθεί βαρύτητα στα πιο ουσιαστικά σημεία καθώς και στις σημαντικές λεπτομέρειες.

Για κάθε απάντηση του χρήστη, είτε σωστή είτε λάθος, υπάρχει μια ανάδραση (feedback), δηλαδή ένας σχολιασμός πάνω στην ερώτηση. Αυτές οι παρατηρήσεις υπάρχουν για να επεξηγούν την σωστή απάντηση.

### **5.3.3 Χρησιμότητα μαθήματος**

Αυτό που επιδιώκεται με την δημιουργία του ειδικού μαθήματος είναι η παροχή ενός πιο ολοκληρωμένου τρόπου εκμάθησης. Ο ενδιαφερόμενος διαθέτει εύκολη πρόσβαση στον ιστότοπο όπου υπάρχει όλο το απαραίτητο υλικό σε θεωρία καθώς και η εφαρμογή των παραδειγμάτων.

Πέραν τούτου, ο χρήστης διευκολύνεται στο να αξιολογήσει τις γνώσεις του, ακόμα και λίγο αφότου ολοκληρώσει κάποια ενότητα μαθημάτων CSS.

Επιπροσθέτως η παροχή υλικού μπορεί να είναι αδιάκοπη και συνεχής. Αυτό σημαίνει πως ο διαχειριστής του μαθήματος μπορεί να προσθέτει και να τροποποιεί υλικό χωρίς περιορισμούς.

Το πιο σημαντικό απ'όλα είναι πως το μάθημα μπορεί να καταστεί προσβάσιμο σε όλους τους φοιτητές και καθηγητές του Τμήματος ή και γενικότερα της σχολής. Έτσι εάν το αποτέλεσμα κρίνεται ικανοποιητικό, από αυτό μπορούν να επωφεληθούν όσοι ενδιαφέρονται να ασχοληθούν με την συγκεκριμένη γλώσσα του διαδικτύου. Επιπλέον



μπορεί να χρησιμοποιηθεί στα πλαίσια κάποιου μαθήματος του Τμήματος, ως συμπληρωματικό μάθημα. Υπάρχει δηλαδή μια ελευθερία στον τρόπο αξιοποίησης του αποτελέσματος.

## 5.4 Σύνοψη

Στην ενότητα που προηγήθηκε έγινε λόγος για τα προγράμματα φυλλομετρητών. Συγκεκριμένα παρουσιάστηκαν οι έννοιες *φυλλομετρητής* και *layout engine*. Αναφέρθηκαν οι πιο δημοφιλείς browsers και οι αντίστοιχοι μηχανισμοί απεικόνισης των στοιχείων των σελίδων.

Ιδιαίτερη σημασία έχει η ασυμφωνία στην υποστήριξη των CSS από τους διάφορους browsers, ζήτημα που απασχόλησε αυτή την ενότητα. Επιπλέον αναφέρθηκαν έννοιες που αφορούν στην διαφορετική λειτουργία και στα προβλήματα που εμφανίζουν τα παραπάνω προγράμματα.

Στη συνέχεια έγινε γενική αναφορά στις τεχνικές αντιμετώπισης των ιδιομορφιών που εμφανίζουν οι browsers.

Ένα τμήμα της ενότητας αφιερώθηκε στους φυλλομετρητές οι οποίοι χρησιμοποιήθηκαν κατά κύριο λόγο για την δοκιμή του CSS Tutor, που είναι ο Firefox και ο Internet Explorer.

Η τελευταία ενότητα του κεφαλαίου αφορά στο μάθημα που δημιουργήθηκε στο σύστημα Blackboard, και που είναι συνοδευτικό του ιστότοπου. Στο μάθημα αυτό βρίσκονται τόσο το site όσο και διαμορφωμένα quiz για την αξιολόγηση των γνώσεων του χρήστη.



## 6 ΕΞΕΛΙΞΕΙΣ ΣΤΟ ΠΕΔΙΟ

Ο χώρος των υπολογιστών και της επιστήμης της Πληροφορικής σημειώνει, ως γνωστόν, τους πιο αλματώδεις ρυθμούς ανάπτυξης. Οι εξελίξεις διαδέχονται η μία την άλλη είτε αφορούν τεχνολογική πρόοδο, είτε κάποια νέα θεωρητική έννοια.

Ο χώρος του Internet και όσων ασχολούνται με αυτό δε μπορεί να αποτελέσει εξαίρεση. Οι αλλαγές που έχουν παρατηρηθεί, ιδίως στην πρώτη δεκαετία του 21<sup>ου</sup> αιώνα, σε σχέση με αυτό που υπήρχε στα τέλη της δεκαετίας του 1980 και στις αρχές του '90 είναι τεράστιες.

Απο τη στιγμή που το ευρύ κοινό αντιλήφθηκε την χρησιμότητα και την δυναμική που προσφέρει ένα τέτοιο μέσο, προέκυψε η ανάγκη για καλύτερη υποστήριξη του (τεχνολογίες Διαδικτύου, ADSL κλπ.). Αυτή η υποστήριξη με τη σειρά της αναβάθμισε αυτό το προϊόν και όλες τις υπηρεσίες του. Έτσι δημιουργήθηκε ένας χώρος που περιέχει πληροφορίες από κάθε λογής πεδίο, σε όγκο και έκταση που δεν μπορεί εύκολα να υπολογιστεί.

Οι έννοιες που αναφέρθηκαν στα προηγούμενα κεφάλαια είναι τμήματα του παραπάνω συστήματος και αφορούν κυρίως το τεχνικό του μέρος. Για να αναβαθμιστεί και να εξελιχθεί λοιπόν ο μηχανισμός αυτός, πρέπει να εξελίσσονται κατάλληλα και τα διάφορα μέρη του ώστε να τον υποστηρίξουν.

Στις επόμενες ενότητες θα λάβει χώρα μια αναφορά σε εξελίξεις που αφορούν στον Παγκόσμιο Ιστό (World Wide Web) και στις γλώσσες των HTML και CSS.

### 6.1 Παγκόσμιος Ιστός

Όταν πρωτοεμφανίστηκε το σύστημα του Παγκοσμίου Ιστού αποτελούνταν από έναν αριθμό σελίδων που ήταν συνδεδεμένες μεταξύ τους. Το περιεχόμενο των σελίδων ήταν στατικό. Το μόνο που μπορούσε να κάνει ο χρήστης ήταν να ανακτά τις σελίδες και να διαβάζει το περιεχόμενό τους, χωρίς να αλληλεπιδρά μαζί τους.

Επιπλέον ο ρόλος του Ιστού τότε ήταν διαφορετικός. Ως ένα νέο σύστημα, μια νέα τεχνολογία, δεν διέθετε τα μέσα ούτε και την απήχηση για να προσφέρει τον αριθμό υπηρεσιών και τις δυνατότητες που προσφέρει σήμερα.

Παρόλα αυτά η χρησιμότητα ενός τέτοιου συστήματος φάνηκε στην πορεία και σε συνδυασμό με την ανάπτυξη των προγραμμάτων φυλλομετρητών σημειώθηκε έκρηξη στη χρήση του.

Σήμερα ο Παγκόσμιος Ιστός αποτελεί ένα μέσο που φιλοξενεί έναν πολύ μεγάλο αριθμό απο υπηρεσίες και εφαρμογές παντός τύπου. Δεν είναι σκόπιμο να γίνει αναφορά σε όλες αυτές, καθώς κάτι τέτοιο δεν εξυπηρετεί τους σκοπούς αυτής της ενότητας. Πρέπει όμως να υπογραμμιστεί το βάθος στο οποίο έχει εισχωρήσει η τεχνολογία αυτή ακόμα και στις συναλλαγές της καθημερινότητας.

Τα τελευταία χρόνια γίνεται πολύ συχνά λόγος για το *Web 2.0*. Δεν πρόκειται βέβαια για κάποια έκδοση του Ιστού με τον τρόπο που ορίζεται για παράδειγμα μια γλώσσα προγραμματισμού. Πρόκειται κατα κάποιο τρόπο για ένα ορόσημο, ή ένα τρόπο να γίνει αναφορά στον τρόπο που χρησιμοποιείται αυτό τον καιρό ο Παγκόσμιος Ιστός.

Συγκεκριμένα το *Web 2.0* χαρακτηρίζεται από την χρήση εφαρμογών και υπηρεσιών που ασχολούνται κυρίως με την προώθηση και διαμοίραση πληροφοριών κάθε είδους μεταξύ των χρηστών. Πιο συγκεκριμένα σχηματίζονται *κοινωνικά δίκτυα* όπου ο κάθε χρήστης διατηρεί ένα λογαριασμό ή ένα προφίλ το οποίο του παρέχει πρόσβαση αλλά και δυνατότητα επικοινωνίας με τους άλλους χρήστες. Ο όρος που αντιστοιχεί στα παραπάνω είναι το *social networking*.

Παραδείγματα τέτοιου τύπου είναι τα blogs, τα διάφορα wiki (σελίδες παροχής πληροφοριών/γνώσεων πάνω σε συγκεκριμένα αντικείμενα) καθώς και σελίδες κοινωνικής δικτύωσης και επικοινωνίας, όπως MySpace, Facebook, Twitter και άλλα.

Παρότι το *Web 2.0* στο έτος που βρισκόμαστε δεν βρίσκεται πολλά χρόνια στο προσκήνιο, γίνεται ήδη λόγος για το *Web 3.0*. Άλλωστε όπως ειπώθηκε οι αλλαγές και οι εξελίξεις στον τομέα του διαδικτύου λαμβάνουν χώρα με τεράστια ταχύτητα.

