



Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης
Σχολή Τεχνολογικών Εφαρμογών Τμήμα
Μηχανικών Πληροφορικής

Πτυχιακή Εργασία

" Σχεδίαση & υλοποίηση ηλεκτρονικού καταστήματος εισιτηρίων για
μουσείο "



Κνέκνας Χριστόδουλος:3677

Επιβλέπων καθηγητής: Παπαδάκης Νικόλαος

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

Περίληψη	3
Summary	3

ΚΕΦΑΛΑΙΟ 1

Στατικές ιστοσελίδες	3
Δυναμικές ιστοσελίδες	4

ΚΕΦΑΛΑΙΟ 2

Γνωριμία με την HTML	5
Δομής της HTML	7
Βασικές ετικέτες	8
CSS	10
PHP	11

ΚΕΦΑΛΑΙΟ 3

Ιστοσελίδα της συγκεκριμένης εργασίας	15
Ευχαριστίες	26
Βιβλιογραφία	26

Περίληψη

Σε αυτό το site, παρουσιάζετε το Κυπριακό Μουσείο στη Λευκωσία. Είναι ένα αρκετά ενημερωμένο πρόγραμμα πλοήγησης που αναμφισβήτητα παρουσιάζει τις πιο σημαντικές πτυχές που θα πρέπει να περιλαμβάνει ένα site για μουσείο. Το site περιλαμβάνει φωτογραφικό υλικό από το μουσείο και πληροφορίες σχετικά με τα εκθέματα της κάθε αίθουσας. Θα πρέπει να σημειωθεί ότι ο χρήστης έχει τη δυνατότητα σύνδεσης στο site με τη χρήση email και θα μπορεί να αγοράζει εισιτήρια για την είσοδο στο μουσείο από το online κατάστημα.

Στο μουσείο εκτίθενται ευρήματα από τις πρωϊμότερες περιόδους της ανθρώπινης παρουσίας στο νησί, από τη 10η χιλιετία π.Χ., μέχρι και τη ρωμαϊκή περίοδο. Η βυζαντινή περίοδος εκπροσωπείται από πολύ λίγα ευρήματα μιας το Κυπριακό Μουσείο είναι κατεξοχήν αρχαιολογικό.

Summary

On this site we presenting the Cyprus Museum of Nicosia. It is a fairly updated browser that undoubtedly presents the most important aspects that should include a museum site. The site includes photographs from the museum and information about the exhibits of each room. It should be noted that the user has the ability to connect to the site using email and will be able to buy entrance tickets from the online store.

The museum exhibits finds from the earliest periods of human presence on the island since the 10th millennium BC, to the Roman period. The Byzantine period is represented by very few findings of the Cyprus Museum is eminently archaeological.

Στατικές Ιστοσελίδες

Στατική ιστοσελίδα (static web-page ή flat web-page) ονομάζεται μια ιστοσελίδα της οποίας το περιεχόμενο μεταφέρεται στον χρήστη ακριβώς στην μορφή που είναι αποθηκευμένο στον εξυπηρετητή ιστοσελίδων (web server), σε αντίθεση με τις δυναμική ιστοσελίδα όπου το περιεχόμενο δημιουργείται από μια εφαρμογή η οποία εκτελείται στον εξυπηρετητή ιστοσελίδων. Τα περιεχόμενα μιας στατικής ιστοσελίδας εμφανίζονται με την ίδια μορφή σε όλους του χρήστες με την μορφή που είναι αποθηκευμένα στο σύστημα αρχείων του εξυπηρετητή ιστοσελίδων. Οι στατικές ιστοσελίδες είναι αποθηκευμένες συνήθως σε μορφή HTML και μεταφέρονται χρησιμοποιώντας το πρωτόκολλο HTTP.

Στα πλεονεκτήματα των στατικών σελίδων περιλαμβάνεται το γεγονός ότι η σελίδα μπορεί να βρίσκεται με εύκολο τρόπο κλωνοποιημένη σε περισσότερους από ένα εξυπηρετητές. Επίσης δε χρειάζεται ειδικό λογισμικό στον εξυπηρετητή ιστοσελίδων για τη δημοσίευση στατικών σελίδων. Τέλος, η σελίδα μπορεί να είναι διαθέσιμη στον φυλλομετρητή κατευθείαν από ένα αποθηκευτικό μέσο όπως ένα CD-ROM ή ένα USB stick μνήμης χωρίς να χρειάζεται να διαμεσολαβήσει ένας εξυπηρετητής ιστοσελίδων με κατάλληλο λογισμικό χειρισμό ιστοσελίδων (για παράδειγμα οι δυναμικές σελίδες σε γλώσσα PHP θέλουν ένα διεργασμένο PHP ο οποίος τρέχει στον εξυπηρετητή ιστοσελίδων και δημιουργεί δυναμικά τη σελίδα πριν αυτή αποσταλεί στον φυλλομετρητή του χρήστη).

Στον αντίποδα, ένα χαρακτηριστικό μειονέκτημα των στατικών ιστοσελίδων είναι ότι δεν είναι εύκολη η διαδραστικότητα με τον χρήστη και η διαχείριση μεγάλου αριθμού στατικών ιστοσελίδων δεν είναι εύκολη χωρίς αυτόματα εργαλεία.

Δυναμικές Ιστοσελίδες

Δυναμική ιστοσελίδα (dynamic web page) είναι μια ιστοσελίδα η οποία δημιουργείται δυναμικά την στιγμή της πρόσβασης σε αυτή ή την στιγμή που ο χρήστης αλληλεπιδρά με τον εξυπηρετητή ιστοσελίδων. Οι δυναμικές ιστοσελίδες θεωρούνται δομικό στοιχείο της νέας γενιάς του παγκόσμιου ιστού (Web 2.0) όπου η πληροφορία διαμοιράζεται σε πολλαπλές ιστοσελίδες.

Η δυναμική ιστοσελίδα μπορεί να δημιουργείται δυναμικά από ένα σενάριο εντολών, το οποίο εκτελείται τοπικά στο πελάτη ή στον εξυπηρετητή ή και στον πελάτη και στον εξυπηρετητή. Είναι δύσκολο να πει κάποιος με ακρίβεια πότε οι "δυναμικές ιστοσελίδες" ξεκίνησαν αλλά η γενικότερη ιδέα ξεκίνησε μετά την εξάπλωση και ανάπτυξη των ιστοσελίδων. Το πρωτόκολλο HTTP άρχισε να χρησιμοποιείται από το 1990 και η γλώσσα HTML ξεκίνησε να χρησιμοποιείται για τον σχεδιασμό ιστοσελίδων από το 1996. Η εξάπλωση χρήσης των φυλλομετρητών ξεκίνησε το 1993 με τον φυλλομετρητή Mosaic. Η ιδέα της δημιουργίας δυναμικών ιστοσελίδων υπήρχε και πριν την ανάπτυξη των ιστοσελίδων. Για παράδειγμα πριν το 1990 υπήρχαν "Βάσεις Δεδομένων" (Bulletin Board Systems) στα οποία οι χρήστες συνδέονταν μέσω μόντεμ και το περιεχόμενο που έβλεπαν δημιουργούνταν δυναμικά.

Η σελίδα συνήθως περιέχει ένα σενάριο εντολών το οποίο ενσωματώνει μια διαδραστικότητα με τον χρήστη (η διαδραστικότητα γίνεται με χρήση πληκτρολογίου - ποντικιού). Το περιεχόμενο της ιστοσελίδας δημιουργείται δυναμικά στον τοπικό υπολογιστή μετά την εκτέλεση του σεναρίου εντολών το οποίο λαμβάνεται από τον απομακρυσμένο εξυπηρετητή.

Για παράδειγμα η γλώσσα σεναρίου εντολών JavaScript ή ActionScript(γλώσσα σεναρίων της Macromedia - σήμερα είναι η εταιρία Adobe Systems) χρησιμοποιείται για την τοπική δημιουργία δυναμικών ιστοσελίδων τεχνολογίας DHTML ή Flash οι οποίες συχνά χρησιμοποιούνται για την ενσωμάτωση ήχων, κινούμενων εικόνων ή αλλαγές και διαμόρφωση του κείμενου της σελίδας. Η γλώσσα σεναρίου η οποία εκτελείται τοπικά μπορεί να καλεί απομακρυσμένα σενάρια εντολών στον εξυπηρετητή ιστοσελίδων χρησιμοποιώντας τεχνολογίες όπως hidden Frame, XMLHttpRequest ή άλλες υπηρεσίες ιστού (web services).

