

ΘΡΑΣΥΒΟΥΛΟΣ-ΚΩΝΣΤΑΝΤΙΝΟΣ ΤΣΙΑΤΣΟΣ

ΕΚΠΑΙΔΕΥΤΙΚΑ ΠΕΡΙΒΑΛΛΟΝΤΑ ΔΙΑΔΙΚΤΥΟΥ



Ελληνικά Ακαδημαϊκά Ηλεκτρονικά
Συγγράμματα και Βοηθήματα
www.kallipos.gr

HEALINK
Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο

ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
ΕΠΕΝΔΥΣΗ ΣΤΗΝ ΚΟΙΝΩΝΙΑ ΤΗΣ ΓΝΩΣΗΣ
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

ΕΣΠΑ
2007-2013
Πρόγραμμα για το μέλλον
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

ΘΡΑΣΥΒΟΥΛΟΣ-ΚΩΝΣΤΑΝΤΙΝΟΣ ΤΣΙΑΤΣΟΣ

***ΕΚΠΑΙΔΕΥΤΙΚΑ ΠΕΡΙΒΑΛΛΟΝΤΑ
ΔΙΑΔΙΚΤΥΟΥ***



Ελληνικά Ακαδημαϊκά Ηλεκτρονικά
Συγγράμματα και Βοηθήματα
www.kallipos.gr

ΕΚΠΑΙΔΕΥΤΙΚΑ ΠΕΡΙΒΑΛΛΟΝΤΑ ΔΙΑΔΙΚΤΥΟΥ

Συγγραφή

Θρασύβουλος-Κωνσταντίνος Τσιάτσος

Κριτικός αναγνώστης

Στάυρος Δημητριάδης
Χρήστος Μπούρας

Συντελεστές έκδοσης

ΤΕΧΝΙΚΗ ΕΠΕΞΕΡΓΑΣΙΑ: Απόστολος Μαυρίδης και Στέργιος Τέγος

Copyright © ΣΕΑΒ, 2015



Το παρόν έργο αδειοδοτείται υπό τους όρους της άδειας Creative Commons Αναφορά Δημιουργού - Μη Εμπορική Χρήση - Παρόμοια Διανομή 3.0. Για να δείτε ένα αντίγραφο της άδειας αυτής επισκεφτείτε τον ιστότοπο <https://creativecommons.org/licenses/by-nc-sa/3.0/gr/>

ΣΥΝΔΕΣΜΟΣ ΕΛΛΗΝΙΚΩΝ ΑΚΑΔΗΜΑΪΚΩΝ ΒΙΒΛΙΟΘΗΚΩΝ

Εθνικό Μετσόβιο Πολυτεχνείο
Ηρώων Πολυτεχνείου 9, 15780 Ζωγράφου

www.kallipos.gr

ISBN: 978-960-603-130-4

Στη Φένια, την Άννα και τους Γονείς μου

Πίνακας περιεχομένων

Πίνακας περιεχομένων.....	4
Πίνακας συντομεύσεων-ακρωνύμια	9
Πρόλογος.....	11
Κεφάλαιο 1. Εισαγωγή στην Εκπαιδευτική Τεχνολογία	14
1. Εισαγωγή στην Εκπαιδευτική Τεχνολογία.....	14
1.1. Η εκπαιδευτική τεχνολογία.....	14
1.2. Παιδαγωγικές προσεγγίσεις στην ηλεκτρονική μάθηση	16
1.2.1. Συμπεριφοριστικό μοντέλο.....	17
1.2.2. Μάθηση Μέσω Διαθέσιμων Πόρων (Resource Based Learning)	18
1.2.3. Εξατομικευμένη ή προσωποποιημένη μάθηση	18
1.2.4. Μάθηση μεταξύ ομοτίμων (Peer learning)	18
1.2.5. Προβληματοκεντρική Μάθηση (Problem Based Learning).....	19
1.2.6. Ανακαλυπτική Μάθηση ή Μάθηση μέσω Διερεύνησης (Inquiry Based Learning)	20
1.2.7. Μάθηση με ανάπτυξη έργου (Project Based Learning)	21
1.2.8. Κοινότητες πρακτικής	21
1.2.9. Συνεργατική μάθηση και συνεργατική μάθηση από απόσταση	22
1.2.10. Άλλες προσεγγίσεις: Μεικτή Μάθηση.....	24
Βιβλιογραφία/Αναφορές	24
Κριτήρια αξιολόγησης.....	26
Κεφάλαιο 2. Εκπαιδευτικά Περιβάλλοντα Διαδικτύου: Χαρακτηριστικά & Είδη	27
2. Εκπαιδευτικά Περιβάλλοντα Διαδικτύου: Χαρακτηριστικά & Είδη.....	27
2.1. Εικονικά Μαθησιακά Περιβάλλοντα	27
2.1.1. Σαφής και προμελετημένη σχεδίαση	28
2.1.2. Τόποι επικοινωνίας	28
2.1.3. Διάφοροι τρόποι αναπαράστασης	29
2.1.4. Συμμετοχή των χρηστών	29
2.1.5. Υποστήριξη διάφορων μοντέλων μάθησης	31
2.1.6. Αξιοποίηση διάφορων τεχνολογιών και υποστήριξη ποικίλες παιδαγωγικών προσεγγίσεων	32
2.2. Συστήματα Διαχείρισης Μάθησης	34
2.2.1. Ο Ρόλος και οι Λειτουργίες των Συστημάτων Διαχείρισης Μάθησης.....	34
2.2.2. Η Αρχιτεκτονική των Συστημάτων Διαχείρισης Μάθησης.....	35
2.3. Περιβάλλοντα Υποστήριξης Συνεργατικής Μάθησης Μέσω Υπολογιστή	36
2.3.1. Ηλεκτρονική συνεργασία και συνεργατικά περιβάλλοντα	36
2.3.2. Σχεδιαστικά χαρακτηριστικά συνεργατικών περιβαλλόντων μάθησης	45
2.4. Προσωπικά Μαθησιακά Περιβάλλοντα & Προσαρμοστικά Εκπαιδευτικά Συστήματα	47

2.4.1.	Προσωπικά Μαθησιακά Περιβάλλοντα	47
2.4.2.	Προσαρμοστικά Εκπαιδευτικά Συστήματα	50
2.5.	Συστήματα Υποστήριξης Μαζικών Ελεύθερων Διαδικτυακών Μαθημάτων	52
2.5.1.	Στόχευση των MOOCs και Είδη	52
2.5.2.	Τα MOOCs στην Ελλάδα και Διεθνώς	53
2.5.3.	Λειτουργικά χαρακτηριστικά μιας πλατφόρμας MOOC.....	55
2.5.4.	Ανοιχτά Θέματα: Το πρόβλημα της πιστοποίησης στα MOOCs	55
	Βιβλιογραφία/Αναφορές	57
	Κριτήρια αξιολόγησης.....	61
	Κεφάλαιο 3. Χρήση, Σχεδιασμός και Επιλογή Εκπαιδευτικών Περιβαλλόντων Διαδικτύου.	62
3.	Χρήση, Σχεδιασμός και Επιλογή Εκπαιδευτικών Περιβαλλόντων Διαδικτύου	62
3.1.	Εισαγωγικά Θέματα	62
3.2.	Χρήση Εκπαιδευτικών Περιβαλλόντων Διαδικτύου	62
3.2.1.	Υποστήριξη Συμπεριφοριστικού Μοντέλου	65
3.2.2.	Υποστήριξη Μάθησης Μέσω Διαθέσιμων Πόρων	66
3.2.3.	Υποστήριξη Εξατομικευμένης/Προσωποποιημένης Μάθησης	67
3.2.4.	Υποστήριξη Μάθησης Μεταξύ Ομότιμων (Peer learning)	68
3.2.5.	Υποστήριξη Μάθησης με Ανάπτυξη Έργου, Προβληματοκεντρικής Μάθησης, Ανακαλυπτικής Μάθησης και Συνεργατικής μάθησης.....	69
3.2.6.	Υποστήριξη Κοινότητας Πρακτικής	70
3.3.	Επιλογή Εκπαιδευτικών Περιβαλλόντων Διαδικτύου.....	71
3.3.1.	Μέθοδοι για την Επιλογή Εκπαιδευτικών Περιβαλλόντων Διαδικτύου	72
3.3.2.	Μέθοδος Σύγκρισης και Επιλογής Εκπαιδευτικών Περιβαλλόντων Διαδικτύου.....	73
3.4.	Πλαίσιο Σχεδιασμού και Υλοποίησης Εκπαιδευτικών Περιβαλλόντων Διαδικτύου.....	76
3.4.1.	Στάδιο 0: Ανάλυση των απαιτήσεων	77
3.4.2.	Στάδιο 1: Εξαγωγή των βασικών τεχνικών και λειτουργικών χαρακτηριστικών του συστήματος	80
3.4.3.	Στάδιο 2: Επιλογή τεχνολογιών και προτύπων.....	82
3.4.4.	Στάδιο 3: Τεχνικές προδιαγραφές και αρχιτεκτονική του συστήματος	83
3.4.5.	Στάδιο 4: Ανάπτυξη συστήματος.....	85
3.4.6.	Στάδιο 5: Αξιολόγηση συστήματος.....	86
	Βιβλιογραφία/Αναφορές	86
	Κριτήρια αξιολόγησης.....	86
	Κεφάλαιο 4. Τεχνολογίες και Πρότυπα για την Υποστήριξη Εκπαιδευτικών Περιβαλλόντων Διαδικτύου	87
4.	Τεχνολογίες και Πρότυπα για την Υποστήριξη Εκπαιδευτικών Περιβαλλόντων Διαδικτύου	87
4.1.	Αρχιτεκτονικές Εκπαιδευτικών Περιβαλλόντων Διαδικτύου.....	87
4.1.1.	Αρχιτεκτονικές Εφαρμογών Παγκόσμιου Ιστού (Web 1.0).....	87
4.1.2.	Βασικές αρχιτεκτονικές συστημάτων συνεργασίας από απόσταση	88

4.2.	Τεχνολογίες.....	89
4.2.1.	Πλατφόρμες και Συστήματα Διαχείρισης Μάθησης.....	90
4.2.2.	Πλατφόρμες και Συστήματα Διαχείρισης Μαθησιακού Περιεχομένου.....	92
4.2.3.	Πλατφόρμες και Τεχνολογίες Ροής Δεδομένων	94
4.2.4.	Πλατφόρμες Τηλεδιάσκεψης και Σύγχρονης Τηλεκπαίδευσης	95
4.3.	Εκπαιδευτικά Πρότυπα και Προδιαγραφές	96
4.3.1.	Διαλειτουργικότητα Ερωτήσεων και Τεστ	98
4.3.2.	Διαλειτουργικότητα Μαθησιακών Εργαλείων	98
4.3.3.	Διαλειτουργικότητα Μαθησιακών Αναλύσεων	99
4.3.4.	Προσβασιμότητα	100
4.3.5.	Διαλειτουργικότητα Περιεχομένου & Συστημάτων.....	100
4.3.6.	Καθορισμός Δεξιοτήτων ή Εκπαιδευτικών Στόχων.....	100
4.3.7.	Προδιαγραφές για Πακετάρισμα και Διαχείριση Περιεχομένου	101
4.3.8.	Σχεδιασμός & Προγραμματισμός Μαθημάτων	101
4.3.9.	Προδιαγραφές για Εταιρικές Υπηρεσίες.....	102
4.3.10.	Ηλεκτρονικό Χαρτοφυλάκιο (ePortfolio).....	102
4.3.11.	Πακετάρισμα Πληροφοριών Μαθητή	102
4.3.12.	Σχεδίαση Μάθησης (Learning Design)	102
4.3.13.	Γενικές Υπηρεσίες Παγκόσμιου Ιστού	102
4.3.14.	Υπηρεσίες Μαθησιακών Πληροφοριών	103
4.3.15.	Ανταλλαγή Μαθησιακών Αντικειμένων.....	103
4.3.16.	Μεταδομένα	103
4.3.17.	Διαλειτουργικότητα Καταλόγων Πόρων.....	104
4.3.18.	Ακολουθία Μαθησιακών Δραστηριοτήτων (Sequencing Specification)	104
4.3.19.	Ηλεκτρονικά Έγγραφα	104
	Βιβλιογραφία/Αναφορές	104
	Κριτήρια αξιολόγησης.....	104
	Κεφάλαιο 5. Βασικές Τεχνολογίες για την Υποστήριξη ΕΠΔ	105
5.	Βασικές Τεχνολογίες για την Υποστήριξη ΕΠΔ.....	105
5.1.	Εισαγωγικά στοιχεία	105
5.1.1.	PHP.....	105
5.1.2.	MySQL.....	106
5.1.3.	Apache web server	106
5.1.4.	Συνεργασία Apache, MySQL PHP	107
5.2.	Εγκατάσταση Apache, MySQL και PHP σε Linux	108
5.2.1.	Προαπαιτούμενα.....	108
5.2.2.	Εγκατάσταση της MySQL 5.....	108
5.2.3.	Εγκατάσταση του Apache2	108
5.2.4.	Εγκατάσταση της PHP5	109
5.2.5.	Έλεγχος της PHP5.....	109

5.2.6.	Υποστήριξη MySQL από την PHP5	110
5.2.7.	Διαχείριση βάσεων δεδομένων με τη χρήση του phpMyAdmin.....	111
5.3.	Εγκατάσταση Apache, MySQL και PHP σε Windows.....	111
5.3.1.	Εγκατάσταση σε εξυπηρετητή Apache2	112
5.3.2.	Αντιμετώπιση Προβλημάτων.....	112
5.3.3.	Εγκατάσταση της PHP5	112
5.3.4.	Αντιμετώπιση Προβλημάτων.....	113
5.3.5.	Εγκατάσταση της MySQL5.....	113
	Βιβλιογραφία/Αναφορές	114
	Κριτήρια αξιολόγησης.....	115
	Κεφάλαιο 6. Εγκατάσταση και χρήση της πλατφόρμας Moodle.....	116
6.	Εγκατάσταση και χρήση της πλατφόρμας Moodle.....	116
6.1.	Τι είναι το Moodle – Ιστορικά στοιχεία	116
6.2.	Φιλοσοφία του Moodle.....	117
6.3.	Επισκόπηση χαρακτηριστικών του Moodle	118
6.4.	Χρήση και δομή της πλατφόρμας Moodle.....	120
6.5.	Εγκατάσταση του Moodle.....	122
6.5.1.	Εγκατάσταση με χρήση του Moodle Installer Package	122
6.5.2.	Εγκατάσταση με χρήση Bitnami.....	125
6.6.	Παραμετροποίηση Moodle.....	125
6.6.1.	Αλλαγή/προσθήκη γλώσσας στην πλατφόρμα Moodle	125
6.6.2.	Εγκατάσταση πρόσθετων και αλλαγή εμφάνισης	127
6.6.3.	Εγκατάσταση πρόσθετων και αλλαγή εμφάνισης	132
6.6.4.	Προσθήκη χρηστών στην πλατφόρμα Moodle.....	134
6.6.5.	Εμφάνιση ημερολογίου	136
6.6.6.	Εμφάνιση/τροποποίηση block.....	137
	Βιβλιογραφία/Αναφορές	138
	Κριτήρια αξιολόγησης.....	139
	Κεφάλαιο 7. Υλοποίηση Δυναμικού Εκπαιδευτικού Ιστοτόπου	140
7.	Υλοποίηση Δυναμικού Εκπαιδευτικού Ιστοτόπου.....	140
7.1.	Σχεδιασμός Ιστοτόπου.....	140
7.1.1.	Δομή ιστοτόπου	140
7.1.2.	Λειτουργίες και χαρακτηριστικά της βάσης δεδομένων.....	141
7.1.3.	Πίνακες.....	141
7.1.4.	Λειτουργίες και χαρακτηριστικά της βάσης δεδομένων.....	142
7.1.5.	Γενική δομή της ιστοσελίδας	142
7.2.	Εισαγωγή σε Διαδικτυακές Τεχνολογίες.....	150
7.2.1.	Εισαγωγή στην HTML	150
7.2.2.	Εισαγωγή στη δημιουργία ιστοσελίδων με PHP	170

7.2.3.	Εισαγωγή στο CSS	191
7.2.4.	Εισαγωγή στην MySQL.....	209
7.3.	Υλοποίηση Δυναμικού Εκπαιδευτικού Ιστοτόπου	216
7.3.1.	Είσοδος και έλεγχος χρήστη	222
7.3.2.	Αποσύνδεση χρήστη	229
7.3.3.	Εισαγωγή δεδομένων σε έναν πίνακα μιας βάσης δεδομένων	229
7.3.4.	Επεξεργασία δεδομένων σε έναν πίνακα μιας βάσης δεδομένων.....	232
7.3.5.	Αποστολή e-mail με χρήση PHP	233
7.3.6.	Αποστολή και αποθήκευση αρχείου στον εξυπηρετητή με χρήση PHP	234
	Βιβλιογραφία/Αναφορές	239
	Κριτήρια αξιολόγησης.....	239
	Κεφάλαιο 8. Αξιολόγηση Εκπαιδευτικών Δικτυακών Τύπων	240
8.	Αξιολόγηση Εκπαιδευτικών Δικτυακών Τύπων	240
8.1.	Τι είναι η αξιολόγηση γενικά	240
8.2.	Στόχευση αξιολόγησης εκπαιδευτικών περιβαλλόντων διαδικτύου	241
8.3.	Στρατηγικές, μέθοδοι και εργαλεία αξιολόγησης εκπαιδευτικών περιβαλλόντων διαδικτύου υπό την ερευνητική, ακαδημαϊκή και μαθησιακή διάσταση	243
8.4.	Εργαλεία και κριτήρια αξιολόγησης εκπαιδευτικών περιβαλλόντων διαδικτύου και ψηφιακού εκπαιδευτικού περιεχομένου	251
8.4.1.	Αποτίμηση της διδακτικής πράξης των διδασκόντων με εστίαση στις καταγεγραμμένες διαλέξεις	251
8.4.2.	Κριτήρια οργάνωσης και παρουσίασης του εκπαιδευτικού περιεχομένου.....	252
8.4.3.	Αξιολόγηση ευχρηστίας	254
8.5.	Παράδειγμα αξιολόγησης ευχρηστίας: Η περίπτωση του OpenDelos	257
8.6.	Παράδειγμα τελικής αξιολόγησης εκπαιδευτικού δικτυακού τύπου: Η περίπτωση των απομακρυσμένων εργαστηρίων του eScience.....	263
	Βιβλιογραφία/Αναφορές	271
	Κριτήρια αξιολόγησης.....	272
	Ευρετήριο - Γλωσσάρι.....	273

Πίνακας συντομεύσεων-ακρωνύμια

ACE	American Council on Education
ADL	Advanced Distributed Learning
AEH	Adaptive Educational Hypermedia
AES	Adaptive Educational Systems
AES	Automated Essay Scoring
APIP	Accessible Portable Item Protocol
AUF	Agence Universitaire de la Francophonie
CAI	Computer Assisted Instruction
CMC	Computer Mediated Communication
cMOOC	Connectivist MOOC
CMS	Content Management System
CPR	Calibrated Peer Review
CPS	Course Planning And Scheduling
CSCL	Computer Supported Collaborative Learning
CSCW	Computer Supported Collaborative Work
CSS	Cascading Style Sheets
CVE	Collaborative Virtual Environment
EDI	Electronic Data Interchange
HTML	Hypertext Markup Language
IBL	Inquiry Based Learning
ICT	Information and Communication Technologies
ILS	Individual Learning Style
KLM	Keystroke Level Model
LDAP	Lightweight Directory Access Protocol
LETSI	International Federation for Learning-Education-Training Systems Interoperability
LMS	Learning Management Systems
LODE	Learning Object Discovery & Exchange
LOM	Learning Object Metadata
LTS	Learning Technology System
LTSC	Learning Technology Standards Committee
MCU	Multipoint Conference Unit
MOOCs	Massive Open Online Courses
MOO	MUD Object Oriented
MUD	Multi User Dungeon
MUVE	Multi-User Virtual Environment
OCW	Open Courseware
OER	Open Educational Resource

PLE	Personal Learning Environment
QTI	Question & Test Interoperability
RBL	Resource Based Learning
RLI	Resource List Interoperability
RTMP	Real Time Message Protocol
RTSP	Real Time Streaming Protocol
SaaS	Software as a Service
SCORM	Sharable Content Object Reference Model
SVG	Scalable Vector Graphics
UML	Unified Modeling Language
VDEX	Vocabulary Definition Exchange
VET	Vocational Education and Training
VLE	Virtual Learning Environment
WHATWG	Web Hypertext Application Technology Working Group
WSDL	Web Services Description Language
WYSIWIS	What You See Is What I See
XCP	Xen Cloud Platform
xMOOC	Extended MOOC
EMΠ	Εικονικό Μαθησιακό Περιβάλλον
ΕΠΑ	Εκπαιδευτικό Περιβάλλον Διαδικτύου
ΜΑΔΜ	Μαζικό ανοικτό Διαδικτυακό Μάθημα
ΠΜΠ	Προσωπικό Μαθησιακό Περιβάλλον
ΠΕΣ	Προσαρμοστικό Εκπαιδευτικό Σύστημα
ΣΜΜΥ	Συνεργατική Μάθηση Μέσω Υπολογιστή
ΣΔΜ	Σύστημα Διαχείρισης Μάθησης
ΤΠΕ	Τεχνολογίες Πληροφορίας και Επικοινωνιών

Κεφάλαιο 7. Υλοποίηση Δυναμικού Εκπαιδευτικού Ιστοτόπου

Σύνοψη

Σε αυτό το κεφάλαιο θα παρουσιαστούν όλα τα απαραίτητα βήματα που χρειάζονται προκειμένου να σχεδιάσουμε και να υλοποιήσουμε ένα δυναμικό εκπαιδευτικό ιστοτόπο, χρησιμοποιώντας τις διαθέσιμες τεχνολογίες που περιγράψαμε στο κεφάλαιο 5 και τις οποίες θα δούμε αναλυτικά στη συνέχεια, καθώς θα προχωράει η υλοποίησή μας. Συμπεριλήφθηκαν όλες εκείνες οι λειτουργίες που είναι απαραίτητες για τη δημιουργία δυναμικών ιστοσελίδων, προς την κατεύθυνση της εξάσκησης και της απόκτησης συγκεκριμένων δεξιοτήτων προγραμματισμού.

Προαπαιτούμενη γνώση

Βασική χρήση υπολογιστών, περιηγητή ιστοσελίδων, HTML & CSS

7. Υλοποίηση Δυναμικού Εκπαιδευτικού Ιστοτόπου

7.1. Σχεδιασμός Ιστοτόπου

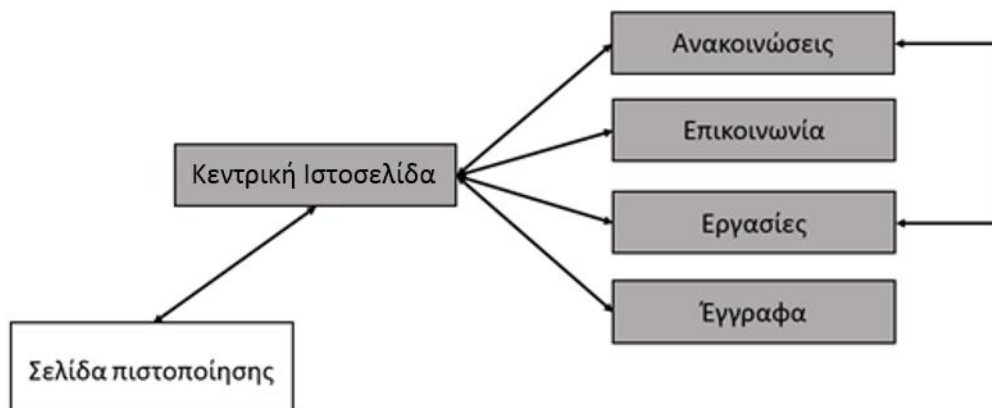
Στόχος είναι η δημιουργία ενός δυναμικού ιστοτόπου (website) που να μπορεί να υποστηρίξει ένα ηλεκτρονικό μάθημα και να προσφέρει ένα σύνολο λειτουργιών και εργαλείων, τόσο για το διδάσκοντα-διαχειριστή του ιστοτόπου, όσο και για τους μαθητές-επισκέπτες του ιστοτόπου. Οπότε, γίνεται αμέσως αντιληπτή η ανάγκη για υποστήριξη χρηστών με διαφορετικούς ρόλους, προκειμένου να διακρίνεται ο χρήστης με την ιδιότητα του διαχειριστή, από το χρήστη με την ιδιότητα του επισκέπτη. Είναι λογικό πως κάθε διαχειριστής θα έχει επαυξημένες δυνατότητες και προσβασιμότητα, σε σύγκριση με τον απλό χρήστη-επισκέπτη. Σε κάθε περίπτωση, όλοι οι χρήστες θα πρέπει να είναι πιστοποιημένοι, δηλαδή θα πρέπει απαραίτητα να είναι εγγεγραμμένοι στη βάση δεδομένων και για να εισέλθουν στον ιστότοπο θα πρέπει κάθε φορά να πιστοποιηθούν (δηλαδή να «κάνουν login») με το email που δήλωσαν και τον προσωπικό κωδικό (password) της επιλογής τους.

7.1.1. Δομή ιστοτόπου

Ο ιστότοπος θα αποτελείται από πέντε βασικές σελίδες:

- Αρχική σελίδα (index).
- Ανακοινώσεις (announcement).
- Επικοινωνία (communication).
- Έγγραφα μαθήματος (documents).
- Εργασίες μαθήματος (homework).
- Είσοδος χρήστη (login).

Πριν να εμφανιστούν οι παραπάνω ιστοσελίδες, τόσο στον καθηγητή όσο και στο μαθητή, θα πρέπει να εμφανίζεται αρχικά η ιστοσελίδα πιστοποίησης στην οποία ο κάθε εγγεγραμμένος χρήστης θα εισάγει τα προσωπικά στοιχεία εισόδου. Επομένως, συνολικά θα εμφανίζονται 6 επιμέρους ιστοσελίδες. Στο παρακάτω σχήμα προβάλλεται πιο παραστατικά η δομή του ιστοτόπου:



Εικόνα 7.1. Η δομή του ιστοτόπου της εργασίας

Το περιεχόμενο του «Τίτλου Ιστοσελίδας» θα πρέπει να είναι μορφοποιημένο σαν «heading 1». Ο τίτλος αυτός θα πρέπει να εμφανίζεται και στο παράθυρο του φυλλομετρητή σας (χρησιμοποιήστε την ετικέτα <title> της HTML). Όλες οι ιστοσελίδες του ιστοτόπου θα έχουν χρώμα της επιλογής σας στο φόντο τους. Η κάθε ιστοσελίδα θα πρέπει να έχει τη μορφή και τη δομή του περιεχομένου που περιγράφεται στις ακόλουθες παραγράφους. Η μορφοποίηση αυτή θα πρέπει να γίνει με χρήση style sheets (CSS) αποκλειστικά.

7.1.2. Λειτουργίες και χαρακτηριστικά της βάσης δεδομένων

Σε αυτή την ενότητα κομμάτι θα προσδιοριστεί η δομή και η σχεδίαση της βάσης δεδομένων, πάνω στην οποία θα βασιστεί η ιστοσελίδα που θα δημιουργηθεί. Για περισσότερες πληροφορίες σχετικά με την εγκατάσταση του εξυπηρετητή της βάσης δεδομένων θα πρέπει να ανατρέξετε στην ενότητα 5.2.2 του βιβλίου.

Έστω ότι η βάση που θα δουλέψετε θα λέγεται «mydb». Σε αυτήν θα πρέπει να έχει δικαιώματα πρόσβασης και επεξεργασίας ένας συγκεκριμένος χρήστης της βάσης δεδομένων και έστω ότι αυτός είναι ο «myuser» με password «myrsw». Τα στοιχεία αυτά, θα χρησιμοποιηθούν στις αντίστοιχες κλήσεις της MySQL όταν θα γράφετε τον κώδικα PHP, αλλά για αυτά θα πούμε αναλυτικά σε επόμενη ενότητα.

7.1.3. Πίνακες

Εφόσον, λοιπόν, έχει εγκατασταθεί η βάση δεδομένων, θα προχωρήσουμε στη δημιουργία πινάκων, οι οποίοι θα χρησιμεύσουν στην αποθήκευση των δεδομένων μας. Παρακάτω περιγράφονται οι απαραίτητοι πίνακες μαζί με τα χαρακτηριστικά τους.

7.1.3.1 Πίνακας για την αποθήκευση χρηστών

Για κάθε πιστοποιημένο χρήστη θα αποθηκεύονται τα παρακάτω δεδομένα της βάσης δεδομένων σε αντίστοιχο πίνακα και έστω ότι ονομάζουμε τον πίνακα αυτό «user»:

- Όνομα.
- Επώνυμο.
- Login name (που θα είναι το e-mail του και θα είναι μοναδικό).
- Password.
- Ρόλος (που θα μπορεί να πάρει μόνο δύο τιμές: Tutor ή Student).

7.1.3.2 Πίνακας για την αποθήκευση ανακοινώσεων

Οι ανακοινώσεις του μαθήματος θα αποθηκεύονται στον αντίστοιχο πίνακα «announcements» της βάσης δεδομένων. Τα βασικά δεδομένα, άρα και αντίστοιχα πεδία, του πίνακα θα είναι τα παρακάτω:

- Αύξων αριθμός
- Ημερομηνία
- Θέμα
- Κυρίως κείμενο

7.1.3.3 Πίνακας για την αποθήκευση εγγράφων μαθήματος

Αντίστοιχα, τα έγγραφα του μαθήματος θα αποθηκεύονται στον πίνακα «documents» της βάσης δεδομένων. Τα βασικά δεδομένα (άρα και αντίστοιχα πεδία) του πίνακα θα είναι τα παρακάτω:

- Αύξων αριθμός.
- Τίτλος εγγράφου.
- Περιγραφή.
- Όνομα/θέση αρχείου.

7.1.3.4 Πίνακας για την αποθήκευση εργασιών μαθήματος

Οι εργασίες του μαθήματος θα αποθηκεύονται στον πίνακα «assignments» της βάσης δεδομένων. Τα βασικά δεδομένα (άρα και αντίστοιχα πεδία) του πίνακα θα είναι τα παρακάτω:

- Αύξων αριθμός.

- Στόχοι.
- Εκφώνηση (όνομα/θέση αρχείου).
- Παραδοτέα.
- Ημερομηνία παράδοσης.

7.1.4. Λειτουργίες και χαρακτηριστικά της βάσης δεδομένων

Οι άμεσες λειτουργίες που θα συντελούνται στη βάση δεδομένων και ταυτόχρονα αποτελούν τις δυνατές επιλογές του διαχειριστή της ιστοσελίδας (όποιος δηλαδή κατέχει το ρόλο του tutor) είναι οι παρακάτω:

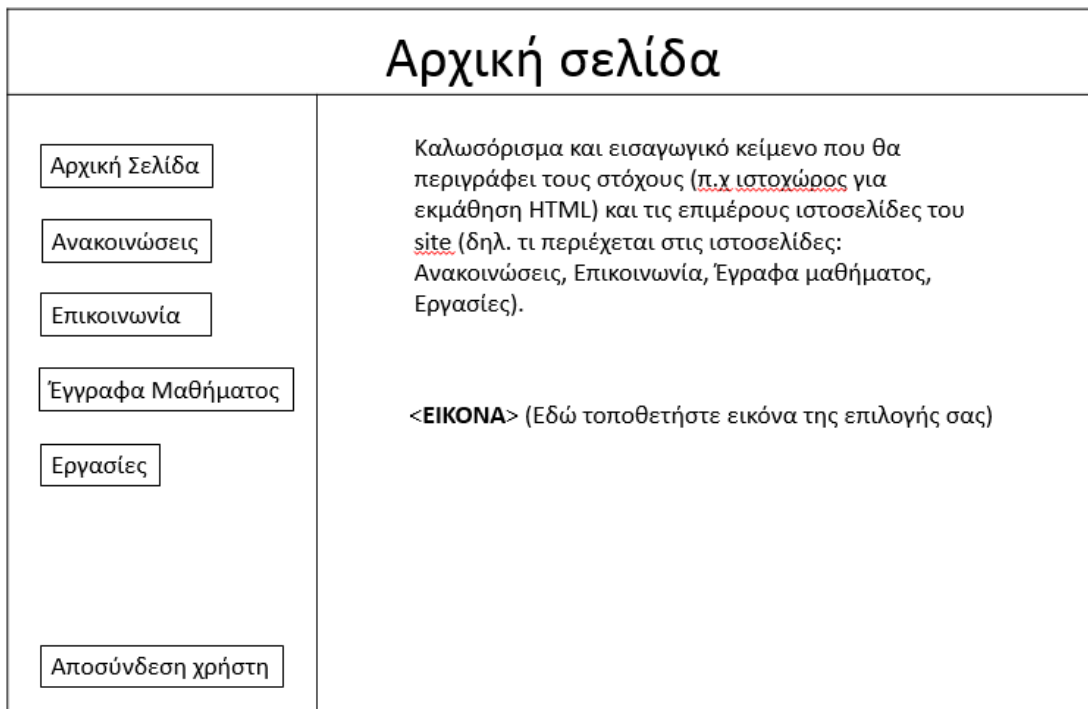
- Προσθήκη/επεξεργασία/διαγραφή χρήστη. Ο διαχειριστής θα πρέπει να μπορεί να έχει τον απόλυτο έλεγχο σχετικά με τους υπόλοιπους χρήστες. Θα έχει τη δυνατότητα να εισάγει νέους, να τους προσδίδει τα αντίστοιχα δικαιώματα (tutor/student), να διαγράφει όποιον χρήστη επιθυμεί και να τροποποιεί τους ήδη εγγεγραμμένους.
- Προσθήκη/επεξεργασία/διαγραφή ανακοινώσεων. Αντίστοιχα, ο διαχειριστής θα μπορεί να δημοσιεύει τις ανακοινώσεις που επιθυμεί, στην αντίστοιχη ιστοσελίδα του ιστοτόπου, να διαγράφει παλιότερες ή/και να τροποποιεί όποιες θέλει.
- Προσθήκη/επεξεργασία/διαγραφή εγγράφων. Οι ίδιες επιλογές πρέπει να υπάρχουν και για τα έγγραφα του μαθήματος τα οποία θα αναρτώνται στην ιστοσελίδα και θα είναι διαθέσιμα για λήψη από τους χρήστες.
- Προσθήκη/επεξεργασία/διαγραφή εργασιών. Και σε αυτό το κομμάτι θα πρέπει να υπάρχουν οι ίδιες δυνατότητες για τις εργασίες που θα αναρτώνται στον ιστότοπο και επιπλέον, για κάθε νέα εργασία, θα πρέπει να δημοσιεύεται και η αντίστοιχη ανακοίνωση, προκειμένου να ενημερώνονται οι μαθητές.

7.1.5. Γενική δομή της ιστοσελίδας

Η δομή του ιστοτόπου φαίνεται στο σχήμα που συναντήσατε παραπάνω. Τα βέλη που φαίνονται δείχνουν τη διασύνδεση μεταξύ των ιστοσελίδων γενικά. Εφόσον υπάρχουν δύο διαφορετικά ήδη χρηστών, είναι απαραίτητο να διατίθεται και διαφορετικό περιεχόμενο για κάθε κατηγορία χρήστη. Πιο συγκεκριμένα, οι χρήστες με ρόλο student θα μπορούν να βλέπουν το περιεχόμενο, χωρίς όμως να έχουν τις δυνατότητες και τις επιλογές που θα έχει ένας χρήστης με ρόλο tutor και οι οποίες περιγράφηκαν παραπάνω. Οι δυνατότητες προσθήκης/επεξεργασίας/διαγραφής από τη βάση δεδομένων παρέχονται μόνο για το ρόλο tutor και δε θα πρέπει να έχουν πρόσβαση σε αυτές οι students. Παρακάτω παρουσιάζεται ξεχωριστά, για κάθε κατηγορία χρήστη, το περιεχόμενο που θα προβάλλεται σε κάθε ιστοσελίδα.

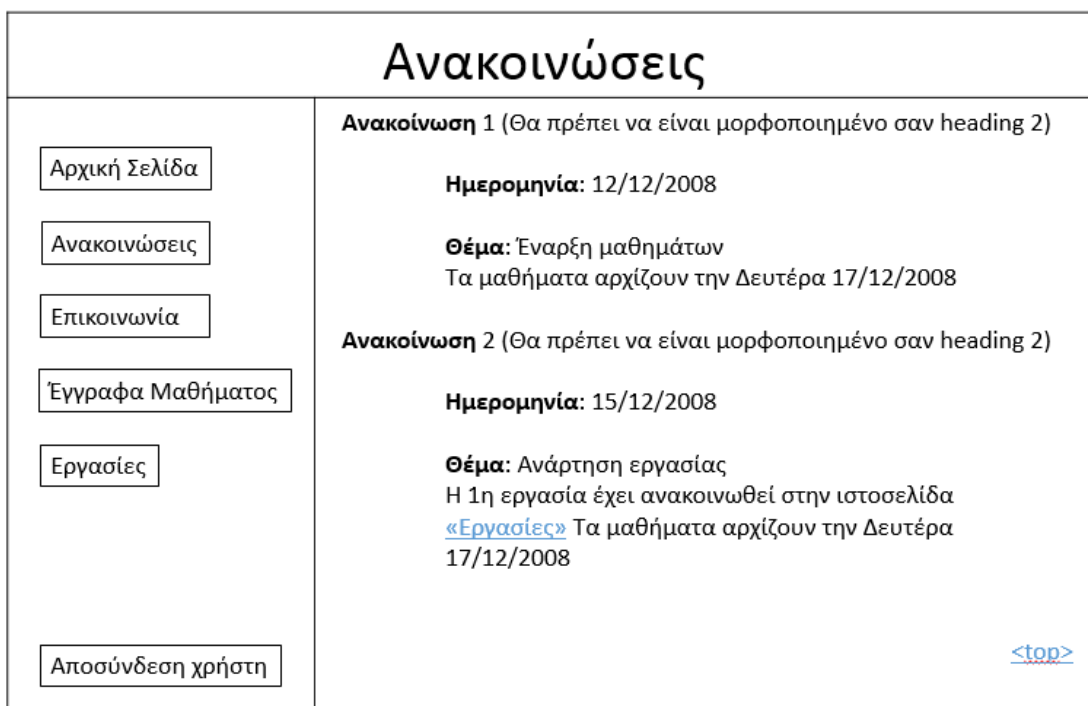
7.1.5.1 Student

index.html: Αποτελεί την αρχική ιστοσελίδα στην οποία θα εισέρχεται ο χρήστης.