Το *Web 3.0* ως έννοια περιέχει μια πολύ σημαντική καινοτομία: την δυνατότητα προγράμματα να χρησιμοποιούν τον Ιστό όπως οι άνθρωποι. Πρόκειται για ένα συνδυασμό του *Web 2.0* όπως είναι γνωστό σήμερα, με την έννοια και τις τεχνικές των *Semantics*.

*Semantics* σημαίνει *σημασιολογία*. Αφορά το νόημα που περιέχει η πληροφορία και όχι τόσο τη δομή της ή τον τύπο της. Αυτό που επιδιώκεται με χρήση των *semantics* στο *Web* είναι να μπορεί λογισμικό (αποκαλείται συχνά *agents* δηλ. πράκτορες) να

αντιλαμβάνεται την πληροφορία που υπάρχει αποθηκευμένη όπως την αντιλαμβάνεται ο άνθρωπος. Θυμίζει αρκετά την έννοια της Τεχνητής Νοημοσύνης.

Πιο συγκεκριμένα, αυτά τα προγράμματα θα έχουν ρόλο να αναζητούν, να οργανώνουν και να παρουσιάζουν την πληροφορία που υπάρχει στον Ιστό. Για να το πετύχουν αυτό, χρησιμοποιούνται κάποιες τεχνολογίες. Τρεις από τις πιο σημαντικές είναι οι παρακάτω:

- *Resource Description Definition* ή απλά RDF ονομάζεται ένα framework που χρησιμοποιείται για την περιγραφή των πηγών και πόρων, δηλαδή της πληροφορίας
- *Extensible Markup Language* ή XML, πρόκειται για μια γλώσσα με βασικό της χαρακτηριστικό την δυνατότητα ορισμού tags από τον χρήστη. Έτσι δίνεται έμφαση στην απόδοση σημασίας στην δομή του εγγράφου αλλά και στο περιεχόμενό του και όχι τόσο στην απλή αναπαράσταση κειμένου, όπως με την HTML.
- *Web Ontology Language* (OWL), που αποτελεί μηχανισμό για την αναπαράσταση γνώσης.

Όπως γίνεται εύκολα αντιληπτό, όλες οι παραπάνω τεχνολογίες έχουν σκοπό την περιγραφή της πληροφορίας.

Μια χαρακτηριστική εφαρμογή του Web 3.0 είναι η πιο «έξυπνη» αναζήτηση. Σήμερα οι αναζητήσεις λαμβάνουν χώρα με τη χρήση λέξεων κλειδιών (keywords). Έτσι το αποτέλεσμα της αναζήτησης εξαρτάται αποκλειστικά από το κείμενο που υπάρχει στις σελίδες και όχι από την σημασία του κειμένου και το θέμα του. Για παράδειγμα, σε περιπτώσεις όπου κάποια λέξη-κλειδί έχει διπλό νόημα ή χρησιμοποιείται μεταφορικά το αποτέλεσμα της αναζήτησης πρέπει να ελεγχθεί. Με τη σημασιολογία πίσω από την πληροφορία κάτι τέτοιο δε θα είναι απαραίτητο.

Επιπλέον των παραπάνω, προωθείται και η έννοια της *αναζήτησης μέσωων* (media search). Αυτό σημαίνει πως η αναζήτηση για κάποια μέσα, όπως ήχο και εικόνα, δε θα γίνεται με χρήση keywords αλλά με άλλα μέσα. Με λίγα λόγια, για να αναζητηθεί μια εικόνα ή ένα τραγούδι δε θα δίνεται ως όρος αναζήτησης κάποια λέξη, αλλά μια παρόμοια εικόνα ή ένα ηχητικό απόσπασμα αντίστοιχα.

Τα παραπάνω είναι ένα μέρος των όσων προωθούνται με την έλευση του Web 3.0. Η βασική ιδέα είναι πάντως να δημιουργηθεί ένα πιο «έξυπνο» δίκτυο.

## 6.2 HTML5

Η τρέχουσα έκδοση της HTML είναι η 4 και υπάρχει από το 1998. Αυτή η έκδοση, αν και εξυπηρετεί ακόμα το Web, παρουσιάζει κάποιες «ανεπάρκειες» απέναντι στις νέες σχεδιαστικές τάσεις και προκλήσεις.

Για να καλυφθούν αυτές οι ανεπάρκειες αναπτύσσονται άλλες τεχνολογίες που λειτουργούν συμπληρωματικά με την HTML. Χαρακτηριστικό παράδειγμα είναι η υποστήριξη βίντεο, όπου υπάρχουν διάφορα πρόσθετα προγράμματα (plugins) για τους browsers (όπως ο Flash Player της Adobe).

Η επερχόμενη έκδοση της HTML είναι η 5. Οι καινοτομίες σε αυτή την έκδοση είναι πολλές και ποικίλες. Το βασικό που πρέπει να γνωρίζει κάποιος για αυτή την έκδοση είναι η ενσωμάτωση στοιχείων στην ίδια τη γλώσσα καθώς και η μετακίνησή της προς την κατεύθυνση που δίνει η XML με ταυτόχρονη απομάκρυνση από την SGML. Τα έγγραφα δηλαδή δεν αποτελούν τόσο αναπαράσταση κειμένου όσο λογικές δομές με πληροφορία.

Εισάγονται λοιπόν νέα στοιχεία και tags που εξυπηρετούν τη δομή (structure) του εγγράφου. Μέχρι σήμερα η οργάνωση ενός εγγράφου γίνεται με χρήση στοιχείων όπως πίνακες και div στα οποία αποδίδεται ένας ρόλος. Για παράδειγμα στις σελίδες του CSS Tutor υπάρχουν τα div στοιχεία με id header, sidebar, content, footer. Στην HTML5 υπάρχουν νέα tags με αυτό το όνομα που εξυπηρετούν τον ίδιο σκοπό. Δημιουργούνται λοιπόν τα tags *header*, *footer*, *nav*, *aside*, *section*, *article*. Η εικόνα 12 δίνει μια ιδέα για μια ενδεικτική δομή εγγράφου HTML5.

Άλλο ένα σημαντικό χαρακτηριστικό της νέας γλώσσας είναι η ενσωματωμένη υποστήριξη (native support) μέσω τύπου βίντεο και ήχου. Με τα tags *video* και *audio*, που θα χρησιμοποιούνται όπως περίπου οι εικόνες, δεν χρειάζεται πλέον κάποιο πρόγραμμα να εγκατασταθεί στον browser για την αναπαραγωγή τους. Η ίδια η HTML5 θα διαθέτει τα κατάλληλα εργαλεία (αλγόριθμους, codecs) για να τα αναπαράγει.

Επιπλέον διευκολύνεται η δημιουργία γραφικών μέσω της ίδιας της HTML με δύο νέα API: *SVG* και *canvas*. Στην πρώτη περίπτωση των Scalable Vector Graphics δύναται η δυνατότητα να σχεδιάζονται εικόνες τύπου vector μέσω του tag `<svg>` και δίνοντας τιμές σε διάφορες ιδιότητες, όπως πλάτος, ύψος, χρώμα γεμίσματος, χρώμα περιγράμματος κ.ά. Στην περίπτωση του canvas API, στην ουσία ορίζεται μια περιοχή στο έγγραφο με το tag `<canvas>` και τις ιδιότητες πλάτος και ύψος, στην οποία

σχεδιάζονται αντικείμενα με χρήση Javascript. Η επιλογή για χρήση του ενός ή του άλλου στοιχείου εξαρτάται από τη φύση του γραφικού και το ρόλο του στη σελίδα.



Εικόνα 12: Τα νέα στοιχεία της HTML5 ορίζουν τη δομή των εγγράφων

Μια ακόμη πολύ σημαντική καινοτομία της HTML5 αφορά στις εφαρμογές που εκτελούνται εκτός σύνδεσης (offline). Συγκεκριμένα παρέχεται μια ξεχωριστή μνήμη cache για την φύλαξη των δεδομένων κάθε εφαρμογής. Έτσι μια εφαρμογή μπορεί να εκτελείται και χωρίς ενεργή σύνδεση στο διαδίκτυο. Μια εφαρμογή webmail για παράδειγμα μπορεί να χρησιμοποιείται για να αναγνωστούν mail ήδη ληφθέντα απο την τελευταία φορά που εκτελέστηκε.

Επιπροσθέτως, η HTML5 φέρνει ένα νέο API που αφορά τον γεωεντοπισμό (geolocation). Με λίγα λόγια πρόκειται για έναν τρόπο να δίνονται οι συντεταγμένες της φυσικής θέσης του χρήστη σε άλλους χρήστες, εφόσον το επιθυμεί. Αυτό θα επιτυγχάνεται εγγενώς από την ίδια την HTML.

Για την καλύτερη και ποιο σημασιολογική σήμανση του κειμένου παρέχεται μια σειρά απο νέα inline tags όπως τα `<mark>` για την επισήμανση, `<time>` για ορισμό πεδίων ώρας και ημερομηνίας, `<progress>` για την εμφάνιση μπάρας προόδου και άλλα.