Η πρώτη "ευρέως διαδεδομένη" έκδοση της γλώσσας σεναρίου εντολών JavaScript ενσωματώθηκε το 1996 στο φυλλομετρητή Netscape 3.

Μια δυναμική ιστοσελίδα συνήθως δημιουργείται από ένα λογισμικό το οποίο τρέχει στον απομακρυσμένο εξυπηρετητή ιστοσελίδων. Ο χρήστης-πελάτης επισκέπτεται τον εξυπηρετητή ιστοσελίδων ο οποίος δημιουργεί δυναμικά την ιστοσελίδα την οποία επιστρέφει σε μορφή HTML στον πελάτη. Τέτοιες τεχνολογίες για την δημιουργία δυναμικών ιστοσελίδων στον εξυπηρετητή είναι η τεχνολογία ASP, ColdFusion, Perl, PHP, WebDNA και άλλες. Οι γλώσσες αυτές συνήθως (με εξαίρεση την ASP.NET και την JSP) χρησιμοποιούν ένα σενάριο εντολών CGI κατά την δυναμική δημιουργία των σελίδων.

Πολλές φορές οι σελίδες είναι αποθηκευμένες στατικά σε κρυφή-προσωρινή μνήμη και αποστέλλονται κατευθείαν στο πελάτη. Με αυτό το τρόπο αποφεύγεται η υπερφόρτωση του εξυπηρετητή ιστοσελίδων και καλείται η μηχανή δημιουργία δυναμικών ιστοσελίδων μόνο όταν το αποθηκευμένο περιεχόμενο έχει αλλάξει.

Η τεχνολογία AJAX χρησιμοποιεί σενάρια εντολών και στον πελάτη και στον απομακρυσμένο εξυπηρετητή για την δημιουργία της δυναμικής ιστοσελίδας. Στην τεχνολογία AJAX γίνεται ανταλλαγή περιεχομένου και δεδομένων από τον υπολογιστή του πελάτη και του απομακρυσμένου εξυπηρετητή και η σελίδα διαμορφώνεται από το σενάριο εντολών το οποίο τρέχει τοπικά στο πελάτη. Το πλεονέκτημα της τεχνολογίας αυτής είναι ότι περιορίζεται η χρήση του εξυπηρετητή ιστοσελίδων και αποστέλλεται μόνο το περιεχόμενο. Για παράδειγμα οι χάρτες της Google είναι ένα παράδειγμα δικτυακής εφαρμογής η οποία χρησιμοποιεί την τεχνολογία AJAX.

HTML

Η HTML (HyperText Markup Language, Γλώσσα Σήμανσης Υπερκειμένου) είναι η κύρια γλώσσα σήμανσης για τις ιστοσελίδες, και τα στοιχεία της είναι τα βασικά δομικά στοιχεία των ιστοσελίδων. Η HTML γράφεται υπό μορφή στοιχείων HTML τα οποία αποτελούνται από ετικέτες, οι οποίες περικλείονται μέσα σε σύμβολα «μεγαλύτερο από» και «μικρότερο από» (για παράδειγμα <html>), μέσα στο περιεχόμενο της ιστοσελίδας. Οι ετικέτες HTML συνήθως λειτουργούν ανά ζεύγη (για παράδειγμα <h1> και </h1>), με την πρώτη να ονομάζεται ετικέτα έναρξης και τη δεύτερη ετικέτα λήξης (ή σε άλλες περιπτώσεις ετικέτα ανοίγματος και ετικέτα κλεισίματος αντίστοιχα). Ανάμεσα στις ετικέτες, οι σχεδιαστές ιστοσελίδων μπορούν να τοποθετήσουν κείμενο, πίνακες, εικόνες κλπ. Ο σκοπός ενός web browser είναι να διαβάσει τα έγγραφα HTML και τα συνθέσει σε σελίδες που μπορεί κανείς να διαβάσει ή να ακούσει. Ο browser δεν εμφανίζει τις ετικέτες HTML, αλλά τις χρησιμοποιεί για να ερμηνεύσει το περιεχόμενο της σελίδας.

Τα στοιχεία της HTML χρησιμοποιούνται για να κτίσουν όλους του ιστότοπους. Η HTML επιτρέπει την ενσωμάτωση εικόνων και άλλων αντικειμένων μέσα στη σελίδα, και μπορεί να χρησιμοποιηθεί για να εμφανίσει διαδραστικές φόρμες. Παρέχει τις μεθόδους δημιουργίας δομημένων εγγράφων (δηλαδή εγγράφων που αποτελούνται από το περιεχόμενο που μεταφέρουν και από τον κώδικα μορφοποίησης του περιεχομένου) καθορίζοντας δομικά σημαντικά στοιχεία για το κείμενο, όπως κεφαλίδες, παραγράφους, λίστες, συνδέσμους, παραθέσεις και άλλα. Μπορούν επίσης να ενσωματώνονται σενάρια εντολών σε γλώσσες όπως η JavaScript, τα οποία επηρεάζουν τη συμπεριφορά των ιστοσελίδων HTML.

Οι Web browsers μπορούν επίσης να αναφέρονται σε στυλ μορφοποίησης CSS για να ορίζουν την εμφάνιση και τη διάταξη του κειμένου και του υπόλοιπου υλικού. Ο οργανισμός W3C, ο οποίος δημιουργεί και συντηρεί τα πρότυπα για την HTML και τα CSS, ενθαρρύνει τη χρήση των CSS αντί διαφόρων στοιχείων της HTML για σκοπούς παρουσίασης του περιεχομένου.

Τι είναι οι ετικέτες;

Οι **ετικέτες** ελέγχουν την δομή και την μορφή του κειμένου της ιστοσελίδας. Επίσης παρέχουν πληροφορίες προς τον web browser για την σελίδα που πρόκειται να εμφανίσουν, όπως ο τίτλος της σελίδας ή ο συγγραφέας της, κ.α.

Πώς είναι μια ετικέτα ;

Οι HTML **ετικέτες** γράφονται ανάμεσα στα σύμβολα < και >
πχ. <όνομα-ετικέτας>

Οι περισσότερες HTML ετικέτες αποτελούνται από μια **ετικέτα αρχής** και μια **ετικέτα τέλους** και ανάμεσα σε αυτές υπάρχει το κείμενο που χαρακτηρίζεται από τις ετικέτες αυτές. Η ετικέτα τέλους περιέχει τον χαρακτήρα /πριν το όνομα της ετικέτας.

πχ. <όνομα-ετικέτας> ... κείμενο ... </όνομα-ετικέτας>

πχ. Κείμενο με έμφαση

Υπάρχουν ορισμένες ετικέτες που δεν έχουν ετικέτες τέλους. Στις ετικέτες αυτές, πριν από το σύμβολο >τοποθετούμε τον χαρακτήρα /

πχ. <όνομα-ετικέτας />

πχ.

Το αρχείο που περιέχει HTML ετικέτες λέγεται HTML αρχείο και έχει επέκταση **.html** ή **.htm** (εκτός βέβαια κι' αν η ιστοσελίδα είναι δυναμική οπότε έχει επεκτάσεις όπως .php, asp, jsr κτλ.). Τα αρχεία

αυτά είναι απλά αρχεία κειμένου σε μορφή ASCII και δεν περιέχουν πληροφορίες για το περιβάλλον ή τα προγράμματα με τα οποία θα λειτουργήσουν. Μπορείτε να ανοίξετε και να δείτε τα αρχεία htm ή html με οποιονδήποτε επεξεργαστή κειμένου, π.χ. Σημειωματάριο (Notepad) των Windows

Πώς είναι μια ετικέτα :

Οι HTML **ετικέτες** γράφονται ανάμεσα στα σύμβολα < και >
πχ. <όνομα-ετικέτας>

Οι περισσότερες HTML ετικέτες αποτελούνται από μια **ετικέτα αρχής** και μια **ετικέτα τέλους** και ανάμεσα σε αυτές υπάρχει το κείμενο που χαρακτηρίζεται από τις ετικέτες αυτές. Η ετικέτα τέλους περιέχει τον χαρακτήρα /πριν το όνομα της ετικέτας.