Εικόνα 7.2. Η οθόνη της αρχικής ιστοσελίδας

announcements.html: Η συγκεκριμένη ιστοσελίδα θα περιέχει ανακοινώσεις σχετικές με το μάθημα. Στην περίπτωση που πρόκειται για ανακοίνωση εργασίας θα πρέπει να γίνεται σύνδεση με την ιστοσελίδα «homework» Μια μορφή της είναι η ακόλουθη:



Εικόνα 7.3. Η οθόνη της ιστοσελίδας με τις ανακοινώσεις

Η επιλογή της «<top>» οδηγεί στην κορυφή της συγκεκριμένης ιστοσελίδας

communication.html: Η συγκεκριμένη ιστοσελίδα θα παρέχει δύο δυνατότητες για την αποστολή e-mail στον καθηγητή:

- Μέσω web φόρμας.
 - Με χρήση e-mail διεύθυνσης.
- Μια μορφή της είναι η ακόλουθη:

<h2>Επικοινωνία</h2>	
<div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Αρχική Σελίδα</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Ανακοινώσεις</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Επικοινωνία</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Έγγραφα Μαθήματος</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Εργασίες</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Αποσύνδεση χρήστη</div>	<p>Η συγκεκριμένη ιστοσελίδα θα περιέχει δύο δυνατότητες για την αποστολή e-mail στον καθηγητή:</p> <ul style="list-style-type: none"> • Μέσω web φόρμας • Με χρήση e-mail διεύθυνσης <p>Αποστολή e-mail μέσω <u>web φόρμας</u> (Θα πρέπει να είναι μορφοποιημένο σαν <u>heading 2</u>)</p> <p>Αποστολέας: <πεδίο για την <u>δν/ση</u> του αποστολέα> Θέμα: <πεδίο για το θέμα> Κείμενο: <πεδίο για το κείμενο></p> <p>Αποστολή e-mail ε χρήση <u>e-mail διεύθυνσης</u> (Θα πρέπει να είναι μορφοποιημένο σαν <u>heading 2</u>)</p> <p>Εναλλακτικά μπορείτε να αποστείλετε e-mail στην παρακάτω διεύθυνση ηλεκτρονικού ταχυδρομείου tutor@csd.auth.test.gr (Εδώ πρόκειται για <u>υπερσύνδεσμο</u> με τον οποίο ανοίγει το πρόγραμμα αποστολής e-mail. Πχ. MS <u>outlook</u>)</p> <p style="text-align: right;"><top></p>

Εικόνα 7.4. Η οθόνη της ιστοσελίδας με τις δυνατότητες επικοινωνίας

documents.html: Η συγκεκριμένη ιστοσελίδα θα περιέχει έγγραφα σχετικά με το μάθημα τα οποία μπορούν να κατεβάσουν (download) οι μαθητές. Μια μορφή της είναι η ακόλουθη:

<h2>Έγγραφα μαθήματος</h2>	
<div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Αρχική Σελίδα</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Ανακοινώσεις</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Επικοινωνία</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Έγγραφα Μαθήματος</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Εργασίες</div> <div style="border: 1px solid black; padding: 2px; margin-bottom: 5px;">Αποσύνδεση χρήστη</div>	<p>Τίτλος εγγράφου 1 (Θα πρέπει να είναι μορφοποιημένο σαν <u>heading 2</u>)</p> <p style="text-align: center;"><i>Περιγραφή:</i> <Περιγραφή του περιεχομένου></p> <p style="text-align: center;">Download (<u>Υπερσύνδεσμος</u> σε ένα αρχείο file1.doc)</p> <p>Τίτλος εγγράφου 2 (Θα πρέπει να είναι μορφοποιημένο σαν <u>heading 2</u>)</p> <p style="text-align: center;"><i>Περιγραφή:</i> <Περιγραφή του περιεχομένου></p> <p style="text-align: center;">Download (<u>Υπερσύνδεσμος</u> σε ένα αρχείο file2.doc)</p> <p style="text-align: right;"><top></p>

Εικόνα 7.5. Η οθόνη της ιστοσελίδας με τα έγγραφα του μαθήματος

Η επιλογή «<top>» οδηγεί στην κορυφή της συγκεκριμένης ιστοσελίδας

homework.html: Η συγκεκριμένη ιστοσελίδα θα περιέχει εκφωνήσεις εργασιών που μπορούν να κα-
ταβιβιάσουν (download) οι μαθητές. Μια μορφή της είναι η ακόλουθη:

The screenshot shows a web page with a title 'Εργασίες' (Homework) at the top. On the left is a sidebar with five buttons: 'Αρχική Σελίδα', 'Ανακοινώσεις', 'Επικοινωνία', 'Έγγραφα Μαθήματος', and 'Εργασίες'. At the bottom of the sidebar is a button 'Αποσύνδεση χρήστη'. The main content area contains two sections for assignments, 'Εργασία 1' and 'Εργασία 2'. Each section includes a heading, a list of objectives, a list of deliverables, and a submission date. A '<top>' link is located at the bottom right of the page.

Εργασίες	
Αρχική Σελίδα Ανακοινώσεις Επικοινωνία Έγγραφα Μαθήματος Εργασίες Αποσύνδεση χρήστη	Εργασία 1 (Θα πρέπει να είναι μορφοποιημένο σαν <u>heading 2</u>) Στόχοι: Οι στόχοι τις εργασίας είναι <ol style="list-style-type: none">1. <στόχος 1>2. <στόχος 2>3. ... Εκφώνηση: Κατεβάστε την εκφώνηση της εργασίας από εδώ (Υπερσύνδεσμος σε ένα αρχείο ergasia1.doc) Παραδοτέα: <ol style="list-style-type: none">1. Γραπτή αναφορά σε word2. Παρουσίαση σε powerpoint Ημερομηνία παράδοσης: 12/5/2009
	Εργασία 2 (Θα πρέπει να είναι μορφοποιημένο σαν <u>heading 2</u>) Στόχοι: Οι στόχοι τις εργασίας είναι <ol style="list-style-type: none">1. <στόχος 1>2. <στόχος 2>3. ... Εκφώνηση: Κατεβάστε την εκφώνηση της εργασίας από εδώ (Υπερσύνδεσμος σε ένα αρχείο ergasia2.doc) Παραδοτέα: <ol style="list-style-type: none">1. Γραπτή αναφορά σε word2. Παρουσίαση σε powerpoint Ημερομηνία παράδοσης: 15/5/2009
	<top>

Εικόνα 7.6. Η οθόνη της ιστοσελίδας με τις εργασίες του μαθήματος

Η επιλογή «<top>» οδηγεί στην κορυφή της συγκεκριμένης ιστοσελίδας

login.php: Η ιστοσελίδα πιστοποίησης θα είναι όπως παρακάτω:

Είσοδος χρήστη

Εισάγετε τα στοιχεία σας στην παρακάτω
φόρμα για να συνδεθείτε:

Όνομα χρήστη:

Password:

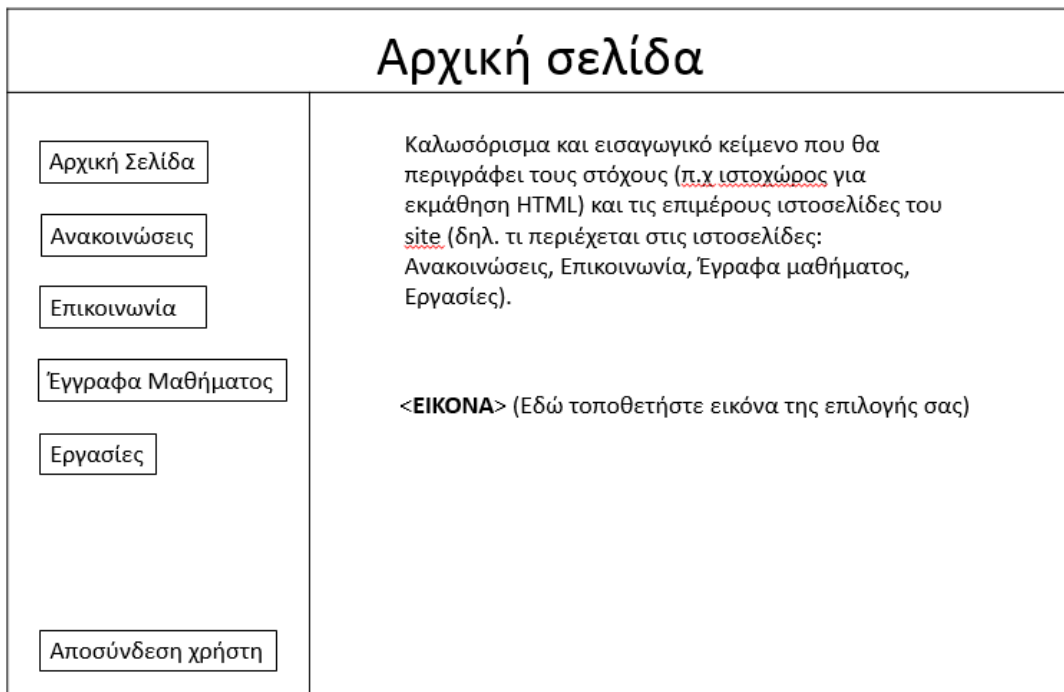
Εικόνα 7.7. Η οθόνη εισόδου χρηστών (κοινή για όλους τους χρήστες)

7.1.5.2 Tutor

Ο χρήστης που έχει αναλάβει το ρόλο tutor θα προσπελαύνει πληροφορία αντίστοιχη με αυτή που βλέπει και ο student και επιπλέον θα έχει στη διάθεσή του όλες τις επιλογές και τις δυνατότητες διαχείρισης της βάσης δεδομένων.

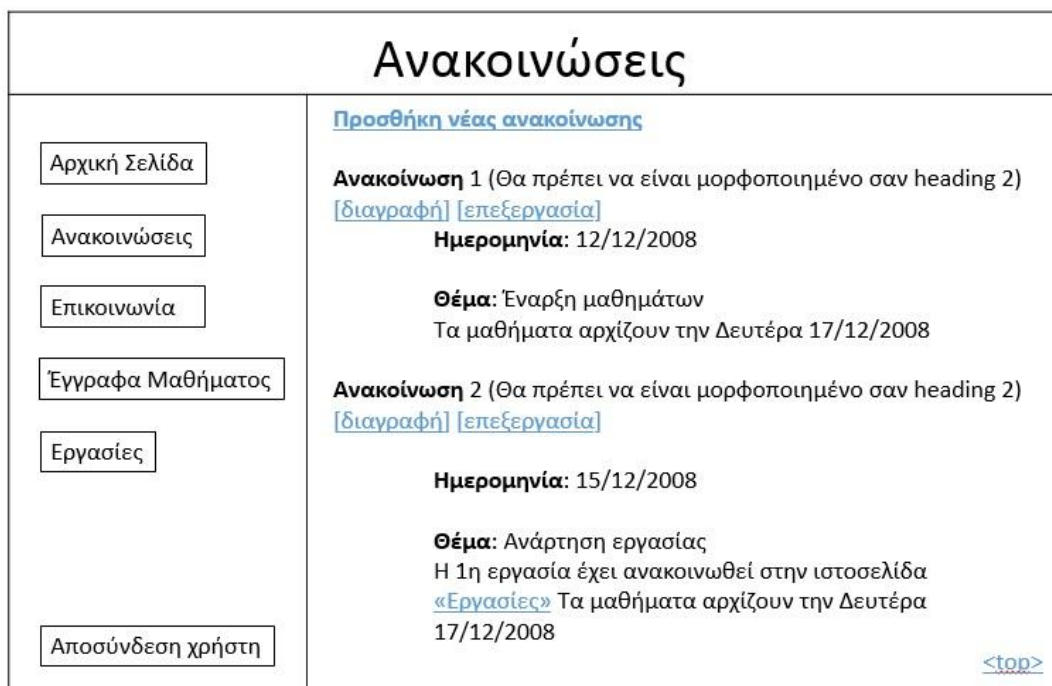
Κάθε ιστοσελίδα θα έχει την ακόλουθη μορφή που θα πρέπει να υλοποιηθεί με χρήση <div>:

Κεντρική_σελίδα (index.php): Αποτελεί την αρχική ιστοσελίδα στην οποία θα εισέρχεται ο χρήστης μετά την πιστοποίησή του.



Εικόνα 7.8. Η οθόνη της αρχικής σελίδας (όμοια με χρήση κατηγορίας «student»)

Ανακοινώσεις (announcements.php): Η συγκεκριμένη ιστοσελίδα θα περιέχει ανακοινώσεις σχετικές με το μάθημα. Στην περίπτωση που πρόκειται για ανακοίνωση εργασίας θα πρέπει να γίνεται σύνδεση με την ιστοσελίδα «Εργασίες». Μια μορφή της για χρήστες που έχουν αναλάβει ρόλο tutor είναι η ακόλουθη:



Εικόνα 7.9. Η οθόνη των ανακοινώσεων

Η επιλογή «**Προσθήκη νέας ανακοίνωσης**» θα οδηγεί τον Καθηγητή σε αντίστοιχη φόρμα για την προσθήκη νέας ανακοίνωσης. Τα δεδομένα που θα συμπληρώνονται στη φόρμα θα αποθηκεύονται στη βάση δεδομένων. Η επιλογή «**[διαγραφή]**» θα διαγράφει τη συγκεκριμένη ανακοίνωση από τη βάση δεδομένων. Η

επιλογή «[επεξεργασία]» θα οδηγεί τον tutor σε αντίστοιχη φόρμα για τη διόρθωση των στοιχείων μιας υπάρχουσας ανακοίνωσης. Τα δεδομένα που θα συμπληρώνονται στη φόρμα θα αποθηκεύονται στη βάση δεδομένων.

Επικοινωνία (communication.php): Η συγκεκριμένη ιστοσελίδα θα παρέχει δύο δυνατότητες για την αποστολή e-mail στον καθηγητή:

- Μέσω web φόρμας.
- Με χρήση e-mail διεύθυνσης.

Μια μορφή της είναι η ακόλουθη:

Επικοινωνία	
<p>Αρχική Σελίδα</p> <p>Ανακοινώσεις</p> <p>Επικοινωνία</p> <p>Έγγραφα Μαθήματος</p> <p>Εργασίες</p> <p>Αποσύνδεση χρήστη</p>	<p>Η συγκεκριμένη ιστοσελίδα θα περιέχει δύο δυνατότητες για την αποστολή e-mail στον καθηγητή:</p> <ul style="list-style-type: none">• Μέσω web φόρμας• Με χρήση e-mail διεύθυνσης <p>Αποστολή e-mail μέσω web φόρμας (Θα πρέπει να είναι μορφοποιημένο σαν heading 2)</p> <p>Αποστολέας: <πεδίο για την δν/ση του αποστολέα></p> <p>Θέμα: <πεδίο για το θέμα></p> <p>Κείμενο: <πεδίο για το κείμενο></p> <p>Αποστολή e-mail ε χρήση e-mail διεύθυνσης (Θα πρέπει να είναι μορφοποιημένο σαν heading 2)</p> <p>Εναλλακτικά μπορείτε να αποστείλετε e-mail στην παρακάτω διεύθυνση ηλεκτρονικού ταχυδρομείου tutor@csd.auth.test.gr (Εδώ πρόκειται για υπερσύνδεσμο με τον οποίο ανοίγει το πρόγραμμα αποστολής e-mail. Πχ. MS outlook)</p> <p style="text-align: right;"><top></p>

Εικόνα 7.10. Η οθόνη της ιστοσελίδας επικοινωνίας (όμοια με χρήση κατηγορίας «student»)

Στη συγκεκριμένη περίπτωση η φόρμα θα λειτουργεί σε συνδυασμό με PHP script και θα αποστέλλει e-mail σε όλους τους χρήστες που είναι πιστοποιημένοι σαν tutors.

Έγγραφα (documents.php): Η συγκεκριμένη ιστοσελίδα θα περιέχει έγγραφα σχετικά με το μάθημα τα οποία μπορούν να κατεβάσουν (download) οι φοιτητές. Επιπλέον, οι καθηγητές μπορούν να ανεβάσουν αρχεία.

Έγγραφα μαθήματος	
<p>Αρχική Σελίδα</p> <p>Ανακοινώσεις</p> <p>Επικοινωνία</p> <p>Έγγραφα Μαθήματος</p> <p>Εργασίες</p> <p>Αποσύνδεση χρήστη</p>	<p>Προσθήκη νέου εγγράφου</p> <p>Τίτλος εγγράφου 1 (Θα πρέπει να είναι μορφοποιημένο σαν <u>heading 2</u>)</p> <p>Περιγραφή: <Περιγραφή του περιεχομένου></p> <p>Download (Υπερσύνδεσμος σε ένα αρχείο file1.doc)</p> <p>Τίτλος εγγράφου 2 (Θα πρέπει να είναι μορφοποιημένο σαν <u>heading 2</u>)</p> <p>Περιγραφή: <Περιγραφή του περιεχομένου></p> <p>Download (Υπερσύνδεσμος σε ένα αρχείο file2.doc)</p> <p><top></p>

Εικόνα 7.11. Η οθόνη της ιστοσελίδας με τα έγγραφα μαθήματος

Η επιλογή «**Προσθήκη νέου εγγράφου**» θα οδηγεί τον καθηγητή σε αντίστοιχη φόρμα για την προσθήκη νέου εγγράφου. Τα δεδομένα που θα συμπληρώνονται στη φόρμα θα αποθηκεύονται στη βάση δεδομένων.

Εργασίες: Η συγκεκριμένη ιστοσελίδα θα περιέχει εκφωνήσεις εργασιών που μπορούν να κατεβάσουν (download) οι μαθητές. Επιπλέον, οι καθηγητές μπορούν να ανεβάσουν εκφωνήσεις.

Εργασίες

Αρχική Σελίδα

Ανακοινώσεις

Επικοινωνία

Έγγραφα Μαθήματος

Εργασίες

Αποσύνδεση χρήστη

Προσθήκη νέας εργασίας

Εργασία 1 (Θα πρέπει να είναι μορφοποιημένο σαν heading 2)
 Στόχοι: Οι στόχοι τις εργασίας είναι
 1. <στόχος 1>
 2. <στόχος 2>
 3. ...
 Εκφώνηση: Κατεβάστε την εκφώνηση της εργασίας από [εδώ](#) (Υπερσύνδεσμος σε ένα αρχείο ergasia1.doc)
 Παραδοτέα:
 1. Γραπτή αναφορά σε word
 2. Παρουσίαση σε powerpoint
 Ημερομηνία παράδοσης: 12/5/2009

Εργασία 2 (Θα πρέπει να είναι μορφοποιημένο σαν heading 2)
 Στόχοι: Οι στόχοι τις εργασίας είναι
 1. <στόχος 1>
 2. <στόχος 2>
 3. ...
 Εκφώνηση: Κατεβάστε την εκφώνηση της εργασίας από [εδώ](#) (Υπερσύνδεσμος σε ένα αρχείο ergasia2.doc)
 Παραδοτέα:
 1. Γραπτή αναφορά σε word
 2. Παρουσίαση σε powerpoint
 Ημερομηνία παράδοσης: 15/5/2009

[<top>](#)

Εικόνα 7.12. Η οθόνη της ιστοσελίδας με τις εργασίες του μαθήματος

Η επιλογή «**Προσθήκη νέας εργασίας**» θα οδηγεί τον Καθηγητή σε αντίστοιχη φόρμα για την προσθήκη νέου εγγράφου. Τα δεδομένα που θα συμπληρώνονται στη φόρμα θα αποθηκεύονται στη βάση δεδομένων. Επιπλέον, όταν ανακοινώνεται μια νέα εργασία θα πρέπει να δημιουργείται και νέα εγγραφή στον αντίστοιχο πίνακα της βάσης δεδομένων όπου αποθηκεύονται οι ανακοινώσεις. Στα παρακάτω πεδία θα αποθηκεύονται οι παρακάτω αντίστοιχες τιμές:

- Ημερομηνία: η ημερομηνία ανάρτησης της εργασίας.
- Θέμα: Υποβλήθηκε η εργασία <αύξων αριθμός εργασίας>.
- Κυρίως κείμενο: Η ημερομηνία παράδοσης της εργασίας είναι <ημερομηνία παράδοσης εργασίας>.

7.2. Εισαγωγή σε Διαδικτυακές Τεχνολογίες

7.2.1. Εισαγωγή στην HTML

Στην κεφάλαιο αυτό θα παρουσιαστούν οι βασικές αρχές για τη σχεδίαση μιας ιστοσελίδας με τη χρήση της γλώσσας HTML. Η σχεδίαση θα ξεκινήσει με την παρουσίαση της διαδικασίας με την οποία μπορούμε να εμφανίσουμε απλό κείμενο σε έναν περιηγητή ιστοσελίδων. Στη συνέχεια θα παρουσιαστούν τρόποι με τους οποίους μπορεί να μορφοποιηθεί το κείμενο, τρόποι για τη βέλτιστη διάταξη των στοιχείων που περιλαμβάνει μια ιστοσελίδα αλλά επίσης και πως είναι δυνατή η εισαγωγή πολυμεσικών στοιχείων όπως εικόνες. Επιπλέον, θα παρουσιαστεί η διαδικασία με την οποία μπορούμε να δημιουργήσουμε φόρμες αποστολής δεδομένων με τη χρήση HTML και θα γίνει εισαγωγή σε βασικές έννοιες της HTML5 παρουσιάζοντας κάποιες βασικές οντότητές της.

7.2.1.1 Τι είναι η HTML;

Η **HTML** (**H**yper**T**ext **M**arkup **L**anguage) ή Γλώσσα Σήμανσης Υπερκειμένου, αποτελεί την κύρια γλώσσα σήμανσης για τη δημιουργία ιστοσελίδων. Δεν αποτελεί γλώσσα προγραμματισμού αλλά γλώσσα αναπαράστασης κειμένου και εικόνων στον περιηγητή ιστοσελίδων. Αρχικά, αναπτύχθηκε ως τρόπος περιγραφής και διαμοίρασης επιστημονικών κειμένων. Σήμερα, χρησιμοποιείται τόσο για τη σήμανση κειμένων διάφορων τύπων όσο και για τη δημιουργία διεπαφών.

Η HTML βασίστηκε στην *Standard General Markup Language* (SGML), η οποία χρησιμεύει για τον καθορισμό γλωσσών σήμανσης (*markup languages*) για έγγραφα. Ουσιαστικά, η SGML μπορεί να περιγραφεί ως ένα σύνολο κανόνων για τη δημιουργία γλωσσών σήμανσης. Η HTML, λοιπόν, ξεκίνησε αρχικά ως εφαρμογή της SGML, στις αρχές της δεκαετίας του 1990, και αποτελεί έως σήμερα τυποποιημένο τρόπο για την περιγραφή της δομής εγγράφων υπερκειμένου. Το πρότυπο SGML αξιοποιεί τη γενικευμένη σήμανση βασισμένη σε στοιχεία τα οποία έχουν διάφορες ιδιότητες και μπορεί να είναι εμφωλευμένα το ένα μέσα στο άλλο. Επιπλέον, το πρότυπο SGML διαχωρίζει τη δομή του εγγράφου από το περιεχόμενό του, κατεύθυνση στην οποία κινήθηκε και η HTML.

Χαρακτηριστικό γνώρισμα της HTML είναι ότι για την περιγραφή της δομής και της εμφάνισης των ιστοσελίδων, καθώς και κάποιων ειδικών λειτουργιών, χρησιμοποιούνται *ετικέτες* (tags), οι οποίες έχουν κάποιες ιδιότητες.

Για παράδειγμα, η ετικέτα <p> συμβολίζει την αλλαγή παραγράφου.

Παρατηρούμε ότι οι ετικέτες περικλείονται από τα σύμβολα «μεγαλύτερο από» («>») και «μικρότερο από» («<»). Συνήθως, λειτουργούν ανά ζεύγη (για παράδειγμα <form> και </form>), με την πρώτη να ονομάζεται *ετικέτα έναρξης* (start tag) και τη δεύτερη *ετικέτα λήξης* (end tag). Το περιεχόμενο της ιστοσελίδας τοποθετείται ανάμεσα στις ετικέτες. Οι διάφορες ετικέτες HTML θα αναλυθούν περισσότερο στη συνέχεια.

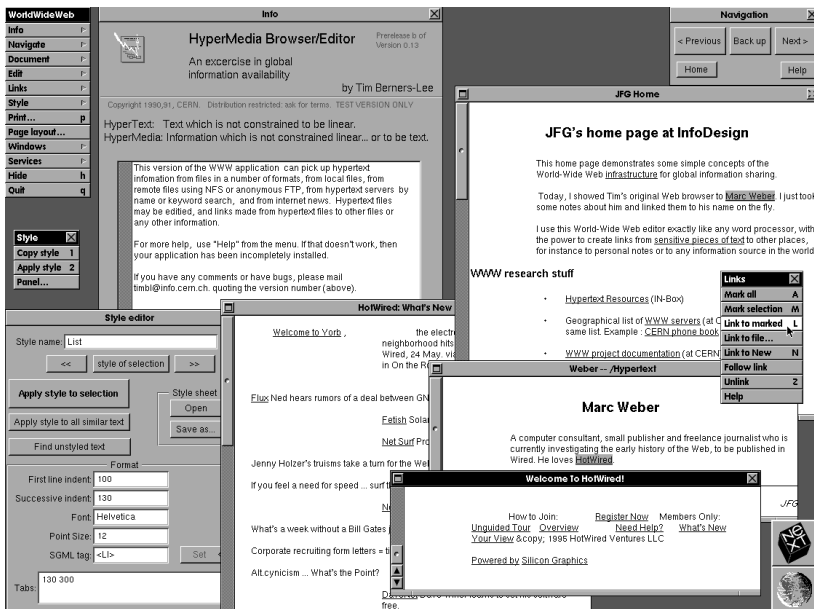
Το πρόγραμμα περιήγησης ιστοσελίδων (Web browser) δεν εμφανίζει τις ετικέτες HTML στον τελικό χρήστη. Αντίθετα, διαβάζει τα έγγραφα HTML και χρησιμοποιεί τις ετικέτες HTML ώστε να ερμηνεύσει το περιεχόμενό τους σε ιστοσελίδες που μπορεί να δει και να διαβάσει κάποιος. Η HTML δίνει στο σχεδιαστή της ιστοσελίδας τη δυνατότητα να ενσωματώσει σε αυτή εικόνες, υπερσυνδέσμους, βίντεο, καθώς και scripts (σύνολα εντολών) τα οποία επηρεάζουν τη λειτουργία της ιστοσελίδας. Η περιγραφή μόνο της δομής του εγγράφου, καθώς και η αποσύνδεσή του από το πώς θα εμφανίζεται στον τελικό χρήστη, καθιστά τη διαμόρφωση του εγγράφου πολύ πιο εύκολη. Τα προγράμματα περιήγησης ιστοσελίδων μπορούν επίσης να αναφέρονται σε στυλ μορφοποίησης (CSS) για τον ορισμό της εμφάνισης και της δομής του εγγράφου.

Στο σημείο αυτό αξίζει να αναφερθούμε σε μία άλλη σημαντική γλώσσα σήμανσης, την XML. Η XML είναι «μετα-γλώσσα» (meta-language), δηλαδή γλώσσα που χρησιμοποιείται για να καθορίσει νέες γλώσσες σήμανσης. Έτσι, η XML μπορεί να χρησιμοποιηθεί για να συμπληρώσει την HTML: ενώ η HTML χρησιμοποιείται για τη διατύπωση και την εμφάνιση των στοιχείων της ιστοσελίδας, η XML προσδιορίζει τη συναφή έννοια των δεδομένων. Στην XML υπάρχει η δυνατότητα να καθορίσει ο χρήστης τις ετικέτες και τις μεταξύ τους σχέσεις, ενώ στην HTML οι ετικέτες είναι προκαθορισμένες. Δεν θα επεκταθούμε άλλο στην περιγραφή της XML, καθώς αυτό είναι έξω από τη θεματολογία του παρόντος βιβλίου. Περισσότερες πληροφορίες για την XML είναι διαθέσιμες στην ιστοσελίδα <http://www.w3.org/XML/>.

7.2.1.2 HTML: Ιστορικά Στοιχεία

Η ιστορία της HTML ξεκίνησε το 1980, όταν ο φυσικός του Παγκοσμίου Επιστημονικού Οργανισμού CERN Tim Berners-Lee επινόησε και κατασκεύασε ένα σύστημα διαμοιρασμού εγγράφων για τους φυσικούς του CERN. Το 1989 πρότεινε τη δημιουργία ενός συστήματος που θα χρησιμοποιούσε υπερκείμενο και θα βασιζόταν στο Διαδίκτυο, και στη συνέχεια δημιούργησε την ίδια την προδιαγραφή της HTML, τον περιηγητή ιστοσελίδων, καθώς και το λογισμικό του εξυπηρετητή ιστοσελίδων.

Στην Εικόνα 7.13 βλέπουμε ένα στιγμιότυπο από τον περιηγητή ιστοσελίδων που δημιούργησε ο Tim Berners-Lee το 1991-92. Αυτός ο περιηγητής ιστοσελίδων υλοποιήθηκε στη γλώσσα προγραμματισμού Objective-C, έτρεχε σε ένα NeXt σταθμό εργασίας, και διευκόλυνε τη δημιουργία και την επεξεργασία web εγγράφων.



Εικόνα 7.13. Ο περιηγητής του Tim Berners-Lee (Πηγή: <http://www.w3.org/>)

Η πρώτη δημόσια διαθέσιμη περιγραφή της HTML (τέλη του 1991) ήταν ένα έγγραφο με όνομα «Ετικέτες HTML», το οποίο περιέγραφε τα 20 εκείνα στοιχεία τα οποία αποτέλεσαν τον αρχικό σχεδιασμό της HTML. Στη συνέχεια, το 1993, το Internet Engineering Task Force (IETF) προχώρησε σε δημοσίευση της πρώτης πρότασης για μια προδιαγραφή της HTML. Το 1994, το IETF δημιούργησε μια ομάδα εργασίας για την HTML η οποία ολοκλήρωσε την πρώτη προδιαγραφή της HTML, την «HTML 2.0», το 1995.

Έκτοτε, υπεύθυνο για τις προδιαγραφές της HTML είναι το World Wide Web Consortium (W3C), το οποίο ιδρύθηκε και διοικείται από τον Tim Berners-Lee. Αποτελείται από οργανισμούς-μέλη, οι οποίοι έχουν εργαζομένους πλήρους απασχόλησης, με στόχο την από κοινού ανάπτυξη προδιαγραφών για τον Παγκόσμιο Ιστό. Από το Μάιο του 2004 έχει 385 μέλη. Η HTML προτυποποιήθηκε παγκόσμια το 2000 (ISO/IEC15445:2000). Η προδιαγραφή 4.01 της HTML δημοσιεύτηκε από το W3C το 1999, ενώ το 2001 δημοσιεύτηκαν αλλαγές που κάλυπταν παραλείψεις της. Το 2004 άρχισε η εξέλιξη της HTML5, η οποία ολοκληρώθηκε και προτυποποιήθηκε στις 28 Οκτωβρίου 2014 και δημοσιεύτηκε ως W3C σύσταση (recommendation).

7.2.1.3 Βασικές Έννοιες & Στοιχεία HTML

Η σήμανση (markup) της HTML αποτελείται από μερικά βασικά συστατικά, τα οποία είναι: η δήλωση τύπου εγγράφου, οι (βασισμένοι σε χαρακτήρες) τύποι δεδομένων, οι αναφορές χαρακτήρων, οι αναφορές οντοτήτων, καθώς και τα στοιχεία (attributes) και οι ιδιότητές τους.

Τα στοιχεία HTML αποτελούνται, στην πιο γενική τους μορφή, από τρία συστατικά: ένα ζεύγος από ετικέτες, οι οποίες ονομάζονται *ετικέτα έναρξης* (start tag) και *ετικέτα λήξης* (end tag), μερικές *ιδιότητες* μέσα στην ετικέτα έναρξης, και το κείμενο ή το γραφικό *περιεχόμενο* μεταξύ των ετικετών, το οποίο μπορεί να εμφολεύει μέσα του και άλλα στοιχεία. Το στοιχείο HTML μπορεί να είναι οτιδήποτε ανάμεσα στις ετικέτες έναρξης και λήξης. Κάθε ετικέτα περικλείεται σε σύμβολα «μεγαλύτερο από» (>) και «μικρότερο από» (<). Οι ιδιότητες του στοιχείου παρέχουν επιπρόσθετη πληροφορία γι' αυτό το στοιχείο και βρίσκονται πάντα στην ετικέτα έναρξης. Το όνομα κάθε στοιχείου HTML είναι το όνομα που χρησιμοποιείται στις αντίστοιχες ετικέτες. Το όνομα της ετικέτας λήξης ξεκινά με μια κάθετο «/», η οποία παραλείπεται στα άδεια στοιχεία (στοιχεία HTML που δεν έχουν καθόλου περιεχόμενο).

Επομένως, η γενική μορφή ενός στοιχείου HTML είναι: <ετικέτα ιδιότητα1="τιμή1" ιδιότητα2="τιμή2">Περιεχόμενο</ετικέτα>. Παραδείγματος χάρη, για τη δήλωση μιας παραγράφου σε HTML, θα μπορούσαμε να γραφεί ο εξής κώδικας:

```
<p>Μια παράγραφος σε HTML. </p>
```

Στη συνέχεια, θα αναλυθεί βήμα προς βήμα ένα πιο σύνθετο παράδειγμα σε HTML. Να σημειωθεί, σε αυτό το σημείο, ότι η HTML μπορεί να γραφεί με χρήση κάποιου HTML editor, όπως είναι ο Adobe Dreamweaver ή ο CoffeeCup HTML Editor. Εναλλακτικά, μπορεί να χρησιμοποιηθεί ένας απλός text editor, όπως είναι το Notepad σε περιβάλλον Windows ή το TextEdit σε Mac. Αυτό γίνεται με τη δημιουργία ενός νέου αρχείου, το οποίο πρέπει να αποθηκευθεί με κατάληξη .htm ή .html. Η προτεινόμενη κωδικοποίηση είναι η UTF-8, καθώς η κωδικοποίηση ANSI καλύπτει μόνο US και Western European χαρακτήρες. Για να δείτε πώς θα εμφανίζεται η ιστοσελίδα, αρκεί να την ανοίξετε στον περιηγητή ιστοσελίδων, σαν οποιαδήποτε άλλη ιστοσελίδα. Ο κώδικας του παραδείγματος είναι ο ακόλουθος:

```
<!DOCTYPE html>
<html lang="en-US">
  <head>
    <title>Hello HTML</title>
  </head>
  <body>
    <p>Hello world</p>
  </body>
</html>
```

Οι ετικέτες της HTML δεν είναι case sensitive (δηλαδή δεν οι κεφαλαίοι και πεζοί χαρακτήρες δεν έχουν διαφορά). Για παράδειγμα, το <P> είναι το ίδιο με το <p>. Αν και η προδιαγραφή της HTML5 δεν απαιτεί τη χρήση ετικετών με πεζούς χαρακτήρες, το W3C προτείνει τη χρήση πεζών στην HTML4.

Ας δούμε τώρα αναλυτικότερα τα στοιχεία HTML και τις ιδιότητές τους. Το ακόλουθο παράδειγμα ξεκινάει με τη δήλωση τύπου εγγράφου (DOCTYPE - Document Type Declaration), το οποίο στον παραπάνω κώδικα είναι για την HTML5. Στη συνέχεια, ακολουθεί το στοιχείο <html>, το οποίο ορίζει ολόκληρο το έγγραφο. Με αυτήν την ετικέτα, ο περιηγητής «καταλαβαίνει» ότι ξεκινάει ένα HTML έγγραφο. Η γλώσσα του εγγράφου συνίσταται να δηλώνεται στην ιδιότητα «lang» του στοιχείου <html>. Τα πρώτα δύο γράμματα προσδιορίζουν τη γλώσσα (εδώ «en»), ενώ τα επόμενα δύο τη διάλεκτο («US»). Η δήλωση της γλώσσας είναι σημαντική για λόγους εφαρμογές προσβασιμότητας (screen readers), καθώς και μηχανές αναζήτησης.

Το περιεχόμενο του στοιχείου <html> περιέχει άλλα στοιχεία HTML. Αρχικά, το στοιχείο <head> είναι «container» για HTML για μεταδεδομένα (metadata), τα οποία αποτελούν δεδομένα σχετικά με το HTML έγγραφο. Τα metadata δεν εμφανίζονται στην ιστοσελίδα. Ορίζουν τον τίτλο του εγγράφου, τα στυλ (που θα δούμε στη συνέχεια), υπερσυνδέσμους, scripts, κ.α. Οι αντίστοιχες ετικέτες που περιγράφουν metadata είναι οι ακόλουθες: <title>, <style>, <meta>, <link>, <script>, and <base>. Στο συγκεκριμένο παράδειγμα χρησιμοποιήθηκε επίσης η ετικέτα <title> για να δηλωθεί ο τίτλος «Hello HTML». Το στοιχείο αυτό είναι απαραίτητο για κάθε HTML έγγραφο, καθώς: (α) ορίζει έναν τίτλο για την ιστοσελίδα στον περιηγητή και (β) εμφανίζει τον τίτλο της ιστοσελίδας στα αποτελέσματα των μηχανών αναζήτησης.

Στη συνέχεια, χρησιμοποιείται το στοιχείο <body> το οποίο ορίζει το κυρίως «σώμα» του εγγράφου. Έχει ετικέτα έναρξης <body> και ετικέτα λήξης </body> Σε αυτό περιέχεται ό,τι θέλουμε να εμφανίζεται στην ιστοσελίδα. Μέσα σε αυτό υπάρχει το στοιχείο <p> το οποίο ορίζει μια παράγραφο. Το περιεχόμενο της παραγράφου είναι «Hello world». Να σημειωθεί εδώ ότι κάποιες φορές η ετικέτα λήξης μπορεί να παραλειφθεί, χωρίς πρόβλημα, όπως στην περίπτωση του στοιχείου <p>. Παρόλα αυτά, δε συνίσταται, καθώς μπορεί να προκύψουν απρόβλεπτα αποτελέσματα ή και διάφορα λάθη. Προτείνεται να κλείνετε τις ετικέτες μόλις τις ανοίξετε και μετά να συμπληρώνετε το περιεχόμενό τους, ώστε να μην γίνει κάποιο λάθος.

Επιπλέον, όπως έχει ήδη αναφερθεί, υπάρχουν στοιχεία HTML τα οποία δεν έχουν περιεχόμενο και ονομάζονται κενά στοιχεία (empty elements). Για παράδειγμα, η αλλαγή γραμμής γίνεται με χρήση του στοιχείου
 το οποίο είναι ένα κενό στοιχείο, χωρίς ετικέτα λήξης. Στη συγκεκριμένη περίπτωση μπορεί να χρησιμοποιηθεί η ετικέτα έναρξης «
», η οποία «κλείνει» κιόλας το στοιχείο HTML. Σε κάθε στοιχείο HTML μπορούν να εφαρμοστούν global HTML attributes (καθολικές HTML ιδιότητες) που δίνουν νόημα στα στοιχεία. Παραδείγματα attributes είναι τα εξής: id (προσδιορίζει ένα μοναδικό κωδικό για ένα στοιχείο), lang (προσδιορίζει τη γλώσσα του περιεχομένου ενός στοιχείου), και draggable (προσδιορίζει εάν ένα στοιχείο θα είναι μεταφέρσιμο ή όχι).

Τέλος, να σημειωθεί εδώ ότι τα ζεύγη ετικετών `<html>...</html>`, `<head>...</head>` και `<body>...</body>` είναι αναγκαία για τη δόμηση κάθε ιστοσελίδας.

Στον ακόλουθο πίνακα (Πίνακας 7.2) παρατίθενται μερικές βασικές ετικέτες HTML με την περιγραφή τους.

Ετικέτα	Περιγραφή
<code><head></code>	Ορίζει πληροφορίες σχετικά με το έγγραφο.
<code><title></code>	Ορίζει τον τίτλο του εγγράφου.
<code><base></code>	Ορίζει μια προεπιλεγμένη διεύθυνση για όλους τους υπερσυνδέσμους της ιστοσελίδας.
<code><link></code>	Ορίζει τη σχέση μεταξύ ενός εγγράφου και μιας εξωτερικής πηγής.
<code><meta></code>	Ορίζει μεταδεδομένα σχετικά με το HTML έγγραφο.
<code><script></code>	Ορίζει client-side script.
<code><style></code>	Ορίζει πληροφορίες σχετικά με το στυλ του εγγράφου.
<code><a></code>	Ορίζει anchor ή υπερσύνδεσμο.
<code></code>	Για bold γραμματοσειρά.
<code><div></code>	Ορίζει ενότητα σε έγγραφο.

Πίνακας 7.2 **Βασικές ετικέτες HTML**

7.2.1.4 Σχόλια στην HTML

Το ζεύγος ετικετών `<!-- ... -->` χρησιμοποιείται στην HTML για να ορίσει σχόλια στον πηγαίο κώδικα. Παραδείγματος χάρη,

```
<!-- This is a comment.-->
```

Τα σχόλια δεν εμφανίζονται από τον περιηγητή ιστοσελίδων. Είναι χρήσιμα για την επεξήγηση του κώδικα, ώστε να είναι ευκολότερο να τον αλλάξουμε κάποια άλλη στιγμή (που πιθανά δε θα θυμόμαστε πλέον τι έχουμε γράψει, ιδίως αν έχει περάσει καιρός), ή κάποιος άλλος που τον βλέπει για πρώτη φορά. Η ετικέτα του σχολίου δεν υποστηρίζει κάποια ιδιότητα HTML.

7.2.1.5 Υπερσύνδεσμοι στην HTML

Ως υπερσύνδεσμοι (hyperlinks) ορίζονται λέξεις, προτάσεις ή εικόνες σε μια ιστοσελίδα, πατώντας στα οποία ο χρήστης μεταφέρεται σε κάποιες άλλες ιστοσελίδες. Συμβολίζονται με την ετικέτα <a>, η οποία μπορεί να χρησιμοποιηθεί με τους εξής τρόπους:

1. Δημιουργώντας έναν σύνδεσμο προς ένα άλλο έγγραφο χρησιμοποιώντας το HTML στοιχείο «href».
2. Δημιουργώντας σελιδοδείκτες σε ένα έγγραφο με χρήση του HTML στοιχείου «name».