Τέλος, μια εκ των σημαντικότερων αλλαγών στην HTML5 αφορά τον τρόπο που δηλώνεται η έκδοση εγγράφου. Ως γνωστόν στην αρχή κάθε εγγράφου της HTML δηλώνεται το `!DOCTYPE`, που είναι μια οδηγία προς τον browser ως προς το πώς να αντιμετωπίσει το έγγραφο, δηλαδή ποια έκδοση της HTML χρησιμοποιεί, με ποιους

περιορισμούς και σε τί mode να το προβάλλει (υπάρχουν τα λεγόμενα quirk modes για συμβατότητα με παλαιότερους browsers). Η δήλωση αυτή είναι πολύπλοκη ενώ και οι εκδόσεις είναι αρκετές. Για παράδειγμα στις σελίδες του CSS Tutor υπάρχει η παρακάτω:

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

Τα έγγραφα της HTML5 δηλώνονται με την απλή πρόταση `<!DOCTYPE html>`. Αυτό και μόνο αρκεί για να καταλάβει ο browser ότι πρόκειται για έγγραφο της HTML5.

Τα παραπάνω είναι κάποια από τα σημαντικά χαρακτηριστικά της νέας HTML. Υπάρχουν και άλλα, όπως το Web Workers API και ένας αριθμός από άλλα elements. Τα παραπάνω όμως δίνουν μια εικόνα για το που οδεύει η γλώσσα.

Η ανάπτυξη της HTML5 γίνεται προοδευτικά, δηλαδή κάποια από τα παραπάνω ήδη υποστηρίζονται από κάποιους browsers. Αυτό σημαίνει πως ολική υποστήριξη ίσως καθυστερήσει, αλλά έγκειται και σε μεγάλο βαθμό στις εταιρείες των φυλλομετρητών.

Το συμπέρασμα που βγαίνει από τα παραπάνω είναι πως πρόκειται για μια γλώσσα πιο ισχυρή από τις προηγούμενες εκδόσεις της λόγω της ενσωμάτωσης στον πυρήνα της νέων χρήσιμων στοιχείων. Τα έγγραφα της «λογικοποιούνται», κάτι που πιθανότατα θα βοηθήσει και στην έλευση του Web 3.0 για το οποίο έγινε λόγος στην περασμένη υποενότητα. Επιπλέον απαιτεί πιο αυστηρή σύνταξη για σωστή απεικόνιση (strict rendering) καθώς απομακρύνεται από την SGML.

## 6.3 CSS3

Η ανάπτυξη των CSS μοιάζει αρκετά με την ανάπτυξη της HTML. Συγκεκριμένα η γλώσσα έχει χωριστεί σε *modules*, δηλαδή σε τμήματα. Το κάθε τμήμα αφορά μια ομάδα ιδιοτήτων ή κάποιο άλλο κομμάτι της γλώσσας. Για παράδειγμα, η σύνταξη και η γραμματική της γλώσσας είναι ένα module, η ανάθεση τιμών με την κληρονομικότητα και την διαδοχή (cascading) ένα άλλο, το box model ένα τρίτο κ.ο.κ. Αυτή η τμηματική πρόοδος ωφελεί στο ότι και η υποστήριξη μπορεί να την ακολουθήσει. Δε χρειάζεται να κυκλοφορήσουν οι τελικές προδιαγραφές της γλώσσας



για να υπάρξει υποστήριξη από τους browsers. Ήδη μάλιστα κάποιες από τις νέες ιδιότητες υποστηρίζονται από μερικούς φυλλομετρητές.

Μια καινοτομία της νέας γλώσσας είναι η δυνατότητα για πολλαπλές εικόνες στο φόντο ενός στοιχείου. Η κάθε εικόνα τοποθετείται σε κάποιο σημείο του φόντου με χρήση ιδιοτήτων, όπως στο παρακάτω παράδειγμα:

```
background:{url(img1.gif) top left no-repeat,  
url(img2.jpg) top 11px no-repeat,  
url(img3.gif) bottom left no-repeat,  
url(img4.gif) left repeat-y;}
```

Άλλη μια νέα ιδιότητα είναι η *border-radius*. Αυτή δίνει τη δυνατότητα να οριστούν καμπυλωτές γωνίες στα στοιχεία. Αυτή η ιδιότητα λύνει τα χέρια πολλών σχεδιαστών και προγραμματιστών, καθώς μέχρι σήμερα επιστρατεύονταν διάφοροι τρόποι για να επιτευχθεί αυτό το αποτέλεσμα. Να σημειωθεί πως το συγκεκριμένο αποτέλεσμα μπορεί να επιτευχθεί και με την χρήση της HTML5 (SVG).

Σχετικά με την μορφοποίηση των borders, η νέα γλώσσα επιτρέπει τη χρήση εικόνων. Η βασική ιδιότητα για να επιτευχθεί αυτό είναι η *border-image*. Επιπλέον επιτρέπεται η χρήση χρώματος gradient (βαθμωτής μεταβολής) για τα borders.

Άλλες ιδιότητες σχετικές με τη μορφοποίηση είναι οι box-shadow και opacity. Η πρώτη αφορά τη σκίαση στοιχείων ενώ η δεύτερη την διαφάνεια. Και σε αυτές τις περιπτώσεις ο σχεδιαστής διευκολύνεται ιδιαίτερα στην επίτευξη του επιθυμητού αποτελέσματος μόνο με CSS.

Εμφανίζεται επίσης η έννοια του animation με χρήση αποκλειστικά CSS. Εκτός αυτού υποστηρίζονται πλέον καλύτερα τα λεγόμενα Web Fonts. Η ιδιότητα *@font-face* υπάρχει μεν πριν την έλευση της νέας έκδοσης (από τα CSS2), αλλά χρησιμοποιούσε ένα τύπο γραμματοσειράς που δεν υποστηριζόταν από πολλούς, το Embedded Open Type (eot).

Οι παραπάνω ιδιότητες είναι μερικές μόνο από αυτές που εισάγονται με τα modules της νέας έκδοσης της γλώσσας. Ο σκοπός της παρούσας ενότητας είναι να προβάλλει κάποιες βασικές ιδιότητες της νέας προδιαγραφής και να δείξει πόσο εύκολη, εφόσον υλοποιηθεί σωστά η προδιαγραφή, θα είναι η μορφοποίηση των στοιχείων με τρόπους που μέχρι πρότινος προκαλούσαν δυσκολίες.



# 7 Συμπεράσματα και Επεκτάσεις

Στην παρούσα ενότητα τοποθετούνται συμπεράσματα που αφορούν την ενασχόληση με την ανάπτυξη του συγκεκριμένου ιστότοπου αλλά και γενικότερα τον διαδικτυακό προγραμματισμό. Γενικώς, εντοπίζονται στοιχεία που πρέπει να έχει κατανοητό ο προγραμματιστής ενώ αναφέρονται και κάποιες εναλλακτικές μορφές προσέγγισης του θέματος.

## 7.1 Συμπεράσματα

Κατα τη διάρκεια της ενασχόλησης με την ανάπτυξη του ιστότοπου CSS Tutor αποκτήθηκαν και εφαρμόστηκαν τεχνικές γνώσεις και αντιμετωπίστηκαν κάποια ζητήματα και δυσκολίες. Μέσα από όλη αυτή τη διαδικασία προέκυψαν κάποια συμπεράσματα, τα οποία με συντομία και περιεκτικότητα θα παρατεθούν σε αυτό το σημείο.

### 7.1.1 Σχεδιασμός Ιστοσελίδων

Αναφέρθηκε και στο τρίτο κεφάλαιο πως ο σχεδιασμός ιστοσελίδων πρέπει να γίνεται με ιδιαίτερη προσοχή.

Συγκεκριμένα πρέπει να μελετάται πρώτα ο σκοπός που επιτελεί μια ιστοσελίδα και μαζί το κοινό στο οποίο στοχεύει, και ύστερα να ερμηνεύονται οι γενικότερες τάσεις που επικρατούν. Αυτό σημαίνει πως αν μια ιστοσελίδα στοχεύει σε ένα κοινό που ασχολείται με μια συγκεκριμένη θεματολογία, και είναι γνωστό πως ένα τέτοιο κοινό έχει κάποια χαρακτηριστικά, αυτά θα πρέπει να λαμβάνονται υπόψη.

Για παράδειγμα, οι χρήστες που ασχολούνται με τα παιχνίδια υπολογιστών κατα πάσα πιθανότητα θα έχουν μεγάλες οθόνες και αναλύσεις οθόνης. Αυτό σημαίνει κάτι για τον σχεδιαστή μιας σελίδας με θέμα το gaming. Αντίστοιχα σε μια εκπαιδευτική εφαρμογή ενδιαφέρει κυρίως το περιεχόμενο να είναι προσβάσιμο και ευανάγνωστο. Έτσι πρέπει να γίνουν κάποιοι συμβιβασμοί ώστε όσο το δυνατόν περισσότεροι χρήστες να μπορούν να επωφεληθούν από την σελίδα που φιλοξενεί την εφαρμογή.

Συμπερασματικά λοιπόν ο σχεδιασμός απαιτεί την κατανόηση του σκοπού του ιστότοπου, της κοινότητας χρηστών στην οποία απευθύνεται με τα χαρακτηριστικά που αυτή φέρει και τέλος την συμμόρφωση ως προς κάποιες σχεδιαστικές αρχές. Τα web standards είναι μια προσπάθεια προς αυτή την κατεύθυνση και η τήρησή τους με χρήση templates και frameworks είναι το βήμα προς αυτή.

### **7.1.2 Διαδικτυακός Προγραμματισμός**

Ο προγραμματισμός που έλαβε χώρα ήταν client-side, οπότε τα συμπεράσματα αφορούν κυρίως αυτού του τύπου τον προγραμματισμό.