πχ. <όνομα-ετικέτας> ... κείμενο ... </όνομα-ετικέτας>

πχ. Κείμενο με έμφαση

Υπάρχουν ορισμένες ετικέτες που δεν έχουν ετικέτες τέλους. Στις ετικέτες αυτές, πριν από το σύμβολο >τοποθετούμε τον χαρακτήρα /

πχ. <όνομα-ετικέτας />

πχ.

Το αρχείο που περιέχει HTML ετικέτες λέγεται HTML αρχείο και έχει επέκταση **.html** ή **.htm** (εκτός βέβαια κι' αν η ιστοσελίδα είναι δυναμική οπότε έχει επεκτάσεις όπως .php, asp, jsp κτλ.). Τα αρχεία αυτά είναι απλά αρχεία κειμένου σε μορφή ASCII και δεν περιέχουν πληροφορίες για το περιβάλλον ή τα προγράμματα με τα οποία θα λειτουργήσουν. Μπορείτε να ανοίξετε και να δείτε τα αρχεία htm ή html με οποιονδήποτε επεξεργαστή κειμένου, π.χ. Σημειωματάριο (Notepad) των Windows

Πώς λειτουργεί μια ετικέτα μέσα σε ένα HTML αρχείο :

Ένα **ζευγάρι ετικετών** καθορίζει την μορφή ενός κειμένου ή μιας και μόνο λέξης. Για παράδειγμα αν ήθελα να γράψω στην γλώσσα HTML το κείμενο 'Μαθαίνω HTML' με έντονη γραφή, θα το έγραφα έτσι:

Μαθαίνω HTML. Μόλις συναντήσαμε την πρώτη HTML ετικέτα. Το γράμμα **b** είναι το αρχικό από την λέξη **bold**. Το ζευγάρι των ετικετών **** και ****, δηλώνει στον web browser, ότι το κείμενο που είναι γραμμένο μεταξύ των ετικετών αυτών θα εμφανιστεί με έντονα (bold) γράμματα.

Αν πάλι ήθελα να γράψω την λέξη wlearn με πλάγιους χαρακτήρες θα έγραφα:

<i>wlearn</i>

Το ζευγάρι των ετικετών **<i>** και **</i>**, λέει στον browser να εμφανίσει το κείμενο που είναι γραμμένο ανάμεσα σε αυτές τις ετικέτες, με πλάγιους χαρακτήρες. Το γράμμα **i** είναι το πρώτο γράμμα από την λέξη **italics**.

Οι ετικέτες μέσα σε ένα HTML αρχείο μπορούν να είναι γραμμένες είτε με μικρά γράμματα (πεζά), είτε με κεφαλαία. Η ετικέτα **** είναι ίδια με τη ετικέτα ****. Μόνο πρέπει να προσέχουμε τις ετικέτες τέλους να είναι γραμμένες όπως η ετικέτες αρχής. Δηλαδή αν μια ετικέτα αρχής είναι γραμμένη με πεζά τότε και η ετικέτα τέλους πρέπει να γραφεί με πεζά.

Λάθος τρόπος:

wlearn

Σωστός τρόπος:

wlearn

ή

wlearn

Η δομή της HTML

Ένα αρχείο HTML αρχίζει πάντα με την ετικέτα `<html>` και αποτελείται από δύο ενότητες: την κεφαλή (**HEAD**) και το κυρίως περιεχόμενο (**BODY**) ή αλλιώς το "σώμα" της σελίδας όπως συνήθως το αποκαλούμε

```
Ενότητα head
<html>
<head>
<title>My first web site</title>
</head>

Ενότητα body
<body>
This is <b>Great</b>!!! <b>YEAH!!!</b><br>
I can build my own <i>web site</i>. <b>YEAH!!!</b><br>
<i>Hey Ma look!!!</i> I can do it by <b>myself</b>
</body>
</html>
```

Η ετικέτα <HTML>

Με την ετικέτα `<html>` αρχίζουμε πάντα τον κώδικα μας και με την ετικέτα `</html>` τον τερματίζουμε. Με αυτόν τον τρόπο πληροφορούμε τον browser ότι οι γραμμές που περικλείονται μέσα σε αυτές τις δύο ετικέτες είναι κώδικας γραμμένος σε γλώσσα HTML.

Η Ενότητα HEAD

Η πρώτη ενότητα (ενότητα HEAD) μιας HTML σελίδας ορίζεται με τις ετικέτες `<head>...</head>`. Οι ετικέτες που γράφονται στην ενότητα **HEAD**, αποτελούν τον πρόλογο για την HTML σελίδα. Υπάρχουν μόνο λίγες ετικέτες που γράφονται στην ενότητα αυτή. Η πιο βασική από αυτές είναι η ετικέτα `<title>`, η οποία καθορίζει τον τίτλο της σελίδας ο οποίος εμφανίζεται στο πάνω μέρος του παραθύρου του web browser.

```
<title>WLearn</title>
```

Η Ενότητα BODY

Η δεύτερη ενότητα (ενότητα body) ορίζεται με τις ετικέτες `<body>...</body>`. Το ζευγάρι των ετικετών `<body>` και `</body>` ορίζει το **κυρίως περιεχόμενο της σελίδας** μέσα στο οποίο γράφουμε το κείμενο που θέλουμε να εμφανιστεί μαζί με τις HTML ετικέτες που το μορφοποιούν. Στην ενότητα αυτή τοποθετούμε επίσης εικόνες, video και ότι άλλο θέλουμε να εμφανιστεί στην σελίδα.

Μπορείτε να δείτε την [Λίστα ετικετών](#), όπου περιγράφονται αναλυτικά όλες οι ετικέτες της HTML

Βασικές ετικέτες

Οι πιο βασικές HTML ετικέτες καθορίζουν επικεφαλίδες, παραγράφους και γραμμές

Επικεφαλίδες

Οι HTML επικεφαλίδες είναι κείμενο που εμφανίζετε με μεγάλα και έντονα γράμματα. Οι επικεφαλίδες ορίζονται από τις ετικέτες `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>` και `<h6>`. Με την `<h1>` ορίζουμε την μεγαλύτερη ετικέτα ενώ με την `<h6>` την μικρότερη.

Παράγραφοι

Οι παράγραφοι ορίζονται από το ζευγάρι ετικετών `<p>` και `</p>`

Αλλαγή γραμμής

Η ετικέτα `
` χρησιμοποιείται όταν θέλουμε να τελειώσουμε μια γραμμή κειμένου και να αρχίσουμε μια καινούργια

Οριζόντια γραμμή

Η οριζόντια γραμμή είναι ένα απλό γραφικό που μπορούμε να χρησιμοποιήσουμε στην σελίδα μας (κυρίως σαν διαχωριστικό). Με την ετικέτα `<hr>` τοποθετούμε μια οριζόντια γραμμή στην σελίδα μας.

Ιδιότητες ετικετών

Οι *ιδιότητες* (attributes) των ετικετών είναι τιμές που δίνουν στην ετικέτα διάφορα χαρακτηριστικά. Κάθε μια από αυτές τις τιμές επιδρά διαφορετικά στην εμφάνιση ή την λειτουργία των ετικετών. Μια ιδιότητα μπαίνει αμέσως μετά το όνομα της ετικέτας και αποτελείται από το όνομα της και μια τιμή μέσα σε διπλά εισαγωγικά.