Ένας υπερσύνδεσμος συντάσσεται με τον ακόλουθο τρόπο: Link. Παραδείγματος χάρη:

```
<a href="http://en.wikipedia.org/wiki/Main_Page">Wikipedia</a>
```

Μπορεί να χρησιμοποιηθεί το στοιχείο target="_blank", σε περίπτωση η επιθυμητή ενέργεια είναι να ανοίγει ο υπερσύνδεσμος σε νέο παράθυρο του περιηγητή:

```
<a href="http://en.wikipedia.org/wiki/Main_Page" target="_blank">Wikipedia</a>
```

Εναλλακτικά, για να ανοίξει στην ίδια καρτέλα στην οποία βρίσκεται ο χρήστης, χρησιμοποιείται το στοιχείο target="_self":

```
<a href="http://en.wikipedia.org/wiki/Main_Page" target="_self">Wikipedia</a>
```

7.2.1.6 Παράγραφοι & Μορφοποίηση

Τα HTML έγγραφα χωρίζονται σε παραγράφους. Όπως αναφέρθηκε και προηγουμένως, οι παράγραφοι στην HTML ορίζονται με τον ακόλουθο τρόπο:

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

Η ετικέτα <p> χρησιμεύει για τη δήλωση της παραγράφου. Οι περιηγητές προσθέτουν αυτόματα περιθώριο (margin) πριν και μετά από κάθε στοιχείο παραγράφου (μια κενή γραμμή). Τα κενά αυτά μπορεί να τα επεξεργαστεί ο σχεδιαστής της ιστοσελίδας μέσω του CSS. Οι περισσότεροι περιηγητές θα εμφανίσουν σωστά τις παραγράφους, ακόμα κι αν παραληφθεί η ετικέτα λήξης </p>. Παρόλα αυτά, δεν ενδείκνυται, καθώς μπορεί να προκληθούν απρόβλεπτα αποτελέσματα.

Η ετικέτα
 ορίζει μια αλλαγή γραμμής και μπορεί να χρησιμοποιηθεί για νέα γραμμή χωρίς όμως να δημιουργηθεί μια νέα παράγραφος. Το συγκεκριμένο στοιχείο είναι κενό, δηλαδή δεν χρειάζεται ετικέτα λήξης. Μπορεί να δηλωθεί με το σύντομο τρόπο
.

Μια άλλη χρήσιμη ετικέτα είναι η ετικέτα <hr> η οποία δημιουργεί μια οριζόντια διαχωριστική γραμμή κειμένου στην ιστοσελίδα μας. Όπως και η ετικέτα
, έτσι κι αυτή μπορεί να γραφεί με το σύντομο τρόπο <hr/>.

Τέλος, η ετικέτα `<pre>` χρησιμοποιείται για να ορίσει ένα προ-μορφοποιημένο (pre-formatted) κείμενο. Το κείμενο αυτό έχει δομή, δηλαδή προκαθορισμένα κενά, αλλαγές γραμμής και δεξιά στοίχιση. Παρακάτω, παρατίθεται ένα παράδειγμα προ-μορφοποιημένου κειμένου.

```
<!DOCTYPE html>
<html>
<body>
<p>Not pre-formatted text.</p>
<pre>
  Pre-formatted text.

  More pre-formatted text.
</pre>
</body>
</html>
```

Το αποτέλεσμα αυτού του παραδείγματος στην οθόνη θα είναι το ακόλουθο:

```
Not pre-formatted text.
  Pre-formatted text.

  More pre-formatted text.
```

Όσον αφορά στη μορφοποίηση του κειμένου, μπορεί να γίνει μέσω ειδικών στοιχείων της HTML, τα οποία σχεδιάστηκαν για την εμφάνιση ειδικών τύπων κειμένου, όπως έντονο (bold) κείμενο, σημαντικό (important) κείμενο, πλάγιο (italic), τονισμένο (emphasized), μικρό (small), υποσημειώσεις (subscript), κ.α. Η πλήρης λίστα των στοιχείων μορφοποίησης κειμένου παρατίθεται στον Πίνακα 7.3 (Πηγή: w3schools):

Ετικέτα	Περιγραφή
<code></code>	Έντονο κείμενο
<code></code>	Τονισμένο κείμενο
<code><i></code>	Πλάγιο κείμενο
<code><small></code>	Μικρότερο κείμενο
<code></code>	Σημαντικό κείμενο
<code><sub></code>	Υποσημείωση
<code><sup></code>	Κείμενο σε εκθέτη
<code><ins></code>	Κείμενο που παρεμβάλλεται

	Διαγεγραμμένο κείμενο
<mark>	Σημειωμένο κείμενο

Πίνακας 7.3. Στοιχεία μορφοποίησης κειμένου HTML

Ιδιαίτερα σημαντικές είναι και οι επικεφαλίδες, καθώς παρέχουν επαυξημένες δυνατότητες μορφοποίησης μιας ιστοσελίδας και χρησιμοποιούνται από τις μηχανές αναζήτησης. Η σημαντικότερη επικεφαλίδα ορίζεται από το ζεύγος ετικετών <h1>...</h1>. Οι υπόλοιπες ετικέτες επικεφαλίδων χρησιμοποιούνται με φθίνουσα σειρά ως την ετικέτα <h6> η οποία είναι και η λιγότερο σημαντική. Όσο σημαντικότερη η επικεφαλίδα τόσο εντονότερα και μεγαλύτερα τα γράμματα του περιεχομένου της. Φυσικά, το στυλ κάθε επικεφαλίδας μπορεί να αλλάξει μέσω CSS.

7.2.1.7 HTML στυλ

Κάθε HTML στοιχείο έχει ένα προεπιλεγμένο στυλ, δηλαδή προκαθορισμένο χρώμα φόντου, κειμένου, μέγεθος κειμένου, κ.λπ.) Η αλλαγή του προεπιλεγμένου στυλ του HTML στοιχείου γίνεται μέσω του style attribute. Η σύνταξη του HTML style attribute στη γενική της μορφή γίνεται ως εξής: style="property:value". Παραδείγματος χάρη, η παρακάτω γραμμή δείχνει πώς μπορεί να αλλάξει το χρώμα φόντου από άσπρο σε ανοιχτό γκρι:

```
<body style="background color:lightgrey">
```

Η ιδιότητα color (χρώμα) ορίζει το χρώμα κειμένου που θα χρησιμοποιηθεί σε ένα στοιχείο HTML. Συντάσσεται ως εξής:

```
<p style="color:red">The text color here is red.</p>
```

Αντίστοιχα, με χρήση της ιδιότητας font-family ορίζεται η γραμματοσειρά που θα χρησιμοποιηθεί σε ένα HTML στοιχείο.

```
<h1 style="font family:verdana">This is a heading written in verdana. </h1>
```

Άλλες δύο χρήσιμες ιδιότητες για αλλαγή του μεγέθους της γραμματοσειράς και της στοίχισης του κειμένου είναι οι ιδιότητες font-size και text-align, αντίστοιχα.

Ο παραπάνω τρόπος αλλαγής του στυλ (δηλαδή μέσω της χρήσης του style attribute των στοιχείων HTML) ονομάζεται inline. Η αλλαγή στυλ μπορεί να γίνει με ακόμα δύο τρόπους: τον internal τρόπο (χρησιμοποιώντας, δηλαδή, ένα <style> element στο κομμάτι <head> του HTML εγγράφου μας, ή με τον external τρόπο) χρησιμοποιώντας ένα ή περισσότερα εξωτερικά αρχεία CSS. Το ακρωνύμιο CSS σημαίνει Cascading Style Sheets. Η χρήση εξωτερικών αρχείων CSS είναι ο πιο συνηθισμένος τρόπος για αλλαγή του στυλ της ιστοσελίδας μας. Θα γίνει αναλυτικότερη περιγραφή του CSS σε επόμενη ενότητα.

Όσον αφορά στα χρώματα στην HTML, υπάρχουν 140 ονόματα χρωμάτων που ορίζονται από την προδιαγραφή της HTML5 και του CSS3. Τα 17 που ορίζονται από την HTML είναι τα εξής: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, orange, purple, red, silver, teal, white, και yellow. Οι τιμές των χρωμάτων ορίζονται χρησιμοποιώντας δεκαεξαδική σήμανση (Hexadecimal) για τις τρεις τιμές RGB (Red,

Green, Blue). Οι δεκαεξαδικές τιμές συντάσσονται με μια δέση (#), ακολουθούμενη από τρεις ή έξι δεκαεξαδικούς χαρακτήρες. Παραδείγματος χάρη, το χρώμα μαύρο ορίζεται ως #000 ή #000000 με δεκαεξαδική σήμανση και rgb(0,0,0) με RGB σήμανση. Το άσπρο ορίζεται ως #FFF ή #FFFFFF και rgb(255, 255, 255).

7.2.1.8 Εικόνες στην HTML

Για την ενσωμάτωση εικόνων, χρησιμοποιείται η ετικέτα , η οποία είναι κενή ετικέτα, αλλά περιέχει ιδιότητες (attributes) της εικόνας. Αυτές είναι οι ακόλουθες:

- **src:** ορίζει το url, δηλαδή τη διεύθυνση που βρίσκεται αποθηκευμένη η εικόνα στον Παγκόσμιο Ιστό. Η σύνταξή του, δηλαδή είναι ως εξής: .
- **alt:** για τον προσδιορισμό εναλλακτικού κειμένου για την περίπτωση που για κάποιο λόγο η εικόνα δεν μπορεί να εμφανιστεί. Επιπλέον, χρειάζεται για screen readers - προγράμματα που διαβάζουν ό,τι εμφανίζεται σε μια οθόνη και το αναπαράγουν ως ομιλία, εικονίδια ήχου ή και braille. Χρησιμοποιούνται από άτομα με προβλήματα όρασης ή μαθησιακές δυσκολίες. Συνεπώς, η συμπερίληψή του alt attribute είναι απαραίτητη. Παραδείγματος χάρη:

```

```

- **width & height:** Μπορεί να γίνει χρήση του style attribute για τον ορισμό του πλάτους και του ύψους μιας εικόνας. Η τιμή του προσδιορίζεται με pixels. Στο ακόλουθο παράδειγμα ορίζεται ένα λογότυπο με πλάτος και ύψος 100 pixels:

```

```

Εναλλακτικά, μπορούν να χρησιμοποιηθούν τα width και height attributes. Οι τιμές και εδώ προσδιορίζονται σε pixels, χωρίς όμως «px» μετά τις αριθμητικές τιμές: . Η χρήση του πρώτου τρόπου (style attribute) συστήνεται, καθώς αποτρέπει τα style sheets (CSS) να αλλάζουν το προεπιλεγμένο μέγεθος των εικόνων. Να σημειωθεί ότι πρέπει πάντα να προσδιορίζεται το μέγεθος της εικόνας, γιατί αλλιώς μπορεί να υπάρξουν προβλήματα κατά τη φόρτωσή της (π.χ. να «τρεμοπαίζει» η ιστοσελίδα).

- **floating:** Μια εικόνα μπορεί να βρίσκεται στο αριστερό ή στο δεξί μέρος μιας παραγράφου. Αυτό ορίζεται ως εξής:

```
<p>  
  
A paragraph with an image. The image floats to the left of the text.  
</p>
```

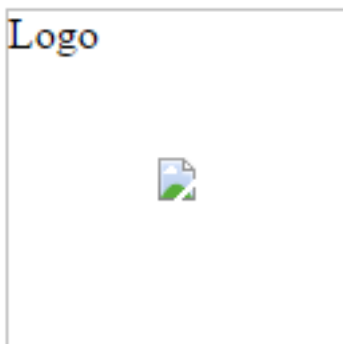
Εάν δεν έχει προηγουμένως προσδιοριστεί ρητά, ο περιηγητής ιστοσελίδων αναμένει να βρει την εικόνα στον ίδιο φάκελο με την ιστοσελίδα. Φυσικά, σε ένα ρεαλιστικό σενάριο, η εικόνα μπορεί να βρίσκεται σε έναν φάκελο όπου αποθηκεύουμε τις εικόνες, οπότε θα πρέπει να αναφερθεί ο φάκελος στο όνομα της εικόνας, στο src attribute: . Για πρόσβαση σε μια εικόνα που βρίσκεται σε άλλον εξυπηρετητή Παγκόσμιου Ιστού πρέπει να γίνει παράθεση της διεύθυνσης (web address) της εικόνας.

Εάν ο περιηγητής δε μπορεί να βρει την εικόνα, θα δείξει έναν εικονίδιο για «broken link» και θα εμφανίσει το κείμενο που ορίστηκε με την ιδιότητα alt (Εικόνα 7.14). Να σημειωθεί σε αυτό το σημείο ότι μια

εικόνα μπορεί να χρησιμοποιηθεί και ως υπερσύνδεσμος παραθέτοντάς τη μέσα στην ετικέτα για υπερσύνδεσμο ως ακολούθως:

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

</a>
```



Εικόνα 7.14. *Broken link icon*

7.2.1.9 Πίνακες στην HTML

Οι πίνακες αποτελούν δομή της HTML με χρήση της οποίας γίνεται η εμφάνιση κειμένου και εικόνων στοιχισμένων σε γραμμές και στήλες. Ορίζονται με το ζεύγος ετικετών `<table>...</table>`. Χωρίζονται σε table rows (γραμμές) με τις ετικέτες `<tr>...</tr>`, οι οποίες μπορούν να χωριστούν σε table data (στήλες με δεδομένα) με τις ετικέτες `<td>...</td>` και table headings (επικεφαλίδες στήλης) με τις ετικέτες `<th>...</th>`. Τα table data `<td>` περιέχουν τα δεδομένα του πίνακα (κείμενο, εικόνες, λίστες, άλλους πίνακες, κ.λπ.). Τα table headings από προεπιλογή εμφανίζονται έντονα (bold) και στοιχισμένα στο κέντρο (centered). Στο ακόλουθο παράδειγμα δηλώνεται ένας πίνακας με τρεις γραμμές και τρεις στήλες.

```

<!DOCTYPE html>
<html>
<body>

<table style="width:100%">
  <tr>
    <td>Name</td>
    <td>Surname</td>
    <td>Age</td>
  </tr>
  <tr>
    <td>Linus</td>
    <td>Torvalds</td>
    <td>45</td>
  </tr>
  <tr>
    <td>Richard</td>
    <td>Stallman</td>
    <td>61</td>
  </tr>
</table>

</body>
</html>

```

Ο πίνακας δε θα έχει περίγραμμα, εκτός κι αν προστεθεί με χρήση του border attribute ή μέσω CSS:
`<table border="1" style="width:100%">`

Μπορούμε, επίσης, μέσω της ιδιότητας *width* να ορίσουμε το μήκος του πίνακα και με την ιδιότητα *height* το ύψος του πίνακα. Οι ιδιότητες αυτές μπορούν να τοποθετηθούν και στις ετικέτες `<td>` και `<th>`. Να σημειωθεί εδώ ότι η ιδιότητα *width* τοποθετείται μόνο στα κελιά της πρώτης γραμμής του πίνακα και εφαρμόζεται και για όλα τα υπόλοιπα κελιά. Αντίστοιχα, η ιδιότητα *height* τοποθετείται μόνο στο πρώτο κελί μιας γραμμής του πίνακα και εφαρμόζεται για όλα τα κελιά της γραμμής αυτής.

Μια άλλη χρήσιμη ετικέτα που μπορεί να χρησιμοποιηθεί σε πίνακες είναι η ετικέτα `<caption>`, η οποία προσθέτει μια λεζάντα στον πίνακα και εισάγεται αμέσως μετά την ετικέτα `<table>`. Σε περίπτωση που θέλουμε ένα κελί να επεκτείνεται σε περισσότερες από μία στήλες, μπορεί να χρησιμοποιηθεί η ιδιότητα *colspan*. Παραδείγματος χάρη, εδώ, η επικεφαλίδα «Title» θα εμφανίζεται σε δύο κελιά, αντί για ένα:

```
<th colspan="2">Title</th>
```

Αντίστοιχα, για να επεκταθεί ένα κελί σε περισσότερες από μία γραμμές, γίνεται χρήση της ιδιότητας *rowspan*:

```
<th rowspan="2">Title:</th>
```

Τέλος, αξίζει να γίνει αναφορά στις ιδιότητες *cellspacing* και *cellpadding* της ετικέτας `<table>`. Η ιδιότητα *cellspacing* χρησιμοποιείται για να οριστεί το πάχος των γραμμών που διαχωρίζουν τα κελιά, ενώ η ιδιότητα *cellpadding* για να οριστεί ο κενός χώρος μεταξύ του πλαισίου των κελιών και του περιεχομένου τους.

7.2.1.10 HTML λίστες

Στην HTML είναι δυνατό να υπάρχουν λίστες χωρίς διάταξη (unordered lists), λίστες με διάταξη (ordered lists), και λίστες περιγραφής (description lists). Οι λίστες χωρίς διάταξη δηλώνονται με την ετικέτα έναρξης ``.

Κάθε στοιχείο της λίστας ξεκινά με την ετικέτα . Και οι δύο αυτές ετικέτες ακολουθούνται από τις αντίστοιχες τους ετικέτες λήξης. Από προεπιλογή, το κάθε αντικείμενο αυτής της λίστας θα έχει κουκκίδες ως σύμβολο (marker) αριστερά από κάθε στοιχείο της λίστας. Συνεπώς, τα περιεχόμενα της λίστας αυτής δεν έχουν κάποια αριθμημένη σειρά. Στις λίστες χωρίς διάταξη υπάρχει η δυνατότητα προσθήκης style attribute για να αλλαχθεί το σχήμα του συμβόλου. Παραδείγματος χάρη:

```
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
</ul>
```

Αποτέλεσμα:

- Item 1
- Item 2
- Item 3

Εικόνα 7.15. Μη διατεταγμένη λίστα

Μια διατεταγμένη λίστα ξεκινά με την ετικέτα , ενώ κάθε αντικείμενο της λίστας προηγείται από την ετικέτα . Τα αντικείμενα της διατεταγμένης λίστας μπορούν να έχουν αριθμούς, πεζά/κεφαλαία γράμματα ή πεζή/κεφαλαία ρωμαϊκή αρίθμηση ως marker. Συνεπώς, αν αλλαχθεί η σειρά των αντικειμένων, αλλάζει και το νόημα της λίστας. Παραδείγματος χάρη:

```
<ol type="I">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

Αποτέλεσμα:

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

Εικόνα 7.16. Διατεταγμένη λίστα

Υπάρχουν και λίστες περιγραφής (description lists), οι οποίες είναι λίστες από όρους (terms) με μια περιγραφή του κάθε όρου. Η λίστα περιγραφής ορίζεται με την ετικέτα <dl>, ενώ οι ετικέτες <dt> και <dd> ορίζουν το όνομα του όρου και της περιγραφής του, αντίστοιχα. Επιπροσθέτως, μια λίστα μπορεί να είναι εμφωλευμένη (λίστα μέσα σε λίστα) ή να περιέχει επιπρόσθετα HTML στοιχεία, όπως υπερσυνδέσμους και εικόνες. Τέλος, να σημειωθεί ότι το στυλ των λιστών μπορεί να αλλάξει μέσω CSS.

7.2.1.11 Η ετικέτα <div>

Το ζεύγος ετικετών <div>...</div> χρησιμοποιείται για να οριστεί μια ενότητα σε ένα έγγραφο HTML. Ουσιαστικά, χρησιμεύει στην ομαδοποίηση στοιχείων στα οποία μετά θα εφαρμοστεί ενιαία μορφοποίηση (χρώματα, γραμματοσειρές, κ.λπ.) με CSS, οπότε διευκολύνεται η σχεδίαση της διάταξης (layout) της ιστοσελίδας. Παραδείγματος χάρη, μπορεί να γίνει ομαδοποίηση κάποιων στοιχείων παραγράφου σε ένα στοιχείο <div>, και με χρήση CSS να εφαρμοστεί ένα στυλ γραμματοσειράς στην ετικέτα <div>, αντί να ξαναγράφεται το ίδιο στυλ για κάθε στοιχείο της παραγράφου.

Η ετικέτα <div> μπορεί, λοιπόν, να παρομοιαστεί με ένα κενό πλαίσιο ή «δοχείο», στο οποίο υπάρχουν να έχουμε τα επιθυμητά στοιχεία της ιστοσελίδας, ή ακόμα και άλλα στοιχεία div. Η ετικέτα <div> υποστηρίζεται από όλους τους περιηγητές. Στην πιο συνηθισμένη διάταξη ιστοσελίδας, υπάρχουν ενότητες για header (επικεφαλίδα της ιστοσελίδας), menu (γραμμή πλοήγησης), content (περιεχόμενο ιστοσελίδας), και footer (βάση ιστοσελίδας). Να επισημανθεί ότι οι περιηγητές προσθέτουν μια κενή γραμμή πριν και μετά την ενότητα. Αυτό μπορεί να αλλάχθει με χρήση CSS.

Ακολουθεί ένα παράδειγμα με χρήση div.

```
<div id="myDiv" name="myDiv" title="Example Div" style="color: red; border: 1px solid black;">

  <h1>Subtitle</h1>

  <p>Your content here...</p>

  <h6>Subtitle</h6>

</div>
```

Όπως αναφέρθηκε παραπάνω, τα στοιχεία που είναι μέσα στο ζεύγος ετικετών <div>...</div> αποκτούν τις ιδιότητες και το στυλ που εφαρμόζονται στο στοιχείο div. Συνεπώς, η παράγραφος και οι τίτλοι θα έχουν τώρα κόκκινο χρώμα. Προστέθηκε, επίσης, στο στοιχείο div μαύρη γραμμή περιγράμματος με πάχος ενός pixel.

7.2.1.12 HTML forms

Οι HTML φόρμες αποτελούν σύνθεση από κουμπιά (buttons), πλαίσια ελέγχου (checkboxes) και πεδία εισαγωγής κειμένου (text input fields), τα οποία είναι ενσωματωμένα σε έγγραφο HTML. Στόχος είναι να καταγραφεί ό,τι εισάγει ο χρήστης σε αυτά τα στοιχεία και να του δοθεί η δυνατότητα να αλληλεπιδράσει έμμεσα με την ιστοσελίδα. Έτσι, παρέχονται διάφορα πεδία στα οποία ο χρήστης θα συμπληρώσει τα στοιχεία του, όπως το όνομά του, το τηλέφωνό του και τη διεύθυνση του ηλεκτρονικού ταχυδρομείου του.

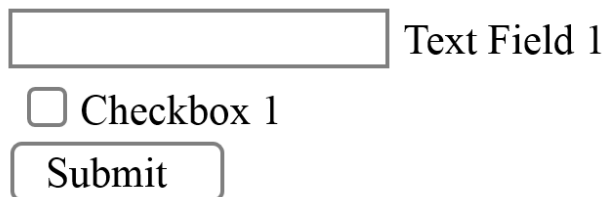
Οι HTML φόρμες τοποθετούνται σε ιστοσελίδες χρησιμοποιώντας το ζεύγος ετικετών `<form>...</form>`. Οι ετικέτες αυτές θα πρέπει να περιλαμβάνουν κάποια άλλα στοιχεία φορμών HTML, δημιουργώντας μια έγκυρη και με συνοχή HTML φόρμα. Στη συνέχεια, παρατίθεται ένα απλό παράδειγμα φόρμας.

```
<form name="myWebForm" action="myServerSideScript.php" method="post">

  <input type="text" /> Text Field 1<br />
  <input type="checkbox" /> Checkbox 1<br />
  <input type="submit" value="SUBMIT" />

</form>
```

Παρατηρούμε ότι χρησιμοποιήθηκαν οι ιδιότητες `action` και `method` στην ετικέτα `form`. Οι ιδιότητες αυτές είναι απαραίτητες για τα στοιχεία των HTML φορμών, καθώς υποδεικνύουν που να στείλουν τα δεδομένα της HTML φόρμας (`action`) και πώς να γίνει η επεξεργασία των δεδομένων (`method`). Φυσικά, η HTML από μόνη της δε μπορεί να επεξεργαστεί δεδομένα. Στην περίπτωση αυτή χρειάζεται μια scripting γλώσσα όπως PERL ή PHP σε συνδυασμό με την HTML ώστε γίνει μετά επεξεργασία των δεδομένων που συλλέγει η HTML φόρμα. Παρακάτω, φαίνεται το πώς θα εμφανιστεί η φόρμα στον περιηγητή (Εικόνα 7.17). Σε αυτό το σημείο, να σημειωθεί ότι είναι ιδιαίτερα σημαντικό να υπάρχει η ιδιότητα `name` στη φόρμα για καλύτερη οργάνωση του εγγράφου, καθώς και `submit` button μέσα στο ζεύγος των ετικετών `<form>...</form>` ώστε να γίνεται σωστά η υποβολή των δεδομένων.



The image shows a rendered HTML form. It consists of three elements: a rectangular text input field, a checkbox followed by the text 'Checkbox 1', and a rectangular button with the text 'Submit' inside it.

Εικόνα 7.17. Η έξοδος του παραπάνω παραδείγματος

Τα HTML στοιχεία εισαγωγής (`input` elements) είναι στοιχεία φορμών, όπως πεδία κειμένου, κουμπιά και πλαίσια ελέγχου. Ονομάζονται έτσι καθώς έχουν την ετικέτα `<input>`, καθώς και κάποιες ιδιότητες για την κατηγοριοποίηση και την ονομασία κάθε στοιχείου, όπως είδαμε και στο παραπάνω παράδειγμα. Η ιδιότητα `type` προσδιορίζει τον τύπο του `input` element και μπορεί να πάρει τις ακόλουθες τιμές: `text`, `checkbox`, `radio`, `button`, `submit`, `reset`, `password`, και `hidden`. Κάθε ένα από αυτά έχει διαφορετική λειτουργικότητα και εμφάνιση. Στο παραπάνω παράδειγμα χρησιμοποιήθηκε ένα πεδίο κειμένου (`input type="text"`), ένα πεδίο ελέγχου (`input type="checkbox"`) και ένα κουμπί (`input type="submit"`) για υποβολή των δεδομένων που εισήγαγε ο χρήστης μέσω των παραπάνω στοιχείων της φόρμας.

Η ιδιότητα `value` έχει διαφορετικό ρόλο ανάλογα με τον τύπο του στοιχείου εισαγωγής. Για παράδειγμα, όταν χρησιμοποιείται με ένα HTML κουμπί, καθορίζει το κείμενο στο κουμπί, ενώ όταν χρησιμοποιείται με ένα πεδίο κειμένου συμπληρώνει το πεδίο με την προκαθορισμένη τιμή. Άλλες χρήσιμες ιδιότητες των στοιχείων της φόρμας είναι οι ιδιότητες `name` και `id`, οι οποίες λειτουργούν ως σύνδεσμος ανάμεσα στην HTML φόρμα και το `script` που θα χρησιμοποιηθεί αργότερα για την επεξεργασία των δεδομένων που υπεβλήθησαν μέσω της φόρμας. Μια καλή τακτική είναι να χρησιμοποιούνται και οι δύο ιδιότητες, καθώς κάποιες scripting γλώσσες απαιτούν τη μία ιδιότητα, ενώ άλλες απαιτούν την άλλη.

7.2.1.13 HTML5 – Εισαγωγικά Στοιχεία

Η HTML5 αποτελεί την πιο πρόσφατη έκδοση της γλώσσας σήμανσης HTML. Είναι η ολοκληρωμένη πέμπτη αναθεώρηση του προτύπου της HTML από το World Wide Web Consortium (W3C) (Οκτώβριος 2014). Η προηγούμενη έκδοση της HTML (HTML4) είχε προτυποποιηθεί το 1997. Η HTML5 θα αντικαταστήσει τις HTML 4, XHTML 1 και DOM Level 2 HTML. Σκοπός της HTML5 είναι η βελτίωση της γλώσσας με την ενσωμάτωση της υποστήριξης πολυμέσων, τη μείωση της ανάγκης για εμπορικές πρόσθετες εφαρμογές (plugins), την αύξηση της αναγνωσιμότητάς της, καθώς και τη δημιουργία πλουσίων διαδικτυακών εφαρμογών (Rich Internet Applications), όπως παράγονταν έως τώρα με χρήση του Adobe Flash, του Microsoft Silverlight, κ.α.

Επιπλέον, η μεγαλύτερη αλλαγή που ίσως επιφέρει η HTML5 είναι η χρήση APIs (Application Programming Interfaces) ως μέρος του HTML document object model (DOM), χωρίς να έχουν αντίστοιχα στοιχεία σήμανσης. Είναι καθαρά DOM APIs τα οποία μπορούν να χρησιμοποιηθούν με JavaScript για τη διαμοίραση - κατανάλωση δεδομένων και τη δημιουργία εφαρμογών.



Εικόνα 7.18. Το λογότυπο της HTML5

Όσον αφορά στη διαδικασία προτυποποίησης της HTML5, τα Mozilla Foundation και Opera Software παρουσίασαν ένα άρθρο τοποθέτησης στο W3C Workshop τον Ιούνιο του 2004 επικεντρωμένο στην ανάπτυξη τεχνολογιών οι οποίες είναι συμβατές προς τα πίσω με υπάρχοντες περιηγητές ιστοσελίδων και συμπεριλάμβανε μια αρχική προδιαγραφή των Web Forms 2.0. Η ομάδα εργασίας Web Hypertext Application Technology Working Group (WHATWG) δημιουργήθηκε και άρχισε να εργάζεται πάνω σε αυτό το άρθρο τοποθέτησης, με αποτέλεσμα την ανακοίνωση μιας δεύτερης προδιαγραφής (Web Applications 1.0). Οι δύο προδιαγραφές συγχωνεύτηκαν στην προδιαγραφή HTML5 η οποία αποτέλεσε σημείο έναρξης της εργασίας του νέου HTML Working Group του W3C το 2007. Να σημειωθεί σε αυτό το σημείο ότι η έκδοση 4.01 της HTML δεν είχε ανανεωθεί από το 2000.

Η ομάδα εργασίας Web Hypertext Application Technology Working Group (WHATWG) δημοσίευσε το πρώτο δημόσιο working draft της προδιαγραφής στις 22 Ιανουαρίου 2008. Το Σεπτέμβριο του 2012, το W3C πρότεινε ένα πλάνο για την έκδοση της προδιαγραφής της HTML5 ως το τέλος του 2014 και της HTML5.1 ως το τέλος του 2016. Πλέον, η HTML5 υποστηρίζεται από όλους τους σύγχρονους περιηγητές ιστοσελίδων, ενώ οι παλαιότερες εκδόσεις τους μπορούν να αυτόματα να χειριστούν τα στοιχεία που δεν αναγνωρίζουν ως ενσωματωμένα στοιχεία (inline elements). Μπορείτε να δείτε πόσα από τα χαρακτηριστικά της HTML5 υποστηρίζει ο περιηγητής σας στην ιστοσελίδα <https://html5test.com/>.

Η HTML5 συγκρίνεται συχνά με το Adobe Flash, καθώς και οι δύο τεχνολογίες παρέχουν τη δυνατότητα για αναπαραγωγή ήχου και βίντεο σε ιστοσελίδες και μπορούν να χρησιμοποιήσουν Scalable Vector Graphics (SVG). Παρόλα αυτά, οι δύο αυτές τεχνολογίες είναι πολύ διαφορετικές, διότι: (α) η HTML5 δε μπορεί να χρησιμοποιηθεί από μόνη της για σχεδιοκίνηση και διαδραστικότητα, αλλά πρέπει να συνδυαστεί με CSS3 ή JavaScript, και (β) υπάρχουν πολλές δυνατότητες του Flash που δεν έχουν άμεση αντιστοιχία στην HTML5.

Η HTML5 αναπτύχθηκε με βάση τους παρακάτω κανόνες και απαιτήσεις:

- Έπρεπε να προστεθούν νέα χαρακτηριστικά στη δομή των HTML, CSS και JavaScript.
- Έπρεπε να μειωθούν οι περιπτώσεις στις οποίες χρειαζόταν εγκατάσταση πρόσθετων εφαρμογών στον περιηγητή ιστοσελίδων για στοιχεία συγκεκριμένου τύπου, όπως για παράδειγμα βίντεο και ήχος.
- Ο τρόπος διαχείρισης σφαλμάτων έπρεπε να βελτιωθεί.
- Έπρεπε να γίνει προσθήκη περισσότερων ετικετών οι οποίες θα αντικαθιστούσαν τμήματα κώδικα JavaScript που χρησιμοποιούνταν συχνά από τους σχεδιαστές ιστοσελίδων.

- Το νέο πρότυπο θα έπρεπε να είναι αυτόνομο, χωρίς να χρησιμοποιεί τμήματα κώδικα από άλλα πρότυπα.
- Τα βήματα του σχεδιασμού και της ανάπτυξης του νέου πρότυπου θα έπρεπε να είναι ανοιχτά στο κοινό.

7.2.1.14 Χαρακτηριστικά της HTML5

Σε αυτό το σημείο θα γίνει συνοπτική παρουσίαση κάποιων νέων χαρακτηριστικών της HTML5.

- Σήμανση - προσθήκη νέων ετικετών: η HTML5 εισάγει κάποια νέα στοιχεία που κάνουν τη δημιουργία και τη διαχείριση των ιστοσελίδων ακόμη πιο εύκολες. Κάποιες από αυτές είναι οι ετικέτες <footer>, η οποία αναφέρεται στο κάτω μέρος της ιστοσελίδας, <article>, <header>, κ.α. Επίσης, καταργείται η χρήση κάποιων μη αποδεκτών στοιχείων της HTML 4.01, (π.χ. <center>), τα οποία χρησιμοποιούνταν καθαρά για λόγους παρουσίασης και μπορούν να αντικατασταθούν από CSS.
- Δυνατότητα σχεδιασμού γραφικών με χρήση JavaScript και της νέας ετικέτας <canvas>.
- Αναπαραγωγή βίντεο και ήχου χωρίς να χρειάζεται η εγκατάσταση πρόσθετων με τις νέες ετικέτες <video> και <audio>.
- Προσθήκη νέων στοιχείων στις HTML φόρμες, όπως <calendar>, <date>, <time>, κ.λπ.
- Η σύνταξη της HTML5 δε βασίζεται πλέον στην SGML, αν και η σήμανση μοιάζει. Έχει σχεδιαστεί να είναι συμβατή προς τα πίσω με τις παλαιότερες εκδόσεις της HTML.
- Επιπλέον, η HTML5 ορίζει scripting application programming interfaces (APIs) τα οποία μπορούν να χρησιμοποιηθούν με τη JavaScript. Κάποια από τα νέα APIs περιλαμβάνουν Offline Web Applications, Drag-and-drop, Document editing, Web Storage, κ.λπ.
- Διαχείριση λαθών: η HTML5 είναι σχεδιασμένη έτσι ώστε οι παλαιότεροι περιηγητές να μπορούν πλέον να αγνοούν με ασφάλεια νέες HTML5 δομές. Η προδιαγραφή της HTML5 περιλαμβάνει λεπτομερείς κανόνες με στόχο διαφορετικά συμβατοί περιηγητές να παράγουν το ίδιο αποτέλεσμα στην περίπτωση που η σύνταξη είναι λάθος.

7.2.1.15 Νέα στοιχεία HTML5

Πολλές από τις ετικέτες του προτύπου HTML 4.01, δεν χρησιμοποιήθηκαν ποτέ ή δεν χρησιμοποιήθηκαν για το σκοπό που αυτές είχαν αρχικά δημιουργηθεί. Οι ετικέτες αυτές αφαιρέθηκαν ή ξαναγράφηκαν από την αρχή στην HTML5. Επίσης, προστέθηκαν πολλές νέες ετικέτες (Πλήρης λίστα είναι διαθέσιμη στην ιστοσελίδα http://www.w3schools.com/html/html5_new_elements.asp). Ας δούμε κάποιες από αυτές με περισσότερες λεπτομέρειες.

Στοιχεία για τη δομή του εγγράφου

- <article> - ορίζει εξωτερικό περιεχόμενο, όπως π.χ. κείμενο από μια άλλη ιστοσελίδα.
- <aside> - ορίζει περιεχόμενο ανεξάρτητο από το περιεχόμενο της ιστοσελίδας.
- <details> - ορίζει επιπρόσθετες λεπτομέρειες τις οποίες ο χρήστης μπορεί να δει ή να κρύψει.
- <dialog> - ορίζει ένα πλαίσιο ή παράθυρο διαλόγου.
- <figure> - ορίζει περιεχόμενο όπως εικονογραφήσεις, διαγράμματα, φωτογραφίες, κ.λπ.
- <figcaption> - ορίζει μια λεζάντα για ένα στοιχείο <figure>.
- <header> - ορίζει ένα τμήμα «header» για το έγγραφο.
- <main> - ορίζει το κυρίως περιεχόμενο του εγγράφου.
- <nav> - ορίζει συνδέσμους πλοήγησης στο έγγραφο.
- <section> - ορίζει μια ενότητα στο έγγραφο.

Στοιχεία για φόρμες

- <output> - ορίζει το αποτέλεσμα ενός υπολογισμού.
- <datalist> - ορίζει προκαθορισμένες επιλογές για έλεγχο της εισόδου του χρήστη
- <keygen> - ορίζει ένα πεδίο generator για key-pair: Προσδιορίζει ένα πεδίο γεννήτρια (generator) ζευγών κλειδιών (key-pair) για χρήση σε φόρμες Ιστού (Web forms). Χρησιμοποιείται για την επεξεργασία φορμών Ιστού με συστήματα διαχείρισης πιστοποιητικών. Το στοιχείο αυτό δημιουργεί ένα ασφαλές ιδιωτικό

κλειδί (private key), καθώς και ένα δημόσιο κλειδί (public key). Όταν υποβάλλεται η φόρμα, το ιδιωτικό κλειδί αποθηκεύεται τοπικά, ενώ το δημόσιο κλειδί αποστέλλεται στον εξυπηρετητή. Στόχος είναι η παροχή ασφαλούς τρόπου για την πιστοποίηση των χρηστών.

Γραφικά

<canvas> - για σχεδιασμό γραφικών με JavaScript

<svg> - για σχεδιασμό γραφικών με SVG

Ειδικότερα, η ετικέτα <canvas> ορίζει μια ορθογώνια περιοχή στην οποία μπορεί να γίνει σχεδίαση γραφικών, όπως παραδείγματος χάρη κύκλων, ορθογωνίων, χαρακτήρων, αλλά και να γίνει εμφάνιση εικόνων από εξωτερικές πηγές. Η σχεδίαση γραφικών μέσα σε αυτήν την περιοχή γίνεται «on the fly», με χρήση scripting (συνήθως με JavaScript). Το στοιχείο canvas έχει πολλές μεθόδους για τη σχεδίαση σχημάτων, κειμένων και την προθήκη εικόνων. Από προεπιλογή, ένα στοιχείο canvas δεν έχει περίγραμμα και περιεχόμενο. Ο παρακάτω κώδικας ορίζει ένα στοιχείο canvas με id «myCanvas», μαύρο περίγραμμα και διαστάσεις 300x200 pixels:

```
<canvas id="myCanvas" width="300" height="200" style="border:1px solid #000000;"></canvas>
```

Στη συνέχεια, γίνεται η σχεδίαση σε αυτήν την περιοχή με χρήση JavaScript. Ο κώδικας JavaScript τοποθετείται στις ετικέτες <script>...</script>. Αρχικά, στην πρώτη γραμμή του κώδικα JavaScript καθορίζεται η περιοχή canvas στην οποία θα σχεδιαστούν γραφικά, με το id της επιθυμητής περιοχής. Στη συνέχεια, δημιουργείται ένα αντικείμενο getContext("2d"), το οποίο είναι μέρος της HTML5. Οι δύο επόμενες γραμμές ορίζουν ένα ορθογώνιο κόκκινου χρώματος με διαστάσεις 150x75 pixels. Η αριστερή γωνία του ορθογωνίου θα είναι στο σημείο 0, 0 της περιοχής canvas.

```
<!DOCTYPE html>
<html>
<body>

<canvas id="myCanvas" width="300" height="200" style="border:1px solid #000000;">
Your browser does not support the HTML5 canvas tag.</canvas>

<script>
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.fillStyle = "#FF0000";
ctx.fillRect(0,0,150,75);
</script>

</body>
</HTML>
```

Αποτέλεσμα:



Εικόνα 7.19. Η έξοδος του παραπάνω παραδείγματος

Όσον αφορά στα SVG, είναι ακρωνύμιο των Scalable Vector Graphics, χρησιμεύουν για τον ορισμό γραφικών σε ιστοσελίδες και αποτελούν πρόταση του W3C. Η SVG αποτελεί γλώσσα για την περιγραφή δι-διάστατων (2D) γραφικών σε XML. Το στοιχείο <svg> αποτελεί container για SVG γραφικά και υποστηρίζει αρκετές μεθόδους για το σχεδιασμό κύκλων, μονοπατιών, εικόνων, κ.λπ. Τα SVG γραφικά, εν αντιθέσει με το canvas είναι ανεξάρτητα ανάλυσης (resolution), είναι προτιμότερα για εφαρμογές με μεγάλες περιοχές απόδοσης (rendering) γραφικών (π.χ. Google Maps), αλλά κάνουν slow rendering αν είναι πολύ σύνθετα και δεν προτείνονται για πολύ απαιτητικές εφαρμογές, όπως π.χ. παιχνίδια.