Αναφέρθηκε πολλάκις το ζήτημα της διαφορετικής συμπεριφοράς των browsers. Αν και δόθηκε έμφαση στην απεικόνιση των σελίδων, εν τούτοις κάποιες διαφορές εμφανίζονται και στην εκτέλεση των scripts.

Η διαφορετική υλοποίηση του DOM και οι ιδιοκτησιακές ιδιότητες (proprietary properties) μπορεί να επιτρέπουν ή να μην επιτρέπουν την σωστή λειτουργία του κώδικα.

Γι' αυτό το λόγο είναι υποχρεωτικό να γίνονται δοκιμές του script σε διαφορετικούς browsers. Οι εκδόσεις του Internet Explorer και ο Firefox μαζί καταλαμβάνουν περισσότερο από το 80% του μεριδίου της αγοράς. Λαμβάνοντας υπόψη ότι ο Internet Explorer παρουσιάζει τις περισσότερες ιδιομορφίες μπορεί να ειπωθεί πως οι δοκιμές σε αυτούς τους δύο browsers εγγυώνται ένα ικανοποιητικό αποτέλεσμα. Αυτό δε σημαίνει πως πρέπει να περιορίζονται σε αυτούς τους φυλλομετρητές. Ήδη ο Opera, ο Google Chrome και ο Safari κερδίζουν ποσοστά, οπότε είναι σίγουρα πιο ασφαλές να γίνονται δοκιμές σε όλους.

Η Javascript είναι η κυρίαρχη γλώσσα προγραμματισμού στον Ιστό, τουλάχιστον όσο αφορά στο client-side τμήμα του. Πολλά frameworks και βιβλιοθήκες API έχουν κυκλοφορήσει που διευκολύνουν το έργο του προγραμματιστή, ανάλογα με αυτό που πρέπει να πετύχει. Έτσι η εκμετάλλευση των παραπάνω είναι μια πολύ καλή πρακτική.

### **7.1.3 Τεχνολογίες Παγκοσμίου Ιστού**

Είναι μια περίοδος έντονης κινητικότητας στο πεδίο των τεχνολογιών του Web. Οι πιο δημοφιλείς από αυτές βρίσκονται σε στάδιο αναδιάρθρωσης και εξέλιξης, με τα αποτελέσματά τους να μην είναι ακόμα απολύτως φανερά.

Όλες οι τεχνολογίες που παρουσιάστηκαν φαίνεται να συγκλίνουν προς μια κατεύθυνση: προσανατολίζονται στο να δώσουν σημασία στο περιεχόμενο του Web. Σήμερα ο Ιστός χρησιμοποιείται ως μέσο ενημέρωσης, ψυχαγωγίας, εκπαίδευσης και κοινωνικοποίησης. Μελλοντικά όλα τα παραπάνω θα συμβαίνουν σε ένα πλαίσιο όπου υπάρχει μια πιο λογική δομή των εγγράφων, η οποία διαθέτει σημασιολογικό περιεχόμενο.

Το Web 3.0 φαίνεται να δίνει πιο ενεργό ρόλο στο λογισμικό, καθώς προγράμματα αναλαμβάνουν να φέρουν εις πέρας διαδικασίες που μέχρι πρόσφατα πραγματοποιούσε ο χρήστης.

Η HTML γίνεται ένα πιο ισχυρό εργαλείο στα χέρια σχεδιαστών και προγραμματιστών, δίνοντας τη δυνατότητα εγγενούς παραγωγής γραφικών και υποστήριξης μέσων.

Η γλώσσα των CSS εμπλουτίζεται και προσφέρει νέες δυνατότητες στον προγραμματιστή. Η ανάπτυξή της είναι προοδευτική ενώ η χρηστικότητα των καινοτομιών εξαρτάται σε μεγάλο βαθμό από την υιοθέτησή τους από τα προγράμματα φυλλομετρητών.

#### **7.1.4 Φυλλομετρητές**

Η σημερινή κατάσταση στον χώρο των φυλλομετρητών είναι σίγουρα καλύτερη από αυτή που υπήρχε πριν μερικά χρόνια.

Οι εταιρείες φέρεται να θέλουν να συμβαδίσουν σε ένα βαθμό, όσο αφορά στην υλοποίηση των προγραμμάτων τους. Η ποιότητα και αξιοπιστία των τελευταίων, παρά τα όποια προβλήματα μπορεί να αντιμετωπίζουν, έχει σίγουρα ανέβει.

Αυτό δεν εγγυάται την απόλυτη συμβατότητα μεταξύ των browsers, κάτι το οποίο μπορεί να μη συμβεί ποτέ.

Οι εταιρείες όμως παίζουν σημαντικό ρόλο στο μέλλον του Web. Δεν παράγουν απλώς εργαλεία, προωθούν ή εμποδίζουν την προώθηση και των τεχνολογιών. Ο W3C, ο WHATWG και άλλοι οργανισμοί παράγουν προτάσεις αλλά αν αυτές δεν υλοποιηθούν από τους φυλλομετρητές τότε η αξία τους υποτιμάται.

Το συμπέρασμα λοιπόν είναι πως οι εταιρείες ανάπτυξης τέτοιου λογισμικού έχουν επιρροή αλλά και ευθύνη απέναντι στον Παγκόσμιο Ιστό.

## 7.2 Εναλλακτική προσέγγιση και επεκτάσεις

Στην σύντομη υποενότητα που ακολουθεί γίνεται αναφορά σε εναλλακτικούς τρόπους προσέγγισης του θέματος της πτυχιακής, ενώ αναφέρονται και πιθανές μελλοντικές επεκτάσεις.

### 7.2.1 Server-side προσέγγιση

Όπως αναφέρθηκε σε περασμένο κεφάλαιο η λογική στον προγραμματισμό του ιστότοπου είναι client-side: ό,τι εκτελείται από πλευράς script γίνεται στον υπολογιστή του χρήστη. Αυτός ο τύπος προγραμματισμού είναι από μια άποψη πιο απλός σε σχέση με τον προγραμματισμό που μέρος της εκτέλεσης λαμβάνει χώρα σε κάποιον server. Πολλές παράμετροι απαλείφονται, όπως επικοινωνία server-client, τύπος βάσης δεδομένων, προγραμματισμός AJAX και άλλες. Ταυτόχρονα όμως οι παραπάνω παράμετροι προσφέρουν κάποια πλεονεκτήματα.

Η εναλλακτική προσέγγιση στον προγραμματισμό του ιστότοπου λοιπόν θα ήταν να ακολουθηθεί μια server-side λογική.

Τα πλεονεκτήματα που προσφέρει μια τέτοια κίνηση αφορούν κυρίως στην αναβάθμιση και επέκταση του site. Πιο συγκεκριμένα, η ενημέρωση του υλικού και η προσθήκη περιεχομένου είναι πιο εύκολη και πιο αποδοτική, όταν συμβαίνει σε ένα περιβάλλον πιο «συγκεντρωτικό». Μέσω ενός συστήματος διαχείρισης περιεχομένου είναι εφικτή η αναδιάρθρωση της ύλης, ενώ και η διόρθωση ή ανανέωση της διευκολύνεται. Γενικότερα υπάρχει η δυνατότητα για συντήρηση του ιστότοπου και διατήρησή του σε συμφωνία με τις αλλαγές στο χώρο. Αυτό είναι πιο δύσκολο εάν το περιεχόμενο πρέπει να αλλάζει τμηματικά σε κάθε μία σελίδα.

Επιπλέον δίνεται η δυνατότητα για χρήση της τεχνολογίας AJAX (Asynchronous Javascript and XML). Ως γνωστόν η συγκεκριμένη τεχνολογία επιτρέπει την ανανέωση περιεχομένου χωρίς να χρειάζεται να ανανεωθεί το σύνολο της σελίδας. Έτσι τα τμήματα της σελίδας που χρειάζονται κάποιου είδους αλλαγή δυναμικά τροποποιούνται χωρίς να προκαλούν εκ νέου φόρτωση της σελίδας. Κάτι τέτοιο θα έβρισκε εφαρμογή εάν ο χρήστης είχε την επιλογή να «κατεβάζει» περισσότερο περιεχόμενο από τον server ώστε να πληροφορηθεί περαιτέρω για κάποια ιδιότητα ή κάποιο στοιχείο. Στην ίδια την εφαρμογή, θα ήταν δυνατό η περιοχή παραδειγμάτων να τροποποιείται ώστε να παρουσιάσει καλύτερα μια ιδιότητα.

Με τη χρήση βάσης δεδομένων από την άλλη, θα ήταν δυνατό να αλλάξει ο τρόπος που λειτουργούν κάποια πράγματα στη σελίδα, από απλά εώς και πιο σύνθετα. Για παράδειγμα θα μπορούσαν να αποθηκεύονται παραδείγματα χρήσης των ιδιοτήτων στη μορφή κανόνων τους οποίους θα ανακτούσε ο χρήστης με ένα κλικ. Επιπλέον θα δινόταν η δυνατότητα να κατασκευαστεί κάποιο τεστ στο ίδιο το site αντί για το σύστημα Blackboard, όπου ο κάθε χρήστης θα καταχωρούσε κάποιο όνομα και θα αποθηκευόταν η επίδοσή του.