Μια ετικέτα με ιδιότητες είναι κάπως έτσι:

```
<όνομα-ετικέτας ιδιότητα1="τιμη" ιδιότητα2="τιμη" ιδιότητα3="τιμη">...</όνομα-ετικέτας>
```

Η ετικέτα `<p>`, για παράδειγμα, μπορεί να πάρει την ιδιότητα `align` η οποία ορίζει την στοίχιση του κειμένου μέσα στην παράγραφο. Η ιδιότητα `align` παίρνει μια από τις τιμές: `left`, `center`, `right`, `justify`

Διαμόρφωση κειμένου

Στην HTML υπάρχουν ετικέτες που ορίζουν την εμφάνιση του κειμένου. Παρακάτω παρουσιάζονται αυτές οι ετικέτες

Η ετικέτα ``, `` και `<big>`

Και οι τρεις αυτές ετικέτες εμφανίζουν το κείμενο με έντονη μορφή. Η `` και `` έχουν την ίδια ακριβώς επίδραση στο κείμενο, ενώ η `<big>` εμφανίζει το κείμενο με έντονη μορφή όπως οι `` και `` αλλά με ελαφρώς μεγαλύτερα γράμματα.

Η ετικέτα ``

Η ετικέτα `` εμφανίζει το κείμενο με πλάγιους και κάπως αχνά γραμμένους χαρακτήρες. Το `em` είναι τα δύο πρώτα γράμματα από την λέξη *emphasize*.

Η ετικέτα `<i>`

Η ετικέτα `<i>` εμφανίζει το κείμενο με πλάγιους χαρακτήρες. Το *i* είναι το πρώτο γράμμα από την λέξη *italics*

Η ετικέτα `<u>`

Η ετικέτα `<u>` εμφανίζει το κείμενο υπογραμμισμένο. Το *u* είναι το πρώτο γράμμα από την λέξη *underline*

Η ετικέτα `<small>`

Η ετικέτα `<small>` εμφανίζει το κείμενο με μικρούς χαρακτήρες.

Η ετικέτες `<sub>` και `<sup>`

Η ετικέτα `<sub>` τοποθετεί το κείμενο σαν μαθηματική βάση, δηλαδή σε χαμηλότερο σημείο απ' ότι βρίσκεται το υπόλοιπο κείμενο.

Στοιχεία

Τα έγγραφα HTML αποτελούνται από στοιχεία HTML τα οποία στην πιο γενική μορφή τους έχουν τρία συστατικά: ένα ζεύγος από ετικέτες, την «ετικέτα εκκίνησης» και την «ετικέτα τερματισμού», μερικές ιδιότητες μέσα στην ετικέτα εκκίνησης, και τέλος το κείμενο ή το γραφικό περιεχόμενο μεταξύ των ετικετών, το οποίο μπορεί να περιλαμβάνει και άλλα στοιχεία εμφωλευμένα μέσα του. Το στοιχείο HTML μπορεί να είναι οτιδήποτε ανάμεσα στις ετικέτες εκκίνησης και τερματισμού. Τέλος, κάθε ετικέτα περικλείεται σε σύμβολα «μεγαλύτερο από» και «μικρότερο από», δηλαδή `<` και `>`.

Επομένως, η γενική μορφή ενός στοιχείου HTML είναι: `<tag attribute1="value1" attribute2="value2">content</tag>`. Μερικά στοιχεία HTML περιγράφονται ως *άδεια στοιχεία*, έχουν τη μορφή `<tag attribute1="value1" attribute2="value2" >`, και δεν έχουν καθόλου περιεχόμενο. Το όνομα κάθε στοιχείου HTML είναι το ίδιο όνομα που χρησιμοποιείται στις αντίστοιχες ετικέτες. Το όνομα της ετικέτας τερματισμού ξεκινά με μια κάθετο `/`, η οποία παραλείπεται στα άδεια στοιχεία. Τέλος, αν δεν αναφέρονται ρητά οι ιδιότητες ενός στοιχείου, τότε χρησιμοποιούνται οι προεπιλογές σε κάθε περίπτωση.

Υπάρχουν διάφοροι τύπου στοιχείων στην HTML.

Η δομική σήμανση περιγράφει τον σκοπό του κειμένου

Για παράδειγμα, το `<h2>Golf</h2>` σημαίνει ότι η λέξη «Golf» θα είναι μια επικεφαλίδα δεύτερου επιπέδου. Η δομική σήμανση δεν εμπεριέχει συγκεκριμένο τρόπο εμφάνισης, αλλά οι περισσότεροι browser έχουν δικό τους προκαθορισμένο τρόπο για τη μορφοποίηση όλων των στοιχείων. Το περιεχόμενο μπορεί να μορφοποιηθεί οπτικά με τη χρήση των CSS.

Η σήμανση οπτικής μορφοποίησης περιγράφει την εμφάνιση του κειμένου, άσχετα από τον σκοπό του.

Για παράδειγμα το `έντονο κείμενο` υποδεικνύει ότι οι συσκευές που θα εμφανίσουν το κείμενο θα το κάνουν έντονο, αλλά δεν λέει τι θα κάνουν με το κείμενο οι συσκευές που δεν μπορούν να το εμφανίσουν, όπως για παράδειγμα οι συσκευές φωνητικής ανάγνωσης σελίδων. Τόσο στην περίπτωση του `έντονο` όσο και του `<i>πλάγιο</i>`, υπάρχουν άλλα στοιχεία τα οποία μπορεί να οπτικά να εμφανίζονται ίδια, αλλά είναι πιο *σημαντικά*, όπως τα ``σημαντικό κείμενο και ``κείμενο με έμφαση αντίστοιχα. Είναι ευκολότερα να ερμηνεύσει ένας φωνητικός browser τα τελευταία δύο στοιχεία. Ωστόσο, δεν είναι ισότιμα με τα αντίστοιχα στοιχεία οπτικής μορφοποίησης. Για παράδειγμα ένα πρόγραμμα φωνητικής ανάγνωσης της σελίδας δεν θα πρέπει να προφέρει με έμφαση τον τίτλο ενός βιβλίου, αλλά όταν το ίδιο περιεχόμενο εμφανίζεται σε μια οθόνη τότε ο τίτλος θα είναι με πλάγια γράμματα. Τα περισσότερα στοιχεία οπτικής μορφοποίησης έχουν θεωρούνται ξεπερασμένα μετά την προδιαγραφή HTML 4.0, και έχουν αντικατασταθεί από τα CSS.

Η σήμανση υπερκειμένου κάνει μερικά τμήματα ενός εγγράφου να συνδέουν με άλλα έγγραφα. Τα στοιχεία anchor δημιουργεί έναν υπερσύνδεσμο στο έγγραφο, και η ιδιότητα href ορίζει τον στόχο του συνδέσμου. Για παράδειγμα, η σήμανση HTML `Βικιπαίδεια` εμφανίζει το κείμενο «Wikipedia» ως υπερσύνδεσμο. Για την εμφάνιση μιας εικόνας ως συνδέσμου, μπορεί να εισαχθεί ένα στοιχείο «img» ως περιεχόμενο του στοιχείου «a». Όπως και το «br», το «img» είναι άδειο στοιχείο. Έχει ιδιότητες αλλά δεν έχει περιεχόμενο, ούτε ετικέτα τερματισμού: ``.

CSS

Η CSS (*Cascading Style Sheets-Διαδοχικά Φύλλα Στυλ*) ή (αλληλουχία φύλλων στύλ) είναι μια γλώσσα υπολογιστή που ανήκει στην κατηγορία των γλωσσών φύλλων στυλ που χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που έχει γραφτεί με μια γλώσσα σήμανσης. Χρησιμοποιείται δηλαδή για τον έλεγχο της εμφάνισης ενός εγγράφου που γράφτηκε στις γλώσσες HTML και XHTML, δηλαδή για τον έλεγχο της εμφάνισης μιας ιστοσελίδας και γενικότερα ενός ιστότοπου. Η CSS είναι μια γλώσσα υπολογιστή προορισμένη να αναπτύσσει στυλιστικά μια ιστοσελίδα δηλαδή να διαμορφώνει περισσότερα χαρακτηριστικά, χρώματα, στοίχιση και δίνει περισσότερες δυνατότητες σε σχέση με την html. Για μια όμορφη και καλοσχεδιασμένη ιστοσελίδα η χρήση της CSS κρίνεται ως απαραίτητη.