Ήχος & Βίντεο

<audio> - ορίζει ως περιεχόμενο ήχο ή μουσική.

<video> - ορίζει ως περιεχόμενο βίντεο ή ταινία.

<source> - ορίζει πηγές για ήχο και βίντεο.

<embed> - ορίζει «containers» για συμπερίληψη εξωτερικών εφαρμογών, όπως πρόσθετα.

Αξίζει να τονιστεί σε αυτό το σημείο ότι πριν την HTML5, η αναπαραγωγή ενός βίντεο σε μια ιστοσελίδα απαιτούσε τη χρήση κάποιας πρόσθετης εφαρμογής, όπως Flash ή QuickTime. Με χρήση του στοιχείου video της HTML5, αρκεί ο παρακάτω κώδικας:

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

Η ιδιότητα «controls» χρησιμεύει για την προσθήκη στοιχείων χειρισμού του βίντεο, όπως «play», «pause», και «volume». Το κείμενο που περιλαμβάνεται ανάμεσα στις ετικέτες <video> and </video> θα εμφανιστεί μόνο σε περιηγητές ιστοσελίδων που δεν υποστηρίζουν την ετικέτα <video>. Εάν συμπεριληφθούν πολλαπλές πηγές για το βίντεο (ετικέτα <source>), ο περιηγητής ιστοσελίδων θα χρησιμοποιήσει το αρχείο με τον πρώτο αναγνωρίσιμο τύπο (format). Με αντίστοιχο τρόπο και χρήση της ετικέτας <audio> μπορεί να γίνει ο ορισμός στοιχείων audio για αναπαραγωγή ήχων στην HTML5.

Η HTML5 εισήγαγε, επίσης, διάφορους νέους τύπους εισόδου από το χρήστη, όπως:

- Χρώμα.
- Ημερομηνία.
- Email.
- Μήνα.
- Αριθμό.
- Τηλέφωνο.
- URL.

Οι νέοι τύποι εισόδου που δεν υποστηρίζονται από παλαιότερους περιηγητές ιστοσελίδων θα έχουν ανάλογη συμπεριφορά με τον τύπο εισόδου «text». Στην HTML5 εισήχθησαν και διάφορες χρήσιμες ιδιότητες

για την ετικέτα `<input>`, όπως `autocomplete`, `autofocus`, `required` και `multiple`, καθώς και οι εξής ιδιότητες για το στοιχείο `<form>`: `autocomplete` και `novalidate`.

Τέλος, όσον αφορά στα APIs της HTML, αυτά περιλαμβάνουν τα εξής:

- **Geolocation:** χρησιμεύει για να τη λήψη της γεωγραφικής θέσης του χρήστη της ιστοσελίδας (με έγκρισή του).
- **Drag and Drop:** στην HTML5 το drag and drop αποτελεί μέρος της προδιαγραφής.
- **Local Storage:** για την αποθήκευση δεδομένων τοπικά στον περιηγητή ιστοσελίδων του χρήστη από εφαρμογές του Παγκόσμιου Ιστού (Web applications). Το local storage είναι πιο ασφαλές από τα cookies, και το αποθηκευτικό του όριο είναι πολύ μεγαλύτερο.
- **Application Cache:** χάρη στο application cache, οι χρήστες μπορούν να χρησιμοποιούν μια εφαρμογή του Παγκόσμιου Ιστού και όταν περιηγούνται στον Παγκόσμιο Ιστό είναι εκτός σύνδεσης (offline browsing), με υψηλότερη ταχύτητα, λόγω των αποθηκευμένων πόρων (cached resources) και με μειωμένο φόρτο στον εξυπηρετητή (server load), καθώς ο περιηγητής ιστοσελίδων θα κατεβάσει μόνο ενημερωμένους πόρους από τον εξυπηρετητή.
- **Web workers:** JavaScript το οποίο τρέχει στο παρασκήνιο (background), ανεξάρτητα από άλλα scripts, χωρίς να επηρεάζει την απόδοση της ιστοσελίδας.
- **Server-Sent Events (SSE):** ένα server-sent event συμβαίνει όταν η ιστοσελίδα ενημερώνεται αυτόματα από τον εξυπηρετητή.

Στη συνέχεια, παρατίθεται ένα παράδειγμα draggable αντικειμένου με χρήση του HTML5 drag and drop API. Όπως έχει ήδη αναφερθεί, το drag and drop είναι μέρος της προδιαγραφής της HTML5. Κάθε αντικείμενο μπορεί να «συρθεί» σε μια άλλη θέση στην ιστοσελίδα, αρκεί να τεθεί η ιδιότητα «draggable» του αντικειμένου σε «true». Για να γίνει αυτό, αρκεί η παρακάτω γραμμή κώδικα:

```
<img draggable="true">
```

Στη συνέχεια, πρέπει να προσδιοριστεί τι ακριβώς θα γίνεται όταν το συγκεκριμένο αντικείμενο «συρθεί». Αυτό γίνεται ως εξής:

```
function drag(ev) {  
    ev.dataTransfer.setData("text", ev.target.id);  
}
```

Δημιουργείται, δηλαδή, μια συνάρτηση, η `drag(event)`, η οποία προσδιορίζει ποια δεδομένα θα συρθούν. Στη συγκεκριμένη περίπτωση, ο τύπος δεδομένων είναι «text» και η τιμή των δεδομένων είναι το id του draggable element. Έπειτα, με το `ondragover` event προσδιορίζεται που ακριβώς μπορεί να αφηθεί το στοιχείο που έσυρε ο χρήστης. Από προεπιλογή, τα στοιχεία δε μπορούν να αφηθούν πάνω σε άλλα στοιχεία, οπότε καλείται η μέθοδος `event.preventDefault()` για αναίρεση της προκαθορισμένης (default) συμπεριφοράς.

Στη συνέχεια, όταν το στοιχείο αφηθεί από το χρήστη, προκύπτει ένα drop event, καλείται η συνάρτηση `drop(event)` και παίρνει το στοιχείο που σύρθηκε με τη μέθοδο `dataTransfer.getData()`. Η μέθοδος αυτή επιστρέφει δεδομένα του ίδιου τύπου με της μεθόδου `setData()`. Τα δεδομένα που σύρθηκαν είναι το id του dragged element («drag1»). Τέλος, γίνεται προσάρτηση του dragged element στο drop element.

```
function drop(ev) {
    ev.preventDefault();
    var data = ev.dataTransfer.getData("text");
    ev.target.appendChild(document.getElementById(data));
}

```

Ο κώδικας όλου του παραδείγματος είναι ακόλουθος:

```
<!DOCTYPE HTML>
<html>
<head>
<script>

function allowDrop(ev) {
    ev.preventDefault();
}

function drag(ev) {
    ev.dataTransfer.setData("text", ev.target.id);
}

function drop(ev) {
    ev.preventDefault();
    var data = ev.dataTransfer.getData("text");
    ev.target.appendChild(document.getElementById(data));
}

</script>
</head>
<body>

<div id="div1" ondrop="drop(event)" ondragover="allowDrop(event)"></div>



</body>
</html>

```

Τέλος, παρατίθενται κάποιοι χρήσιμοι σύνδεσμοι για HTML(5):

- W3C Recommendation (<http://www.w3.org/TR/html5/>): Η τελευταία ανανεωμένη προδιαγραφή της HTML.
- W3C schools CSS (<http://www.w3schools.com/html/>): Περιέχει και αναλύει τα χαρακτηριστικά που ορίζει η επίσημη αναφορά του W3C για την HTML. Δίνει τη δυνατότητα να δοκιμάσετε διαδραστικά τον κώδικά σας στον περιηγητή.
- HTML5 doctor (<http://html5doctor.com/>): Άρθρα και επίλυση ερωτήσεων σχετικά με την υλοποίηση των χαρακτηριστικών της HTML5.
- HTML5 rocks (<http://www.html5rocks.com/en/>): Εκπαιδευτικό υλικό (tutorials) και πηγές σχετικά με την HTML5.
- HTML5test (<https://html5test.com/>): Εμφανίζει αναλυτικά πόσα χαρακτηριστικά της HTML5 υποστηρίζει ο εκάστοτε περιηγητής.
- HTML boilerplate (<https://html5boilerplate.com/>): Δημοφιλές front-end template για τη δημιουργία ιστοσελίδων.

- Bootstrap (<http://getbootstrap.com/>): HTML, CSS και JS framework για τη δημιουργία websites και web applications.

7.2.2. Εισαγωγή στη δημιουργία ιστοσελίδων με PHP

7.2.2.1 Ας θυμηθούμε τι είναι η PHP;

Όπως προαναφέραμε η PHP είναι πιθανότατα η πιο διαδεδομένη γλώσσα script στον Παγκόσμιο Ιστό πληροφοριών από την πλευρά του εξυπηρετητή (server). Μπορεί να χρησιμοποιηθεί για τη δημιουργία ιστοσελίδων πρόσβασης σε έναν ιστότοπο (login, sign-in), για λήψη πληροφοριών από φόρμες, δημιουργία αλληλεπιδραστικών και συνεργατικών εφαρμογών, όπως forum, παρουσίαση πληροφοριών από βάσεις δεδομένων και πολλές ακόμη λειτουργίες. Ο κώδικας PHP εκτελείται από την πλευρά του εξυπηρετητή και δημιουργεί κώδικα HTML που παρουσιάζεται στον επισκέπτη μιας ιστοσελίδας.

7.2.2.2 Προαπαιτούμενες γνώσεις

Για την ευκολότερη κατανόηση των εννοιών που παρουσιάζονται σε αυτό το κεφάλαιο θα πρέπει να έχετε βασική γνώση των παρακάτω:

- HTML.
- CSS (προαιρετικά).
- JavaScript (προαιρετικά).

7.2.2.3 Τι χρειάζεται για να χρησιμοποιήσω την PHP;

Για να ξεκινήσετε να δουλεύετε σε PHP μπορείτε:

- Να βρείτε ένα πάροχο για φιλοξενία ιστοσελίδων (web hosting) που υποστηρίζει PHP και MySQL.
- Να εγκαταστήσετε έναν εξυπηρετητή Παγκόσμιου Ιστού Web Server στον υπολογιστή σας, καθώς και PHP και MySQL (βλ. Κεφάλαιο 5).

7.2.2.4 Βασική Σύνταξη PHP

Ένα πρόγραμμα (script) PHP εκτελείται στον server και η HTML που δημιουργείται αποστέλλεται στον browser. Ένα script της PHP πάντα ξεκινά (ανοίγει) με `<?php` και κλείνει με `?>` εναλλακτικά ένα script μπορεί να ανοίγει με `<?>` και να τελειώνει (κλείνει) με `?>`.

Το πιο κλασικό παράδειγμα σε οποιαδήποτε γλώσσα είναι η εμφάνιση της φράσης «Hello World» στην οθόνη του υπολογιστή.

Ας ξεκινήσουμε λοιπόν με αυτό!

Για την εμφάνιση του μηνύματος θα χρησιμοποιηθεί η εντολή **echo**. Το τέλος κάθε εντολής σε PHP υποδηλώνεται με το σύμβολο «;» (semicolon). Στην PHP όλες οι λέξεις κλειδιά (δεσμευμένες λέξεις) με τις οποίες συνάσσουμε τις εντολές του προγράμματος, δεν διαφέρουν για πεζούς ή κεφαλαίους χαρακτήρες, άρα η εντολή **echo** και η εντολή **ECHO** είναι ισοδύναμες.

Ας δούμε το παρακάτω παράδειγμα κώδικα για την εμφάνιση της φράσης «Hello World» στην οθόνη του υπολογιστή.

```
<!DOCTYPE html>
<html>
<body>

<h1>My first PHP page</h1>

<?php
echo "Hello World!";
?>

</body>
</html>
```



Εικόνα 7.20. Η έξοδος του παραπάνω παραδείγματος

7.2.2.5 Σχόλια

Ένα σχόλιο στην PHP είναι μια ή πολλές γραμμές που δεν εκτελούνται σαν μέρος του script. Ο μοναδικός σκοπός τους είναι για να διαβαστούν από κάποιον που διαβάζει τον κώδικα ώστε να καταλάβει ποιος είναι ο σκοπός του ή για να δρα ως υπενθύμιση όταν γράφετε το συγκεκριμένο κομμάτι κώδικα και ποια είναι η λειτουργία του.

Στο παρακάτω παράδειγμα κώδικα παρουσιάζονται διάφοροι τρόποι εισαγωγής σχολίων σε ένα script της PHP.

```
<?php

// Αυτό είναι ένα σχόλιο που καταλαμβάνει μια σειρά

# Αυτό είναι επίσης ένα σχόλιο που καταλαμβάνει μια σειρά

/*
Αυτό είναι ένα σχόλιο
που μπορεί να καταλαμβάνει
παραπάνω από μια σειρά
στο PHP script
*/
?>
```

7.2.2.6 Μεταβλητές και σταθερές

Για τη δημιουργία πιο σύνθετου κώδικα χρειάζεται ένας τρόπος λήψης και αποθήκευσης τιμών. Αυτό μπορεί να γίνει με την χρήση μεταβλητών. Μια μεταβλητή της PHP μπορεί να περιέχει μια συμβολοσειρά (string) ή έναν αριθμό. Παρακάτω παρουσιάζονται μερικοί κανόνες για τη φύση των μεταβλητών και την ονοματολογία τους.

Κανόνες για μεταβλητές της PHP

- Μια μεταβλητή πάντα ξεκινάει με το χαρακτήρα «\$» και ακολουθείται από το όνομα της μεταβλητής.
- Το όνομα μιας μεταβλητής πάντα ξεκινάει με ένα γράμμα ή το χαρακτήρα «_»(underscore/κάτω παύλα).
- Το όνομα μιας μεταβλητής δεν μπορεί να ξεκινάει με αριθμό.
- Μια μεταβλητή μπορεί να περιέχει στο όνομα της μόνο αλφαριθμητικούς χαρακτήρες και κάτω παύλες (underscores).
- Το όνομα μιας μεταβλητής περιέχει διάκριση πεζών-κεφαλαίων, δηλαδή η μεταβλητή \$car είναι διαφορετική από τη μεταβλητή \$CaR.
- Δεν χρειάζεται να δηλώσουμε μια μεταβλητή στην αρχή του script. Η δημιουργία και ο τύπος της γίνεται την ώρα της κλήσης της στον κώδικα

Οι μεταβλητές στην PHP μπορούν να αποθηκεύσουν διαφορετικούς τύπους τιμών. Παρακάτω παρουσιάζονται οι τύποι δεδομένων που υποστηρίζονται από την PHP.

- String.
- Integer.
- Float (floating point numbers - also called double).
- Boolean.
- Array.
- Object.
- NULL.
- Resource.

Για τις πράξεις μεταξύ μεταβλητών (αριθμητικών ή τύπου string) χρησιμοποιούμε τελεστές. Η PHP χωρίζει τους τελεστές στις παρακάτω κατηγορίες.

- Αριθμητικοί τελεστές.
- Τελεστές Ανάθεσης.
- Τελεστές Σύγκρισης.
- Τελεστές Αύξησης/Μείωσης.
- Τελεστές Λογικής.
- Τελεστές Αλφαριθμητικών.
- Τελεστές Πινάκων.

Πίνακας 7.4 Αριθμητικοί τελεστές

Τελεστής	Όνομα	Παράδειγμα	Αποτέλεσμα
+	Πρόσθεση	$\$x + \y	Άθροισμα $\$x$ και $\$y$
-	Αφαίρεση	$\$x - \y	Διαφορά $\$x$ και $\$y$
*	Πολλαπλασιασμός	$\$x * \y	Πολμός $\$x$ και $\$y$
/	Διαίρεση	$\$x / \y	Διαίρεση $\$x$ και $\$y$
%	Ακέραιο υπόλοιπο	$\$x \% \y	Υπόλοιπο διαίρεσης $\$x$ και $\$y$
**	Ύψωση σε δύναμη	$\$x ** \y	Το αποτέλεσμα του να υψώσουμε το $\$x$ στην $\$y$ -οστή δύναμη

Πίνακας 7.5. Τελεστές ανάθεσης

Τελεστής	Όμοιος με..	Αποτέλεσμα
$x=y$	$x=y$	Στον αριστερό τελεστή ανατίθεται η τιμή της έκφρασης στα δεξιά
$x+=y$	$x=x+y$	Πρόσθεση
$x-=y$	$x=x-y$	Αφαίρεση
$x*=y$	$x=x*y$	Πολλαπλασιασμός
$x/=y$	$x=x/y$	Διαίρεση
$x%=y$	$x=x\%y$	Ακέραιο υπόλοιπο

Πίνακας 7.6. Τελεστές Σύγκρισης

Τελεστής	Όνομα	Παράδειγμα	Αποτέλεσμα
==	Ισότητα	$\$x==\y	Αληθής όταν το \$x είναι ίσο με το \$y
===	Ταυτόσημα	$\$x===\y	Αληθής όταν το \$x είναι ίσο με το \$y και είναι και ίδιου τύπου
!=	Ανισότητα	$\$x!=\y	Αληθής όταν το \$x δεν είναι ίσο με το \$y
<>	Ανισότητα	$\$x<>\y	Αληθής όταν το \$x δεν είναι ίσο με το \$y
!==	Μη ταυτόσημα	$\$x!==\y	Αληθής όταν το \$x δεν είναι ίσο με το \$y ή δεν είναι ίδιου τύπου
>	Μεγαλύτερο	$\$x>\y	Αληθής όταν το \$x είναι μεγαλύτερο από το \$y
<	Μικρότερο	$\$x<\y	Αληθής όταν το \$x είναι μικρότερο από το \$y
>=	Μεγαλύτερο ή ίσο	$\$x>=\y	Αληθής όταν το \$x είναι μεγαλύτερο ή ίσο από το \$y
<=	Μικρότερο ή ίσο	$\$x<=\y	Αληθής όταν το \$x είναι μικρότερο ή ίσο από το \$y

Πίνακας 7.7. Τελεστές Αύξησης/Μείωσης

Τελεστής	Όνομα	Περιγραφή
++\$x	Αύξηση πριν	Αύξηση του \$x κατά ένα, επιστροφή του \$x
\$x++	Αύξηση μετά	Επιστροφή του \$x, αύξηση του \$x κατά ένα
--\$x	Μείωση πριν	Μείωση του \$x κατά ένα, επιστροφή του \$x
\$x--	Μείωση μετά	Επιστροφή του \$x, μείωση του \$x κατά ένα

Πίνακας 7.8. Λογικοί Τελεστές

Τελεστής	Όνομα	Παράδειγμα	Αποτέλεσμα
----------	-------	------------	------------

And	Και	$\$x$ and $\$y$	Αληθής όταν και το $\$x$ και το $\$y$ είναι αληθή
Or	Ή	$\$x$ or $\$y$	Αληθής όταν είτε το $\$x$ ή το $\$y$ είναι αληθή
Xor	Xor	$\$x$ xor $\$y$	Αληθής όταν είτε το $\$x$ ή το $\$y$ είναι αληθή, αλλά όχι μαζί
&&	Και	$\$x$ && $\$y$	Αληθής όταν και το $\$x$ και το $\$y$ είναι αληθή
	Ή	$\$x$ $\$y$	Αληθής όταν είτε το $\$x$ ή το $\$y$ είναι αληθή
!	Όχι	! $\$x$	Αληθής όταν το $\$x$ δεν είναι αληθές

Πίνακας 7.9. Τελεστές Αλφαριθμητικών

Τελεστής	Όνομα	Παράδειγμα	Αποτέλεσμα
.	Συνένωση	$\$txt1.\$txt2$	Συνένωση του $\$txt1$ με το $\$txt2$
.=	Προσάρτηση	$\$txt1.=\$txt2$	Προσάρτηση του $\$txt2$ στο $\$txt1$

Πίνακας 7.10. Τελεστές Πινάκων

Τελεστής	Όνομα	Παράδειγμα	Αποτέλεσμα
+	Ένωση	$\$x + \y	Ένωση $\$x$ και $\$y$
==	Ισότητα	$\$x==\y	Επιστρέφει αληθής όταν το $\$x$ και το $\$y$ έχουν τα ίδια ζευγάρια κλειδιών/τιμών
===	Ταυτόσημα	$\$x===\y	Επιστρέφει αληθής όταν το $\$x$ και το $\$y$ έχουν τα ίδια ζευγάρια κλειδιών/τιμών, με την ίδια σειρά και ίδιους τύπους
!=	Ανισότητα	$\$x!=\y	Επιστρέφει αληθής όταν το $\$x$ δεν είναι ίσο με το $\$y$

<>	Ανισότητα	$\$x \neq \y	Επιστρέφει αληθής όταν το \$x δεν είναι ίσο με το \$y
!==	Μη ταυτόσημα	$\$x !== \y	Επιστρέφει αληθής όταν το \$x δεν είναι ταυτόσημο με το \$y

Η εμβέλεια μιας μεταβλητής ορίζεται από το σημείο που δημιουργείται και ορίζει το μέρος του script που μπορεί αυτή η μεταβλητή να χρησιμοποιηθεί. Έτσι, αν μια μεταβλητή έχει δημιουργηθεί εκτός μιας συνάρτησης (function), η εμβέλειά της θεωρείται καθολική (global), οπότε δεν μπορεί να χρησιμοποιηθεί μέσα στη συνάρτηση και αντίστοιχα αν μια μεταβλητή δημιουργηθεί μέσα σε μια συνάρτηση, η εμβέλειά της θεωρείται τοπική (local), τότε αν χρησιμοποιηθεί έξω από αυτήν θα εμφανιστεί μήνυμα λάθους. Για να καλέσουμε την τιμή μιας καθολικής μεταβλητής μέσα σε μια συνάρτηση θα πρέπει πρώτα να προσθέσουμε κατά τη δημιουργία της τη λέξη κλειδί global.

Παρακάτω παρουσιάζονται μερικά παραδείγματα κώδικα που επιδεικνύουν την εμβέλεια μιας μεταβλητής.

```
<?php
$x = 5; // καθολική εμβέλεια

function Test1() {
// χρησιμοποιώντας τη μεταβλητή x εδώ θα οδηγήσει σε μήνυμα λάθους
    echo "<p>Variable x inside function is: $x</p>";
}
Test1();
// χρησιμοποιώντας τη μεταβλητή x εδώ θα λάβουμε την τιμή 5
echo "<p>Variable x outside function is: $x</p>";
?>
```

```
<?php
function Test2() {
    $x = 5; // τοπική εμβέλεια
    echo "<p>Variable x inside function is: $x</p>";
}
Test2();

// χρησιμοποιώντας τη μεταβλητή x εδώ θα οδηγήσει σε μήνυμα λάθους
echo "<p>Variable x outside function is: $x</p>";
?>
```

```
<?php
$x = 5;
$y = 10;

function Test3() {
    global $x, $y;
    $y = $x + $y;
}

Test3();
echo $y; // η έξοδος θα είναι 15
?>
```

Κανονικά όταν μια συνάρτηση έχει εκτελεστεί οι τοπικές μεταβλητές διαγράφονται. Υπάρχουν περιπτώσεις όμως που δεν είναι επιθυμητό μια μεταβλητή να διαγραφεί, τότε χρησιμοποιείται η λέξη κλειδί static.

Στο παρακάτω παράδειγμα παρουσιάζεται η χρήση της λέξης κλειδί static:

```
<?php
function static_example() {
    static $x = 0;
    echo $x;
    $x++;
}

static_example ();
static_example ();
static_example ();
//Η έξοδος θα είναι 0 , 1 , 2
?>
```

Οι σταθερές λειτουργούν παρόμοια με τις μεταβλητές με τη διαφορά ότι δεν μπορούν να αλλάξουν τιμή κατά την εκτέλεση του php script. Έτσι, μπορούμε να πούμε ότι μια σταθερά είναι ένα αναγνωριστικό για μια συγκεκριμένη τιμή. Οι σταθερές έχουν αυτόματα καθολική εμβέλεια από τη στιγμή που δημιουργούνται. Για τη δημιουργία μιας σταθεράς χρησιμοποιείται η συνάρτηση `define()` με την παρακάτω σύνταξη:

```
define(name, value, case-insensitive)
```

Όπου `name` το όνομα της σταθεράς, `value` η τιμή που μπορεί να είναι αριθμός ή και ένα `string` και τέλος η παράμετρος `case-insensitive` με προεπιλεγμένη τιμή `false` επιτρέπει σε μια σταθερά να έχει όνομα που δεν περιέχει διάκριση πεζών-κεφαλαίων.

Παρακάτω παρουσιάζεται ένα παράδειγμα δημιουργίας μιας σταθεράς τύπου `string` με όνομα «welcome» που περιέχει την πρόταση «welcome to PHP 101»

Παράδειγμα:

```
<?php
define("welcome ", "Welcome to PHP 101");
echo welcome;
?>
```

7.2.2.7 Συναρτήσεις για δεδομένα τύπου `string`

Σε αυτή την ενότητα παρουσιάζονται μερικές από τις πιο συνηθισμένες συναρτήσεις για την επεξεργασία δεδομένων τύπου `string`.

Η συνάρτηση `strlen()`;

Η συνάρτηση της PHP `strlen()`; επιστρέφει τον αριθμό χαρακτήρων μιας συμβολοσειράς ή μιας μεταβλητής που την εμπεριέχει.

Παράδειγμα:

```
<?php
echo strlen("Hello world!"); // Η έξοδος θα είναι 12
?>

<?php
$teststring = "Hello there!";
echo strlen($teststring); // Η έξοδος θα είναι 11
?>
```

Η συνάρτηση `str_word_count()`;

Η συνάρτηση `str_word_count()`; επιστρέφει τον αριθμό λέξεων μίας συμβολοσειράς ή μιας μεταβλητής που την εμπεριέχει.

Παράδειγμα:

```
<?php
echo str_word_count ("Hello world!"); // Η έξοδος θα είναι 2
?>

<?php
$teststring = "Hello there!";
echo str_word_count ($teststring); // Η έξοδος θα είναι 2
?>
```

Η συνάρτηση `strrev()`;

Η συνάρτηση `strrev()`; Αντιστρέφει μια συμβολοσειρά ή μια μεταβλητή που την εμπεριέχει.

Παράδειγμα:

```
<?php
echo strrev("Hello world!"); // Η έξοδος θα είναι !dlrow olleH
?>

<?php
$teststring = "Hello there!";
echo strrev($teststring); // Η έξοδος θα είναι !ereht olleH
?>
```

Η συνάρτηση `strpos()`;

Η συνάρτηση `strpos()`; ψάχνει για συγκεκριμένο κείμενο μέσα σε μια συμβολοσειρά και αν το βρει επιστρέφει τη θέση του χαρακτήρα όπου βρέθηκε το κείμενο για πρώτη φορά, αλλιώς επιστρέφει την τιμή `FALSE`.

Παράδειγμα:

```
<?php
echo strpos("Hello world!", "world"); // Η έξοδος θα είναι 6
?>

<?php
$teststring = "Hello there!";
$string_to_find = " there";
echo strpos($teststring, $string_to_find);
// Η έξοδος θα είναι 6
?>
```

Μεταβλητές τύπου `Array`

Μερικές φορές χρειάζεται να αποθηκευτούν παραπάνω από μια τιμές σε μια μεταβλητή. Σε αυτήν την περίπτωση πρέπει να χρησιμοποιηθεί ένας πίνακας με την χρήση μιας μεταβλητής τύπου `array`.

Για τη δήλωση μιας μεταβλητής τύπου `array` υπάρχουν δυο τρόποι:

```
<?php
$fruit = array("Apple", "Pear", "Banana");
?>

<?php
$fruit[0] = "Apple";
$fruit[1] = "Pear";
$fruit[2] = "Banana";
?>
```

Σε κάθε περίπτωση ο κώδικας θα δημιουργήσει μια μεταβλητή τύπου array με όνομα fruit που θα περιέχει τις τιμές «Apple», «Pear» και «Banana».

Για την εύρεση του αριθμού των τιμών σε ένα array μπορεί να χρησιμοποιηθεί η συνάρτηση count()
Παράδειγμα:

```
<?php
$fruit = array("Apple", "Pear", "Banana");
Echo count($fruit);
//Η έξοδος θα είναι 3
?>
```

Για την προσπέλαση (διάσχιση) ενός array και για την εκτύπωση των τιμών μπορεί να χρησιμοποιηθεί μια δομή επανάληψης.

```
<?php
$fruit = array("Apple", "Pear", "Banana");
$length_of_array = count($fruit);

for($x = 0; $x < $length_of_array; $x++) {
    echo $fruit[$x];
    echo "<br>";
}
?>
```

7.2.2.8 Δομές επιλογής/επανάληψης στην PHP

Δομές Επανάληψης

Μερικές φορές χρειάζεται να εκτελεστεί μια διαδικασία με τον ίδιο τρόπο πολλές φορές. Σε αυτήν την περίπτωση μπορεί να χρησιμοποιηθεί μια από τις δομές επανάληψης για την αυτοματοποίηση μέρους ή ακόμα και ολόκληρης τη διαδικασίας.

Ο βρόχος *do...while*

Ο βρόχος *do...while* πάντα εκτελεί μια φορά το μπλοκ κώδικα που εμπεριέχει και μετά ελέγχει τη συνθήκη που τέθηκε. Αν η συνθήκη είναι αληθής τότε ο κώδικας που εμπεριέχεται θα συνεχίσει να εκτελείται μέχρι η συνθήκη να είναι ψευδής. Παράδειγμα:

```
<?php
$x = 1;

do {
    echo $x;
    $x++;
} while ($x <= 5);
// Η έξοδος θα είναι 1, 2, 3, 4, 5
?>
```

Ο βρόχος *for*

Ο βρόχος *for* χρησιμοποιείται όταν είναι γνωστό εκ των προτέρων το πόσες φορές θα πρέπει να εκτελεστεί ένα κομμάτι κώδικα.

Παράδειγμα:

```
<?php
for ($x = 0; $x <= 10; $x++) {
    echo "The number is: $x <br>";
    // Η έξοδος θα είναι από το 0 μέχρι και το 10
}
?>
```

Επίσης, μπορεί ο αριθμός στην παραπάνω συνθήκη να αντικατασταθεί με μια μεταβλητή στην οποία ανατίθεται τιμή σε άλλο μέρος του κώδικα και έτσι να μεταβληθεί δυναμικά ο αριθμός επαναλήψεων στο βρόχο.

Ο βρόχος foreach

Ο βρόχος foreach χρησιμοποιείται μόνο σε μεταβλητές τύπου array και χρησιμοποιείται για να προσελάσει ζευγάρια θέσης/τιμής. Στο παρακάτω παράδειγμα επιδεικνύεται πως με έναν βρόχο foreach εκτυπώνονται οι τιμές της μεταβλητής τύπου array «fruit».

Παράδειγμα:

```
<?php
$fruit = array("Apple", "Pear", "Banana");

foreach ($fruit as $value) {
    echo "$value <br>";
    // Η έξοδος θα είναι Apple, Pear, Banana
}
?>
```

Δομές επιλογής

Οι δομές επιλογής/ελέγχου επιτρέπουν την αλλαγή της ροής εκτέλεσης του κώδικα θέτοντας συγκεκριμένες συνθήκες και ανάλογα με το αν οι συνθήκες αυτές είναι αληθείς ή όχι.

Η δομή ελέγχου if και οι παραλλαγές της

Η δομή ελέγχου if στην πιο απλή μορφή της χρησιμοποιείται για τον έλεγχο αν ένα τμήμα κώδικα θα εκτελεστεί σύμφωνα με μια συνθήκη. Εναλλακτικά με την παραλλαγή if...else ορίζεται μια συνθήκη και τα τμήματα κώδικα που θα εκτελεστούν ανάλογα με το αν αυτή η συνθήκη είναι αληθής ή ψευδής. Η τελευταία παραλλαγή είναι η if...elseif...else, που χρησιμοποιείται όταν πέρα από την πρώτη συνθήκη πρέπει να ελεγχθεί άλλη μια (ή και παραπάνω, καθώς δεν υπάρχει περιορισμός στο πόσες συνθήκες elseif μπορούν να παρεμβληθούν) συνθήκη πριν την εκτέλεση της else που σημαίνει ότι όλες οι προηγούμενες συνθήκες ήταν ψευδείς. Παρακάτω παρατίθενται παραδείγματα για κάθε μια από τις παραλλαγές της δομής επιλογής if.

Παράδειγμα:

```

<?php
$time_hour = date("H");

if ($time_hour < "20")
{
    echo "καλημέρα!";
/* Το κομμάτι κώδικα θα εκτελεστεί μόνο αν η ώρα του υπολογιστή είναι πριν τις 8 το
βράδυ */
}
?>

<?php
$ time_hour = date("H");

if ($time_hour < "20")
{
    echo "Καλημέρα!";
/* Το κομμάτι κώδικα θα εκτελεστεί μόνο αν η ώρα του υπολογιστή είναι πριν τις 8 το
βράδυ */
} else
{
    echo "Καληνύχτα!";
/* Το κομμάτι κώδικα θα εκτελεστεί αν η ώρα του υπολογιστή είναι μετά τις 8 το βράδυ */
}
?>

<?php
$ time_hour = date("H");

if ($time_hour < "10")
{
    echo "Καλημέρα!";
/* Το κομμάτι κώδικα θα εκτελεστεί μόνο αν η ώρα του υπολογιστή είναι πριν τις 10 το
πρωί */
} elseif ($time_hour < "20")
{
    echo "Καλό μεσημέρι/απόγευμα!";
/* Το κομμάτι κώδικα θα εκτελεστεί μόνο αν η ώρα του υπολογιστή είναι πριν τις 8 το
βράδυ και η πρώτη συνθήκη είναι αναληθής άρα είναι μετά τις 10 το πρωί */
} else
{
    echo "Καληνύχτα!";
/* Το κομμάτι κώδικα θα εκτελεστεί αν η ώρα του υπολογιστή είναι μετά τις 8 το βράδυ */
}
?>

```

Η δομή επιλογής switch

Η δομή επιλογής switch χρησιμοποιείται όταν χρειάζεται να εκτελεστούν διαφορετικά τμήματα κώδικα ανάλογα με το αποτέλεσμα μιας συνθήκης (παρόμοια αλλά όχι ίδια με μια δομή if...elseif...else).

Στην πράξη υπάρχει μια συνθήκη (συνήθως μια μεταβλητή) η οποία αφού επιστρέφει μια τιμή, συγκρίνεται με τις τιμές για κάθε περίπτωση (case). Αν βρεθεί κάποια ισότητα τότε εκτελείται το αντίστοιχο τμήμα κώδικα και η δομή σταματάει να εκτελείται χωρίς να προχωρήσει στην επόμενη σύγκριση (αυτή είναι η διαφορά με τη δομή if...elseif...else). Σε περίπτωση που καμία από τις περιπτώσεις δεν ταιριάζει τότε εκτελείται το τμήμα κώδικα που αντιστοιχεί στην περίπτωση default. Στο παρακάτω παράδειγμα παρουσιάζεται η σύνταξη της δομής switch.

Παράδειγμα:


```

<?php
$best_fruit = "Apple";

switch ($best_fruit) {
    case "Apple":
        echo "Το καλύτερο φρούτο είναι το μήλο!";
        break;
    case "Pear":
        echo "Το καλύτερο φρούτο είναι το αχλάδι!";
        break;
    case "Banana":
        echo "Το καλύτερο φρούτο είναι η μπανάνα!";
        break;
    default:
        echo "Το καλύτερο φρούτο δεν είναι ούτε το μήλο ούτε το αχλάδι ούτε και η μπανάνα!";
}
?>

```

7.2.2.9 Οι δηλώσεις include και require

Οι δηλώσεις include και require αντιγράφουν όλο τον κώδικα που υπάρχει σε ένα συγκεκριμένο αρχείο και τον παραθέτουν στο αρχείο που χρησιμοποιεί τη δήλωση. Αυτό είναι πολύ χρήσιμο στην περίπτωση που χρειάζεται σε πολλαπλές σελίδες ενός ιστοτόπου να υπάρχει το ίδιο κομμάτι κώδικα.

Η διαφορά μεταξύ της include και της require είναι ότι όταν ένα αρχείο που δηλώνεται σε μια δήλωση include δεν μπορεί να εντοπιστεί από την PHP, ο κώδικας που ακολουθεί θα εκτελεστεί, ενώ στην περίπτωση της require με το που εκτελεστεί η εντολή και το αρχείο δεν εντοπιστεί, η εκτέλεση θα σταματήσει και θα εμφανιστεί μήνυμα λάθους.

Κανόνες για τις δηλώσεις include και require:

- Όταν ένα αρχείο είναι σημαντικό και πρέπει οπωσδήποτε να υπάρχει για την ομαλή λειτουργία του script χρησιμοποιείται η δήλωση require.
- Όταν ένα αρχείο δεν είναι σημαντικό και η συνέχεια του script δεν επηρεάζεται από την απουσία του χρησιμοποιείται η δήλωση include.

Παράδειγμα:

Υπάρχει ένα αρχείο με το μενού ενός ιστοτόπου με όνομα menu.php:

```

<?php
echo '<a href="/default.asp">Home</a> -
<a href="/html/default.asp">HTML Tutorial</a> -
<a href="/css/default.asp">CSS Tutorial</a> -
<a href="default.asp">PHP Tutorial</a>';
?>

```

Αντί σε κάθε ξεχωριστή ιστοσελίδα να πρέπει να γραφεί ο ίδιος κώδικας, η χρήση της δήλωσης require θα κάνει πιο εύκολη και την τροποποίηση του χωρίς να χρειάζεται να γίνουν αλλαγές σε κάθε μια ιστοσελίδα ξεχωριστά:

```

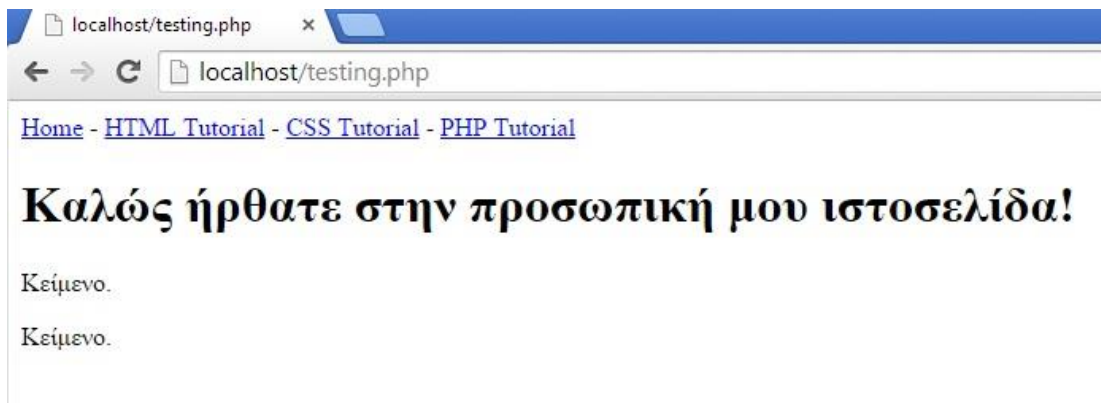
<html>
<body>

<div class="menu">
<?php require 'menu.php';?>
</div>

<h1>Καλώς ήρθατε στην προσωπική μου ιστοσελίδα!</h1>
<p>Κείμενο.</p>
<p>Κείμενο.</p>

</body>
</html>

```



Εικόνα 7.21. Η έξοδος του παραπάνω παραδείγματος

7.2.2.10 Sessions

Τα sessions (σύνοδοι) είναι ένας τρόπος για να αποθηκεύονται προσωρινά μεταβλητές και δεδομένα για την αξιοποίηση τους σε πολλαπλές σελίδες ενός ιστοτόπου. Τα δεδομένα αυτά από τη στιγμή που θα δημιουργηθούν αποθηκεύονται προσωρινά στη μνήμη του υπολογιστή μέχρι ο χρήστης να κλείσει τον περιηγητή ιστοσελίδων ή ο προγραμματιστής να δηλώσει το κλείσιμο του session σε κάποιο σημείο του κώδικα (π.χ. κατά την αποσύνδεση του χρήστη, log out από ένα ιστότοπο).