Θα ήταν δυνατή επίσης η κατασκευή ενός προσαρμογέα σχεδίασης (theme selector) με μόνιμο αποτέλεσμα. Μια αναφορά στο θέμα που επιλέγει ο χρήστης για το site μπορεί να βρισκόταν στη βάση δεδομένων, με το αρχείο css που το κατασκευάζει να αλλάζει για όλα τα HTML έγγραφα αντί μόνο για το συγκεκριμένο. Η δυνατότητα αποθήκευσης του σημείου στο οποίο βρίσκεται ο χρήστης θα επέτρεπε την κατασκευή ενός δυναμικού breadcrumb. Επιπροσθέτως θα έδινε τη δυνατότητα κατασκευής μιας φόρμας επικοινωνίας (feedback form) όπου ο χρήστης θα μπορούσε να στείλει σχόλια και παρατηρήσεις.

Οι παραπάνω αλλαγές είναι ενδεικτικές των δυνατοτήτων που προσφέρει ο δεύτερος τύπος προγραμματισμού. Κάποιες από αυτές μπορούν να υλοποιηθούν και με client-side προγραμματισμό, αλλά κάτι τέτοιο θα ήταν πιθανώς όχι και τόσο αποδοτικό.

## **7.2.2 Πιθανές επεκτάσεις**

Όλες οι παραπάνω σκέψεις αποτελούν προτάσεις για πιθανή μελλοντική εξέλιξη του ιστότοπου.

Κάτι ακόμα το οποίο θα βοηθούσε τον χρήστη να κατανοήσει καλύτερα ιδιότητες και στοιχεία θα ήταν ίσως η δυνατότητα να συντάσσει δικούς του κανόνες εξαρχής και να βλέπει το αποτέλεσμα τους σε μια νέα σελίδα ή σε κάποια συγκεκριμένη περιοχή της τρέχουσας.

Επιπροσθέτα, παρότι το γραφικό περιβάλλον που κατασκευάστηκε καλύπτει τις ανάγκες του σκοπού για τον οποίο δημιουργήθηκε, ένα διαφορετικό πιο «ζωντανό» περιβάλλον με χρήση animation, drag and drop, preview thumbnails πιθανώς να βελτίωνε την αλληλεπίδραση και να τραβούσε περισσότερο την προσοχή του χρήστη. Το site άλλωστε έχει εκπαιδευτικό χαρακτήρα.

## 7.3 Σύνοψη

Στην τελευταία ενότητα της πτυχιακής εργασίας παρουσιάζονται τα συμπεράσματα του συγγραφέα σχετικά με τους τομείς της εργασίας, όπως προέκυψαν μετά την ενασχόλησή του με αυτή.

Επιπλέον αναφέρονται άλλοι τρόποι που θα μπορούσαν να υιοθετηθούν για την κάλυψη του ίδιου ζητήματος και την κατασκευή της εφαρμογής.

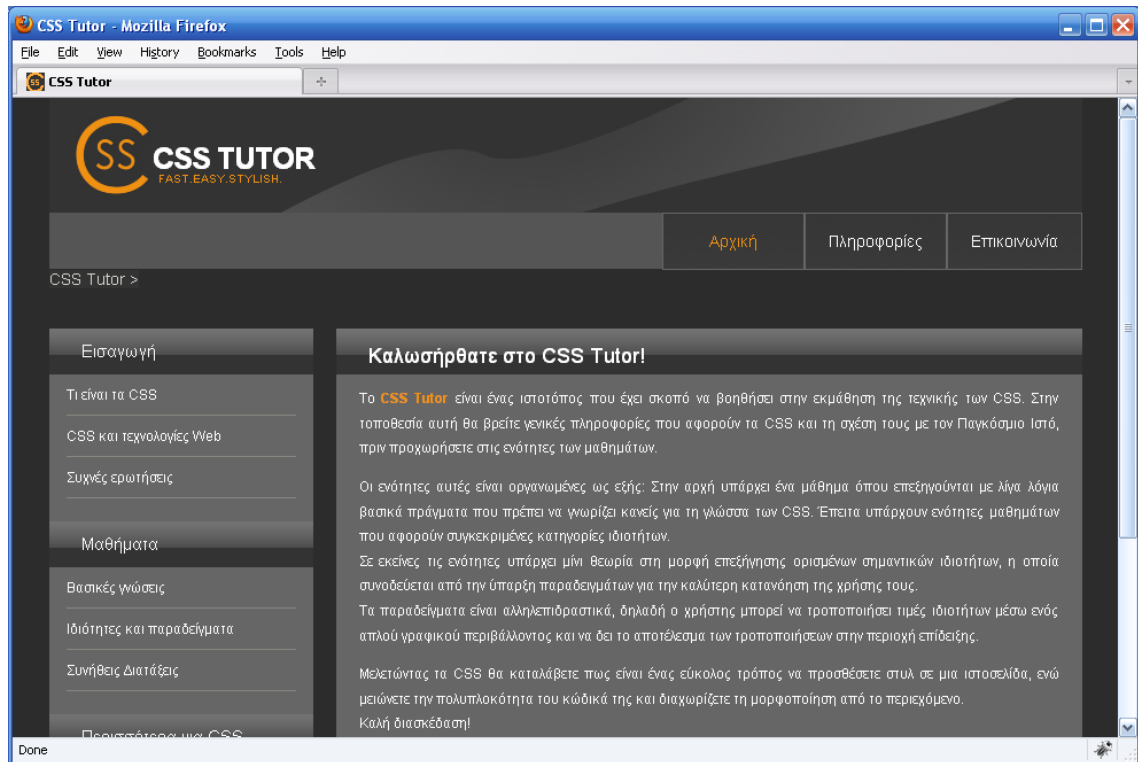


# Βιβλιογραφία

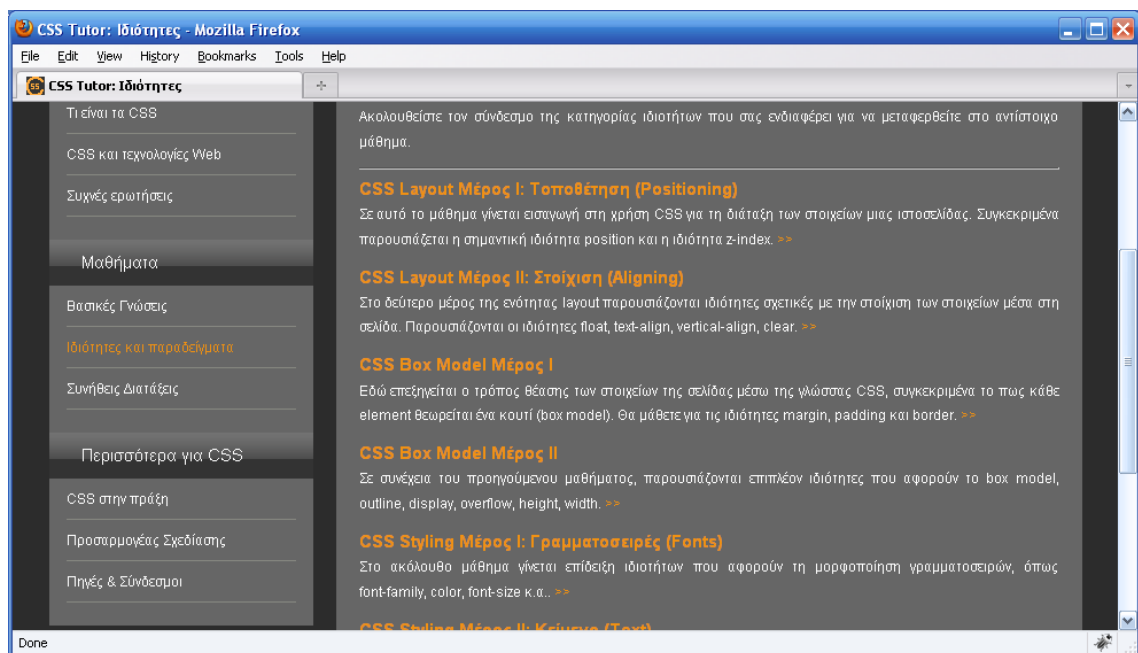
- [1] Μιχάλης Σαλαμπάσης, Σημειώσεις μαθήματος "Προγραμματισμός Διαδικτυακών Εφαρμογών", ΑΤΕΙΘ Σχολή Τεχνολογικών Εφαρμογών, Τμήμα Πληροφορικής, Φεβρουάριος 2005
- [2] Dick Oliver, Michael Morrison, "Sams Teach Yourself HTML and CSS in 24 Hours", Sams Publishing, December 14, 2005
- [3] Michael Moncur, "Sams Teach Yourself Javascript in 24 Hours", Sams Publishing, June 21, 2006
- [4] Jonathan Chaffer, Karl Swedberg, "Learning jQuery Better Interaction Design and Web Development with Simple JavaScript Techniques", PACKT Publishing, June 2007
- [5] Jeff Croft, Ian Lloyd and Dan Rubin, "Pro CSS Techniques", Apress, 2006
- [6] Eric Meyer, "CSS: The Definitive Guide, 3<sup>rd</sup> Edition", O'Reilly, November 2006
- [7] Kynn Bartlett, "Sams Teach Yourself CSS in 24 Hours", Sams Publishing, June 23, 2006
- [8] <http://www.w3c.org>, Επίσημη ιστοσελίδα οργανισμού World Wide Web Consortium
- [9] <http://www.w3schools.com>, Εκπαιδευτική ιστοσελίδα για διαδικτυακό προγραμματισμό υπό την ευθύνη του W3C
- [10] <http://jquery.com>, Επίσημη ιστοσελίδα του Javascript framework JQuery
- [11] <http://www.css3.gr>, Ιστοσελίδα με ενημερωτικό/εκπαιδευτικό χαρακτήρα σχετικά με τις προδιαγραφές της γλώσσας CSS3
- [12] <http://www.hongkiat.com>, Ιστοσελίδα με υλικό και συνδέσμους για τον σχεδιασμό ιστοσελίδων
- [13] <http://www.alistapart.com>, Ιστοσελίδα με υλικό και συνδέσμους για τον σχεδιασμό ιστοσελίδων και τον διαδικτυακό προγραμματισμό