Η αλληλουχία εφαρμογής των φύλλων στύλ

Για ένα έγγραφο πχ xhtml θα υπάρχουν παραπάνω από ένα φύλλα στυλ τα οποία περιέχουν δηλώσεις για την εμφάνιση ενός συγκεκριμένου στοιχείου. Το Φύλλο στύλ που εφαρμόζεται σε ένα έγγραφο μπορεί να προέρχεται από :

- ☑ το συγγραφέα μιας ιστοσελίδας
- ☑ το χρήστη του πλοηγού
- ☑ τον ίδιο τον πλοηγό, αν έχει το δικό του προκαθορισμένο φύλλο στυλ .

Συνεπώς για ένα xhtml στοιχείο θα υπάρχουν παραπάνω από μια δηλώσεις που πιθανόν να είναι συγκρουόμενες. Το πρότυπο css για να επιλύσει παρόμοιες συγκρούσεις έχει καθορίσει μια αλληλουχία-σειρά στην οποία θα μπουν αυτές οι δηλώσεις και με βάση την οποία θα επιλεγεί πχ η δήλωση που είναι πρώτη στη σειρά.

Ο αλγόριθμος δημιουργίας αυτής της σειράς-αλληλουχίας είναι ο ακόλουθος:

1. Βρες όλες τις δηλώσεις που εφαρμόζονται στο στοιχείο που μας ενδιαφέρει. Οι δηλώσεις εφαρμόζονται στο στοιχείο αν ο επιλογέας του το επιλέξει (ταιριάζει με αυτό).
2. Ταξινόμησε με βάση τη σημασία (κανονική ή σημαντική) και προέλευση (συγγραφέας , χρήστη ή πλοηγός χρήστη). Με αύξουσα σειρά προτεραιότητας:
 - a. Δηλώσεις πλοηγού χρήστη
 - b. Κανονικές δηλώσεις χρήστη
 - c. Κανονικές δηλώσεις συγγραφέα
 - d. Σημαντικές δηλώσεις συγγραφέα
 - e. Σημαντικές δηλώσεις χρήστη
3. Ταξινόμησε τις δηλώσεις ίδιας σημασίας και προέλευσης με κριτήριο την εξειδίκευση του επιλογέα: οι πιο εξειδικευμένοι επιλογείς υπερισχύουν των πιο γενικών. Τα ψευδό-στοιχεία και οι ψευδο-κλάσεις λογαριάζονται σαν κανονικά στοιχεία και κλάσεις αντίστοιχα.
4. Τέλος ταξινόμησε ανάλογα με τη σειρά καθορισμού: αν δύο δηλώσεις έχουν το ίδιο βάρος , προέλευση και εξειδίκευση , αυτή που προσδιορίστηκε τελευταία επικρατεί. Οι δηλώσεις σε εισαγόμενα φύλλα στυλ θεωρούνται ότι δηλώνονται πριν από τις δηλώσεις στο ίδιο το φύλλο στυλ . Αφού λοιπόν προκύψει μια σειρά-αλληλουχία κανόνων εμφάνισης που αφορούν το ίδιο στοιχείο θα επιλεγεί προς εφαρμογή (για την αποφυγή συγκρούσεων) η δήλωση που θα είναι τελευταία στην σειρά που αναλύθηκε πιο πάνω.

PHP

Η **PHP** είναι μια [γλώσσα προγραμματισμού](#) για τη δημιουργία σελίδων web με δυναμικό περιεχόμενο. Μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό [διακομιστή](#) του Παγκόσμιου Ιστού (π.χ. [Apache](#)), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα [HTML](#).

Ένα αρχείο με κώδικα PHP θα πρέπει να έχει την κατάλληλη [επέκταση](#) (π.χ. *.php, *.php4, *.html κ.ά.). Η ενσωμάτωση κώδικα σε ένα αρχείο επέκτασης .html δεν θα λειτουργήσει και θα εμφανίσει στον browser τον κώδικα χωρίς καμία επεξεργασία, εκτός αν έχει γίνει η κατάλληλη ρύθμιση στα MIME types του server. Επίσης ακόμη κι όταν ένα αρχείο έχει την επέκταση .php, θα πρέπει ο server να είναι ρυθμισμένος για να επεξεργάζεται και να μεταγλωττίζει τον κώδικα PHP σε HTML που καταλαβαίνει το πρόγραμμα πελάτη. Ο διακομιστής Apache, που χρησιμοποιείται σήμερα ευρέως σε συστήματα με τα [λειτουργικά συστήματα GNU/Linux, Microsoft Windows, Mac OS X](#) υποστηρίζει εξ ορισμού την εκτέλεση κώδικα PHP, είτε με την χρήση ενός πρόσθετου (mod_php) ή με την αποστολή του κώδικα προς εκτέλεση σε εξωτερική διεργασία [CGI](#) ή [FCGI](#) ή με την έλευση της php5.4 υποστηρίζονται η εκτέλεση σε πολυάσχολους ιστοχώρους, FastCGI Process Manager (FPM).

Η ιστορία της PHP ξεκινά από το [1994](#), όταν ένας φοιτητής, ο [Rasmus Lerdorf](#) δημιούργησε χρησιμοποιώντας τη γλώσσα προγραμματισμού Perl ένα απλό [script](#) με όνομα php.cgi, για προσωπική χρήση. Το script αυτό είχε σαν σκοπό να διατηρεί μια λίστα στατιστικών για τα άτομα που έβλεπαν το online βιογραφικό του σημείωμα. Αργότερα αυτό το script το διέθεσε και σε φίλους του, οι οποίοι άρχισαν να του ζητούν να προσθέσει περισσότερες δυνατότητες. Η γλώσσα τότε ονομαζόταν PHP/FI από τα αρχικά Personal Home Page/Form Interpreter. Το [1997](#) η PHP/FI έφθασε στην έκδοση 2.0, βασιζόμενη αυτή τη φορά στη γλώσσα [C](#) και αριθμώντας περισσότερους από 50.000 [ιστότοπους](#) που τη χρησιμοποιούσαν, ενώ αργότερα την ίδια χρονιά οι [Andi Gutmans](#) και [Zeev Suraski](#) ξαναέγραψαν τη γλώσσα από την αρχή, βασιζόμενοι όμως αρκετά στην PHP/FI 2.0. Έτσι η PHP έφθασε στην έκδοση 3.0 η οποία θύμιζε περισσότερο τη σημερινή μορφή της. Στη συνέχεια, οι Zeev και Andi δημιούργησαν την εταιρεία Zend (από τα αρχικά των ονομάτων τους), η οποία συνεχίζει μέχρι και σήμερα την ανάπτυξη και εξέλιξη της γλώσσας PHP. Ακολούθησε το [1998](#) η έκδοση 4 της PHP, τον Ιούλιο του [2004](#) διατέθηκε η έκδοση 5, ενώ αυτή τη στιγμή έχουν ήδη διατεθεί και οι πρώτες δοκιμαστικές εκδόσεις της επερχόμενης PHP 6, για οποιονδήποτε προγραμματιστή θέλει να τη χρησιμοποιήσει. Οι περισσότεροι ιστότοποι επί του παρόντος χρησιμοποιούν κυρίως τις εκδόσεις 4 και 5 της PHP.

Φόρμες

Αν και ο κύριος σκοπός μιας ιστοσελίδας είναι η δημοσίευση πληροφοριών στο Internet, δεν είναι ο μοναδικός. Όλο και περισσότερα sites χρησιμοποιούν την αμφίδρομη επικοινωνία, δηλαδή δεν παρέχουν απλά πληροφορίες προς τους επισκέπτες, αλλά ζητούν από αυτούς να επιλέγουν ή να πληκτρολογούν στοιχεία. Για παράδειγμα, πολλά sites περιέχουν polls τα οποία παρουσιάζουν μια ερώτηση και ζητούν από τον επισκέπτη να επιλέξει μια από τις απαντήσεις. Στην περίπτωση αυτή έχουμε αμφίδρομη επικοινωνία, καθώς η ροή της πληροφορίας δεν γίνεται μόνο από τον Server που φιλοξενεί την ιστοσελίδα προς τον επισκέπτη αλλά και το αντίθετο.