Ένα session ξεκινάει με τη συνάρτηση `session_start()` και οι μεταβλητές υποδηλώνονται με τη μορφή `$_SESSION["variable_name"]`.

Παράδειγμα:

Δημιουργία μιας ιστοσελίδας με όνομα «`session_test1.php`»

```
<?php
// Εκκίνηση του session
session_start();
?>
<!DOCTYPE html>
<html>
<body>

<?php
// Δήλωση μεταβλητών
$_SESSION["name"] = "James";
$_SESSION["occupation"] = "Programmer";
echo "Session variables are set.";
?>

</body>
</html>
```

Στην παραπάνω ιστοσελίδα έχει ξεκινήσει ένα session και έχουν δηλωθεί δυο μεταβλητές (όνομα και επάγγελμα). Στην επόμενη ιστοσελίδα με όνομα «`session_test2.php`» χρησιμοποιείται η εντολή `echo` για την εμφάνιση των μεταβλητών που δηλώθηκαν στην `session_test1`.

```

<?php
session_start();
?>
<!DOCTYPE html>
<html>
<body>
<?php
/* Εμφάνιση των μεταβλητών του session που δηλώθηκαν στην προηγούμενη ιστοσελίδα */
echo "Your name is " . $_SESSION["name"] . "<br>";
echo "Your occupation is " . $_SESSION["occupation"] . ".";
?>

</body>
</html>

```

Η συνάρτηση `session_start()` πρέπει πάντα να δηλώνεται στην αρχή του εγγράφου πριν από τα HTML tags. Τέλος, για τη διαγραφή του session και των δεδομένων που εμπεριέχονται σε αυτό χρησιμοποιούνται οι συναρτήσεις `session_unset()` `session_destroy()` όπως στον κώδικα παρακάτω:

```

<?php
// Εκκίνηση του session
session_start();
?>
<!DOCTYPE html>
<html>
<body>

<?php
// Διαγραφή των μεταβλητών
session_unset();

// Καταστροφή του session
session_destroy();
?>

</body>
</html>

```

7.2.2.11 Φόρμες και επικύρωση δεδομένων

Οι φόρμες είναι ο τρόπος με τον οποίο γίνεται η συλλογή δεδομένων από το χρήστη στην PHP. Μπορεί να είναι ένα απλό πλαίσιο κειμένου, μια σειρά από πλαίσια ελέγχου (checkboxes), κουμπιά επιλογής (radio buttons) και άλλα.

Σε ένα απλό παράδειγμα φόρμας ο χρήστης εισάγει το όνομα του σε ένα πλαίσιο κειμένου και αφού επιλέξει το κουμπί «submit» τα δεδομένα αποστέλλονται με τη μέθοδο HTTP POST σε ένα άλλο αρχείο «sentdatatest.php» για επεξεργασία ή στη συγκεκριμένη περίπτωση απλή εμφάνιση.

Παράδειγμα:

Ο κώδικας του αρχείου με τη φόρμα έχει ως εξής:

```

<html>
<body>

<form action=" sentdatatest.php" method="post">
Name: <input type="text" name="name"><br>
<input type="submit">
</form>

</body>
</html>

```

Και ο κώδικας του αρχείου «sentdatatest.php» θα είναι:

```
<html>
<body>

Καλώς ήρθατε <br>
Το όνομα σας είναι : <?php echo $_POST["name"]; ?><br>

</body>
</html>
```

Υπάρχει άλλη μια μέθοδος με την οποία αποστέλλονται τα δεδομένα μεταξύ ιστοσελίδων που έχει σύνταξη ίδια με την `$_POST` αλλά κάποιες πολύ σημαντικές διαφορές. Η μέθοδος `$_GET` μεταφέρει δεδομένα σαν παραμέτρους του URL σε αντίθεση με την `$_POST` που τα μεταφέρει μέσω της μεθόδου HTTP POST. Δεδομένα που έχουν σταλεί με τη μέθοδο `$_GET` είναι ορατά σε όλους, καθώς εμφανίζονται στο URL αλλά αυτό είναι θεμιτό σε μερικές περιπτώσεις που τα δεδομένα δεν είναι ευαίσθητα και υπάρχει περίπτωση να χρειάζεται η ιστοσελίδα με τα συγκεκριμένα δεδομένα να μπει σαν σελιδοδείκτης (π.χ. bookmark σε συγκεκριμένο αντικείμενο ενός e-shop). Επίσης, η μέθοδος `$_GET` έχει όριο 2000 χαρακτήρων για τις μεταβλητές/δεδομένα που αποστέλλονται μέσω αυτής. Η μέθοδος `$_POST` δεν έχει όριο χαρακτήρων και τα δεδομένα που αποστέλλονται μέσω αυτής είναι αόρατα προς άλλους οπότε είναι ιδανική για ευαίσθητα δεδομένα όπως κωδικί.

Ένα ακόμα σημαντικό κομμάτι όσον αφορά την ασφάλεια στις φόρμες και τα δεδομένα είναι η επικύρωση τους. Όταν υπάρχει η ανάγκη για υποχρεωτικά πεδία σε φόρμες είναι χρήσιμη η καθολική μεταβλητή `$_SERVER["PHP_SELF"]` η οποία αποστέλλει τα δεδομένα στην ιστοσελίδα που έγινε η υποβολή τους, επιτρέποντας τον έλεγχο για λάθη και για δεδομένα που ήταν υποχρεωτικά αλλά λείπουν. Η `$_SERVER["PHP_SELF"]` όμως από μόνη της αφήνει ένα σημαντικό κενό ασφάλειας στη μορφή του cross site scripting (XSS), γι' αυτό είναι σημαντικό τα δεδομένα να περνάνε πρώτα από τη συνάρτηση `htmlspecialchars()` η οποία μετατρέπει ειδικούς χαρακτήρες που χρησιμοποιούνται σε επιθέσεις XSS σε ονόματα HTML και τους καθιστά άχρηστους.

Παράδειγμα:

Έστω ότι υπάρχει η ιστοσελίδα «test.php» με την ακόλουθη φόρμα:

```
<form method="post" action="<?php echo $_SERVER["PHP_SELF"]; ?>">
```

Αν ένας χρήστης εισάγει στο URL την ιστοσελίδα «www.webserver.com/test.php» ο παραπάνω κώδικας θα μεταφραστεί σε:

```
<form method="post" action="test.php">
```

Έστω όμως ότι ο χρήστης εισάγει στο URL τα επόμενα:

```
http://www.webserver.com/test.php/%22%3E%3Cscript%3E%3Emalicious_script%3C/script%3E
```

Στην παραπάνω περίπτωση λόγω της `$_SERVER["PHP_SELF"]` ο κώδικας θα μεταφραστεί σε:

```
<form method="post" action="test_form.php/"><script>malicious_script</script>
```

Όπου `malicious_script`, μπορεί να είναι κώδικας βλαβερός προς άλλους χρήστες. Αν όμως χρησιμοποιηθεί η `htmlspecialchars()` όπως παρακάτω:

```
<form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">
```

Τότε ο κώδικας του URL μεταφράζεται σε:

```
<form method="post" action="test_form.php/&quot;&gt;&lt;script&gt;(malicious_script)&lt;/script&gt;">
```

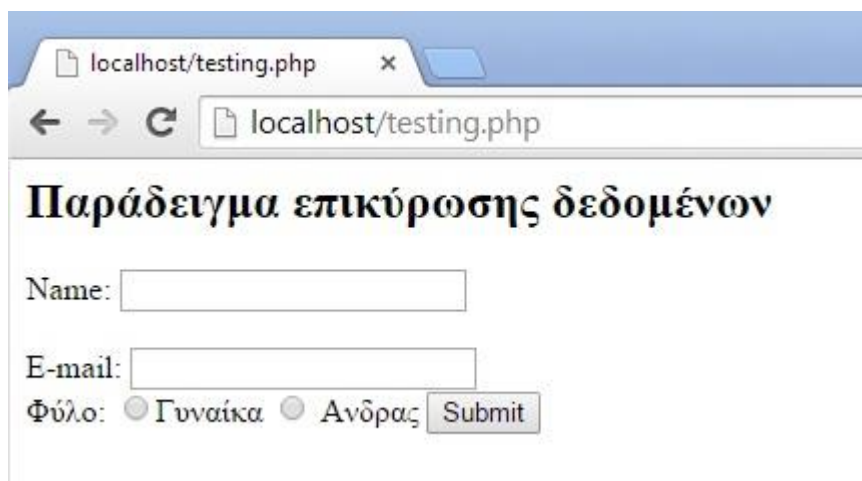
που δεν έχει ιδιαίτερο νόημα και η επίθεση αποτυγχάνει.

Στο παρακάτω παράδειγμα αναδεικνύεται ο τρόπος με τον οποίο λειτουργεί η επικύρωση δεδομένων σε μια φόρμα με τρία στοιχεία, δυο πλαίσια κειμένου για όνομα και e-mail και μια επιλογή από 2 κουμπιά επιλογής για φύλο.

Παράδειγμα:

Η φόρμα στην απλή της μορφή θα έχει ως εξής:

```
<h2>Παράδειγμα επικύρωσης δεδομένων</h2>
<form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">
  Name: <input type="text" name="name" value">
  <br><br>
  E-mail: <input type="text" name="email" value">
  Φύλο:
  <input type="radio" name="gender" value="female">Γυναίκα
  <input type="radio" name="gender" value="male">Ανδρας
  <input type="submit" name="submit" value="Submit">
</form>
```



Εικόνα 7.22. Η έξοδος του παραπάνω παραδείγματος

Το επόμενο βήμα είναι η δημιουργία μιας συνάρτησης που θα λαμβάνει δεδομένα και χρησιμοποιεί τη συνάρτηση:

- htmlspecialchars() για να μειώσει το ρίσκο επίθεσης μέσω δεδομένων.
- stripslashes() για την αφαίρεση ανάστροφων καθέτων από τα δεδομένα.
- trim() για την αφαίρεση περιττών χαρακτήρων (ειδικούς χαρακτήρες, στηλοθέτες(tab) κ.α.).

Η συνάρτηση θα έχει ως εξής:

```

<?php
// Θέτουμε τις μεταβλητές σε κενή κατάσταση
$name = $email = $gender = "";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $name = test_input($_POST["name"]);
    $email = test_input($_POST["email"]);
    $gender = test_input($_POST["gender"]);
}

function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}
?>

```

Έτσι, τώρα κάθε φορά που ο χρήστης πατάει το κουμπί submit τα δεδομένα πρώτα περνάνε από τον έλεγχο της συνάρτησης test_input().

Επίσης, το `$_SERVER["REQUEST_METHOD"] == "POST"` ενεργοποιείται αν ο χρήστης πατήσει το κουμπί submit αλλιώς το κομμάτι κώδικα αγνοείται και εμφανίζεται η κενή φόρμα (π.χ. την πρώτη φορά που ο χρήστης θα εισέλθει στην ιστοσελίδα).

Το επόμενο βήμα είναι η μετατροπή των πεδίων που θεωρούνται σημαντικά σε υποχρεωτικά και η δημιουργία μηνυμάτων λάθους στην περίπτωση που ο χρήστης δεν εισάγει δεδομένα ή εισάγει λάθος δεδομένα σε ένα υποχρεωτικό πεδίο.

Στον παρακάτω κώδικα εισάγονται μερικές καινούριες μεταβλητές (`$nameErr`, `$emailErr` και `$genderErr`) που θα περιέχουν τα μηνύματα λάθους, και κανόνες για την περίπτωση που ο χρήστης αφήσει ένα πεδίο που είναι υποχρεωτικό κενό:

```

<?php
// Θέτουμε τις μεταβλητές σε κενή κατάσταση
$nameErr = $emailErr = $genderErr = "";
$name = $email = $gender = "";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (empty($_POST["name"])) {
        $nameErr = "Name is required";
    } else {
        $name = test_input($_POST["name"]);
    }

    if (empty($_POST["email"])) {
        $emailErr = "Email is required";
    } else {
        $email = test_input($_POST["email"]);
    }

    if (empty($_POST["gender"])) {
        $genderErr = "Gender is required";
    } else {
        $gender = test_input($_POST["gender"]);
    }
}
?>

```

Επιπλέον, στο κομμάτι κώδικα της φόρμας προστίθενται τμήματα PHP κώδικα για την εμφάνιση του μηνύματος λάθους αν ο χρήστης δεν εισάγει δεδομένα:

```

<form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">
Name: <input type="text" name="name">
<span class="error">* <?php echo $nameErr;?></span>
<br><br>
E-mail:
<input type="text" name="email">
<span class="error">* <?php echo $emailErr;?></span>
<br><br>
Gender:
<input type="radio" name="gender" value="female">Female
<input type="radio" name="gender" value="male">Male
<span class="error">* <?php echo $genderErr;?></span>
<br><br>
<input type="submit" name="submit" value="Submit">
</form>

```

Στο επόμενο βήμα πρέπει να ελεγχθεί αν το όνομα και το e-mail είναι σωστά. Το όνομα θα πρέπει να περιέχει μόνο χαρακτήρες και κενά και το e-mail να έχει σωστή μορφή. Για το όνομα θα χρησιμοποιηθεί μια κανονική έκφραση και η συνάρτηση της PHP «preg_match()» για τον έλεγχο των χαρακτήρων και για το e-mail η συνάρτηση της PHP «filter_var()».

Άρα ο κώδικας ελέγχου θα πάρει την εξής μορφή:

```

<?php
// Θέτουμε τις μεταβλητές σε κενή κατάσταση
$nameErr = $emailErr = $genderErr = "";
$name = $email = $gender = "";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (empty($_POST["name"])) {
        $nameErr = "Name is required";
    } else {
        $name = test_input($_POST["name"]);
        // ελέγχουμε το όνομα να περιέχει μόνο χαρακτήρες και κενά
        if (!preg_match("/^[a-zA-Z ]*$/", $name)) {
            $nameErr = "Only letters and white space allowed";
        }
    }

    if (empty($_POST["email"])) {
        $emailErr = "Email is required";
    } else {
        $email = test_input($_POST["email"]);
        // Ελέγχουμε αν το email έχει σωστή μορφή
        if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
            $emailErr = "Invalid email format";
        }
    }

    if (empty($_POST["gender"])) {
        $genderErr = "Gender is required";
    } else {
        $gender = test_input($_POST["gender"]);
    }
}
?>

```

Τέλος, θα ήταν καλό, όταν ο χρήστης επιλέξει το κουμπί submit, τα δεδομένα να παραμένουν στις φόρμες για την περίπτωση που κάνει λάθος, έτσι ώστε να μην χρειάζεται να τα ξαναγράφει από την αρχή. Αυτό επιτυγχάνεται με ένα μικρό τμήμα κώδικα PHP μέσα σε κάθε φόρμα.

Από κάτω παρουσιάζεται ο πλήρης κώδικας της φόρμας με έλεγχο και έξοδο του τι έχει γράψει ο χρήστης:

```

<!DOCTYPE HTML>
<html>
<head>
<style>
.error {color: #FF0000;}
</style>
</head>
<body>

<?php
// Θέτουμε τις μεταβλητές σε κενή κατάσταση
$nameErr = $emailErr = $genderErr = "";
$name = $email = $gender = "";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (empty($_POST["name"])) {
        $nameErr = "Name is required";
    } else {
        $name = test_input($_POST["name"]);
        // ελέγχουμε το όνομα να περιέχει μόνο χαρακτήρες και κενά
        if (!preg_match("/^[a-zA-Z ]*$/", $name)) {
            $nameErr = "Only letters and white space allowed";
        }
    }

    if (empty($_POST["email"])) {
        $emailErr = "Email is required";
    } else {
        $email = test_input($_POST["email"]);
        // Ελέγχουμε αν το email έχει σωστή μορφή
        if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
            $emailErr = "Invalid email format";
        }
    }

    if (empty($_POST["gender"])) {
        $genderErr = "Gender is required";
    } else {
        $gender = test_input($_POST["gender"]);
    }
}

function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}
?>

<h2> Παράδειγμα επικύρωσης δεδομένων </h2>
<p><span class="error">* required field.</span></p>
<form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">
    Name: <input type="text" name="name" value="<?php echo $name;?>">
    <span class="error">* <?php echo $nameErr;?></span>
    <br><br>
    E-mail: <input type="text" name="email" value="<?php echo $email;?>">
    <span class="error">* <?php echo $emailErr;?></span>
    <br><br>
    Gender:
    <input type="radio" name="gender" <?php if (isset($gender) && $gender=="female") echo
"checked";?> value="female">Female
    <input type="radio" name="gender" <?php if (isset($gender) && $gender=="male") echo
"checked";?> value="male">Male
    <span class="error">* <?php echo $genderErr;?></span>
    <br><br>
    <input type="submit" name="submit" value="Submit">
</form>

```



```
<?php
echo "<h2>Τα δεδομένα που εισήγατε είναι :</h2>";
echo $name;
echo "<br>";
echo $email;
echo "<br>";
echo $gender;
?>

</body>
</html>
```

7.2.2.12 Μεταφόρτωση αρχείων στην PHP

Ένα ακόμα είδος φόρμας που έχει όμως αρκετή διαφορά με τα είδη που παρουσιάστηκαν παραπάνω είναι η φόρμα μεταφόρτωσης (uploading) αρχείων και ο κώδικας που είναι απαραίτητος για τη διαχείρισή/αποθήκευσή τους.

Στο παρακάτω απόσπασμα κώδικα παρουσιάζεται το html τμήμα της φόρμας μεταφόρτωσης αρχείων (στο συγκεκριμένο παράδειγμα μόνο εικόνες):

```
<!DOCTYPE html>
<html>
<body>

<form action="upload.php" method="post" enctype="multipart/form-data">
  Select image to upload:
  <input type="file" name="fileToUpload" id="fileToUpload">
  <input type="submit" value="Upload Image" name="submit">
</form>

</body>
</html>
```

Σημαντικό είναι να σημειωθεί ότι αν δεν τεθεί το method= "post" και το enctype= "multipart/form-data" η μεταφόρτωση δεν λειτουργεί. Παρακάτω παρουσιάζεται το αρχείο «upload.php» που αναλαμβάνει τη μεταφόρτωση του αρχείου που επιλέχθηκε από την παραπάνω φόρμα:

```

<?php
$target_dir = "uploads/";
$target_file = $target_dir . basename($_FILES["fileToUpload"]["name"]);
$uploadOk = 1;
$imageFileType = pathinfo($target_file,PATHINFO_EXTENSION);
// Check if image file is a actual image or fake image
if(isset($_POST["submit"])) {
    $check = getimagesize($_FILES["fileToUpload"]["tmp_name"]);
    if($check !== false) {
        echo "File is an image - " . $check["mime"] . ".";
        $uploadOk = 1;
    } else {
        echo "File is not an image.";
        $uploadOk = 0;
    }
}
// Check if file already exists
if (file_exists($target_file)) {
    echo "Sorry, file already exists.";
    $uploadOk = 0;
}
// Check file size
if ($_FILES["fileToUpload"]["size"] > 5000000) {
    echo "Sorry, your file is too large.";
    $uploadOk = 0;
}
// Allow certain file formats
if($imageFileType != "jpg" && $imageFileType != "png" && $imageFileType != "jpeg"
&& $imageFileType != "gif") {
    echo "Sorry, only JPG, JPEG, PNG & GIF files are allowed.";
    $uploadOk = 0;
}
// Check if $uploadOk is set to 0 by an error
if ($uploadOk == 0) {
    echo "Sorry, your file was not uploaded.";
// if everything is ok, try to upload file
} else {
    if (move_uploaded_file($_FILES["fileToUpload"]["tmp_name"], $target_file)) {
        echo "The file " . basename($_FILES["fileToUpload"]["name"]) . " has been up-
loaded.";
    } else {
        echo "Sorry, there was an error uploading your file.";
    }
}
?>

```

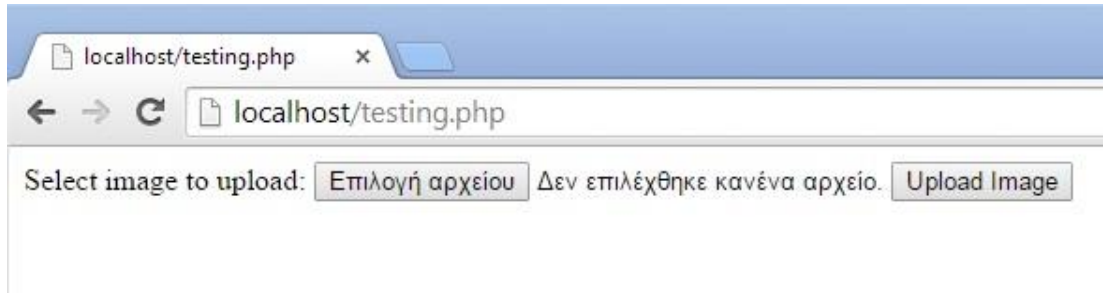
Η μεταβλητή \$target_dir καθορίζει το φάκελο στον οποίο θα αποθηκευτεί το αρχείο. Ο φάκελος αυτός θα πρέπει, στη συγκεκριμένη περίπτωση, να είναι στο ίδιο μέρος που βρίσκεται το αρχείο «upload.php» και να έχει δημιουργηθεί από διαχειριστή γιατί η PHP δεν έχει δικαιώματα δημιουργίας φακέλων για λόγους ασφάλειας.

Η μεταβλητή \$target_file περιέχει τη διαδρομή και το όνομα του αρχείου προς μεταφόρτωση. Η μεταβλητή \$imageFileType περιέχει την επέκταση του αρχείου προς μεταφόρτωση και χρησιμοποιείται για να καθορίσουμε ότι το αρχείο είναι στην πραγματικότητα αρχείο εικόνας. Η συνθήκη file_exists(\$target_file) ελέγχει αν υπάρχει αρχείο με ίδιο όνομα στο φάκελο με τις μεταφορτώσεις. Αν υπάρχει εμφανίζει μήνυμα λάθους. Η συνθήκη (\$_FILES["fileToUpload"]["size"] > 5000000) ελέγχει για το αν το μέγεθος του αρχείου είναι μεγαλύτερο από περίπου 5Mb και αν είναι εμφανίζει μήνυμα λάθους.

Η επόμενη συνθήκη μας επιτρέπει να θέσουμε συγκεκριμένους τύπους αρχείων εικόνας που είναι επιτρεπτοί για μεταφόρτωση και αν η εικόνα που μεταφορτώνεται δεν ανήκει σε κάποιον από αυτούς τότε εμφανίζεται μήνυμα λάθους. Η τελευταία συνθήκη περιέχει τη μεταβλητή \$uploadOk που χρησιμοποιείται σαν σημάι για το αν συνέβη κάποιο πρόβλημα σε όλους τους προηγούμενους ελέγχους. Τέλος, αφού ξεπεράστηκαν όλοι οι έλεγχοι η παρακάτω εντολή:

```
move_uploaded_file($_FILES["fileToUpload"]["tmp_name"], $target_file
```

μεταφέρει το αρχείο στον προκαθορισμένο φάκελο. Πάλι, αν υπάρξει κάποιο πρόβλημα (π.χ. δεν υπάρχει αρκετός χώρος στο σκληρό δίσκο του server) εμφανίζεται μήνυμα λάθους, αλλιώς εμφανίζεται μήνυμα επιτυχούς μεταφόρτωσης.



Εικόνα 7.23. Η έξοδος του παραπάνω παραδείγματος

7.2.3. Εισαγωγή στο CSS

Στο κεφάλαιο αυτό θα παρουσιαστεί η γλώσσα CSS (Cascading Style Sheets) που επιτρέπει τον ακριβή ορισμό του τρόπου εμφάνισης των στοιχείων που περιλαμβάνει μία ιστοσελίδα, η οποία έχει αναπτυχθεί με χρήση HTML. Η χρήση της γλώσσας CSS προσφέρει μεγάλη ευελιξία και πολλά άλλα πλεονεκτήματα έναντι της μορφοποίησης μέσω HTML ιδιοτήτων (attributes). Το συγκεκριμένο κεφάλαιο θα εστιάσει αρχικά στη δομή των βασικών κανόνων σύνταξης της CSS, καθώς και στον τρόπο χρήσης των κανόνων αυτών για τη μορφοποίηση μιας ιστοσελίδας. Στη συνέχεια θα παρουσιαστούν οι συγκεκριμένοι κανόνες και οι διάφοροι τρόποι με τους οποίους μπορούν να μορφοποιηθούν τα επιμέρους στοιχεία αλλά και η γενικότερη διάταξη μιας HTML σελίδας. Τέλος, θα γίνει αναφορά σε βασικές ιδιότητες του CSS.

7.2.3.1 Τι είναι το CSS;

Το ακρωνύμιο CSS σημαίνει Cascading Style Sheets (ή στα ελληνικά Διαδοχικά Φύλλα Στυλ). Όπως έχει ήδη αναφερθεί στην ενότητα για την HTML, κάθε HTML στοιχείο έχει ένα προκαθορισμένο στυλ. Η αλλαγή του προκαθορισμένου στυλ ενός HTML στοιχείου γίνεται μέσω του style attribute. Αυτός ο τρόπος αλλαγής του στυλ (δηλαδή μέσω της χρήσης του style attribute των στοιχείων HTML) ονομάζεται *inline*. Η αλλαγή στυλ μπορεί να γίνει με ακόμα δύο τρόπους: τον *internal* τρόπο (χρησιμοποιώντας, δηλαδή, ένα <style> element στο κομμάτι <head> του HTML εγγράφου, όπου δηλώνονται τα επιθυμητά στυλ των στοιχείων του HTML εγγράφου, ή με τον *external* τρόπο) χρησιμοποιώντας, δηλαδή, ένα ή περισσότερα εξωτερικά (external) Style Sheets τα οποία αποθηκεύονται σε αρχεία CSS.

Η χρήση εξωτερικών αρχείων CSS είναι ο πιο συνηθισμένος και ο προτεινόμενος τρόπος για αλλαγή του στυλ μιας ιστοσελίδας. Το CSS είναι μια απλή style sheet γλώσσα η οποία περιγράφει την εμφάνιση και τη μορφοποίηση ενός εγγράφου το οποίο έχει γραφεί με χρήση γλώσσας σήμανσης. Μπορεί να χρησιμοποιηθεί σε κάθε είδους αρχείου XML, όπως plain XML και SVG. Η πιο συχνή χρήση του είναι για την αλλαγή του στυλ ιστοσελίδων γραμμένων σε HTML. Μαζί με την HTML και την JavaScript, το CSS αποτελεί βασική τεχνολογία που χρησιμοποιείται κατά κόρον σήμερα για τη δημιουργία ιστοσελίδων και διεπαφών χρήστη για web και mobile applications.



Εικόνα 7.24. Τα λογότυπα των CSS3, HTML5 και JavaScript

Το CSS σχεδιάστηκε κυρίως για να διευκολύνει το διαχωρισμό του περιεχομένου του εγγράφου από την παρουσίασή του, για λόγους προσβασιμότητας, ευελιξίας, ευκολότερης διαμοίρασης και μικρότερης πολυπλοκότητας λόγω της ελαχιστοποίησης της επανάληψης κώδικα στη δόμηση του περιεχομένου. Η χρήση CSS δίνει στο σχεδιαστή/ προγραμματιστή τη δυνατότητα να διαχωρίσει τις οδηγίες σχετικά με το πώς θα εμφανίζονται τα διάφορα στοιχεία στην ιστοσελίδα σε ένα ξεχωριστό αρχείο ή σε μια ενότητα του εγγράφου HTML. Έτσι, διατηρώντας τις οδηγίες στυλ σε εξωτερικά .css αρχεία, μπορεί να αλλάξει με πολύ εύκολο τρόπο την εμφάνιση πολλών ιστοσελίδων, γλιτώνοντας κόπο και χρόνο.

Άλλωστε, η γλώσσα σήμανσης HTML δεν είχε σχεδιαστεί αρχικά ώστε να περιλαμβάνει ετικέτες μορφοποίησης του εγγράφου, παρά μόνο ορισμού της δομής και του περιεχομένου του. Στη συνέχεια, προστέθηκαν ετικέτες όπως, για παράδειγμα, οι ``, ``, `<i>` στην προδιαγραφή 3.2 της HTML, προκαλώντας πολλά προβλήματα, και καθιστώντας τη δημιουργία μεγάλων ιστοσελίδων χρονοβόρα και κουραστική. Επιπλέον, οι στυλιστικές απαιτήσεις στη δημιουργία ιστοσελίδων αυξάνονταν, καθιστώντας τη δημιουργία τους με HTML όχι μόνο δύσκολες στο να γραφούν, αλλά και να συντηρηθούν. Το τελικό αποτέλεσμα ήταν να μην υπάρχει συνέπεια στην εμφάνιση των ιστοσελίδων, καθώς εξαρτιόνταν από τις δυνατότητες των περιηγητών.

Για το λόγο αυτό, το World Wide Web Consortium (W3C) δημιούργησε το CSS, με στόχο το διαχωρισμό όλης της μορφοποίησης από το HTML έγγραφο, και την αποθήκευσή του σε ένα ξεχωριστό αρχείο CSS. Συνεπώς, το CSS:

- Καθορίζει το επίπεδο παρουσίασης (presentation layer) της διεπαφής χρήστη
- Ορίζει στον περιηγητή το πώς πρέπει να παρουσιαστεί το κάθε στοιχείο στην οθόνη του χρήστη.

7.2.3.2 Ιστορικά στοιχεία

Τα style sheets υπήρχαν με διάφορες μορφές από τις απαρχές της SGML (Standard Generalized Markup Language) τη δεκαετία του 1980. Όσον αφορά όμως στη χρήση τους για τον Παγκόσμιο Ιστό, τέθηκε η απαίτηση να μπορούν να χρησιμοποιηθούν style sheets προερχόμενα από διαφορετικές πηγές. Έτσι, οι ήδη υπάρχουσες style sheet γλώσσες (π.χ. DSSSL, FOSI) κρίθηκαν ακατάλληλες.

Το CSS προτάθηκε για πρώτη φορά το 1994 από τον Håkon Wium Lie, συνεργάτη του Tim Berners-Lee στο CERN. Την ίδια περίοδο εξετάζονταν κι άλλες style sheet languages για τον Παγκόσμιο Ιστό. Προτάθηκαν 9 διαφορετικές style sheet languages στο W3C, από τις οποίες επιλέχθηκαν δύο ως βάση για το CSS, το CHSS και το SSP. Μετά από συζητήσεις, προέκυψε το 1996 η πρώτη W3C Recommendation για το CSS (CSS1). Συν-συγγραφέας του CSS1 και συν-δημιουργός του CSS θεωρείται ο Bert Bos.

Το CSS επιτρέπει να επηρεάζεται το στυλ ενός εγγράφου από πολλαπλά style sheets («cascading»). Κάποιες από τις δυνατότητες που παρείχε η CSS1 προδιαγραφή ήταν:

- Υποστήριξη ιδιοτήτων γραμματοσειράς.
- Προσδιορισμός χρώματος γραμματοσειράς, φόντου και άλλων στοιχείων.
- Διευθέτηση στοίχισης γραμματοσειράς, εικόνων, πινάκων, κ.λπ.

Η προδιαγραφή του CSS1 ολοκληρώθηκε το 1996. Εκείνον το χρόνο κυκλοφόρησε ο περιηγητής ιστοσελίδων Internet Explorer 3 της Microsoft, ο οποίος υποστήριζε σε μικρό βαθμό το CSS. Παρόλα αυτά, χρειάστηκε να περάσουν πάνω από τρία χρόνια ώσπου κάποιος περιηγητής ιστοσελίδων να υποστηρίξει (σχεδόν)

πλήρως την προδιαγραφή. Έως τον Αύγουστο του 2010 κανένας περιηγητής δεν είχε πλήρη υποστήριξη του CSS2, το οποίο είχε δημιουργηθεί το Νοέμβριο του 1997 και δημοσιευτεί ως W3C Recommendation το Μάιο του 1998. Επιπλέον, καθώς ο βαθμός υποστήριξης διέφερε σε κάθε περιηγητή, οι σχεδιαστές/ προγραμματιστές κατέφευγαν συχνά σε CSS hacks (τεχνάσματα), filters (φίλτρα) και workarounds (παρακάμψεις), ώστε να αντιμετωπίσουν ασυμβατότητες και την πιθανά διαφορετική εμφάνιση των ιστοσελίδων ανάλογα με τους περιηγητές και τις πλατφόρμες.

Για το λόγο αυτό, το W3C ξεκίνησε την αναθεώρηση της προδιαγραφής CSS2 και τη δημιουργία της προδιαγραφής CSS 2.1. Κάποιες ιδιότητες που δεν υποστηρίζονταν από περιηγητές καταργήθηκαν, και άλλαξαν κάποιες καθορισμένες συμπεριφορές. Η δημιουργία της CSS3 προδιαγραφής άρχισε το 1998 και είναι υπό ανάπτυξη. Ενώ η CSS2 είναι μια μεγάλη προδιαγραφή η οποία ορίζει διάφορα χαρακτηριστικά, η CSS3 χωρίζεται σε πολλά ξεχωριστά αρχεία («modules»). Κάθε ένα από αυτά τα modules προσθέτει ή επεκτείνει χαρακτηριστικά ήδη ορισμένα στην CSS2, διατηρώντας έτσι συμβατότητα «προς τα πίσω» (backwards compatibility).

7.2.3.3 CSS Version History

Στον ακόλουθο πίνακα (Πίνακας 7.11) παρατίθενται συνοπτικά τα χαρακτηριστικά της κάθε έκδοσης του CSS.

Πίνακας 7.11 CSS Ιστορικό Εκδόσεων (Version History)

Έκδοση	Χαρακτηριστικά
CSS 1	Έγινε επίσημη σύσταση (recommendation) του W3C το Δεκέμβριο του 1996. Άρχισε να χρησιμοποιείται στους περισσότερους περιηγητές από το 2000 και μετά. Ανάμεσα σε άλλα υποστήριζε: <ul style="list-style-type: none"> • Στοιχίση κειμένου, εικόνων, πινάκων, κ.λπ. • Ιδιότητες γραμματοσειράς. • Μοναδικά αναγνωριστικά (ids) για στοιχεία. • Κλάσεις για ομάδες ιδιοτήτων.
CSS 2	Δημοσιεύτηκε ως W3C Recommendation το Μάιο του 1998, αλλά δεν υποστηρίχθηκε ποτέ πλήρως. Κάποιες νέες δυνατότητες που εισήγαγε: <ul style="list-style-type: none"> • Τύπους μέσων (media types). • Z-index • Bidirectional text. • Νέες ιδιότητες για γραμματοσειρές (π.χ. σκιά - shadow).
CSS 2.1	Δημοσιεύτηκε τον Ιούνιο του 2005. Κάποια χαρακτηριστικά του: <ul style="list-style-type: none"> • Διορθώνει λάθη της προδιαγραφής CSS2. • Αφαιρεί στοιχεία που δεν υποστηρίζονται πλήρως από τους περιηγητές. • Επεκτείνει τα χαρακτηριστικά που υποστηρίζονται πλήρως.
CSS 3.0	Υπό ανάπτυξη. Βασικά χαρακτηριστικά: <ul style="list-style-type: none"> • Χωρίζεται σε modules. • Αποτελείται από πολλές διαφορετικές προδιαγραφές. • Εισάγει νέες δυνατότητες και εφέ, όπως: animations, transitions, gradients, multiple backgrounds, 3D transforms, κ.λπ. • Παρέχει τη δυνατότητα για advanced selectors.

7.2.3.4 Πλεονεκτήματα χρήσης CSS

Το CSS είναι μια ιδιαίτερη απλή γλώσσα που μας επιτρέπει στο σχεδιαστή της ιστοσελίδας να ορίσει με σαφήνεια πώς θα εμφανίζονται τα διάφορα στοιχεία της. Ανάμεσα στα πλεονεκτήματα της χρήσης CSS περιλαμβάνονται τα ακόλουθα:

- Μεγαλύτερη ευελιξία, καλύτερα αποτελέσματα: με χρήση του CSS είναι πλέον δυνατές μορφοποιήσεις δύσκολες ή αδύνατες με χρήση μόνο HTML.
- Γρηγορότερες ιστοσελίδες: χρησιμοποιώντας εξωτερικό αρχείο HTML, ο περιηγητής το αποθηκεύει στη μνήμη cache την πρώτη φορά που φορτώνει κάποια ιστοσελίδα ενός ιστοτόπου, οπότε δε χρειάζεται να το κατεβάξει κάθε φορά που ο χρήστης επισκέπτεται μια ιστοσελίδα του ιστοτόπου.
- Η συντήρηση αλλά και η διαμόρφωση των ιστοσελίδων είναι πλέον πολύ ευκολότερη, καθώς η εμφάνιση μιας ολόκληρης ιστοσελίδας μπορεί να ελέγχεται από ένα ή περισσότερα εξωτερικά αρχεία CSS. Σε περίπτωση που ο σχεδιαστής της ιστοσελίδας θέλει να αλλάξει κάτι στην εμφάνιση κάποιου στοιχείου, αρκεί να αλλάξει το αρχείο CSS. Αυτό είναι ιδιαίτερα χρήσιμο στην περίπτωση που πρέπει να γίνει αλλαγή ενός στοιχείου που υπάρχει σε διάφορα σημεία στην ιστοσελίδα, καθώς χρειάζεται πλέον μια μόνο αλλαγή στο CSS.
- Ο κώδικας HTML είναι πλέον λιγότερο πολύπλοκος, πιο ευανάγνωστος και συντομότερος, δεδομένου ότι οι κανόνες μορφοποίησης γράφονται μόνο μια φορά και εφαρμόζονται με ιεραρχικό τρόπο («precedence rules»).
- Ο σχεδιαστής της ιστοσελίδας έχει μεγαλύτερο έλεγχο όσον αφορά στο στυλ της ιστοσελίδας, ενώ μπορεί να χρησιμοποιεί πολλαπλά CSS αρχεία για μια ιστοσελίδα.
- Καλύτερο Search Engine Optimization (SEO): το περιεχόμενο είναι ανεξάρτητο από τη μορφοποίησή του, οπότε είναι ευκολότερο να γίνει καταγραφή και αρχειοθέτηση (indexing) από τις μηχανές αναζήτησης.

7.2.3.5 Τι ονομάζεται «Cascade» στο CSS

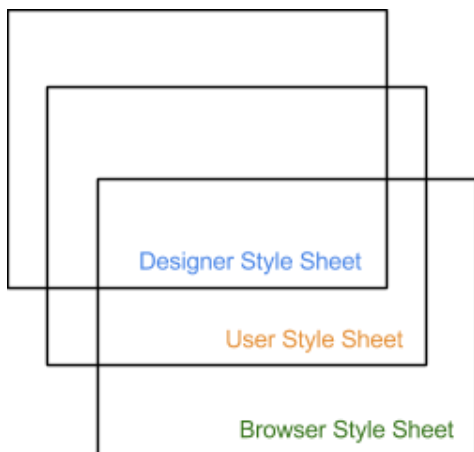
Ο όρος «Cascade» ουσιαστικά αναφέρεται στο συνδυασμό:

- Των προκαθορισμένων στυλ του περιηγητή.
- Των εξωτερικών style sheets.
- Των inline, internal (ή embedded) και user-defined styles.

...σε ένα style sheet.

Στο CSS επιτρέπεται τα στοιχεία να κληρονομούν στυλ από στοιχεία - γονείς. Ένα στοιχείο - παιδί κληρονομεί το στυλ του από το στοιχείο - γονέα, εκτός κι αν οριστεί ένα συγκεκριμένο στυλ για το στοιχείο - παιδί. Έτσι, μειώνεται το CSS που πρέπει να γραφεί για τα στοιχεία - παιδιά. Το «Cascade» ορίζει ότι κάθε ξεχωριστό στυλ έχει διαφορετική προτεραιότητα η οποία επηρεάζει την κληρονομικότητα του CSS. Τα στυλ μπορούν να ταξινομηθούν ως προς την προτεραιότητά τους σε φθίνουσα σειρά, ως εξής:

- Inline styles.
- Internal styles.
- External style sheet.
- Browser default.