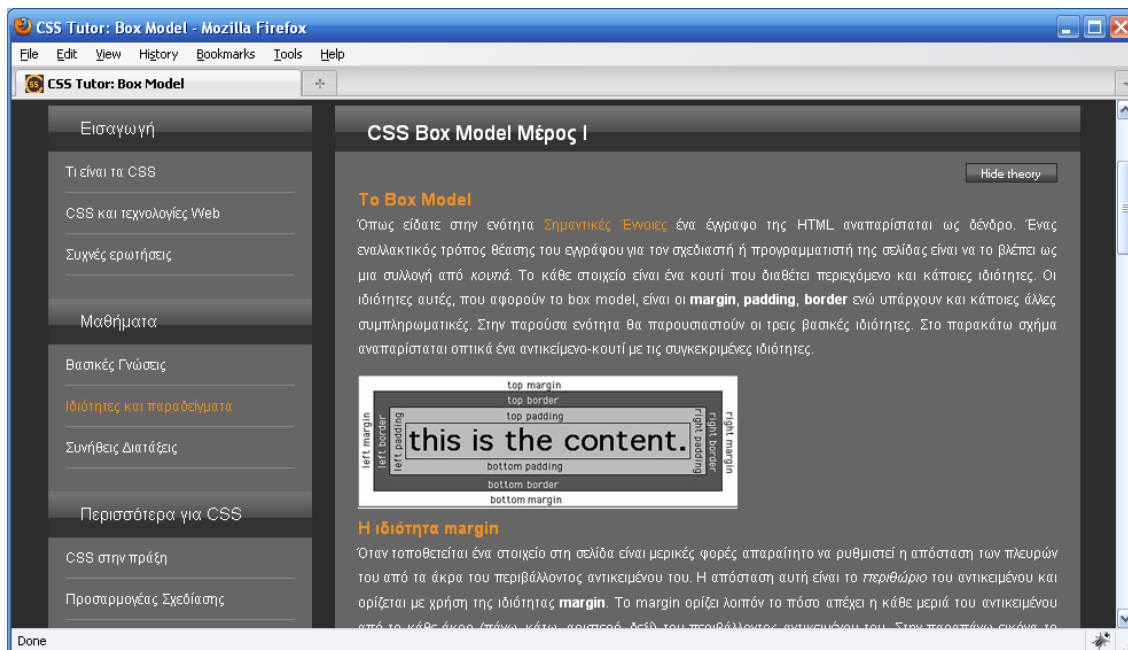
# Παράρτημα Α – Στιγμιότυπα σελίδων



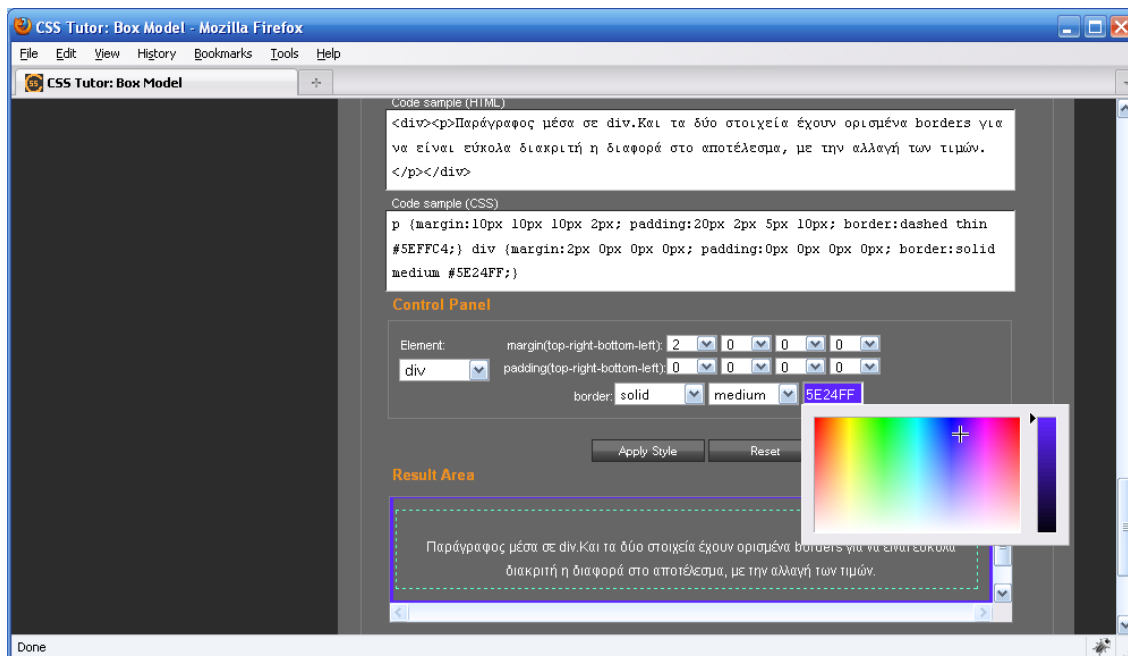
Εικόνα 13: Αρχική σελίδα του CSS Tutor



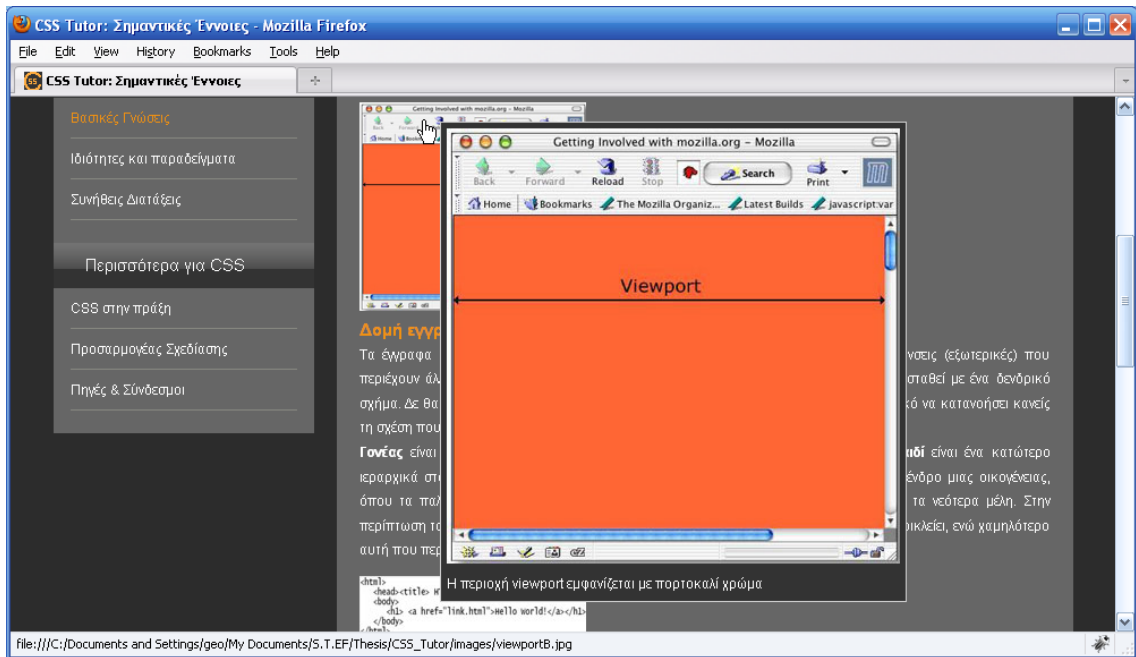
Εικόνα 14: Οι ιδιότητες στην ενότητα μαθημάτων χωρίζονται σε κατηγορίες



Εικόνα 15: Σε κάθε σελίδα μαθήματος υπάρχει θεωρία που επεξηγεί κάποια ιδιότητα ή έννοια



Εικόνα 16: Στις σελίδες μαθημάτων υπάρχει το γραφικό περιβάλλον για την δοκιμή των ιδιοτήτων