Οι Φόρμες μας δίνουν την δυνατότητα να μετασχηματίσουμε την ιστοσελίδα μας από μια μάζα πληροφοριών που ο επισκέπτης διαβάζει παθητικά σε μια αμφίδρομη επικοινωνία όπου έχει έναν πιο ενεργητικό ρόλο πληκτρολογώντας στοιχεία ή κάνοντας κλικ με το ποντίκι του σε διάφορες επιλογές.

Στην παραπάνω Φόρμα ο επισκέπτης, αφού συμπληρώσει τα πεδία πατάει το κουμπί

Με το πάτημα του κουμπιού τα δεδομένα που συμπλήρωσε ο επισκέπτης στέλνονται στον Server

μέσω του πρωτοκόλλου HTTP. Έπειτα ο Server επεξεργάζεται με κάποιο πρόγραμμα script τις τιμές των πεδίων και επιστρέφει στον browser του επισκέπτη μια HTML σελίδα. Τα περιεχόμενα της σελίδας αυτής μπορεί να είναι οτιδήποτε, από μια απλή απάντηση μέχρι επιστροφή αποτελέσματος σε ερώτηση προς μια περίπλοκη Βάση Δεδομένων.

Δοκιμάστε να συμπληρώσετε τα στοιχεία που ζητάει η Φόρμα και έπειτα να πατήσετε το κουμπί *Αποστολή Στοιχείων*. Με το πάτημα του κουμπιού τα δεδομένα που πληκτρολογήσατε στέλνονται στον Server όπου εισάγονται μέσα σε ένα αρχείο script. Το script αυτό (*serverSideScript.php*, το οποίο είναι γραμμένο σε γλώσσα PHP) παίρνει αυτά τα δεδομένα και τα εμφανίζει μέσα σε έναν πίνακα. Στην περίπτωση αυτή δεν γίνεται καμία επεξεργασία των δεδομένων. Το μόνο που κάνει το script εδώ είναι να παίρνει τα δεδομένα από τον browser σας και τα εμφανίζει μέσα σε έναν πίνακα

Λεζάντα των στοιχείων μιας Φόρμας

Όλα τα στοιχεία μιας Φόρμας, είτε αυτά είναι πεδία κειμένου, είτε είναι περιοχές κειμένου, είτε είναι κουμπιά επιλογών κ.ο.κ., έχουν σχεδιαστεί να δέχονται με συγκεκριμένο τρόπο (είτε με πληκτρολόγηση, είτε με επιλογή με το ποντίκι) συγκεκριμένες (σημασιολογικά) τιμές οι οποίες θα επεξεργαστούν από κάποιο script αρχείο.

Για τον λόγο αυτόν κάθε στοιχείο μιας φόρμας θα πρέπει να έχει ένα κείμενο (δίπλα του ή σε οποιοδήποτε άλλο σημείο ώστε να είναι εμφανή η αντιστοιχία του με το στοιχείο της Φόρμας) το οποίο κείμενο να προσδίδει έναν χαρακτηρισμό στο στοιχείο της Φόρμας (π.χ.: όνομα, επώνυμο, τηλέφωνο κ.ο.κ.) ώστε ο επισκέπτης της σελίδας να ξέρει ποια πληροφορία ζητά το στοιχείο αυτό. Για παράδειγμα στην Φόρμα που παρουσιάζεται στην αρχή αυτής της σελίδας, το πεδίο κειμένου το οποίο δέχεται το επώνυμο χαρακτηρίζεται από το κείμενο Επώνυμο. Το κείμενο αυτό ονομάζεται *Λεζάντα* του πεδίου κειμένου που δέχεται το επώνυμο.

Με λίγα λόγια η *Λεζάντα* ενός στοιχείου της Φόρμας είναι το κείμενο που χαρακτηρίζει το περιεχόμενο του πεδίου που πρέπει να πληκτρολογήσει ή να επιλέξει ο επισκέπτης. Το κείμενο αυτό μπορούμε βέβαια να το μορφοποιήσουμε με HTML ετικέτες.

Η ετικέτα <form>

Για να δημιουργήσουμε μια φόρμα χρησιμοποιούμε τις ετικέτες <form> και </form>. Οι κυριότερες ιδιότητες της ετικέτας είναι οι *name*, *method* και *action*.

Με την ιδιότητα *name* δίνουμε ένα όνομα στην φόρμα ώστε να μπορούμε να αναφερόμαστε σε αυτή μέσα από τον κώδικα ενός script. Καλό είναι πάντα να δίνουμε όνομα στις Φόρμες που κατασκευάζουμε, ιδιαίτερα στις περιπτώσεις που έχουμε περισσότερες από μια Φόρμες στην ίδια σελίδα.

Η ιδιότητα *action* περιέχει το URL του αρχείου script στον Server το οποίο θα επεξεργαστεί τα στοιχεία της Φόρμας. Το script είναι ένα πρόγραμμα το οποίο τρέχει στον Server και το οποίο μεταξύ των άλλων μπορεί να δέχεται σαν είσοδο δεδομένα τα οποία λαμβάνει ο Server από τον browser του επισκέπτη (συνήθως από μια Φόρμα). Το script , χρησιμοποιώντας μια script γλώσσα προγραμματισμού (π.χ. asp , php κτλ.) , επεξεργάζεται τα δεδομένα αυτά και έπειτα επιστρέφει στον browser μια HTML σελίδα. Η επεξεργασία αυτή μπορεί να είναι από απλές πράξεις μεταξύ των δεδομένων έως και αναζήτηση σε μια περίπλοκη Βάση Δεδομένων με βάση τα δεδομένα αυτά (ένα τέτοιο script περιέχει κατάλληλες εντολές ώστε να συνδεθεί σε μια Βάση Δεδομένων και να προσπελάσει αλλά και να προσθέσει και να διαγράψει εγγραφές)

Η *method* καθορίζει τον τρόπο με τον οποίο στέλνονται τα δεδομένα της Φόρμας στον Server που βρίσκεται το πρόγραμμα *script* που θα τα επεξεργαστεί και μπορεί να πάρει τις τιμές *get* ή *post*. Με την μέθοδο *get* τα δεδομένα προσθέτονται στο τέλος του URL που "δείχνει" η ιδιότητα *action* και χωρίζονται από το σύμβολο *&*. Με την μέθοδο *post* τα δεδομένα στέλνονται ξεχωριστά από το URL. Στην περίπτωση *post* το αρχείο *script* παίρνει τα δεδομένα της Φόρμας μέσω της στάνταρ εισόδου. Παρακάτω η ιδιότητα *method* αναφέρεται πιο αναλυτικά.

Μέσα στα όρια των ετικετών `<form>` και `</form>` εκτός απο τα στοιχεία της Φόρμας μπορούμε επίσης να προσθέσουμε κείμενο (συνήθως Λεζάντες των στοιχείων) μαζί με ετικέτες μορφοποίησης

Η ετικέτα `<input>`

Με την ετικέτα `<input>` εισάγουμε τα περισσότερα στοιχεία της φόρμας. Οι κυριότερες ιδιότητες της ετικέτας είναι η *type* η οποία καθορίζει τον τύπο του στοιχείου της Φόρμας (Πεδίο Κειμένου ή Περιοχή Κειμένου ή Κουμπί Επιλογών ή Κουτί Πολλαπλών Επιλογών ή Κουμπί), η *name* με την οποία δίνουμε ένα μοναδικό όνομα στο στοιχείο της φόρμας (δεν πρέπει να υπάρχουν στοιχεία φόρμας με τα ίδια ονόματα) και η *value* με την οποία δίνουμε μια αρχική τιμή στο στοιχείο της φόρμας. Η ετικέτα `<input>` δεν έχει ετικέτα τέλους

Πώς εισάγουμε ένα Πεδίο Κειμένου (text field) στην Φόρμα μας

Για να εισάγουμε ένα Πεδίο Κειμένου χρησιμοποιούμε την ετικέτα `<input>` ορίζοντας την τιμή *text* στην ιδιότητα *type*.