Εικόνα 7.25. Cascading Style Sheets

7.2.3.6 Χρήση των Style Sheets

Έχει ήδη γίνει αναφορά σε internal, inline και external style sheets. Στο σημείο αυτό θα αναλυθεί το πώς χρησιμοποιείται κάθε ένα από αυτά, ώστε να γίνουν πιο κατανοητές οι διαφορές μεταξύ τους.

Internal Styles: Χρησιμοποιείται στην περίπτωση που μία μόνο ιστοσελίδα ενός ιστοτόπου έχει διαφορετικό στυλ από τις υπόλοιπες σελίδες. Μπορεί, έτσι, ο σχεδιαστής της ιστοσελίδας να εφαρμόσει κάποιους κανόνες CSS μόνο στη συγκεκριμένη ιστοσελίδα ή να τροποποιήσει τους υπάρχοντες κανόνες CSS του ιστοτόπου. Ορίζεται στο κομμάτι «head» της ιστοσελίδας, μέσα στην ετικέτα <style>. Για παράδειγμα:

```
<style type="text/css">
body {}
</style>
```

Inline Styles: Το inline style ενός μόνο στοιχείου ορίζεται μέσω της ιδιότητας «style» της σχετικής ετικέτας. Χρησιμοποιείται όταν στόχος είναι η μορφοποίηση ενός μόνο στοιχείου της ιστοσελίδας και η εν λόγω μορφοποίηση δε θα χρειαστεί για κάτι άλλο στην ιστοσελίδα. Για παράδειγμα, στον ακόλουθο κώδικα ορίζεται ότι το μέγεθος της γραμματοσειράς στη συγκεκριμένη παράγραφο είναι 12px:

```
<p style="font size: 12px"> MyParagraph!</p>
```

External Style Sheet: Ιδανικό για εφαρμογή του στυλ σε πολλές σελίδες. Κάθε ιστοσελίδα θα πρέπει να συμπεριλαμβάνει έναν σύνδεσμο προς το style sheet. Αυτό γίνεται με χρήση της ετικέτας <link>, η οποία πρέπει να είναι στο κομμάτι «head» της ιστοσελίδας. Αποτελεί τη σωστότερη χρήση CSS. Παράδειγμα χρήσης:

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

Το αρχείο style.css μπορεί να γραφεί σε οποιονδήποτε text editor. Πρέπει να μην περιέχει ετικέτες HTML και να αποθηκευθεί με κατάληξη «.css». Παραδείγματος χάρη, μπορεί να είναι το παρακάτω:

```
body {
  background-color: blue;
}
h1 {
  color: black;
  margin-left: 25px;
}
```

Μπορεί, επίσης, να γίνεται αναφορά σε πολλαπλά external style sheets στο ίδιο HTML έγγραφο. Μεγαλύτερη προτεραιότητα έχει αυτό που έχει γραφεί τελευταίο.

Προκύπτει, λοιπόν, εύλογα το ερώτημα: ποιό στυλ θα χρησιμοποιηθεί εάν ορίζονται περισσότερα από ένα στυλ για κάποιο HTML στοιχείο; Όπως αναφέρθηκε προηγουμένως, κάθε τύπος style έχει διαφορετική προτεραιότητα. Μεγαλύτερη προτεραιότητα έχουν τα inline styles (μέσα σε ετικέτα στοιχείου HTML), οπότε θα παρακάμψει (override) το style που ορίζεται στην ετικέτα <head> (internal style). Έπειτα, θα παρακάμψει το style που ορίζεται σε external style sheets, αλλά και στο προκαθορισμένο στυλ του περιηγητή (default).

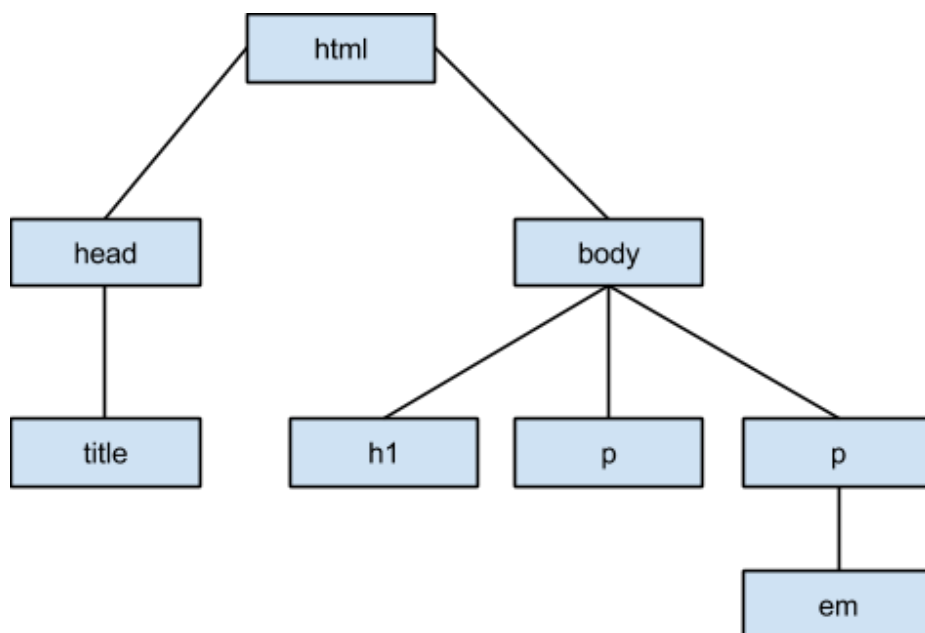
7.2.3.7 Δένδρο εγγράφου (Document Tree)

Κάθε HTML και XHTML έγγραφο είναι ένα Document Tree. Όταν ο περιηγητής φορτώνει μια ιστοσελίδα, δημιουργεί το λεγόμενο «Document Object Model» της ιστοσελίδας, το οποίο κατασκευάζεται ως δέντρο από αντικείμενα. Στο ακόλουθο παράδειγμα αναλύεται η δομή ενός Document Tree.

HTML κώδικας:

```
<html>
  <head>
    <title>Document Tree</title>
  </head>
  <body>
    <h1>Header 1</h1>
    <p>A paragraph</p>
    <p>...and a <em>second</em> one</p>
  </body>
</html>
```

Document Tree:



Εικόνα 7.26. Παράδειγμα Document Tree

7.2.3.8 Κληρονομικότητα

Κάθε HTML και XHTML έγγραφο είναι ένα Document Tree. Όταν ο περιηγητής φορτώνει μια ιστοσελίδα, δημιουργεί το λεγόμενο «Document Object Model» της ιστοσελίδας, το οποίο κατασκευάζεται ως δέντρο από αντικείμενα. Στην HTML, κάθε στοιχείο θεωρείται *απόγονος* (descendant) κάποιου άλλου στοιχείου, επιτρέποντάς το να κληρονομεί ιδιότητες από το στοιχείο-γονέα. Ο κάθε απόγονος μπορεί να έχει και δικούς του απογόνους, εκτός και αν πρόκειται για στοιχεία με κενές ετικέτες, όπως το `img`. Έτσι, διαμορφώνεται μια ιεραρχία, η οποία αποτελεί το DOM Tree (Document Object Model tree). Αναφερθήκαμε στο Document Tree προηγουμένως. Για περισσότερες λεπτομέρειες για το Document Object Model μπορείτε να ανατρέξετε [εδώ](#).

Τα στοιχείο - απόγονος, λοιπόν, μπορεί να κληρονομήσει μια ιδιότητα από το γονέα του. Αυτό συμβαίνει στην περίπτωση που δεν έχει οριστεί κάποιο συγκεκριμένο στυλ για το στοιχείο - παιδί. Κάποιες ιδιότητες CSS, οι οποίες μπορούν να κληρονομηθούν και σε στοιχεία απογόνους είναι οι εξής:

- **Color**
- **font-size**
- **font-family**
- **font-style**
- **font-weight**
- **text-align**
- **text-decoration**

Οι συγκεκριμένες ιδιότητες CSS θα αναλυθούν στη συνέχεια της ενότητας.

Ο σχεδιαστής της ιστοσελίδας μπορεί, επίσης, να ορίσει μια διαφορετική τιμή μέσω κανόνα CSS για μια συγκεκριμένη ιδιότητα ενός στοιχείου. Στην περίπτωση αυτή, θα εφαρμοστεί αυτή η τιμή και όχι η κληρονομημένη (inherited) τιμή του στοιχείου. Εάν, πάλι, θέλουμε ένα στοιχείο να αποκτήσει για μια ιδιότητα την τιμή της ιδιότητας αυτής του γονέα του, ενώ αυτή κανονικά δεν κληρονομείται, τότε υπάρχει η δυνατότητα να οριστεί ως τιμή της ιδιότητας για το στοιχείο αυτό η τιμή «inherit».

7.2.3.9 Κανόνες Σύνταξης CSS

Κανόνες CSS

Ένας κανόνας (rule) CSS αποτελείται από δύο μέρη: τον επιλογέα (CSS selector) και το μπλοκ δήλωσης (declaration block). Ο επιλογέας δείχνει στο στοιχείο που θα μορφοποιηθεί. Το μπλοκ δήλωσης αφορά στο πώς

αυτό θα μορφοποιηθεί. Περιέχει όνομα ιδιότητας (CSS property) και τιμή. Οι ιδιότητες διαχωρίζονται με «;». Η σύνταξη, δηλαδή, είναι η ακόλουθη:

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

Ο επιλογέας μπορεί να περιέχει και πολλαπλές ιδιότητες:

```
selector { property1: value1; property2: value2; }
```

Κάθε μπλοκ δήλωσης περιέχεται σε αγκύλες και κάθε δήλωση πρέπει να τερματίζεται με «;». Για παράδειγμα, ο κώδικας για να οριστεί ότι όλα τα στοιχεία <p> του συγκεκριμένου εγγράφου θα έχουν στοίχιση στο κέντρο και κείμενο χρώματος μαύρο, είναι ο ακόλουθος:

```
p { color:black;text align:center; }
```

Για λόγους αναγνωσιμότητας, μπορεί να γραφεί μία μόνο δήλωση ανά γραμμή. Το προηγούμενο, δηλαδή, παράδειγμα μπορεί να ξαναγραφεί ως εξής:

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

Τέλος, μπορεί να γίνει δήλωση ιδιοτήτων για πολλαπλούς επιλογείς. Αντί, δηλαδή, για τον ακόλουθο κώδικα:

```
h1 { color: yellow; }  
h2 { color: yellow; }  
h3 { color: yellow; }
```

Μπορεί να γραφεί:

```
h1 h2 h3 { color: yellow; }
```

Σχόλια CSS

Τα σχόλια αποτελούν τμήματα κώδικα που «παραβλέπονται» από τον περιηγητή και χρησιμεύουν στην επεξήγηση του κώδικα. Στο CSS η ανάγκη για σχόλια δεν είναι τόσο μεγάλη, αλλά μπορούν να χρησιμοποιηθούν για να διαχωρίζουν νοηματικά το αρχείο και για να παρέχουν χρήσιμες πληροφορίες, όπως το ποιός είναι ο δημιουργός του. Τα σχόλια εισάγονται μέσα σε /* ... */. Μπορούν να εκτείνονται σε περισσότερες από μία γραμμές.

Παράδειγμα:

```
p {
    color: black; /* This is a single-line comment */
    text-align: center;
}
/* This is
a multi-line
comment */
```

Επιλογείς CSS

Οι επιλογείς CSS (CSS selectors) επιτρέπουν την επιλογή των HTML στοιχείων στα οποία θα εφαρμοστεί κάποια μορφοποίηση. Χρησιμοποιούνται για να επιλεγούν στοιχεία βάσει του τύπου τους, της κλάσης τους, του id τους, κ.α. Ουσιαστικά, αποτελούν *patterns* για να αντιστοιχίζονται (match) στοιχεία στο document tree, το οποίο παρουσιάστηκε σε προηγούμενη υποενότητα. Η προδιαγραφή του CSS ορίζει διάφορα είδη επιλογέων. Συνοπτικά, οι βασικότεροι επιλογείς είναι οι ακόλουθοι:

- Επιλογέας «στοιχείο» (element selector ή type selector): αποτελείται μόνο από το όνομα μιας ετικέτας HTML. Επιλέγει το όνομα του στοιχείου βάσει του τύπου του (element type). Για παράδειγμα, έστω ότι πρέπει να εφαρμοστεί το ίδιο στυλ σε όλα τα στοιχεία <p> μιας ιστοσελίδας. Θα επιλεγούν με ακόλουθο τον ακόλουθο τρόπο:

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

- Όλες οι παράγραφοι της ιστοσελίδας θα έχουν πλέον στοίχιση στο κέντρο και μαύρο χρώμα γραμματοσειράς. Αντίστοιχα, ο επιλογέας `img` θα εφαρμοστεί σε όλες τις εικόνες της ιστοσελίδας μας, ο επιλογέας `table` θα εφαρμοστεί σε όλους τους πίνακες, ο επιλογέας `h1` σε όλα τα `h1` στοιχεία της ιστοσελίδας, κ.ο.κ. Με τον επιλογέα `body` ο σχεδιαστής μπορεί να επιλέξει ολόκληρη την ιστοσελίδα. Ο επιλογέας αυτός αντιστοιχίζει, δηλαδή, κάθε στιγμιότυπο (instance) του element type στο document tree.
- Επιλογέας «*» (universal selector): ο επιλογέας αυτός «ταιριάζει» το όνομα κάθε στοιχείου. Οι ιδιότητες που θα συμπεριληφθούν σε αυτόν τον κανόνα CSS θα εφαρμοστούν σε κάθε στοιχείο της ιστοσελίδας. Χρησιμοποιείται, συνήθως, σε συνδυασμό με άλλους επιλογείς. Στο ακόλουθο παράδειγμα, όλοι οι απόγονοι (descendants) του `h1` θα έχουν κόκκινο χρώμα κειμένου:

```
h1 * { color : red }
```

- Επιλογέας «id» (id selector): Χρησιμοποιεί την ιδιότητα «id» ενός HTML στοιχείου για να επιλέξει ένα συγκεκριμένο στοιχείο. Το id θα πρέπει να είναι μοναδικό, δηλαδή δε θα πρέπει να υπάρχουν δύο ή περισσότερα στοιχεία στην ιστοσελίδα μας με το ίδιο id. Η χρήση των ids είναι παρόμοια με τη χρήση κλάσεων. Είναι προτιμότερη η χρήση ids όταν το στοιχείο στο οποίο θα εφαρμοστεί η μορφοποίηση είναι μοναδικό. Επίσης, αξίζει να σημειωθεί ότι το id έχει υψηλότερη προτεραιότητα σε σχέση με τις κλάσεις. Για την επιλογή του επιθυμητού στοιχείου, γράφουμε το χαρακτήρα «#», ακολουθούμενο από το id του στοιχείου. Το id δε μπορεί να ξεκινάει με αριθμό. Για παράδειγμα:

```
#paragraph1 {
    text-align: center;
    color: blue;
}
```

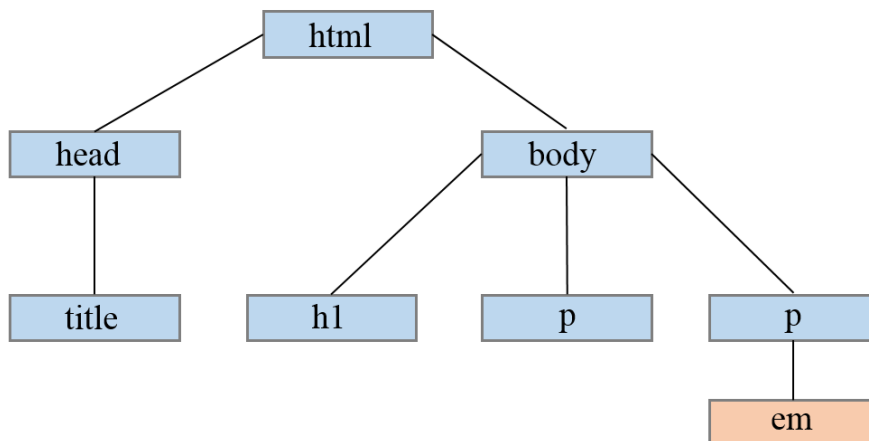
- Επιλογέας «κλάση» (class selector): Ο επιλογέας αυτός ξεκινάει με μια τελεία «.», ακολουθούμενη από το όνομα της επιθυμητής κλάσης. Ο περιηγητής «ταιριάζει» όσα στοιχεία στην ιστοσελίδα έχουν αυτήν την κλάση στην ιδιότητά τους «class», και εφαρμόζει μορφοποίηση βάσει των ιδιοτήτων που θα γράψουμε στον κανόνα CSS. Ένα στοιχείο μπορεί να ανήκει σε περισσότερες από μια κλάσεις, δηλωμένες στην ιδιότητά του class με κενά ανάμεσά τους. Τα ονόματα των κλάσεων δε μπορούν να ξεκινούν από αριθμούς, όπως και τα ids. Με τη χρήση κλάσεων, γίνεται εφικτή η ομαδοποίηση στοιχείων HTML, ώστε να τους αποδοθεί μετά συγκεκριμένη μορφοποίηση μέσω CSS. Δεν υπάρχει περιορισμός για το πόσο συχνά θα χρησιμοποιηθεί μια κλάση σε μία ιστοσελίδα. Στο παρακάτω παράδειγμα, όλα τα στοιχεία με class="center" θα έχουν στοίχιση στο κέντρο:

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

Είναι, επίσης, εφικτό να δηλωθεί ότι μόνο συγκεκριμένα στοιχεία κάποιας κλάσης θα πρέπει να επηρεάζονται από τον CSS κανόνα. Στο επόμενο παράδειγμα θα εφαρμόζεται στοίχιση στο κέντρο στα στοιχεία με ετικέτα <p> και ιδιότητα class= "center".

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

- Επιλογέας «απόγονος» (descendant selector): επιλέγει στοιχεία που είναι απόγονοι άλλων στοιχείων στο document tree. Παράδειγμα:



Εικόνα 7.27. Το document tree του παραδείγματος

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

p em { color: blue; }
```

(Τα έντονα γράμματα σε παράγραφο θα είναι χρώματος μπλε)

Τέλος, αξίζει να αναφερθεί ότι μπορούμε να συνδυάσουμε περισσότερους από έναν επιλογείς, δημιουργώντας σύνθετους επιλογείς CSS. Υπάρχουν διάφοροι τρόποι συνδυασμού επιλογέων. Για παράδειγμα:

- selector1, selector2, selector3, ...: εφαρμόζεται σε κάθε στοιχείο που πληροί τις προϋποθέσεις επιλογής για κάποιον από τους επιλογείς που χρησιμοποιήθηκαν.
- selector1, selector2, selector3, ... , selectorn: εφαρμόζεται σε κάθε στοιχείο που πληροί τις προϋποθέσεις επιλογής του selectorn, το οποίο βρίσκεται σε στοιχείο που πληροί τις προϋποθέσεις του selectorn-1, κ.ο.κ

Ψευδο-κλάσεις & ψευδο-στοιχεία

Με χρήση των ψευδο-κλάσεων (pseudo-classes) είναι δυνατή η επιλογή στοιχείων βάσει χαρακτηριστικών άλλων πέρα από το όνομά τους, τις ιδιότητές τους ή το περιεχόμενό τους. Για παράδειγμα, μπορούν να χρησιμοποιηθούν για να ορίσουμε διαφορετική μορφοποίηση για έναν υπερσύνδεσμο που έχουμε επισκεφθεί (visited) και για έναν υπερσύνδεσμο που δεν έχουμε επισκεφθεί (unvisited). Ουσιαστικά, παρέχουν τη δυνατότητα για μορφοποίηση βασισμένη σε πληροφορία που βρίσκεται εκτός του Document Tree. Επιτρέπονται οπουδήποτε σε επιλογείς, και τα ονόματά τους είναι case-insensitive.

Στη γενική μορφή της, η σύνταξη των ψευδο-κλάσεων είναι η ακόλουθη:

```
selector:pseudo-class {
    property:value;
}
```

Παράδειγμα χρήσης:

```
/* unvisited link */
a:link {
    color: #FFCCCC;
}
/* visited link */
a:visited {
    color: #1A1414;
}
```

Οι συνηθέστερα χρησιμοποιούμενες ψευδο-κλάσεις είναι οι εξής:

- a:link (χρησιμοποιείται για στοιχεία τύπου a).
- a:visited (για στοιχεία τύπου a που έχει ήδη επισκεφθεί ο χρήστης).
- a:active (για στοιχεία τύπου a, τη στιγμή που ο χρήστης έχει πατημένο το ποντίκι πάνω σε αυτά)

- στοιχείο E:hover (χρησιμοποιείται για τα στοιχεία E τη στιγμή που ο χρήστης κάνει «hover», δηλαδή έχει το δείκτη του ποντικιού πάνω σε κάποιο από αυτά).
- στοιχείο E:focus (χρησιμοποιείται κυρίως για στοιχεία φορμών και εφαρμόζεται στα στοιχεία E που εκείνη τη στιγμή έχουν «focus»).
- στοιχείο E: empty (επιλέγει κάθε στοιχείο E που δεν έχει παιδιά).
- Μπορείτε να δείτε την πλήρη λίστα ψευδο-κλάσεων [εδώ](#).

Αντίστοιχα, με χρήση των ψευδο-στοιχείων (pseudo-elements) «ταιριάζουμε» *abstractions* στο Document Tree, πέρα από αυτές που ορίζονται από τη γλώσσα του εγγράφου. Ειδικότερα, μπορεί να γίνει χρήση pseudo-element για να μορφοποιηθούν συγκεκριμένα μέρη ενός στοιχείου, όπως για παράδειγμα μόνο το πρώτο γράμμα ενός στοιχείου. Η σύνταξη των ψευδο-στοιχείων είναι παρόμοια με αυτή των ψευδο-κλάσεων, δηλαδή:

```
selector::pseudo-element {
    property:value;
}
```

Παράδειγμα χρήσης:

```
p::first-letter {
    color: #ff0000;
    font-size: large;
}
```

Συχνά χρησιμοποιούμενα ψευδο-στοιχεία είναι τα:

- στοιχείο E::first-line (επιλέγει την πρώτη γραμμή του στοιχείου E).
- στοιχείο E::selection (επιλέγει το κομμάτι εκείνο του στοιχείου E που είναι επιλεγμένο από το χρήστη).

Για την πλήρη λίστα ψευδο-στοιχείων μπορείτε να ανατρέξετε [εδώ](#).

7.2.3.10 Ιδιότητες CSS (CSS properties)

Στη συνέχεια, θα γίνει αναφορά σε μερικές βασικές ιδιότητες του CSS.

Ιδιότητες για το φόντο

- **background-color:** καθορίζει το χρώμα φόντου ενός στοιχείου. Παράδειγμα χρήσης:

```
body {
    background-color: #000000;
}
```

Τα χρώματα μπορούν να οριστούν είτε σε μορφή Hexadecimal RGB Values (π.χ #FF000), είτε με RGB Values (π.χ. rgb(255, 0, 0), είτε με W3C Standard Color Names (π.χ. aqua, black, blue, gray, κ.λπ.)

- **background-image:** ορίζει μια εικόνα φόντου. Δέχεται ως τιμή τη διεύθυνση της εικόνας σε παρενθέσεις, με «url» μπροστά από αυτή. Από προεπιλογή, η εικόνα επαναλαμβάνεται για να καλύπτει

όλο το στοιχείο στο οποίο θα είναι εικόνα φόντου. Παρακάτω, γίνεται δήλωση εικόνας φόντου για μια ιστοσελίδα:

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

- **background-repeat:** από προεπιλογή, η εικόνα θα επαναλαμβάνεται τόσο στον οριζόντιο, όσο και στον κάθετο άξονα. Αυτό δεν είναι πάντα επιθυμητό. Με την ιδιότητα `background-repeat` ορίζεται ο τρόπος με τον οποίο θα επαναλαμβάνεται η εικόνα που ορίστηκε ως εικόνα φόντου. Οι τιμές που παίρνει είναι:
 - `no-repeat:` καμία επανάληψη.
 - `repeat-x:` η εικόνα επαναλαμβάνεται στον οριζόντιο άξονα.
 - `repeat-y:` η εικόνα επαναλαμβάνεται στον κάθετο άξονα.
 - `repeat:` η εικόνα επαναλαμβάνεται και οριζόντια και κάθετα (προεπιλογή).

Παράδειγμα:

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

- **background-position:** η ιδιότητα αυτή χρησιμοποιείται για να οριστεί το που θα τοποθετηθεί η εικόνα φόντου που ορίστηκε προηγουμένως για εικόνα φόντου, τόσο στον κάθετο όσο και στον οριζόντιο άξονα. Οι τιμές που παίρνει συνήθως είναι:
 - `left, right, center:` οριζόντια θέση εικόνας.
 - `top, bottom, center:` κάθετη θέση εικόνας.

Μπορούν να γίνουν και διάφοροι συνδυασμοί των παραπάνω.

Για παράδειγμα:

```
body {  
    background-image: url("image.png");  
    background-repeat: no-repeat;  
    background-position: right top;  
}
```

Είναι, επίσης, δυνατό να οριστεί και αριθμητικά η θέση της εικόνας φόντου, με ποσοστό επί τοις εκατό (%) ή με κάποια μονάδα μεγέθους (pixels):

```
background position: 20% 35%;
```

```
background position: 100px 30px;
```

Τέλος, μπορεί να χρησιμοποιηθεί η ιδιότητα **background**, για τον ορισμό όλων των ιδιοτήτων που είδαμε παραπάνω σε μία δήλωση. Για παράδειγμα:

```
body {  
  background: #ffffff url("logo.png") no-repeat left top;  
}
```

Εάν παραλειφθεί ο ορισμός κάποιας ιδιότητας κατά τη χρήση της ιδιότητας background, τότε ο περιηγητής ιστοσελίδων θα χρησιμοποιήσει την προεπιλεγμένη γι' αυτήν τιμή.

Ιδιότητες για κείμενο

- **color:** η ιδιότητα αυτή μας επιτρέπει να αλλάξουμε το χρώμα του κειμένου. Όπως είδαμε και προηγουμένως για το χρώμα φόντου, τα χρώματα μπορούν να εισαχθούν σε μορφή Hexadecimal RGB Values (π.χ. #FF000), είτε με RGB Values (π.χ. rgb(255, 0, 0), είτε με W3C Standard Color Names (π.χ. aqua, black, blue, gray, κ.λπ.). Το προεπιλεγμένο χρώμα κειμένου ορίζεται στον body selector:

```
body {  
  color: blue;  
}  
h1 {  
  color: #000000;  
}
```

- **font-size:** αφορά στο μέγεθος της γραμματοσειράς. Οι τιμές που παίρνει ορίζονται συνήθως σε pixels (π.χ. font-size: 10px;) ή στιγμές (π.χ. font-size: 10pt). Εάν το μέγεθος της γραμματοσειράς δεν καθοριστεί, τότε παίρνει την προεπιλεγμένη τιμή, η οποία είναι 16px. Προτείνεται η χρήση των κατάλληλων ετικετών της HTML, όπως <h1> ... <h6> για κεφαλίδες και <p> για παραγράφους.

```
p {  
  font-size: 20px;  
}
```

- **font-family:** για τον ορισμό των γραμματοσειρών που θα χρησιμοποιηθούν για το κείμενο. Ορίζουμε την επιθυμητή γραμματοσειρά για το κείμενο, καθώς και εναλλακτικές επιλογές, για την περίπτωση που η γραμματοσειρά που ορίστηκε δεν υπάρχει στον υπολογιστή του χρήστη. Αν δεν υπάρχει καμία από όλες τις γραμματοσειρές που ορίστηκαν, τότε ο περιηγητής επιλέγει κάποια που να ανήκει στη γενική οικογένεια γραμματοσειρών (generic font-family), η οποία ορίζεται τελευταία με την κατάλληλη λέξη-κλειδί. Για παράδειγμα, μπορεί να οριστεί η γενική οικογένεια «serif», η οποία αποτελείται από γραμματοσειρές με ακρεμόνες, όπως η Garamond και η Georgia. Η σύνταξη της ιδιότητας αυτής είναι η ακόλουθη:

```
font family: font|initial|inherit;
```

όπου στο font δηλώνεται η επιθυμητή γραμματοσειρά, στο initial η προεπιλεγμένη τιμή και στο inherit η τιμή που θα κληρονομήσει από το στοιχείο-γονέα.

- **font-weight:** ορίζει το «βάρος» της γραμματοσειράς. Συνήθως, οι περισσότερες γραμματοσειρές διατίθενται σε δύο βάρη (κανονικό και έντονο), οπότε ουσιαστικά ορίζει αν το κείμενο θα είναι έντονο ή όχι. Η προεπιλεγμένη τιμή είναι «normal». Οι τιμές που μπορεί να πάρει είναι normal, bold, bolder, lighter, initial, inherit, καθώς και 100, 200, 300, 400, 500, 600, 700, 800, 900, όπου το 400 ισοδυναμεί με normal text και το 700 με bold text. Παράδειγμα χρήσης:

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

- **font-style:** η ιδιότητα αυτή ορίζει αν η γραμματοσειρά θα είναι κανονική (normal) ή πλάγια (italic). Οι τιμές που παίρνει είναι τρεις: normal, italic και oblique. Ενώ οι δύο τελευταίες έχουν συνήθως το ίδιο αποτέλεσμα, η χρήση της oblique δε συνίσταται λόγω ασυμβατότητας με παλαιότερους περιηγητές. Για παράδειγμα:

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

- **font:** Όπως η ιδιότητα background που αναλύθηκε παραπάνω, έτσι και η ιδιότητα font δίνει τη δυνατότητα να οριστούν ταυτόχρονα πολλές ιδιότητες που αφορούν σε κείμενο. Αν κάποια ιδιότητα δε συμπεριληφθεί, τότε ο περιηγητής θα χρησιμοποιήσει την προεπιλεγμένη τιμή της.

Παράδειγμα χρήσης:

```
font: italic bold 20px Calibri, Trebuchet MS, Verdana, sans serif;
```

- **text-align:** ορίζει τη στοίχιση του κειμένου. Οι πιθανές τιμές της είναι left, right, center, justify, initial και inherit. Παράδειγμα χρήσης:

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

- **text-decoration:** δίνει τη δυνατότητα προσθήκης διακόσμησης στο κείμενο. Οι τιμές που παίρνει είναι οι ακόλουθες:
 - none: καμία διακόσμηση.
 - underline: υπογράμμιση.
 - overline: γραμμή πάνω από το κείμενο.
 - line-through: διαγράμμιση.
 - initial: προεπιλεγμένη τιμή.
 - inherit: τιμή που κληρονομείται από στοιχείο – γονέα.

Μπορούν να εφαρμοστούν παραπάνω από μία τιμές για το ίδιο κείμενο.

- **text-justify:** για τον ορισμό των κενών ανάμεσα στις λέξεις. Παίρνει διάφορες τιμές, όπως: auto (ο αλγόριθμος για το justification επιλέγεται από τον περιηγητή), inter-word (αυξάνει/μειώνει το κενό ανάμεσα σε λέξεις), kashida (επιμηκύνει χαρακτήρες), κ.λπ. Παράδειγμα χρήσης:

```
div {  
    text-align: justify;  
    text-justify: inter-word;  
}
```

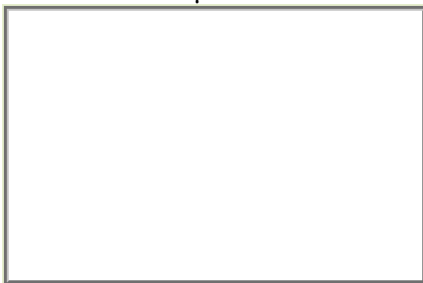
Ιδιότητες για το περίγραμμα

- **border-style:** ορίζει το στυλ του περιγράμματος. Παίρνει διάφορες τιμές, με πιο συχνές τις ακόλουθες:
 - solid: συμπαγές περίγραμμα (μια ενιαία γραμμή).
 - dashed: διακεκομμένο περίγραμμα.
 - dotted: περίγραμμα που αποτελείται από τελείες.
 - double: διπλό περίγραμμα.
 - groove: 3D grooved περίγραμμα.

Παράδειγμα χρήσης:

```
div{  
    border-width:5px;  
    border-style:groove;  
}
```

Αποτέλεσμα:



Εικόνα 7.28. Η έξοδος του παραπάνω παραδείγματος

Επίσης, η ιδιότητα αυτή μπορεί να πάρει έως τέσσερις τιμές που αντιστοιχούν στα τέσσερα περιγράμματα ενός στοιχείου, με τη σειρά (top, right, bottom και left).

- **border-color:** χρησιμεύει για τον ορισμό του χρώματος του περιγράμματος. Το χρώμα ορίζεται με έναν από τους τρόπους που περιγράφηκαν παραπάνω. Αν δεν οριστεί η ιδιότητα border-color, χρησιμοποιείται το χρώμα που ορίστηκε με την ιδιότητα color. Η ιδιότητα border-color πρέπει να οριστεί αφού πρώτα οριστεί το περίγραμμα με την ιδιότητα border-style. Μπορεί να πάρει τέσσερις τιμές, που αντιστοιχούν στα τέσσερα borders ενός στοιχείου, με την εξής σειρά: top, right, bottom και left.

Παράδειγμα χρήσης:

```
p {
  border-style: solid;
  border-color: #ff0000 #0000ff;
}
```

- **border-bottom:** Υπάρχουν πάρα πολλές ιδιότητες για τη δήλωση επιμέρους περιγραμμάτων ενός στοιχείου. Μια από αυτές είναι η `border-bottom`, η οποία χρησιμεύει για τη δήλωση όλων των ιδιοτήτων του `bottom border` σε μία μόνο δήλωση. Οι ιδιότητες που μπορούν να οριστούν, είναι, με τη σειρά:
 - `border-bottom-width`.
 - `border-bottom-style`.
 - `border-bottom-color`.

... συν τις τιμές `initial` και `inherit`.

Για παράδειγμα:

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  border-style: solid;
  border-bottom: thick dotted #ff0000;
}
</style>
</head>
<body>
<p>Bottom border example!</p>
</body>
</html>
```

Αποτέλεσμα:



Εικόνα 7.29. Η έξοδος του παραπάνω παραδείγματος

- **border-width:** η ιδιότητα αυτή ορίζει το πάχος του περιγράμματος, σε κάποια από τις μονάδες μέτρησης που χρησιμοποιούνται στο CSS (π.χ. pixels). Επιπλέον, μπορεί να οριστεί με λέξεις-κλειδιά, όπως `medium`, `thin` και `thick`. Όπως και άλλες ιδιότητες που επεξηγήθηκαν παραπάνω, έτσι κι αυτή μπορεί να πάρει έως τέσσερις τιμές (για κάθε ένα από τα τέσσερα περιγράμματα του στοιχείου).

- **border:** Η ιδιότητα border αποτελεί συντόμευση για να οριστούν με μία μόνο δήλωση όλες οι παραπάνω ιδιότητες που σχετίζονται με το περίγραμμα. Είναι, δηλαδή, αντίστοιχη των ιδιοτήτων font και background. Για παράδειγμα:

```
border: 10px solid #ffffff;
```

Επιπλέον, κάποιες άλλες χρήσιμες ιδιότητες του CSS είναι οι εξής:

- **padding:** ορίζει το κενό μεταξύ των ορίων ενός στοιχείου και το περιεχομένου του. Είναι απαραίτητο να ορίζεται ώστε να μην ενώνεται το περιεχόμενο του στοιχείου με το περίγραμμά του. Μπορεί να οριστεί διαφορετικό padding ανά πλευρά, όπως στην ιδιότητα border, χρησιμοποιώντας έως τέσσερις τιμές με τη σειρά που παρουσιάστηκε παραπάνω. Ακολουθεί ένα παράδειγμα χρήσης αυτής της ιδιότητας:

```
padding: 0px 6px 6px 6px;
```

- **margin:** ορίζει το χώρο μεταξύ των ορίων ενός στοιχείου και όσων το περιβάλλουν. Είναι, δηλαδή, το αντίθετο από το padding. Είναι ιδιαίτερα χρήσιμο για εικόνες, ώστε να μην ενώνονται με το κείμενο γύρω από αυτές. Όπως και προηγουμένως, μπορούν κι εδώ να οριστούν διαφορετικές τιμές margin ανά πλευρά. Παράδειγμα χρήσης:

```
margin: 6px 4px;
```

- **float:** επιτρέπει στα στοιχεία που περιβάλλουν το επιθυμητό στοιχείο να «ρέουν» γύρω του, όπως συμβαίνει, για παράδειγμα, στις εικόνες που συνοδεύουν ένα κείμενο. Δέχεται τις τιμές right, left και none.
- **width, height:** ορίζουν το πλάτος και το ύψος ενός στοιχείου. Παίρνουν τιμές είτε επί τις εκατό (%), είτε σε κάποια μονάδα μήκους (π.χ. pixels). Παραδείγματα χρήσης:

```
width: 100px;
height: 90%;
```

Οι ιδιότητες που αναλύθηκαν παραπάνω αποτελούν κάποιες από τις πιο συχνά χρησιμοποιούμενες ιδιότητες CSS. Παρόλα αυτά, είναι μερικές μόνο από τις ιδιότητες της προδιαγραφής. Για την πλήρη λίστα, δείτε στην ιστοσελίδα <http://www.w3schools.com/cssref/default.asp>. Τέλος, κάποιοι χρήσιμοι σύνδεσμοι στο Διαδίκτυο με υλικό σχετικά με το CSS είναι οι ακόλουθοι:

- W3C schools CSS (<http://www.w3schools.com/css/default.asp>): Η επίσημη αναφορά του W3C για το CSS.
- CSS3.info (<http://www.css3.info/>): Οι τελευταίες εξελίξεις όσον αφορά στο CSS.
- CSS Property Index (<http://www.blooberry.com/indexdot/css/propindex/all.htm>): Πλήρης αναφορά όλων των ιδιοτήτων του CSS.
- W3C validator (<http://jigsaw.w3.org/css-validator/>): Ο επίσημος CSS validator του W3C.
- CSStidy (<http://csstidy.sourceforge.net/>): Open source CSS parser και optimizer.
- Unfolding the Box Model (<http://rupl.github.io/unfold/>): Διαδραστικές διαφάνειες για CSS 3D Transforms.
- Learn CSS Interactively (<http://benhowdle.im/cssselectors/>): Visualization και επεξηγήσεις για τους CSS επιλογείς.

7.2.4. Εισαγωγή στην MySQL

7.2.4.1 Ας θυμηθούμε τι είναι η MySQL;

Όπως προαναφέραμε, η MySQL είναι το πιο διαδεδομένο σύστημα διαχείρισης βάσεων δεδομένων. Γενικότερα, μια βάση δεδομένων επιτρέπει την αποθήκευση, αναζήτηση, ταξινόμηση και ανάκληση δεδομένων αποτελεσματικά. Η MySQL χρησιμοποιεί την τυπική γλώσσα ερωτημάτων για βάσεις δεδομένων SQL (Structured Query Language). Οι βάσεις δεδομένων είναι χρήσιμες γιατί αποθηκεύουν δεδομένα κατηγορικά. Έτσι για παράδειγμα μια αποθήκη μπορεί να έχει μια βάση δεδομένων με τους ακόλουθους πίνακες:

- Προϊόντα.
- Πελάτες.
- Παραγγελίες.
- Υπάλληλοι.

Μια βάση δεδομένων ανακαλεί δεδομένα χρησιμοποιώντας ερωτήματα (Queries). Ένα ερώτημα είναι μια αίτηση προς τη βάση δεδομένων και επιστρέφει δεδομένα τα οποία είναι αποθηκευμένα με έναν δομημένο τρόπο (εγγραφές).