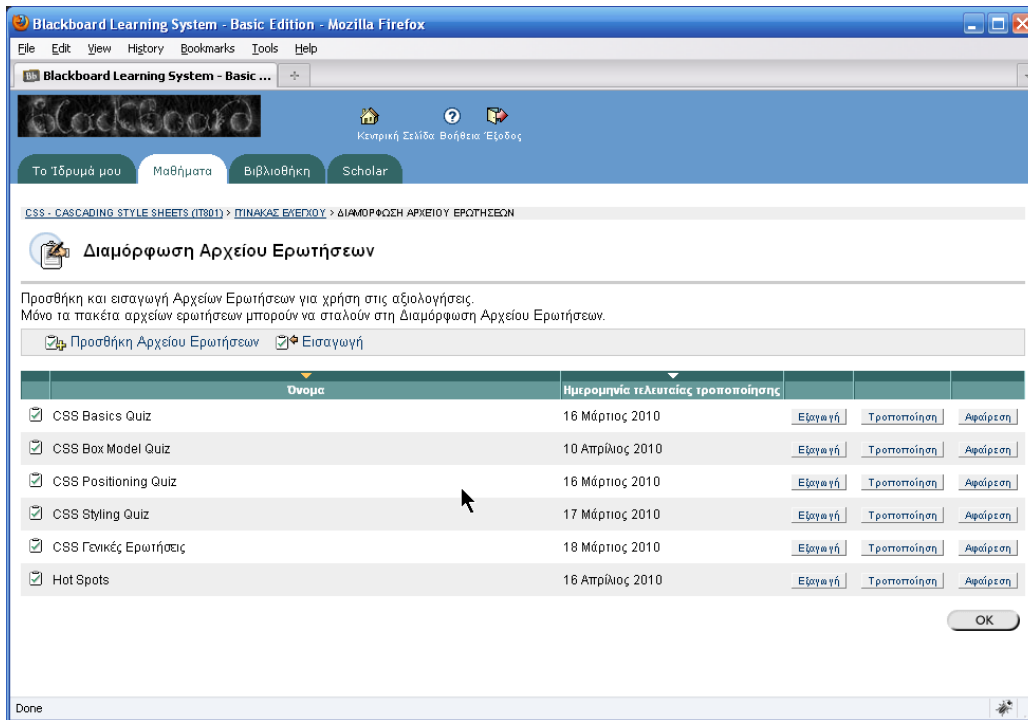
Εικόνα 17: : Μεγέθυνση των εικόνων όταν το ποντίκι περνάει πάνω από αυτές



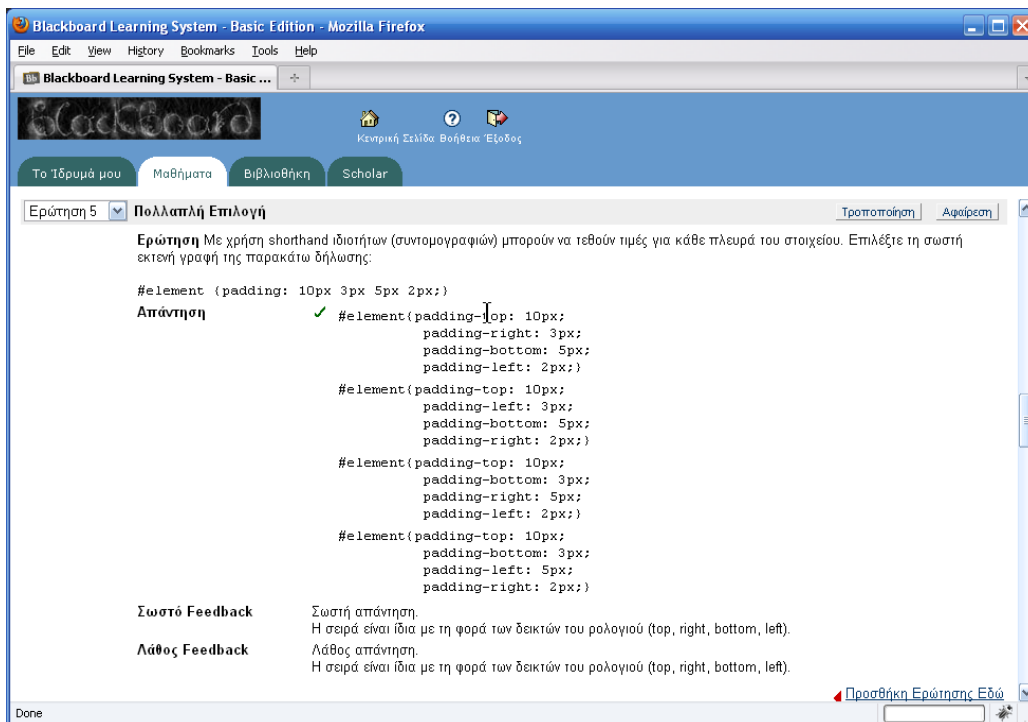
Εικόνα 18: Αλλαγή στην όψη του site με ένα κλικ



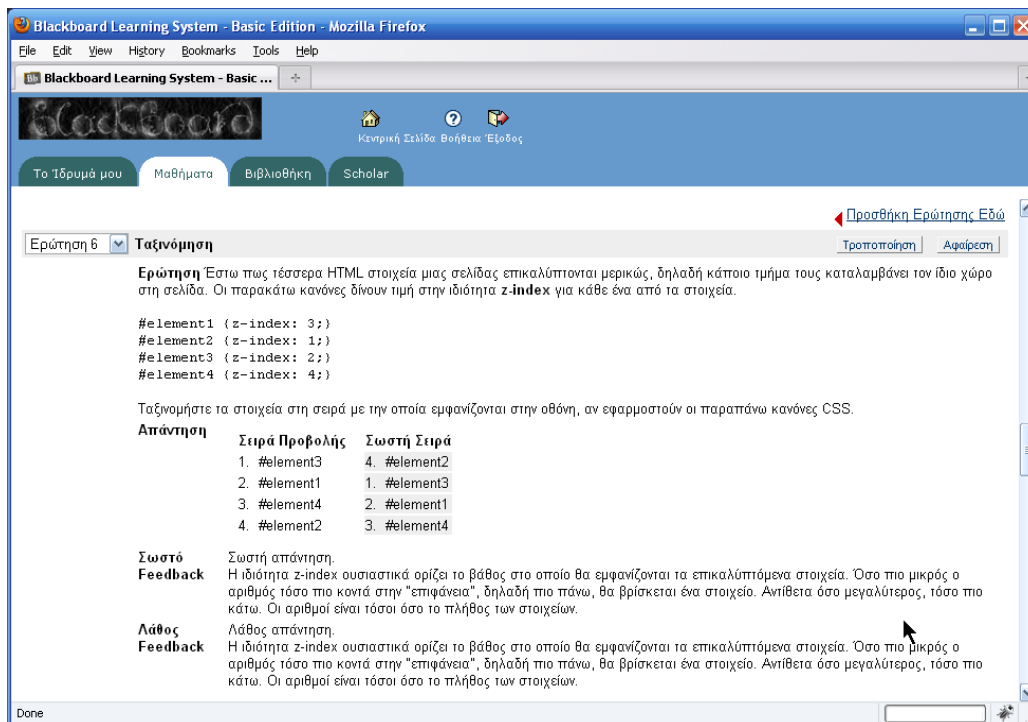
# Παράρτημα Β – Blackboard



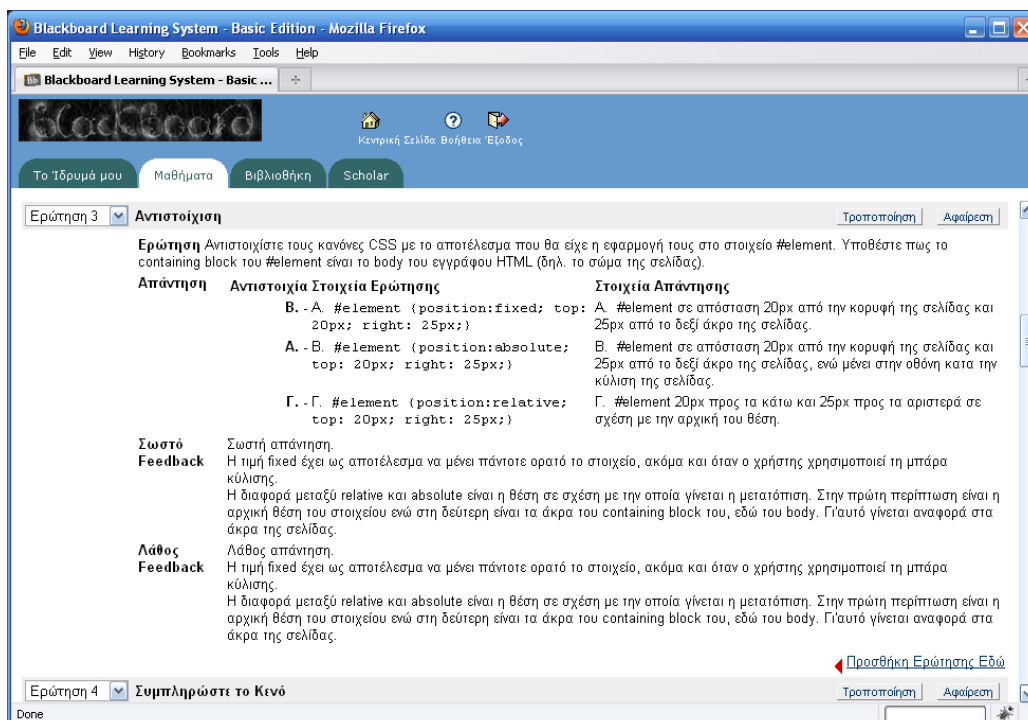
Εικόνα 19: Οι ερωτήσεις χωρίστηκαν σε κατηγορίες (question pools)