Η χρήση της ιδιότητας *value* σε αυτήν την περίπτωση εμφανίζει μέσα στο πλαίσιο κειμένου την τιμή της ιδιότητας. Μια άλλη ιδιότητα που μπορεί να πάρει η ετικέτα `<input>` είναι η *size* η οποία καθορίζει τον αριθμό των ορατών χαρακτήρων που μπορεί να χωρέσει το Πεδίο Κειμένου. Η ιδιότητα *maxlength* καθορίζει τον αριθμό χαρακτήρων που μπορεί να πληκτρολογήσει ο επισκέπτης.

Πώς εισάγουμε Περιοχή Κειμένου στην Φόρμα μας ;

Για να εισάγουμε Περιοχή Κειμένου στην Φόρμα μας χρησιμοποιούμε την ετικέτα `<textarea>`.

Η ιδιότητα *cols* ορίζει τον αριθμό των ορατών χαρακτήρων (αυτοί δηλαδή που φαίνονται χωρίς οριζόντιο scrolling) που θα χωράει κάθε γραμμή στην Περιοχή Κειμένου, ενώ η ιδιότητα *rows* ορίζει τον αριθμό των ορατών γραμμών (αυτές δηλαδή που φαίνονται χωρίς κάθετο scrolling) που θα έχει η Περιοχή Κειμένου. Η χρήση και των δύο αυτών ιδιοτήτων είναι απαραίτητη. Όταν ο επισκέπτης πατάει το Κουμπί Υποβολής της Φόρμας αυτό που θα σταλεί στον Server είναι ολόκληρο το κείμενο που πληκτρολόγησε στη Περιοχή κειμένου

Πώς εισάγουμε ένα Πεδίο Mail στην Φόρμα μας ;

Εισαγωγή πεδίου password σε μια φόρμα

```
<form name="form1" method="post" action="example1.php">
Password: <input type="password" name="mail" size="20" maxlength="20" />

<input type="Submit" name="Submit" value="Αποστολή" />
</form>
```

Πολλές φορές κατά την πλοήγηση μας στο Internet χρειάζεται να πληκτρολογήσουμε ένα Username και κάποιον κωδικό μέσα σε ένα πεδίο κειμένου για να εισέλθουμε σε κάποια ελεγχόμενη περιοχή, όπως Forums , Chats , Web emails κτλ. Όταν γράφουμε μέσα στην περιοχή κειμένου του Password εμφανίζονται ή αστεράκια ή βουλίστσες, ανάλογα τον browser που χρησιμοποιούμε.

Για να εισάγουμε στην δική μας Φόρμα ένα τέτοιο πεδίο στο οποίο όταν ο επισκέπτης της σελίδας μας πληκτρολογεί μέσα σε αυτό, να εμφανίζονται βουλίστσες ή αστεράκια απλά βάζουμε την τιμή *mail* στην ιδιότητα *type* της ετικέτας <input>. Η ιδιότητες *size* και *maxlength* μπορούν να χρησιμοποιηθούν όπως και στην περίπτωση του απλού Πεδίου Κειμένου

Το κουμπί Υποβολής της Φόρμας

Το πάτημα του Κουμπιού Υποβολής της Φόρμας στέλνει στον Server τα δεδομένα που πληκτρολόγησε ο χρήστης. Για να εισάγουμε στην Φόρμα μας Κουμπί Υποβολής χρησιμοποιούμε την ετικέτα <input> ορίζοντας την τιμή **Submit** στην ιδιότητα **type**. Η ιδιότητα **value** ορίζει το κείμενο στο Κουμπί Υποβολής.

Η τοποθέτηση του Κουμπιού Υποβολής στην Φόρμα μας είναι προαιρετική. Σχεδόν όλες οι φόρμες έχουν Κουμπί Υποβολής. Αυτές που δεν έχουν η υποβολή της Φόρμας γίνεται με κώδικα script (συνήθως γραμμένο σε Javascript) που ενεργοποιείται μετά από ένα **HTML γεγονός**

Αποστολή στοιχείων στον Server (Μέθοδος GET και POST)

Η αποστολή των τιμών από την φόρμα του browser προς το αρχείο script του Server που θα τα επεξεργαστεί, γίνεται με βάση την ιδιότητα *method* (μέθοδος) της ετικέτας <form> η οποία μπορεί να πάρει τις τιμές *get* και *post*.

Στην περίπτωση χρήσης της μεθόδου *GET* κάθε όνομα ενός στοιχείου συνοδεύεται με την τιμή που πληκτρολόγησε ή επέλεξε ο επισκέπτης και αυτό το "πακέτο" (όνομα στοιχείου-τιμή στοιχείου) προστίθεται στο τέλος του URL που δείχνει η ιδιότητα *action* της ετικέτας <form>.

και ο επισκέπτης πληκτρολογήσει το όνομα Akis στο πεδίο κειμένου με όνομα *onoma* και έπειτα πατήσει το κουμπί Submit, τότε το URL που θα ανοίξει ο browser θα είναι το:

```
http://www.wlearn.gr/doityourself/get_example.php?onoma=Akis&Submit=Submit
```

Στο τέλος του URL μπαίνει ένα λατινικό ερωτηματικό και στην συνέχεια ακολουθεί το όνομα του στοιχείου την φόρμας, που στην προκειμένη περίπτωση είναι *onoma* , ακολουθεί το σύμβολο = (ίσον) και αμέσως μετά η τιμή που πληκτρολόγησε ο επισκέπτης. Στην περίπτωση αυτή η Φόρμα στέλνει δύο ονόματα στοιχείων (*onoma* και *Submit*) μαζί με τις τιμές τους (*value*) στον Server. Όταν υπάρχουν περισσότερα από ένα πεδία στην Φόρμα, όπως στην περίπτωση αυτή, τότε τα "πακέτα" τιμών στο URL χωρίζονται με το σύμβολο &.

Αν ο επισκέπτης πληκτρολογήσει όνομα και επίθετο και έπειτα πατήσει το Submit Button , τότε το URL που θα ανοίξει ο browser θα είναι το:

```
http://www.wlearn.gr/doityourself/get_example.php?onoma=Akis&eponimo=Kargiofyllis&Submit=Submit
```

Ο Server αποθηκεύει όλη την συμβολοσειρά του URL μετά το λατινικό ερωτηματικό (?) στην μεταβλητή περιβάλλοντος *QUERY_STRING* . Έπειτα η συμβολοσειρά αυτή περνάει μέσα στο script το οποίο με προγραμματιστικό κώδικα διαχωρίζει τα πακέτα τιμών (όνομα_στοιχείο=τιμή) μεταξύ τους.

To site

Στη 1^η σελίδα “το μουσείο” έχουμε πληροφορίες σχετικά με το περιεχόμενο του μουσείου.

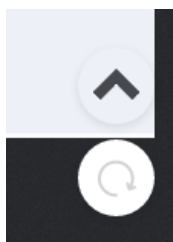


Με τη χρήση script επιτύχαμε την αυτόματη ή και χειροκίνητη εναλλαγή εικόνων και κειμένου.

```
]<script>
]jQuery(function () {
]   jQuery('#camera_wrap_1').camera({
]     height: '317px',
]     loader: false,
]     pagination: false,
]     thumbnails: false
]   });
] });
- </script>
```

```
<div class="camera_wrap camera_azure_skin" id="camera_wrap_1">
  <div data-src="images/slide1.jpg">
    <div class="camera_caption fadeIn">
      <h2>Κυπριακό Μουσείο</h2>
      Η Κύπρος, μια μικρή χώρα με λιγότερο από ένα εκατομμύριο κατοίκους, διαθέτει ένα σχετικά ευτυχεστικά μεγάλο αριθμό μουσείων. Τα μουσεία αυτά ποικίλουν, από κρατικά ιδρύματα με μεγάλες συλλογές και μεγάλο εργατικό δυναμικό, έως μικρά ιδιωτικά μουσεία τα οποία λειτουργούν υπό την εποπτεία ενός αιθέριου και έχουν διατηρηθεί με την επιμονή και την αφυσίαση των ιδιοκτητών τους.
    </div>
  </div>
  <div data-src="images/slide2.jpg">
    <div class="camera_caption fadeIn">
      <h2>Κυπριακό Μουσείο</h2>
      Το Κυπριακό Μουσείο, το πρώτο αρχαιολογικό μουσείο Κύπρου, εκθέτει ευρήματα από τις προϊμότερες περιόδους της ανθρώπινης εγκατάστασης στο νησί, από τη 10η χιλιετία π.Χ. μέχρι και τη Ρωμαϊκή περίοδο.
    </div>
  </div>
</div>
```