Για παράδειγμα, το παρακάτω ερώτημα επιλέγει όλες τις εγγραφές από τη στήλη «Product_name» του πίνακα «Products»:

```
SELECT Product_name FROM Products
```

7.2.4.2 Σύνδεση σε μια βάση MySQL με PHP

Πριν δοθεί πρόσβαση σε μια βάση δεδομένων MySQL πρέπει να γίνει σύνδεση του PHP script με τον server της βάσης. Μια τυπική σύνδεση παρουσιάζεται στο παρακάτω παράδειγμα:

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";

// Δημιουργία σύνδεσης
$conn = new mysqli($servername, $username, $password);

// Έλεγχος σύνδεσης για σφάλματα
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";
?>
```

Η σύνδεση τερματίζει με το τέλος του script αλλά αν χρειάζεται να τερματίσει πιο πριν αυτό είναι δυνατό με την εντολή:

```
$conn->close();
```

7.2.4.3 Δημιουργία βάσης δεδομένων μέσω της PHP

Για τη δημιουργία μιας βάσης με ένα script της PHP χρησιμοποιείται η δήλωση «CREATE DATABASE». Στο παρακάτω παράδειγμα παρουσιάζεται η δημιουργία μιας βάσης με όνομα «TestDB»:

```

<?php
$servername = "localhost";
$username = "username";
$password = "password";

// Δημιουργία σύνδεσης
$conn = new mysqli($servername, $username, $password);
// Έλεγχος σύνδεσης για σφάλματα
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Δημιουργία Βάσης
$sql = "CREATE DATABASE TestDB";
if ($conn->query($sql) === TRUE) {
    echo "Database created successfully";
} else {
    echo "Error creating database: " . $conn->error;
}

$conn->close();
?>

```

7.2.4.4 Δημιουργία πίνακα σε βάση δεδομένων μέσω της PHP

Για τη δημιουργία ενός νέου πίνακα σε μια βάση δεδομένων μέσω της PHP χρησιμοποιείται η δήλωση «CREATE TABLE». Στο παρακάτω παράδειγμα παρουσιάζουμε τη δημιουργία ενός πίνακα με όνομα «Employees» με τέσσερις στήλες: «id», «Firstname», «Lastname» και «email»

Παράδειγμα:

```

CREATE TABLE Employees (
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
firstname VARCHAR(30) NOT NULL,
lastname VARCHAR(30) NOT NULL,
email VARCHAR(50)
)

```

Ο τύπος δεδομένων κάθε στήλης (στη συγκεκριμένη περίπτωση υπάρχουν μόνο τύποι VARCHAR) καθορίζει τι δεδομένα μπορούν να αποθηκευτούν σε κάθε στήλη.

Μετά από τον τύπο μπορούν να καθοριστούν προαιρετικά γνωρίσματα για κάθε στήλη. Τα ορίσματα αυτά έχουν ως εξής:

- NOT NULL – Κάθε σειρά πρέπει να περιέχει μια τιμή για τη συγκεκριμένη στήλη, δεν επιτρέπονται μηδενικές τιμές.
- DEFAULT value – Ορισμός μιας προκαθορισμένης τιμής που προστίθεται αν καμία άλλη τιμή δεν έχει περαστεί.
- UNSIGNED – Χρησιμοποιείται για αριθμητικούς τύπους δεδομένων, περιορίζει τα δεδομένα της στήλης σε θετικούς αριθμούς συμπεριλαμβανομένου και του μηδέν.
- AUTO INCREMENT – Η MySQL αυτόματα αυξάνει την τιμή του πεδίου κατά 1 κάθε φορά που προστίθεται μια νέα εγγραφή.
- PRIMARY KEY – Χρησιμοποιείται για τον επακριβή προσδιορισμό των γραμμών σε ένα πίνακα. Η στήλη με το πρωτεύων κλειδί είναι συνήθως ένα ID και χρησιμοποιείται συχνά με το γνώρισμα AUTO INCREMENT.

Στο παρακάτω παράδειγμα, παρουσιάζεται ο πλήρης κώδικας σε PHP για τη δημιουργία του πίνακα «Employees»:

```

<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = " TestDB ";

// Δημιουργία σύνδεσης
$conn = new mysqli($servername, $username, $password, $dbname);
// Έλεγχος σύνδεσης για σφάλματα
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// κώδικας sql για τη δημιουργία πίνακα
$sql = "CREATE TABLE Employees (
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
firstname VARCHAR(30) NOT NULL,
lastname VARCHAR(30) NOT NULL,
email VARCHAR(50)
)";

if ($conn->query($sql) === TRUE) {
    echo "Table Employees created successfully";
} else {
    echo "Error creating table: " . $conn->error;
}

$conn->close();
?>

```

7.2.4.5 Εισαγωγή δεδομένων σε βάση δεδομένων με PHP

Μετά τη δημιουργία βάσης και πίνακα το επόμενο βήμα είναι η εισαγωγή δεδομένων. Υπάρχουν μερικοί βασικοί κανόνες που πρέπει να ακολουθηθούν σε σχέση με τη σύνταξη ερωτημάτων SQL σε ένα PHP script. Οι κανόνες αυτοί είναι:

- Τα ερωτήματα SQL πρέπει πάντα να βρίσκονται ανάμεσα σε διπλά εισαγωγικά μέσα σε ένα PHP script.
- Τιμές τύπου string πρέπει πάντα να βρίσκονται ανάμεσα σε διπλά εισαγωγικά μέσα σε ένα ερώτημα SQL.
- Αριθμητικές τιμές μέσα σε ένα ερώτημα SQL δεν πρέπει να βρίσκονται ανάμεσα σε διπλά εισαγωγικά.
- Η λέξη κλειδί NULL μέσα σε ένα ερώτημα SQL δεν πρέπει να βρίσκεται ανάμεσα σε διπλά εισαγωγικά.

Για την εισαγωγή δεδομένων μέσω της PHP χρησιμοποιείται η δήλωση INSERT INTO. Στην προηγούμενη ενότητα δημιουργήθηκε ο πίνακας «Employees» με τέσσερις στήλες: «id», «Firstname», «Lastname» και «email». Στο παρακάτω παράδειγμα παρουσιάζεται ο κώδικας για την εισαγωγή δεδομένων σε αυτόν τον πίνακα.

Παράδειγμα:

```

<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = " TestDB ";

// Δημιουργία σύνδεσης
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Έλεγχος σύνδεσης για σφάλματα
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

$sql = "INSERT INTO Employees (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com')";

if (mysqli_query($conn, $sql)) {
    echo "New record created successfully";
} else {
    echo "Error: " . $sql . "<br>" . mysqli_error($conn);
}

mysqli_close($conn);
?>

```

Για την προσθήκη παραπάνω από μιας εγγραφής σε μια βάση δεδομένων μέσω της PHP χρησιμοποιείται η συνάρτηση `mysqli_multi_query()`. Στο παρακάτω παράδειγμα παρουσιάζεται ο κώδικας για την εισαγωγή τριών εγγραφών σε αυτόν τον πίνακα.

Παράδειγμα:

```

<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = " TestDB ";

// Δημιουργία σύνδεσης
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Έλεγχος σύνδεσης για σφάλματα
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

$sql = "INSERT INTO Employees (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com');";
$sql .= "INSERT INTO Employees (firstname, lastname, email)
VALUES ('Mary', 'Moe', 'mary@example.com');";
$sql .= "INSERT INTO Employees (firstname, lastname, email)
VALUES ('Julie', 'Dooley', 'julie@example.com')";

if (mysqli_multi_query($conn, $sql)) {
    echo "New records created successfully";
} else {
    echo "Error: " . $sql . "<br>" . mysqli_error($conn);
}

mysqli_close($conn);
?>

```

7.2.4.6 Prepared statements

Τα prepared statements (έτοιμες δηλώσεις) είναι ένα χαρακτηριστικό που χρησιμοποιείται για την εκτέλεση της ίδιας ή παρόμοιας δήλωσης SQL επανειλημμένα. Ο τρόπος με τον οποίο λειτουργούν είναι ο εξής:

- Δημιουργείται ένα πρότυπο μιας δήλωσης SQL και αποστέλλεται στη βάση. Μερικές τιμές αφήνονται απροσδιόριστες και ονομάζονται παράμετροι.
- Η βάση δεδομένων αναλύει και βελτιστοποιεί τη δήλωση χωρίς όμως να την εκτελέσει και την αποθηκεύει.
- Σε μεταγενέστερο χρόνο οι παράμετροι συνδέονται με μεταβλητές και η βάση εκτελεί τη δήλωση. Η εκτέλεση μπορεί να γίνει πολλές φορές και με διαφορετικές παραμέτρους προσφέροντας προστασία από επιθέσεις τύπου SQL injection και μειωμένη απαίτηση σε πόρους, καθώς αντί για την αποστολή ολόκληρης της δήλωσης, αποστέλλονται μόνο οι παράμετροι κάθε φορά που εκτελείται η δήλωση.

Στο παρακάτω παράδειγμα παρουσιάζεται η εισαγωγή των εγγραφών του προηγούμενου παραδείγματος χρησιμοποιώντας μια prepared statement:

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = " TestDB ";

// Δημιουργία σύνδεσης
$conn = new mysqli($servername, $username, $password, $dbname);

// Έλεγχος σύνδεσης για σφάλματα
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Προετοιμασία μιας έτοιμης δήλωσης (prepared statement) και σύνδεση με μεταβλητές
$stmt = $conn->prepare("INSERT INTO Employees (firstname, lastname, email) VALUES (?, ?, ?)");
$stmt->bind_param("sss", $firstname, $lastname, $email);

// Είσαγωγή παραμέτρων και εκτέλεση
$firstname = "John";
$lastname = "Doe";
$email = "john@example.com";
$stmt->execute();

$firstname = "Mary";
$lastname = "Moe";
$email = "mary@example.com";
$stmt->execute();

$firstname = "Julie";
$lastname = "Dooley";
$email = "julie@example.com";
$stmt->execute();

echo "New records created successfully";

$stmt->close();
$conn->close();
?>
```

Τα σημεία που πρέπει να δοθεί προσοχή είναι τα:

```
$stmt = $conn->prepare("INSERT INTO Employees (firstname, lastname, email) VALUES (?, ?, ?)");
```

Στη δημιουργία μιας έτοιμης δήλωσης (prepared statement) πρέπει να δηλωθούν με αγγλικό ερωτηματικό οι τιμές που θα αφεθούν απροσδιόριστες (στη συγκεκριμένη περίπτωση και οι τρεις), αλλά όπως διαφαίνεται από κάτω με τη δήλωση της σύνδεσης πρέπει να προσδιορίσουμε τον τύπο αυτών των απροσδιόριστων τιμών επιτυγχάνοντας πολύ μεγάλη προστασία απέναντι σε επιθέσεις SQL injection αν τα δεδομένα που συνδέουμε είναι είσοδος από το χρήστη.

```
$stmt->bind_param("sss", $firstname, $lastname, $email);
```

Η παράμετρος «sss» στο συγκεκριμένο παράδειγμα προσδιορίζει ότι οι τιμές που θα σταλούν στη βάση είναι τύπου «string» και για τις τρεις παραμέτρους. Η συγκεκριμένη παράμετρος μπορεί να έχει έναν εκ των παρακάτω τύπων:

- i – integer.
- d – double.
- s – string.
- b – BLOB.

7.2.4.7 Επιλογή δεδομένων από μια βάση δεδομένων

Για την επιλογή και εμφάνιση δεδομένων από μια βάση χρησιμοποιείται η δήλωση SELECT FROM. Και για την εμφάνιση τους χρησιμοποιείται η συνάρτηση num_rows() για τον έλεγχο του αν υπάρχουν δεδομένα και η συνάρτηση fetch_assoc() μέσα σε ένα βρόχο while για την εμφάνιση τους στο χρήστη. Στο παρακάτω παράδειγμα παρουσιάζεται ο τρόπος με τον οποίο επιλέγονται τα πεδία «id», «firstname» και «lastname» από τον πίνακα «Employees» της βάσης «TestDB» και εμφανίζονται στο χρήστη.

Παράδειγμα:

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "TestDB";

// Δημιουργία σύνδεσης
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Έλεγχος σύνδεσης για σφάλματα
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

$sql = "SELECT id, firstname, lastname FROM Employees";
$result = mysqli_query($conn, $sql);
//Έλεγχος την περίπτωση επιστροφής 0 εγγραφών
if (mysqli_num_rows($result) > 0) {
    // Εξαγωγή εγγραφών ανα σειρά
    while($row = mysqli_fetch_assoc($result)) {
        echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " . $row["last-
name"]. "<br>";
    }
} else {
    echo "0 results";
}

mysqli_close($conn);
?>
```

Η δήλωση αποθηκεύεται στη μεταβλητή `$sql` και τα αποτελέσματα της αποστολής στη βάση στη μεταβλητή `$result`. Έπειτα ελέγχεται αν υπάρχει έστω και μια σειρά αποτελεσμάτων και αν ναι ξεκινάει ο βρόχος που τα εμφανίζει ένα-ένα μέχρι να μην υπάρχουν άλλα δεδομένα στην `$result`.

Στην περίπτωση που ο πίνακας με τα δεδομένα έχει χιλιάδες εγγραφές, πράγμα που μπορεί να οδηγήσει σε πολύ αργή ανταπόκριση από τον server η MySQL δίνει τη δυνατότητα με την παράμετρο `LIMIT` να επιλέξουμε το μέγιστο αριθμό των εγγράφων που θα επιστρέψει η βάση. Η δήλωση `SELECT` με την παράμετρο `LIMIT` έχει τη μορφή:

```
$sql = "SELECT lastname FROM Employees LIMIT 30";
```

Σε αυτήν την περίπτωση η βάση θα επιστρέψει τις 30 πρώτες εγγραφές ακόμα και αν υπάρχουν παραπάνω. Αν χρειαστεί όμως για παράδειγμα να επιλεγθούν οι εγγραφές 16-45 αντί για τις πρώτες 30 τότε μπορεί να χρησιμοποιηθεί η παράμετρος `OFFSET`. Άρα η παραπάνω δήλωση θα αλλάξει σε:

```
$sql = "SELECT lastname FROM Employees LIMIT 30 OFFSET 15";
```

7.2.4.8 Διαγραφή δεδομένων από μια βάση με PHP

Για τη διαγραφή εγγράφων από ένα πίνακα της βάσης χρησιμοποιείται η δήλωση `DELETE FROM WHERE`. Αν στη δήλωση ο όρος `WHERE` παραληφθεί τότε θα διαγραφούν όλες οι εγγραφές του πίνακα. Στο παρακάτω παράδειγμα παρουσιάζεται ο τρόπος με τον οποίο γίνεται η διαγραφή της εγγραφής με `id` ίσο με 3 από τον πίνακα `Employees` της βάσης `TestDB`.

Παράδειγμα:

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "TestDB";

// Δημιουργία σύνδεσης
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Έλεγχος σύνδεσης για σφάλματα
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

// Κώδικας sql για τη διαγραφή εγγραφής
$sql = "DELETE FROM Employees WHERE id=3";

if (mysqli_query($conn, $sql)) {
    echo "Record deleted successfully";
} else {
    echo "Error deleting record: " . mysqli_error($conn);
}

mysqli_close($conn);
?>
```

7.2.4.9 Ενημέρωση εγγράφων σε πίνακα βάσης δεδομένων με PHP

Για την ενημέρωση εγγράφων σε ένα πίνακα της βάσης δεδομένων χρησιμοποιείται η δήλωση `UPDATE SET WHERE`. Αν στη δήλωση παραληφθεί ο όρος `WHERE` τότε θα ενημερωθούν όλες οι εγγραφές του πίνακα. Στο παρακάτω παράδειγμα παρουσιάζεται ο τρόπος με τον οποίο γίνεται ενημέρωση της στήλης `lastname` της εγγραφής με `id` ίσο με 2 στον πίνακα `Employees` της βάσης `TestDB` με την τιμή `Doe`.

Παράδειγμα:

```

<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "TestDB";

// Δημιουργία σύνδεσης
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Έλεγχος σύνδεσης για σφάλματα
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
// Κώδικας sql για την ενημέρωση εγγραφής
$sql = "UPDATE Employees SET lastname='Doe' WHERE id=2";

if (mysqli_query($conn, $sql)) {
    echo "Record updated successfully";
} else {
    echo "Error updating record: " . mysqli_error($conn);
}

mysqli_close($conn);
?>

```

7.3. Υλοποίηση Δυναμικού Εκπαιδευτικού Ιστοτόπου

Σε αυτό το σημείο θα παρουσιαστούν τα πιο σημαντικά βήματα που χρειάζεται να γίνουν, προκειμένου να υλοποιηθεί ο δυναμικός ιστότοπος. Πιο συγκεκριμένα, θα παρατεθούν τα αποσπάσματα του κώδικα της γλώσσας PHP για συγκεκριμένες λειτουργίες του δυναμικού ιστοτόπου, καθώς και οι οδηγίες που απαιτούνται για την κατασκευή της βάσης δεδομένων.

Πριν ξεκινήσει, όμως, η συγγραφή του κώδικα, θα πρέπει να ελεγχθεί μια συγκεκριμένη τιμή μεταβλητής, που υπάρχει στο αρχείο με τις ρυθμίσεις της PHP. Σε νεότερες εκδόσεις της PHP (από την 4.2.0 και μετά), η τιμή αυτής της μεταβλητής είναι αυτή που πρέπει (δηλαδή «off») για την περίπτωση που χρειάζεται ο δυναμικός ιστότοπος που θα υλοποιηθεί. Οπότε δε χρειάζεται κάποια αλλαγή. Αλλά αν διατίθεται παλαιότερη έκδοση της PHP (δηλαδή πιο πριν από την έκδοση 4.2.0) ή αν απλά χρειάζεται να γίνει έλεγχος για την τιμή της μεταβλητής τότε θα πρέπει να γίνει η παρακάτω διαδικασία:

Όλες οι ρυθμίσεις που αφορούν στην PHP αποθηκεύονται σε ένα αρχείο που ονομάζεται `php.ini` το οποίο δημιουργείται κατά την εγκατάσταση της PHP στον υπολογιστή (βλ. Κεφάλαιο 5), σε φάκελο, ο οποίος μπορεί να είναι διαφορετικός (και να βρίσκεται σε διαφορετική τοποθεσία) ανάλογα το λειτουργικό σύστημα που εγκαθίσταται. Το αρχείο `php.ini` ουσιαστικά είναι ένα αρχείο τύπου `text` που περιέχει επεξηγηματικό κείμενο, καθώς και ένα σύνολο γραμμών εντολών-οδηγιών με τιμές σε διάφορες παραμέτρους. Το αρχείο μπορεί να ανοίξει με έναν οποιονδήποτε επεξεργαστή κειμένου (`text editor`). Η σημαντική αλλαγή που πρέπει να γίνει, πριν προχωρήσει η υλοποίηση του δυναμικού ιστοτόπου, περιγράφεται στη συνέχεια.

Αρχικά, θα πρέπει να εντοπιστεί μέσα στο αρχείο `php.ini` η παρακάτω γραμμή: `register_globals=Off`. Η γραμμή αυτή δηλώνει ότι η PHP δεν θα δέχεται `global` μεταβλητές. Η τιμή `Off` είναι προεπιλεγμένη γιατί προσδίδει μεγαλύτερη ασφάλεια στις ιστοσελίδες. Ωστόσο, εφόσον η εφαρμογή που θα αναπτυχθεί στη συγκεκριμένη περίπτωση, είναι σχετικά απλή, η τιμή αυτή, μπορεί να αλλαχθεί σε `On` χωρίς να επιφέρει σημαντικές επιπτώσεις. Αντιθέτως μάλιστα ο κώδικας γίνεται πιο απλός. Αυτό, θα φανεί καλύτερα σε επόμενο βήμα, κατά το οποίο θα χρειαστεί να περαστεί η τιμή μίας μεταβλητής από μία ιστοσελίδα σε μία άλλη.

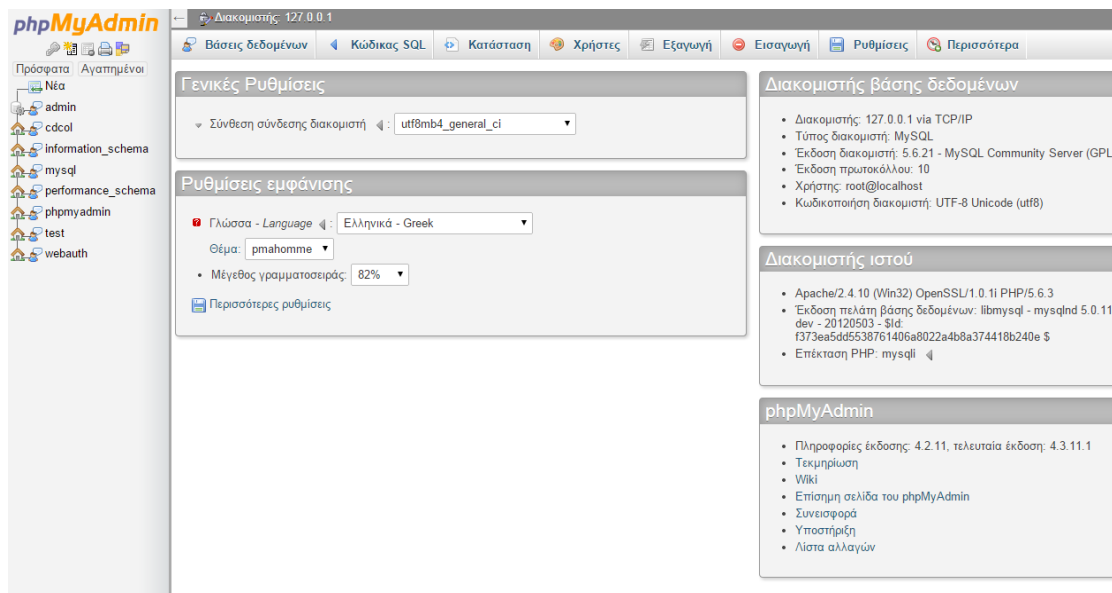
Επομένως, θα πρέπει να αλλαχθεί στο αρχείο `php.ini` η γραμμή: `register_globals=Off` σε `register_globals=On` και να αποθηκευτεί το αρχείο. Σημαντική λεπτομέρεια είναι ότι το αρχείο πρέπει να παραμείνει `php.ini` και όχι το `php.ini.txt`! Όλα τα παραδείγματα και τα τμήματα κώδικα που υπάρχουν στη συγκεκριμένη υλοποίηση, τρέχουν με την επιλογή `register_globals = On`.

Ανοίγοντας έναν οποιονδήποτε περιηγητή και πληκτρολογώντας στη γραμμή διευθύνσεων τη διεύθυνση <http://localhost>, θα εμφανιστούν τα περιεχόμενα του φακέλου που περιέχει όλα τα αρχεία της PHP, του Apache server και της MySQL που δημιουργήθηκαν κατά την εγκατάσταση. Η διεύθυνση <http://localhost> είναι η αρχική ιστοσελίδα που βλέπει ο Apache server. Σε αυτόν το φάκελο θα αποθηκεύονται οι φάκελοι και τα

αρχεία που θα δημιουργούνται προκειμένου να μπορούν να μεταφράζονται από τη μηχανή της PHP και να προβάλλονται από τον Apache.

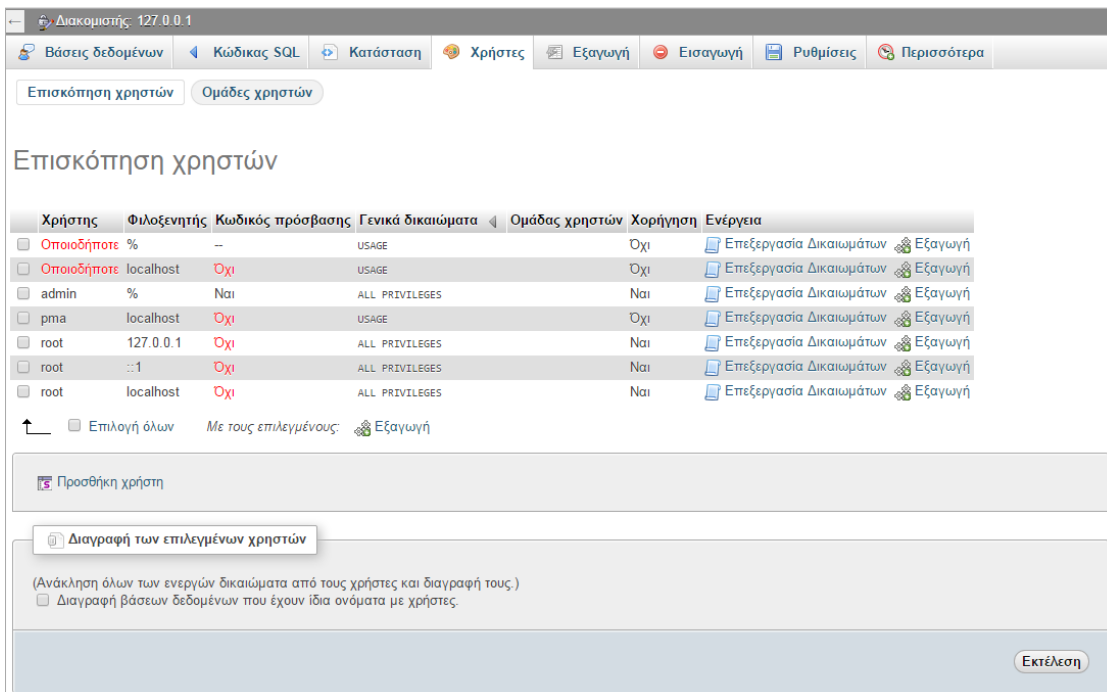
Η περίπτωση κατά την οποία αναπτύσσονται σελίδες με κώδικα PHP και δεν υπάρχει η προοπτική για δημοσίευση αυτών στο Διαδίκτυο, τότε ο Apache server αντιστοιχεί στη διεύθυνση `http://localhost` ή στην ip `127.0.0.1`. Ειδικά, όταν υπάρχει διαθέσιμη σταθερή ip τότε οι ιστοσελίδες που δημιουργούνται, μπορούν να δημοσιευθούν στο Διαδίκτυο, δηλώνοντας στον Apache την ip και το domain name.

Η έκδοση αυτή της PHP περιλαμβάνει και ένα γραφικό περιβάλλον διαχείρισης. Το περιβάλλον αυτό λέγεται «`phpmyadmin`» και η διεύθυνσή του είναι η `http://localhost/phpmyadmin/`. Ο σύνδεσμος για αυτό το περιβάλλον εμφανίζεται και στην αρχική σελίδα του apache server την `http://localhost/`. Αυτό το γραφικό περιβάλλον περιέχει όλες τις λειτουργίες που χρειάζονται για τη διαχείριση μίας βάσης δεδομένων και παρακάτω θα παρουσιαστεί ο τρόπος με τον οποίο μπορεί να δημιουργηθεί μία βάση δεδομένων από την αρχή, μαζί με τους επιμέρους πίνακες έτσι ώστε να αποθηκεύονται τα επιθυμητά δεδομένα, καθώς και να δημιουργούνται οι αντίστοιχοι λογαριασμοί χρηστών με τα κατάλληλα δικαιώματα πρόσβασης. Ανοίγοντας το `phpmyadmin` εμφανίζεται η παρακάτω οθόνη (Εικόνα 7.28):



Εικόνα 7.30. Η κεντρική ιστοσελίδα του περιβάλλοντος `phpmyadmin`

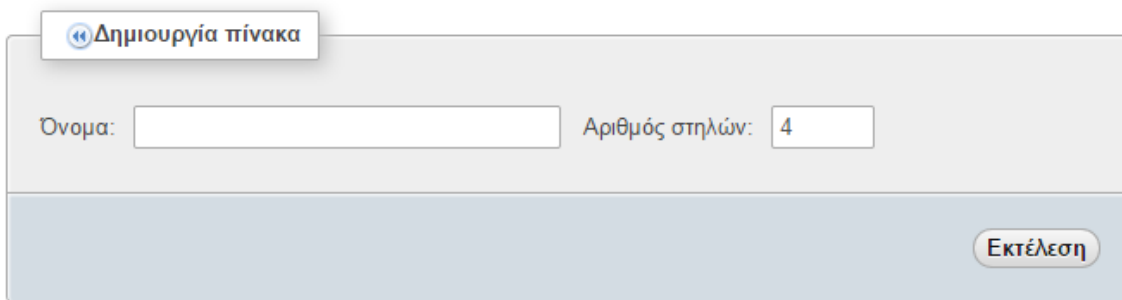
Η δημιουργία χρηστών της βάσης δεδομένων και η παροχή δικαιωμάτων χρήσης, γίνεται από την καρτέλα «Χρήστες». Εκεί προβάλλονται οι ήδη υπάρχοντες χρήστες και επιπλέον το κουμπί προσθήκης νέων χρηστών.



Εικόνα 7.31. Καρτέλα «Χρήστες» όπου φαίνονται οι χρήστες της βάσης δεδομένων

Στην καρτέλα «Βάσεις δεδομένων» φαίνονται οι διαθέσιμες βάσεις δεδομένων και με την επιλογή «Δημιουργία βάσης δεδομένων» δημιουργείται μία νέα βάση με όνομα οτιδήποτε γράφει στο πεδίο «Όνομα βάσης δεδομένων» και κωδικοποίηση χαρακτήρων με ό,τι επιλεγεί στο πεδίο «Σύνθεση». Είναι σημαντικό να επιλεγεί μία κωδικοποίηση χαρακτήρων, η οποία θα δουλεύει αξιόπιστα σε όλα τα συστήματα και μία ασφαλής επιλογή είναι η utf8. Συστήνεται να επιλεγεί η utf8_general_ci. Πατώντας το κουμπί «Δημιουργία» εκτελείται η εντολή για τη δημιουργία της βάσης δεδομένων και στη συνέχεια μπορεί να προχωρήσει η δόμηση της βάσης με τη δημιουργία πινάκων.

Εφόσον έχει δημιουργηθεί η βάση δεδομένων, μπορεί να ξεκινήσει πλέον η δημιουργία των πινάκων στους οποίους θα αποθηκεύονται τα δεδομένα. Για να δημιουργηθεί ένας νέος πίνακας, θα πρέπει αρχικά να οριστεί το όνομά του και ο αριθμός των στηλών που θα περιέχει.



Εικόνα 7.32. Δημιουργία νέου πίνακα στη βάση δεδομένων

Κάθε στήλη θα αντιπροσωπεύει μια μεταβλητή του πίνακα και θα παίρνει τιμές συγκεκριμένου τύπου (ακέραιο, αλφαριθμητικό, κείμενο κ.α.). Τα σημαντικά πεδία που θα πρέπει να συμπληρωθούν είναι το όνομα του πεδίου, ο τύπος του, το μήκος του (ή το σύνολο διακριτών τιμών που μπορεί να πάρει), το αν θα μπορεί να πάρει την τιμή NULL, αν θα έχει προκαθορισμένη τιμή και αν θα είναι ή όχι πρωτεύον κλειδί, ευρετήριο ή μοναδικό.

Για το όνομα του πεδίου, συστήνεται να μην υπάρχουν κενά για να είναι πιο απλά τα ερωτήματα στη βάση δεδομένων (queries) που θα γίνονται στη συνέχεια. Επίσης, τα ονόματα των μεταβλητών στην php είναι case-sensitive. Καλό είναι λοιπόν να ακολουθείται ένας κανόνας. Για παράδειγμα, όλα τα ονόματα των στηλών με μικρά ή με κεφαλαία (ή κάποιος συνδυασμός που θα απομνημονεύεται εύκολα). Όσον αφορά τον τύπο της κάθε στήλης, οι τύποι της MySQL δεν είναι ίδιοι με αυτούς της php, ωστόσο αυτό δεν είναι σοβαρό πρόβλημα,

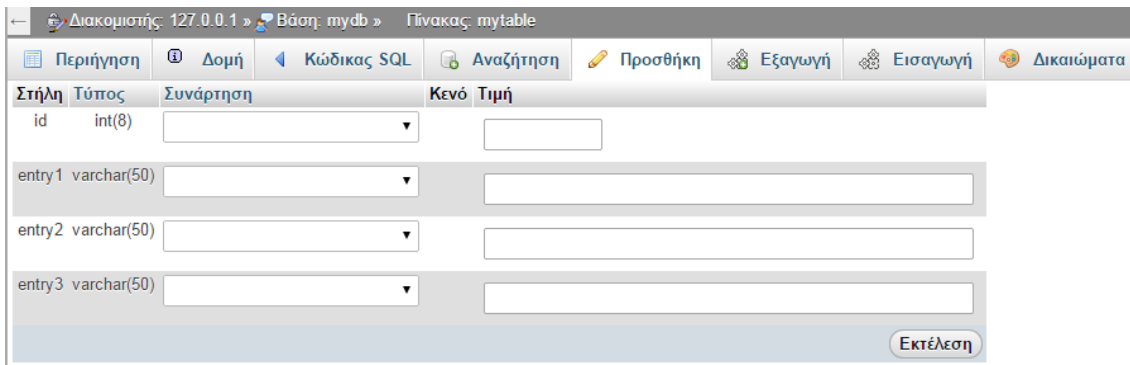
καθώς η PHP αποδίδει αυτόματα τον κατάλληλο τύπο σε κάθε μεταβλητή της. Στο πεδίο «Μήκος» δηλώνεται το μήκος που θα μπορεί να έχει η τιμή στο πεδίο. Δεν είναι απαραίτητο να χρειάζονται όλοι οι τύποι δεδομένων δήλωση του μήκους. Για παράδειγμα, αν το πεδίο είναι τύπου TEXT τότε δεν χρειάζεται να οριστεί το μήκος. Στην επιλογή «Κενό» δηλώνεται αν το πεδίο του πίνακα μπορεί να έχει τιμή NULL. Επίσης, στην επιλογή «Προεπιλογή» μπορεί να οριστεί η προκαθορισμένη τιμή που θα πάρει το πεδίο, αν δεν δοθεί κάποια εξαρχής.

Πολλές φορές μπορεί να χρησιμοποιηθεί μία στήλη ως αναγνωριστικό μίας εγγραφής. Για παράδειγμα, μπορεί μία στήλη με όνομα «id» να περιέχει τον αύξοντα αριθμό της εγγραφής. Αυτό το πεδίο μπορεί να επιλεγεί ότι θα είναι auto_increment (στήλη A_I). Έτσι, η MySQL θα εισάγει αυτόματα τον επόμενο αύξοντα αριθμό σε αυτό το πεδίο κατά την εισαγωγή δεδομένων στον πίνακά σας. Στην επιλογή «Ευρετήριο» μπορεί να οριστεί αν η στήλη θα μπορεί να δέχεται διπλότυπα ή όχι (Unique), αν εκτός από μοναδικό είναι και ευρετήριο (Index) ή αν θα είναι πρωτεύον κλειδί (Primary).

Αν μία στήλη επιλεγεί ως ευρετήριο τότε η στήλη αυτή δεν θα δέχεται διπλότυπα, ενώ παράλληλα η MySQL θα δημιουργήσει ένα ευρετήριο (καταλαμβάνοντας όμως επιπλέον χώρο) και θα το χρησιμοποιεί κάθε φορά που εκτελούνται ερωτήματα στον πίνακα. Τέλος, μπορεί να υπάρξει συνδυασμός στηλών ως πρωτεύον κλειδί. Για να γίνει αυτό, τότε θα πρέπει να επιλεγεί και στις 2 στήλες η επιλογή Primary.

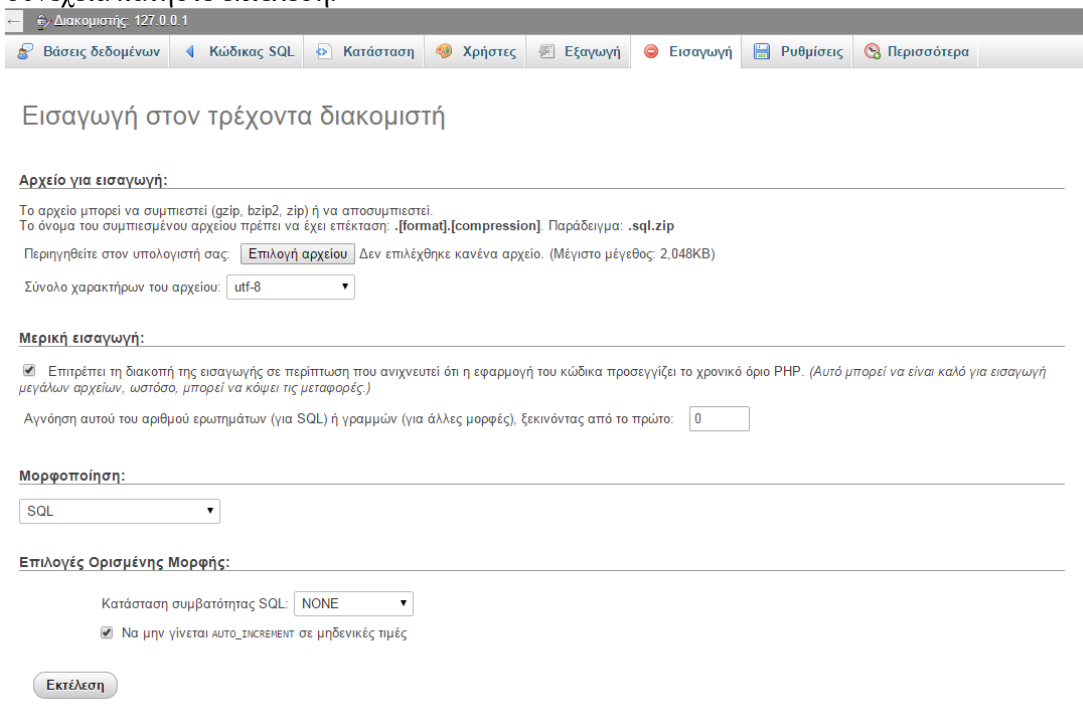
Εικόνα 7.33. Ορισμός στηλών στον πίνακα που πρόκειται να δημιουργηθεί

Αν έχουν γίνει όλα σωστά, τότε μπορείτε να προχωρήσετε στην εισαγωγή των δεδομένων στους πίνακες. Επιλέγοντας τη βάση δεδομένων που θέλετε και στη συνέχεια τον πίνακα που θα εισαχθούν τα δεδομένα, από το μενού που βρίσκεται στα αριστερά, επιλέξτε το πεδίο «Προσθήκη». Θα εμφανιστούν οι στήλες του πίνακα, ο τύπος κάθε στήλης, η δυνατότητα να εισαχθεί κάποια συνάρτηση και τέλος, το πεδίο με τις τιμές που μπορούν να συμπληρωθούν από εσάς. Προσοχή χρειάζεται ώστε τα δεδομένα που εισάγουμε να είναι σε συμφωνία με τη δομή του πίνακα.



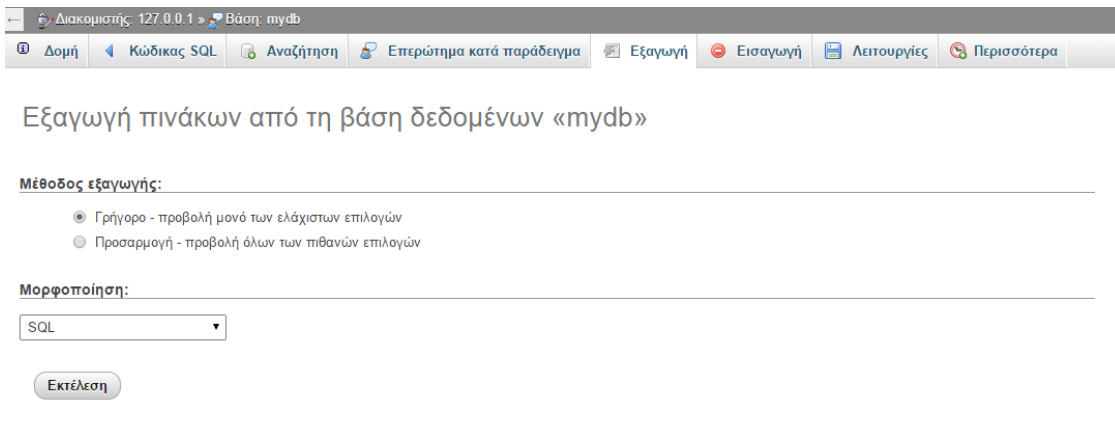
Εικόνα 7.34. Εισαγωγή δεδομένων σε πεδία του πίνακα

Δύο ακόμα σημαντικές λειτουργίες είναι η εισαγωγή και η εξαγωγή βάσης δεδομένων. Αν θέλετε να χρησιμοποιήσετε μία ήδη υπάρχουσα βάση δεδομένων ή ακόμα και ένα συγκεκριμένο πίνακα, θα πρέπει να εισάγετε τη βάση μέσω της επιλογής «Εισαγωγή», από την κεντρική ιστοσελίδα της phpmyadmin. Επιλέξτε απλά το αρχείο που επιθυμείτε και το οποίο θα πρέπει να έχει κατάληξη «.[format].[compression]» και στη συνέχεια πατήστε εκτέλεση.