Εικόνα 20: Ερώτηση πολλαπλής επιλογής με feedback για τον χρήστη

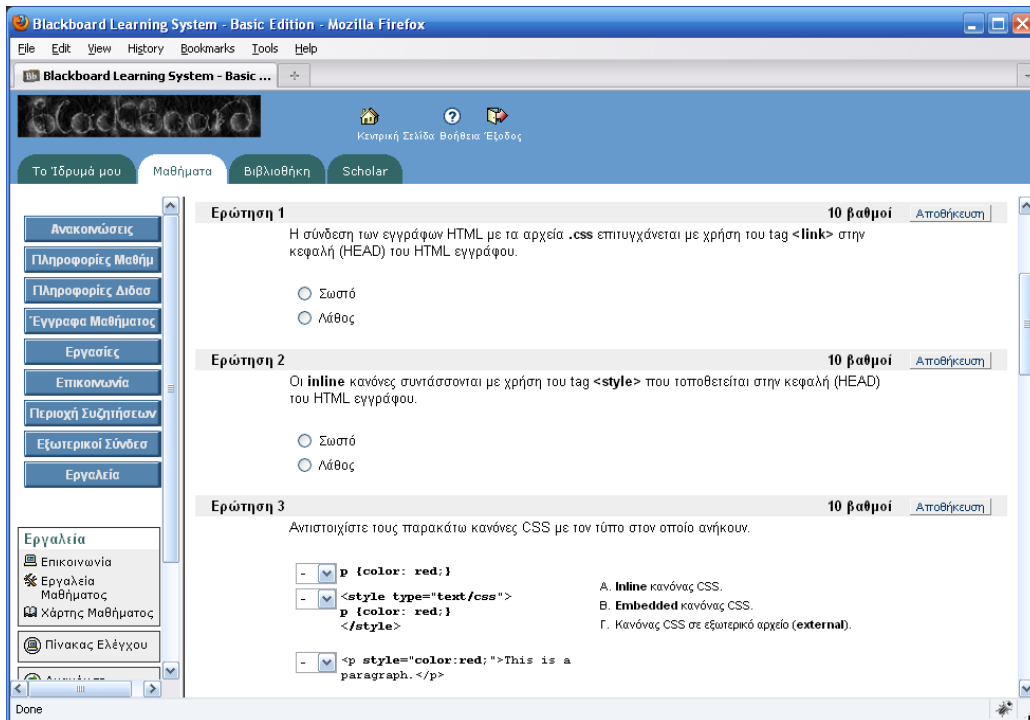


Εικόνα 21: Ερώτηση ταξινόμησης: διατύπωση, απαντήσεις και feedback

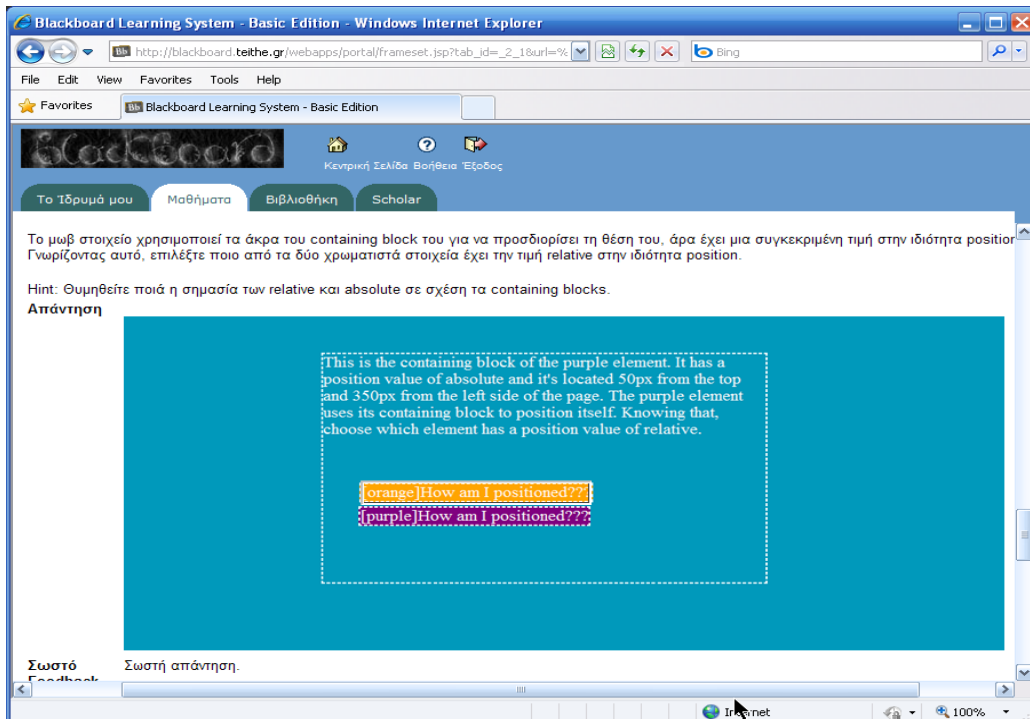


Εικόνα 22: Ερώτηση αντιστοίχισης: διατύπωση, στοιχεία και feedback

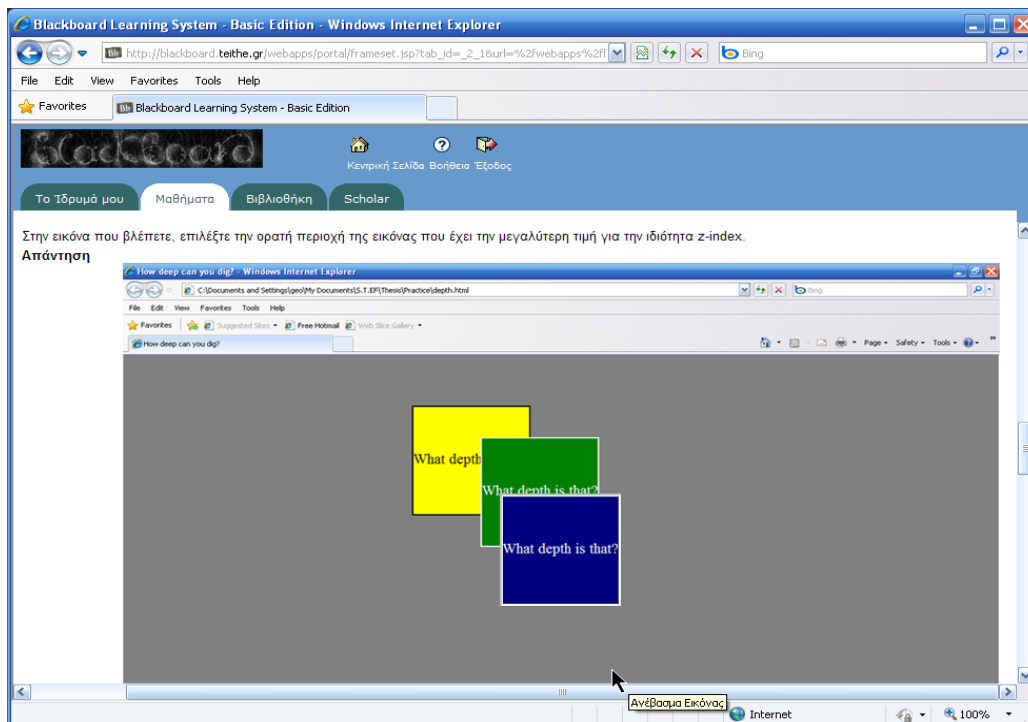




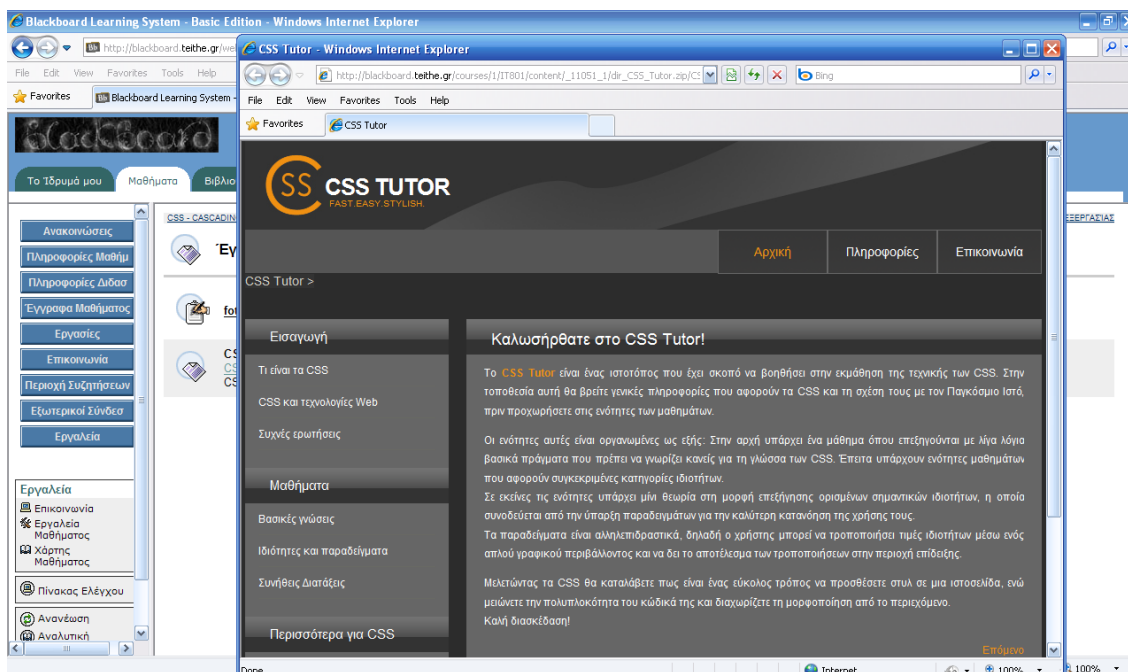
Εικόνα 23: Στις ερωτήσεις αντιστοιχούν βαθμοί για την αξιολόγηση



Εικόνα 24: Ερώτηση τύπου Hot Spot, όπου ο χρήστης απαντά πάνω στην εικόνα



Εικόνα 25: Ερώτηση τύπου Hot Spot για την επίδειξη της ιδιότητας z-index



Εικόνα 26: Το CSS Tutor ανοίγει μέσω του Blackboard