Εικόνα 7.35. Εισαγωγή υπάρχουσας βάσης δεδομένων

Με απλό τρόπο μπορεί επίσης να γίνει και η εξαγωγή μιας βάσης δεδομένων, επιλέγοντας το αντίστοιχο κουμπί «Εξαγωγή». Επιλέξτε απλά την επιθυμητή μορφοποίηση και πατήστε «Εκτέλεση». Το αρχείο θα αποθηκευτεί τοπικά στον υπολογιστή σας και θα είναι διαθέσιμο για εισαγωγή σε οποιοδήποτε άλλο μηχάνημα.



Εικόνα 7.36. Εξαγωγή της βάσης δεδομένων

Εφόσον ξέρετε τα βασικά που χρειάζονται για τη δημιουργία βάσης δεδομένων και πινάκων, μπορείτε να προχωρήσετε στην υλοποίηση των πινάκων για το δυναμικό ιστότοπο που θα δημιουργήσετε.

Αρχικά, θα παρουσιαστεί η γενική μορφή κάθε ιστοσελίδας, με τη διαμόρφωση του μενού πλοήγησης και του περιεχομένου. Το θέμα της σχεδίασης του CSS αφήνεται στην ευχέρεια σας, καθώς αποτελεί καθαρά δική σας επιλογή. Θα δείτε μόνο με ποιον τρόπο μπορείτε να συμπεριλάβετε το αρχείο CSS σε κάθε ιστοσελίδα και πώς να χρησιμοποιήσετε τους κανόνες του. Εφόσον παρουσιάστηκε σε προηγούμενο σημείο η γενική μορφή των ιστοσελίδων, θα παρουσιαστεί κατευθείαν ο κώδικας που θα χρησιμοποιήσετε. Επομένως, για τη γενική μορφή κάθε ιστοσελίδας θα ακολουθήσετε το παρακάτω παράδειγμα:

```
<html>
<head>
  <title>Αρχική σελίδα</title>
  <meta charset="utf-8"/>

  // Σύνδεση με το αρχείο CSS
  <link rel="stylesheet" href="stylesheet.css" type="text/css"/>
</head>
<body class="body">
  <div>
    <nav>

      // Το μενού πλοήγησης

      <a href="index.php">Αρχική σελίδα</a>

      <a href="announcements.php">Ανακοινώσεις</a>

      <a href="communication.php">Επικοινωνία</a>

      <a href="documents.php">Εγγραφα μαθήματος</a>

      <a href="homework.php">Εργασίες</a>

    </nav>

    <content>

      // Το κυρίως περιεχόμενο της ιστοσελίδας

      <div>
        <h1>Τίτλος μαθήματος</h1>
        <p> Καλώς ήρθατε στην ιστοσελίδα του μαθήματος</p>
      </div>
    </content>

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

Αυτή η δομή θα είναι καλό να διατηρηθεί και στις υπόλοιπες ιστοσελίδες και με βάση αυτή θα προστεθεί αργότερα και ο κώδικας PHP που είναι απαραίτητος. Η τελική της μορφή θα φανεί στη συνέχεια.

7.3.1. Είσοδος και έλεγχος χρήστη

Όπως αναφέρθηκε, κάθε χρήστης θα πρέπει να κάνει είσοδο στον ιστότοπο με τα προσωπικά του στοιχεία. Αυτό θα γίνει με μία φόρμα υποβολής στοιχείων, η οποία θα ενσωματωθεί στο αρχείο login.php, το οποίο θα είναι η πρώτη ιστοσελίδα που θα εμφανίζεται στο χρήστη, προκειμένου να εισάγει τα προσωπικά του στοιχεία. όπως η παρακάτω:

```
<html>
<head>
<title>Ιστοσελίδα Login</title>
</head>
<body>
<h1>Ιστοσελίδα Login</h1>
<form action="check.php" method="post">
  Username : <input type="text" name="user">
<br>
  Password : <input type="password" name="psw">
<br>
<input type="submit" value="Login">
</form>
</body>
</html>
```

Η παραπάνω φόρμα στέλνει τις τιμές των μεταβλητών στο αρχείο «check.php» όπως δηλώνεται και στο πεδίο «action». Αν τα στοιχεία είναι σωστά τότε εμφανίζει την ιστοσελίδα «home.php», ανάλογα για κάθε τύπο χρήστη (tutor/student) διαφορετικά, θα πρέπει να εμφανίζει ένα μήνυμα λάθους, ότι τα στοιχεία που έβαλε ο χρήστης δεν είναι σωστά και ότι θα πρέπει να προσπαθήσει ξανά. Για να γίνει ο έλεγχος των στοιχείων που εισήγαγε ο χρήστης, θα πρέπει να προσπελαστεί η βάση δεδομένων και να ελέγξει αν υπάρχει χρήστης με τα συγκεκριμένα στοιχεία.

Το πρώτο βήμα είναι να γίνει μία σύνδεση με τη βάση δεδομένων. Αυτό θα γίνει με την εντολή «mysql_connect». Είχε αναφερθεί σε προηγούμενο σημείο, η δημιουργία του χρήστη της βάσης δεδομένων με όνομα χρήστη «myuser» και κωδικό «mypsw». Αυτά τα στοιχεία θα χρησιμοποιηθούν στην εντολή «mysql_connect». Στη συνέχεια θα πρέπει να επιλεγεί η βάση που έχετε δημιουργήσει χρησιμοποιώντας την εντολή «mysql_select_db». Ο κώδικας για τη σύνδεση με τη βάση είναι ο παρακάτω:

```
$hostname = "localhost";
$database = "mydb";
$username = "myuser";
$password = "mypsw";
$link = mysql_connect($hostname , $username , $password) or
die("Prosoxi!Provlima stin sundesi me ton server : " . mysql_error());
mysql_select_db($database,$link);
```

Το αποτέλεσμα της εντολής «mysql_connect» αποδίδεται στη μεταβλητή \$link. Αυτό γίνεται γιατί όταν η «mysql_connect» εκτελείται σωστά τότε επιστρέφει ένα αναγνωριστικό (ένας αριθμός). Αυτό το χαρακτηρι-

στικό μπορεί να χρησιμοποιηθεί στη συνέχεια για να δηλώσει μια συγκεκριμένη σύνδεση. Θα μπορούσαν, δηλαδή να υπάρχουν ταυτόχρονα πολλές συνδέσεις, σε διαφορετικές βάσεις δεδομένων, και να εκτελούνται ερωτήματα σε κάθε μία ξεχωριστά.

Για να προσδιορίζεται ποιο ερώτημα θα εκτελεστεί κάθε φορά σε κάθε βάση θα πρέπει χρησιμοποιούμε αυτό το αναγνωριστικό. Επίσης, η εντολή «or die», που ακολουθεί τη «mysql_connect», διακόπτει τη σύνδεση, αν υπάρχει κάποιο πρόβλημα, και εμφανίζει ένα μήνυμα σχετικά με το πρόβλημα που προκλήθηκε. Επιπλέον, παρέχεται η δυνατότητα για παρεμβολή μηνυμάτων του ίδιου του προγραμματιστή, προκειμένου να υπάρχει μια καλύτερη κατανόηση για το σημείο και το είδος του σφάλματος που ενδεχομένως να συμβαίνει.

Στη συνέχεια, πρέπει να επιλεγεί η βάση δεδομένων με την εντολή «mysql_select_db» περνώντας ως ορίσματα το όνομα της βάσης και την τιμή της μεταβλητής \$link για να δηλωθεί στο σύστημα πού να ψάξει για τη βάση. Σε αυτό το σημείο δεν είναι απαραίτητη η προσθήκη του \$link ως όρισμα της «mysql_select_db», ωστόσο είναι καλή πρακτική να χρησιμοποιείται σε κάθε διάδραση με τη MySQL.

Εφόσον έχει γίνει επιτυχής σύνδεση με τη βάση, θα πρέπει να ελεγχθούν οι τιμές για το username και το password που έδωσε ο χρήστης στην ιστοσελίδα «login.php». Αυτό γίνεται εκτελώντας ένα ερώτημα προς τη βάση (query) χρησιμοποιώντας την εντολή «mysql_query»:

```
$Qcheck_user = "SELECT user.*
                FROM `user`
                WHERE user.username = '$user'
                AND user.password = '$psw' ";
$check_user = mysql_query($Qcheck_user , $link) or
                die("Prosoxi!Provlima stin epilogi pinaka user : " . mysql_error());
```

Προσοχή θέλει στο γεγονός ότι τα ονόματα των πινάκων μέσα στο query περικλείονται από ` (δηλαδή από πλάγιες αποστρόφους), καθώς και στα ονόματα των μεταβλητών, τα οποία όπως έχει αναφερθεί είναι case sensitive, και περικλείονται από κάθετες αποστρόφους.

Σε περίπτωση που ο χρήστης εισήγαγε έγκυρα στοιχεία και η αναζήτηση στη βάση ήταν επιτυχής η εκτέλεση της εντολής «mysql_query» θα επιστρέψει μια γραμμή με περιεχόμενα την εγγραφή του συγκεκριμένου χρήστη (όνομα, επώνυμο, username, password, role). Αυτή η γραμμή αποθηκεύεται στη μεταβλητή \$check_user και δεν πρόκειται για πίνακα, αλλά για resource (δηλαδή πόρο)! Έτσι, δεν μπορεί κάποιος να διαχειριστεί τη μεταβλητή \$check_user ως πίνακα, αφού είναι τύπου resource. Υπάρχουν πολλοί τρόποι για να ελεγχθεί αν το αποτέλεσμα που επέστρεψε η εκτέλεση του query σημαίνει ότι ο χρήστης υπάρχει στη βάση. Ένας τρόπος είναι να μετρηθούν οι γραμμές που επέστρεψε. Αυτό γίνεται με την εντολή «mysql_num_rows»:

```
$number = mysql_num_rows($check_user);
if ($number == 0) {
//μπλοκ εντολών για αποτυχημένη σύνδεση
echo "Πρόβλημα στη σύνδεση. Παρακαλώ ξαναπροσπαθήστε. <br>";
echo "<a href=\"login.php\">Πίσω</a>";
exit;
}
Else {
//μπλοκ εντολών για επιτυχημένη σύνδεση
}
```

Η εντολή `exit` σταματάει την εκτέλεση του κώδικα. Σε περίπτωση επιτυχημένης σύνδεσης, θα πρέπει να μεταφερθεί ο χρήστης στην αρχική σελίδα του ιστοτόπου, τη `home.php`, η οποία θα είναι διαφορετική για `tutor/student` όπως αναφέρθηκε. Επίσης, είναι επιθυμητό, οι σελίδες να θυμούνται τα στοιχεία του χρήστη, για το λόγο αυτό, θα πρέπει αυτά να αποθηκεύονται χρησιμοποιώντας `sessions`. Για να μπορέσετε όμως να πάρετε τα στοιχεία του `resource` και να τα αποθηκεύσετε σε `session`, θα πρέπει πρώτα να μετατραπούν σε πίνακα με την εντολή `mysql_fetch_array`. Ο κώδικας που ακολουθεί περιέχει το μπλοκ των εντολών σε περίπτωση επιτυχημένης σύνδεσης.

```
$user = mysql_fetch_array($check_user);
$username = $user[0];
$name = $user["name"];
$surname = $user[3];
$email = $user["email"];
$role = $user["role"];
session_register("Username" , "Name" , "Surname" , "Email", "Role");
//Έλεγχος κατηγορίας χρήστη
if($role == "tutor"){
//Αν είναι τύπου tutor, κατεύθυνέ τον στην αρχική ιστοσελίδα για καθηγητές
header("Location: index.php");
}
elseif ($role == "student") {
//Αν είναι τύπου student, κατεύθυνέ τον στην αρχική ιστοσελίδα για μαθητές
header("Location: student/index.php");
}
else {
//Αν δεν είναι τίποτε από τα δύο, κατεύθυνέ τον στην ιστοσελίδα εισόδου ξανά
header("Location: login.php");
}
```

Η εντολή `mysql_fetch_array` παίρνει την πρώτη γραμμή ενός `resource` και τη μετατρέπει σε πίνακα. Στη συνέχεια αποθηκεύεται αυτός ο πίνακας στη μεταβλητή `$user`. Αν καλούνταν η `mysql_fetch_array` δύο φορές τότε αυτή θα έφερνε τη δεύτερη γραμμή κ.ο.κ. Κάθε φορά λοιπόν που καλείται η `mysql_fetch_array` φέρνει την επόμενη γραμμή ενός `resource`, μέχρις ότου να φτάσει στο τέλος, οπότε και επιστρέφει `false`.

Εφόσον η μεταβλητή `$user` περιέχει πίνακα, μπορούν να αποδοθούν οι τιμές της σε άλλες μεταβλητές. Θυμηθείτε ότι η προσπέλαση σε πίνακα μπορεί να γίνει, είτε με αριθμητική τιμή (οι πίνακες ξεκινάνε από το 0), είτε με το όνομα της στήλης. Με την εντολή `session_register` (πρέπει να έχει προηγηθεί η εντολή `session_start`) αποθηκεύονται οι τιμές των `Username`, `Name`, `Surname`, `Email` και `Role` στο `session`.

Τέλος, η εντολή `header` μεταφέρει αυτόματα το χρήστη στην ιστοσελίδα `index.php`, αντίστοιχα για κάθε τύπο χρήστη (`tutor/student`) Χρειάζεται ιδιαίτερη προσοχή το γεγονός ότι η εντολή `header` θα πρέπει να κληθεί πριν εμφανιστούν οποιεσδήποτε ετικέτες HTML στην ιστοσελίδα, διαφορετικά θα εμφανιστεί μήνυμα λάθους. Συνολικά ο κώδικας της ιστοσελίδας `check.php` είναι ο εξής:

```

<?php
session_start();

$hostname = "localhost";
$database = "mydb";
$username = "myuser";
$password = "mypswh";

$link = mysql_connect($hostname , $username , $password) or
die("Prosoxi!Provlima stin sundesi me ton server : " . mysql_error());

mysql_select_db($database,$link);

$query = "SELECT user.*
FROM `user`
WHERE user.username = '$user'
AND user.password = '$psw' ";

$check_user = mysql_query($query , $link) or
die("Prosoxi!Provlima stin epilogi pinaka user : " . mysql_error());

$num_rows = mysql_num_rows($check_user);

if ($num_rows == 0)
{
echo "Πρόβλημα στη σύνδεση. Παρακαλώ ξαναπροσπαθήστε. <br>";

echo "<a href='\"login.php\"'>Πίσω</a>";

exit;
}
else
{
$user = mysql_fetch_array($check_user);

$username = $user[0];

$name = $user["name"];

$surname = $user[3];

$email = $user["email"];

$role = $user ["role"];

session_register("Username" , "Name" , "Surname" , "Email", "Role");

if($role == "tutor"){
header("Location: index.php");}
}
elseif ($role == "student") {
header("Location: student/index.php");}

else {
header("Location: login.php");}
}

```

Αν ο χρήστης εισάγει σωστά τα στοιχεία του, τότε μεταφέρεται στην αντίστοιχη αρχική σελίδα του ιστοτόπου, την ιστοσελίδα home.php. Στο παρακάτω παράδειγμα θα παρουσιαστεί μόνο η ιστοσελίδα home.php για τον καθηγητή (ρόλο tutor), ωστόσο εσείς μπορείτε να εφαρμόσετε αυτά που θα αναλυθούν στη

συνέχεια σε όλες τις σελίδες της εφαρμογής σας. Η ιστοσελίδα home.php στο παράδειγμά μας, εμφανίζει τα στοιχεία του χρήστη, καθώς και τα στοιχεία του πίνακα user. Τα στοιχεία του χρήστη μπορεί να τα εμφανίσει χρησιμοποιώντας το session. Τα στοιχεία του πίνακα user όμως πρέπει να τα ανακτήσει από τη βάση μας. Έτσι λοιπόν θα πρέπει να συνδεθεί ξανά στη βάση με τον ίδιο τρόπο που συνδέθηκε και η ιστοσελίδα check.php. Το μόνο που αλλάζει είναι το query που θα εκτελέσουμε. Επίσης, είναι καλή πρακτική να ανακτάτε τις πληροφορίες που χρειάζεστε από τη βάση στην αρχή του κώδικά σας ώστε να τα έχετε συγκεντρωμένα. Έτσι λοιπόν η ιστοσελίδα index.php ξεκινάει με τον εξής κώδικα:

```
$hostname = "localhost";
$database = "mydb";
$username = "myuser";
$password = "myspw";

$link = mysql_connect($hostname , $username , $password) or

    die("Prosoxi!Provlima stin sundesi me ton server : " . mysql_error());

mysql_select_db($database,$link);

$Qget_data= "SELECT user.* FROM `user` ";
$get_data = mysql_query($Qget_data , $link) or

    die("Prosoxi!Provlima stin epilogi pinaka user : " . mysql_error());
```

Στη συνέχεια, η ιστοσελίδα θα πρέπει να εμφανίζει τα στοιχεία του χρήστη. Έτσι λοιπόν, μετά το πρώτο μπλοκ εντολών της php στο οποίο ανακτούμε δεδομένα από τη βάση μας, μπορούμε να έχουμε τον εξής κώδικα:

```
echo "Καλώς ήρθες $Name $Surname. Έχεις συνδεθεί με το username $Username";
```

Μετά από αυτό το μήνυμα θα πρέπει να εμφανίζονται τα δεδομένα του πίνακα user. Παρακάτω θα παρουσιαστεί ο τρόπος με τον οποίο θα μπορείτε να εμφανίσετε πολλές γραμμές ενός πίνακα, καθώς η εμφάνιση μιας μόνο γραμμής είναι ίδια με την εμφάνιση του παραπάνω μηνύματος (καλούμε μια φορά την «mysql_fetch_array»). Ο πιο απλός τρόπος για να εμφανίσετε πολλές γραμμές ενός πίνακα είναι να καλέσετε πολλές φορές τη συνάρτηση «mysql_fetch_array» μέσα σε μια δομή while. Έτσι, ο παρακάτω κώδικας εμφανίζει όλες τις γραμμές του πίνακα user.

```
while ($newrow = mysql_fetch_array($get_data))
{
echo "<p> Κεφάλαιο $newrow[0] <br> $newrow[1] <hr> </p>";
}
```

Τώρα η ιστοσελίδα index.php εμφανίζει αυτά που πρέπει. Ωστόσο, δεν είναι πλήρης, διότι ο οποιοσδήποτε θα μπορούσε να γράψει τη διεύθυνση της ιστοσελίδας index.php στη γραμμή διευθύνσεων και να δει τα περιεχόμενα της ιστοσελίδας. Επομένως, για να αποφευχθεί κάτι τέτοιο, θα πρέπει, σε κάθε ιστοσελίδα μετά την check.php, να γίνεται ένας έλεγχος για να διαπιστωθεί αν ο χρήστης έχει το δικαίωμα να δει τη συγκεκριμένη ιστοσελίδα ή ότι, με άλλα λόγια, έχει περάσει επιτυχώς από την ιστοσελίδα check.php. Η διαδικασία αυτή λέγεται authorization.

Ένας απλός τρόπος για να ελέγξετε αν ο χρήστης έχει δικαίωμα να δει τη συγκεκριμένη ιστοσελίδα είναι να δείτε αν υπάρχει ορισμένη μεταβλητή \$Username μέσα στο session. Να θυμίσουμε ότι η αποθήκευση της μεταβλητής \$Username μέσα στο session έγινε στην ιστοσελίδα check.php με χρήση της εντολής session_register. Ο κώδικας που ελέγχει αν έχει αποθηκευτεί η μεταβλητή \$Username μέσα στο session είναι ο εξής:

```
if (!isset($_SESSION['Username']))
{
echo "<h1>Απαγορεύεται η πρόσβαση!!!</h1>";
exit();
}
```

Η συνάρτηση isset επιστρέφει true αν η μεταβλητή που δέχεται ως όρισμα έχει οριστεί. Ο πίνακας \$_Session[] περιέχει όλες τις μεταβλητές του session. Τώρα ο κώδικας της ιστοσελίδας home.php είναι ολοκληρωμένος και φαίνεται παρακάτω:

```
<?php
session_start();
if (!isset($_SESSION['Username']))
{
echo "<h1>Access denied!!!</h1>";
exit();
}
$hostname = "localhost";
$databse = "mydb";
$username = "myuser";
$password = "myps";
$link = mysql_connect($hostname , $username , $password) or
die("Prosoxi!Provlima stin sundesi me ton server : " . mysql_error());
mysql_select_db($databse,$link);
$query = "SELECT user.* FROM `user` ";
$get_data = mysql_query($query , $link) or
die("Prosoxi!Provlima stin epilogi pinaka user : " . mysql_error());
echo "Καλώς ήρθες $Name $Surname. Έχεις συνδεθεί με το username $Username";
while ($newrow = mysql_fetch_array($get_data)){
echo "<p> Κεφάλαιο $newrow[0] <br> $newrow[1] <hr> </p>";
}
?>
```

Επειδή θα συμβεί αρκετές φορές να χρησιμοποιήσετε τον ίδιο ακριβώς κώδικα σε διάφορες σελίδες του ιστοτόπου σας, για λόγους οργάνωσης, καλό είναι να αποθηκεύσετε τα επαναλαμβανόμενα τμήματα κώδικα σε ξεχωριστά αρχεία και να τα καλείτε στις σελίδες που χρειάζεται κάθε φορά. Αυτό μπορεί να γίνει με τη συνάρτηση include.

Αρχείο dbconnect.php: Σε αυτό το αρχείο βρίσκεται ο κώδικας που χρησιμοποιείται για τη σύνδεση με τη βάση δεδομένων.

```
<?php
$hostname = "localhost";
$database = "mydb";
$username = "myuser";
$password = "mypswn";
$link = mysql_connect($hostname , $username , $password) or
die("Prosoxi!Provlima stin sundesi me ton server : " . mysql_error());
mysql_select_db($database,$link);
?>
```

Αρχείο auth.php: Σε αυτό το αρχείο βρίσκεται ο κώδικας που χρησιμοποιείται για να ελέγχουμε αν ένας χρήστης έχει δικαίωμα να δει μια συγκεκριμένη ιστοσελίδα. Αυτό γίνεται, διότι υπάρχει η περίπτωση ο χρήστης να μην είναι πιστοποιημένος ή να μην έχει πρόσβαση λόγω περιορισμένων δικαιωμάτων σε μία συγκεκριμένη ιστοσελίδα:

```
<?php
if (!isset($_SESSION['Username']))
{
echo "<h1>Access denied!!!</h1>";
exit();
}
?>
```

Έτσι, ο κώδικας της ιστοσελίδας home.php γίνεται:


```

<?php
session_start();
include("auth.php");
include("dbconenct.php");
$Qget_data = "SELECT user.* FROM `user` ";
$get_data = mysql_query($Qget_data , $link) or
die("Prosoxi!Provlima stin epilogi pinaka user : " . mysql_error());
echo "Καλώς ήρθες $Name $Surname. Έχεις συνδεθεί με το username $Username";
while ($newrow = mysql_fetch_array($get_data))
{ echo "<p> Κεφάλαιο $newrow[0] <br> $newrow[1] <hr> </p>"; }
?>

```

7.3.2. Αποσύνδεση χρήστη

Για να αποσυνδεθεί κάποιος και να τερματίσει τη σύνδεση του στον ιστότοπο, χρησιμοποιήστε το παρακάτω απόσπασμα κώδικα:

```

<?php
    session_start();
    unset($_SESSION);
    session_destroy();
    session_write_close();
    header('Location: login.php');
    die;
?>

```

7.3.3. Εισαγωγή δεδομένων σε έναν πίνακα μιας βάσης δεδομένων

Παρακάτω, θα παρουσιαστεί ο τρόπος με τον οποίο θα μπορέσετε να προσθέσετε μια νέα εγγραφή σε έναν πίνακα μιας βάσης δεδομένων. Για παράδειγμα, στην περίπτωση που θέλετε να προσθέσετε ένα νέο χρήστη στον πίνακα «user». Όπως είδατε στην αντίστοιχη ενότητα για την PHP, η εντολή INSERT INTO προσθέτει μια νέα εγγραφή σε έναν πίνακα μιας βάσης δεδομένων. Η σύνταξή της είναι η ακόλουθη:

```

INSERT INTO table_name
VALUES (value1, value2,....)

```

Επίσης, μπορούμε να ορίσουμε και τις στήλες στις οποίες θέλουμε να εισάγουμε δεδομένα:

```
INSERT INTO table_name (column1, column2,...)
VALUES (value1, value2,...)
```

Το αρχείο "insertuser.php" παρουσιάζει το σχετικό κώδικα για την εισαγωγή των χρηστών με στοιχεία username, password, name, surname, role στον πίνακα user της βάσης δεδομένων mydb: 'UNa', 'PASa', 'NAMEa', 'SURNAMEa', 'tutor' και 'UNbb', 'PASbb', 'NAMEbb', 'SURNAMEbb', 'student' αντίστοιχα.

```
<?php
$hostname = "localhost";
$database = "mydb";
$username = "myuser";
$password = "mypswh";

$link = mysql_connect($hostname , $username , $password) or
die("Prosoxi!Provlima stin sundesi me ton server : " . mysql_error());
if (!$link)
{
die('Could not connect: ' . mysql_error());
}
mysql_select_db("mydb", $link);
mysql_query("INSERT INTO user (username, password, name, surname, role)
VALUES ('UNa', 'PASa', 'NAMEa', 'SURNAMEa', 'tutor')")or
die("1-Prosoxi!Provlima stin epilogi pinaka user : " .
mysql_error());;
mysql_query("INSERT INTO user (username, password, name, surname, role)
VALUES ('UNbb', 'PASbb', 'NAMEbb', 'SURNAMEbb', 'student')")or
die("2-rosoxi!Provlima stin epilogi pinaka user : " . mysql_error());;
mysql_close($link);
?>
```

Προσοχή! Το παραπάνω παράδειγμα θα ολοκληρωθεί μόνο αν δώσετε προνόμια INSERT στο χρήστη (profile) «myuser» για τη βάση δεδομένων mydb (αυτό θα γίνει στο περιβάλλον phpmyadmin). Αλλιώς θα προκύψει το παρακάτω πρόβλημα:

```
Prosoxi!Provlima stin epilogi pinaka user : INSERT command denied to user
'myuser'@'localhost' for table 'user'
```

Ένας εναλλακτικός (και πιο γενικός) τρόπος για να εισάγονται δεδομένα, είναι με τη χρήση φόρμας. Στη συγκεκριμένη περίπτωση το παραπάνω script υλοποιείται σε δύο βήματα:

- Τη δημιουργία της φόρμας.
- Την επεξεργασία των στοιχείων της φόρμας με ένα PHP script.

Αρχικά, θα πρέπει να δημιουργήσετε την HTML φόρμα για την εισαγωγή των χρηστών με στοιχεία username, password, name, surname, role στον πίνακα user της βάσης δεδομένων mydb. Το αρχείο "inserthtml.html" παρουσιάζει το σχετικό κώδικα.

```
<html>
<body>
<form action="insertuserform.php" method="post">
username: <input type="email" name=" username" />
<p>
password: <input type="password" name="password" />
<p>
name: <input type="text" name="name" />
<p>
surname: <input type="text" name="surname " />
<p>
role: <input type="text" name="role" />
<p>
<input type="submit" />
</form>
</body>
</html>
```

Η παραπάνω φόρμα καλεί το PHP script insertuserform.php που αναλαμβάνει την επεξεργασία (εισαγωγή) των στοιχείων της φόρμας. Η αποστολή και λήψη των τιμών των παραμέτρων γίνεται με την χρήση της μεθόδου POST χρησιμοποιώντας τις μεταβλητές \$_POST της PHP. Το αρχείο "insertuserform.php" παρουσιάζει το σχετικό κώδικα.

```

<?php
$hostname = "localhost";
$databse = "mydb";
$username = "myuser";
$password = "myspw";
$link = mysql_connect($hostname , $username , $password) or
        die("Prosoxi!Provlima stin sundesi me ton server : " . mysql_error());
if (!$link)
{
die('Could not connect: ' . mysql_error());
}
mysql_select_db("mydb", $link);
$sql="INSERT INTO user (username, password, name, surname, email)
        VALUES
        ('$_POST[username]', '$_POST[password]', '$_POST[name]', '$_POST[surname]
        ', '$_P
        OST[role]')";
if (!mysql_query($sql,$link))
{
die('Error: ' . mysql_error());
}
echo "1 record added";
mysql_close($link);
?>

```

7.3.4. Επεξεργασία δεδομένων σε έναν πίνακα μιας βάσης δεδομένων

Για την επεξεργασία μίας υπάρχουσας εγγραφής ενός πίνακα στη βάση δεδομένων, θα πρέπει σε πρώτη φάση, να εντοπιστεί η συγκεκριμένη εγγραφή στη βάση και στη συνέχεια να γίνει η επεξεργασία της. Η αναζήτηση θα γίνει όπως φάνηκε και σε προηγούμενα παραδείγματα με την εντολή SELECT. Η επεξεργασία της εγγραφής και η ενημέρωσή της, θα γίνει με την εντολή UPDATE:

```

UPDATE table_name
SET column1=value, column2=value2,...
WHERE some_column=some_value

```

Το αρχείο update_user.php παρουσιάζει τη διαδικασία.

7.3.5. Αποστολή e-mail με χρήση PHP

Η PHP προσφέρει μια πολύ χρήσιμη δυνατότητα σχετικά με την αποστολή e-mail, μέσω ενός PHP script. Ο πιο απλός τρόπος για να στείλει κάποιος ένα e-mail με PHP είναι να στείλει ένα email με απλό κείμενο. Στο παρακάτω παράδειγμα αρχικά δηλώνουμε τις απαραίτητες μεταβλητές:

- \$to
- \$subject
- \$message
- \$from
- \$headers

Στη συνέχεια χρησιμοποιούμε τις μεταβλητές σαν παραμέτρους στη συνάρτηση mail() για να στείλουμε το e-mail:

```
<?php
$to = "tutor@cti.gr";
$subject = "Test mail";
$message = "GEIA XARA. TEST E-EMAIL.";
$from = "tutor@csd.auth.gr";
value
$headers = "From: $from";
mail($to,$subject,$message,$headers);
echo
?>
```

Με την PHP μπορεί να δημιουργηθεί και μια φόρμα αποστολής e-mail. Για παράδειγμα την αποστολή ερωτήσεων από τους φοιτητές στον καθηγητή σε έναν εκπαιδευτικό ιστότοπο. Το παρακάτω παράδειγμα αποστέλλει ένα μήνυμα κειμένου σε μια προκαθορισμένη e-mail διεύθυνση. Το αρχείο "formemail.php" παρουσιάζει το σχετικό κώδικα:

```

<html>
<body>
<?php
if (isset($_REQUEST['email']))
//Αν το "email" έχει συμπληρωθεί τότε γίνεται αποστολή
{
//αποστολή email
$email = $_REQUEST['email'] ;
$subject = $_REQUEST['subject'] ;
$message = $_REQUEST['message'] ;
mail("tsiatsos@cti.gr", "Subject: $subject",
$message, "From: $email");
echo "Thank you for using our mail form";
}
Else
//Αν το "email" /EN έχει συμπληρωθεί ξαναδημιουργείται η φόρμα
{
echo "<form method='post' action='formemail.php'>
Email: <input name='email' type='text' /><br />
Subject: <input name='subject' type='text' /><br />
Message:<br />
<textarea name='message' rows='15' cols='40'>
</textarea><br />
<input type='submit' />
</form>";
}
?>
</body>
</html>

```

7.3.6. Αποστολή και αποθήκευση αρχείου στον εξυπηρετητή με χρήση PHP

Στο παρακάτω παράδειγμα θα παρουσιαστεί μια πολύ χρήσιμη δυνατότητα της PHP σχετικά με την αποστολή (upload) και την αποθήκευση ενός αρχείου στον web server. Το συγκεκριμένο παράδειγμα αποθηκεύει το αρχείο που θα ανεβάσει ο χρήστης σε έναν νέο φάκελο με το όνομα «upload». Τα βήματα που θα ακολουθηθούν για την υλοποίηση του παραδείγματος είναι τα ακόλουθα:

- Δημιουργία της HTML φόρμας για το ανέβασμα του αρχείου.
- Δημιουργία του PHP script για την αποστολή του αρχείου.
- Προσθήκη περιορισμών κατά την αποστολή ενός αρχείου.
- Αποθήκευση του απεσταλμένου αρχείου.

Το πρώτο βήμα στην υλοποίηση του παραδείγματος είναι η δημιουργία HTML φόρμας για το ανέβασμα του αρχείου η οποία θα ονομαστεί «upload_form.html»:

Αρχείο upload_form.html

```
<html>
<body>
<form action="upload_file.php" method="post"
enctype="multipart/form-data">
<label for="file">Filename:</label>
<input type="file" name="file" id="file" />
<br />
<input type="submit" name="submit" value="Submit" />
</form>
</body>
</html>
```

Παρατηρήστε τα παρακάτω σημαντικά στοιχεία σχετικά με τον παραπάνω κώδικα:

- Το χαρακτηριστικό enctype της ετικέτας <form> καθορίζει τον τύπο του περιεχομένου (content-type) που θα χρησιμοποιηθεί κατά την υποβολή της φόρμας. Η τιμή "multipart/form-data" χρησιμοποιείται όταν η φόρμα απαιτεί την υποβολή binary δεδομένων (όπως τα δεδομένα ενός αρχείου).
- Το χαρακτηριστικό type="file" της ετικέτας <input> καθορίζει ότι η είσοδος θα πρέπει να γίνει επεξεργάσιμη σαν ένα αρχείο. Για παράδειγμα, κατά την προβολή της φόρμας σε έναν web browser, θα υπάρχει ένα browse-button δίπλα στο πεδίο input.

Τα δεδομένα που υποβάλλονται με την προηγούμενη φόρμα θα πρέπει να επεξεργαστούν κατάλληλα από το αντίστοιχο PHP script. Το αρχείο "upload_file.php" παρουσιάζει το σχετικό κώδικα:

Αρχείο upload_file.php

```

<?php
if ($_FILES["file"]["error"] > 0){
echo "Error: " . $_FILES["file"]["error"] . "<br />"; }
else {
echo "Upload: " . $_FILES["file"]["name"] . "<br />";
echo "Type: " . $_FILES["file"]["type"] . "<br />";
echo "Size: " . ($_FILES["file"]["size"] / 1024) . " Kb<br />";
echo "Stored in: " . $_FILES["file"]["tmp_name"]; }
?>

```

Όπως φαίνεται από το παραπάνω απόσπασμα κώδικα, για την αποστολή ενός αρχείου από έναν τοπικό υπολογιστή σε έναν απομακρυσμένο εξυπηρετητή χρησιμοποιείται ο σφαιρικός πίνακας της PHP με όνομα `$_FILES`. Η πρώτη παράμετρος είναι το input name της φόρμας (εδώ το "file") και ο δεύτερος δείκτης μπορεί να είναι "name", "type", "size", "tmp_name" ή "error". Για παράδειγμα:

- `$_FILES["file"]["name"]` – το όνομα του αρχείου που αποστέλλεται.
- `$_FILES["file"]["type"]` – ο τύπος του αρχείου που αποστέλλεται.
- `$_FILES["file"]["size"]` – το μέγεθος σε bytes του αρχείου που αποστέλλεται.
- `$_FILES["file"]["tmp_name"]` – το όνομα του προσωρινού αντίγραφου του αρχείου που αποθηκεύεται στον εξυπηρετητή.
- `$_FILES["file"]["error"]` – ο κωδικός λάθους που προκύπτει από την αποστολή του αρχείου.

Πρέπει να σημειωθεί πως αυτός είναι ο απλούστερος τρόπος αποστολής ενός αρχείου. Για λόγους ασφάλειας θα πρέπει συνήθως να επιβάλλουμε περιορισμούς στο τι ακριβώς θα έχει την δυνατότητα να ανεβάσει ο χρήστης. Ένα παράδειγμα φαίνεται στην επόμενη παράγραφο, όπου ο χρήστης θα πρέπει να αποστέλλει μόνο .gif ή .jpeg αρχεία με μέγεθος μέχρι 20 kb. Το αρχείο "upload_filer.php" παρουσιάζει το σχετικό κώδικα:


```

<?php
if ((($_FILES["file"]["type"] == "image/gif")
|| ($_FILES["file"]["type"] == "image/jpeg")
|| ($_FILES["file"]["type"] == "image/pjpeg"))
&& ($_FILES["file"]["size"] < 20000))
{
if ($_FILES["file"]["error"] > 0)
{
echo "Error: " . $_FILES["file"]["error"] . "<br />";
}
else
{
echo "Upload: " . $_FILES["file"]["name"] . "<br />";
echo "Type: " . $_FILES["file"]["type"] . "<br />";
echo "Size: " . ($_FILES["file"]["size"] / 1024) . " Kb<br />";
echo "Stored in: " . $_FILES["file"]["tmp_name"];
}
}
else
{
echo "Invalid file";
}
?>

```

Τα παραπάνω παραδείγματα δημιουργούν ένα προσωρινό αντίγραφο του απεσταλμένου αρχείου στον PHP temp φάκελο του εξυπηρετητή. Το προσωρινό αντίγραφο εξαφανίζεται όταν το script τερματίζει. Για την αποθήκευση του απεσταλμένου αρχείου χρειάζεται να αντιγράψετε το αρχείο σε μια άλλη θέση. Το αρχείο «upload_filef.php» παρουσιάζει το σχετικό κώδικα. Το παρακάτω script ελέγχει αν το αρχείο υπάρχει ήδη και αν όχι αντιγράφει το αρχείο στον προκαθορισμένο φάκελο.

```

<?php
if (($_FILES["file"]["type"] == "image/gif")
|| ($_FILES["file"]["type"] == "image/jpeg")
|| ($_FILES["file"]["type"] == "image/pjpeg")
&& ($_FILES["file"]["size"] < 20000))
{
if ($_FILES["file"]["error"] > 0)
{
echo "Return Code: " . $_FILES["file"]["error"] . "<br />";
}
else
{
echo "Upload: " . $_FILES["file"]["name"] . "<br />";
echo "Type: " . $_FILES["file"]["type"] . "<br />";
echo "Size: " . ($_FILES["file"]["size"] / 1024) . " Kb<br />";
echo "Temp file: " . $_FILES["file"]["tmp_name"] . "<br />";
if (file_exists("upload/" . $_FILES["file"]["name"]))
{
echo $_FILES["file"]["name"] . " already exists. ";
}
else
{
move_uploaded_file($_FILES["file"]["tmp_name"],
"upload/" . $_FILES["file"]["name"]);
echo "Stored in: " . "upload/" . $_FILES["file"]["name"];
}
}
}
else
{
echo "Invalid file";
}
?>

```

Αυτά ήταν τα βασικά scripts τα οποία είναι απαραίτητα για τη δημιουργία του δικού σας δυναμικού ιστοτόπου. Την ίδια λογική και φιλοσοφία θα πρέπει να ακολουθήσετε για την υλοποίηση και των υπόλοιπων ιστοσελίδων. Το γραφικό κομμάτι είναι αποκλειστικά δική σας επιλογή, αλλά οι διαδράσεις με τη βάση δεδομένων καλύφθηκαν σε έναν αρκετά μεγάλο βαθμό.

Για περισσότερες πληροφορίες προτείνονται ορισμένοι χρήσιμοι σύνδεσμοι:

- www.apache.org, σχετικά με τον Apache.
- www.php.net, σχετικά με την PHP.
- www.mysql.com, σχετικά με την MySQL.

Σχετικό εκπαιδευτικό υλικό είναι διαθέσιμο στους παρακάτω δικτυακούς τόπους:

- <http://www.w3schools.com/php/default.asp>, σχετικά με την PHP.
- <http://www.w3schools.com/html/default.asp>, σχετικά με την HTML.
- http://www.w3schools.com/php/php_mysql_intro.asp, σχετικά με την PHP – MySQL.

Βιβλιογραφία/Αναφορές

W3Schools (2015). Ανακτήθηκε από <http://www.w3schools.com/>

Welling, L., & Thomson, L. (2009). *Ανάπτυξη Web Εφαρμογών με PHP και MySQL*. Εκδόσεις Γκιούρδας.

Κριτήρια αξιολόγησης

Τα κριτήρια αξιολόγησης του παρόντος κεφαλαίου είναι διαθέσιμα στον [βοηθητικό ιστότοπο του παρόντος συγγράμματος](#) και στην ιστοσελίδα Κριτήρια αξιολόγησης για το [Κεφάλαιο 7 - Υλοποίηση Δυναμικού Εκπαιδευτικού Ιστοτόπου](#).