Επίσης τοποθετήσαμε 2 κουμπιά στη δεξιά κάτω γωνία το ένα για refresh και το άλλο για να σε μεταφέρει στην κορυφή της σελίδας.



```

<a id="refresh" value="Refresh" onClick="history.go()">
<svg class="refreshicon" version="1.1" id="Capa_1" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" x="0px" y="0px"
width="25px" height="25px" viewBox="0 0 322.447 322.447" style="enable-background:new 0 0 322.447 322.447;"
xml:space="preserve">
<path
d="M321.832,230.327c-2.133-6.565-9.184-10.154-15.75-8.025l-16.254,5.281C299.785,206.991,305,184.347,305,161.224
c0-84.089-68.41-152.5-152.5-152.5C68.411,8.724,0,77.135,0,161.224s68.411,152.5,152.5,152.5c6.903,0,12.5-5.597,12.5-12.5
c0-6.902-5.597-12.5-12.5-12.5c-70.304,0-127.5-57.195-127.5-127.5c0-70.304,57.196-127.5,127.5-127.5
c70.305,0,127.5,57.196,127.5,127.5c0,19.372-4.371,38.337-12.723,55.568l-5.553-17.096c-2.133-6.564-9.186-10.156-15.75-8.025
c-6.566,2.134-10.16,9.186-8.027,15.751l14.74,45.368c1.715,5.283,6.615,8.642,11.885,8.642c1.279,0,2.582-0.198,3.865-0.614
145.369-14.738C320.371,243.946,323.965,236.895,321.832,230.327z"/>
</svg>
</a>

```

```

<script type="text/javascript">
// create the back to top button
$('body').prepend('<a href="#" class="back-to-top">Back to Top</a>');

var amountScrolled = 300;

$(window).scroll(function() {
    if ( $(window).scrollTop() > amountScrolled ) {
        $('.a.back-to-top').fadeIn('slow');
    } else {
        $('.a.back-to-top').fadeOut('slow');
    }
});

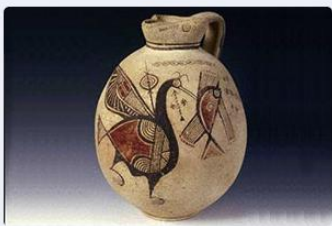
$('.a.back-to-top, a.simple-back-to-top').click(function() {
    $('html, body').animate({
        scrollTop: 0
    }, 700);
    return false;
});
</script>

```


Στη 2^η σελίδα “συλλογές” έχουμε τις αίθουσες του μουσείου με αναλυτική περιγραφή των εκθεμάτων τους.

ΤΟ ΜΟΥΣΕΙΟ
ΣΥΛΛΟΓΕΣ
ΕΓΓΡΑΦΗ
ΕΙΣΙΤΗΡΙΑ
ΕΠΙΚΟΙΝΩΝΙΑ


Αίθουσες



Αίθουσα I



Αίθουσα II



Αίθουσα III

Τοποθετήσαμε και ένα κουμπί “show more” για εξοικονόμηση χώρου.

```
<li class="grid_4 alpha">
  <figure></figure>
  <p><a href="#" class="link">Αίθουσα I</a></p>
  <div>
    <input type="checkbox" class="read-more-state" id="post-1" />
    <p class="read-more-wrap">Στην Αίθουσα I παρουσιάζεται η νεολιθική και χαλκολιθική περίοδος.
    <span class="read-more-target">Η αίθουσα αυτή αναδεικνύει με ποσοστια σύμβαση με τα νέα ανασκαφικά δεδομένα,
    τα οποία διαφοροποιούν την εικόνα που είχαμε για τη νεολιθική Κύπρο.
    Εκτίθενται εργαλεία, σκεύη, κοσμήματα, ειδώλια και άλλα αντικείμενα
    αντιπροσωπευτικά της νεολιθικής και χαλκολιθικής περιόδου της Κύπρου.
    Είναι κεραμικά προϊόντα εκτίθεται ένα εξαιρετικά ενδιαφέρον σύνολο σκευμάτων,
    τα οποία αποτελεί μια από τις πωριότερες ενδείξεις λατρείας στο νασί.
    Η έκθεση συνοδεύεται από ψηφιακό υλικό.</span></p>
    <label for="post-1" class="read-more-trigger"></label>
  </div>
</li>
```

Στη 3η σελίδα “εγγραφή” ο χρήστης μπορεί να κάνει SIGN IN ή SIGN UP και να γραφτεί στο Newsletter του μουσείου για να μένει ενήμερος για όλες τις δραστηριότητες μας.

```
<div class="grid_4 omega">
  <h2>Newsletter</h2>
  <form id="subscribe-form" action="#">
    <fieldset>
      <input type="text" value="Enter Your Email"
      onFocus="if(this.value=='Enter Your Email'){this.value=''}"
      onBlur="if(this.value==''){this.value='Enter Your Email'}">
      <a href="#" class="button1" onClick="document.getElementById('subscribe-form').submit()">Subscribe</a>
    </fieldset>
  </form>
</div>
```

SIGN IN - SIGN UP

SIGN IN SIGN UP

USERNAME

PASSWORD

NO KEEP ME SIGNED IN

SIGN IN

Forgot your password?

```
<div class="container">
  <div class="frame">
    <div class="nav">
      <ul class="links">
        <li class="signin-active"><a class="btn">Sign in</a></li>
        <li class="signup-inactive"><a class="btn">Sign up </a></li>
      </ul>
    </div>
    <div ng-app ng-init="checked = false">
      <form class="form-signin" action="" method="post" name="form">
        <label for="username">Username</label>
        <input class="form-styling" type="text" name="username" placeholder="" />
        <label for="password">Password</label>
        <input class="form-styling" type="text" name="password" placeholder="" />
        <input type="checkbox" id="checkbox" />
        <label for="checkbox" ><span class="ui"></span>Keep me signed in</label>
        <div class="btn-animate">
          <a class="btn-signin">Sign in</a>
        </div>
      </form>
      <form class="form-signup" action="" method="post" name="form">
        <label for="fullname">Full name</label>
        <input class="form-styling" type="text" name="fullname" placeholder="" />
        <label for="email">Email</label>
        <input class="form-styling" type="text" name="email" placeholder="" />
        <label for="password">Password</label>
        <input class="form-styling" type="text" name="password" placeholder="" />
        <label for="confirmpassword">Confirm password</label>
        <input class="form-styling" type="text" name="confirmpassword" placeholder="" />
        <a ng-click="checked = !checked" class="btn-signup">Sign Up</a>
      </form>
    </div>
  </div>
</div>
```

```

<div class="success">
  <svg width="270" height="270" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" x="0px" y="0px"
  viewBox="0 0 60 60" id="check" pg-class="checked ? 'checked' : ''">
    <path fill="#ffffff" d="M40.61,23.03L26.67,36.97L13.495,23.788c-1.146-1.147-1.359-2.936-0.504-4.314
    c3.894-6.28,11.169-10.243,19.283-9.348c9.258,1.021,16.694,8.542,17.622,17.81c1.232,12.295-8.683,22.607-20.849,22.042
    c-9.9-0.46-18.128-8.344-18.972-18.218c-0.292-3.416,0.276-6.673,1.51-9.578" />
  <div class="success&#x27E9">
    <p> Thanks for signing up! Check your email for confirmation.</p>
  </div>
</div>

<div class="forgot">
  <a href="#">Forgot your password?</a>
</div>

<div>
  <div class="cover-photo"></div>
  <div class="profile-photo"></div>
  <h1 class="welcome">Welcome, Test</h1>
  <a class="btn-goback" value="Refresh" onClick="history.go()">Go back</a>
</div>
</div>

```

```

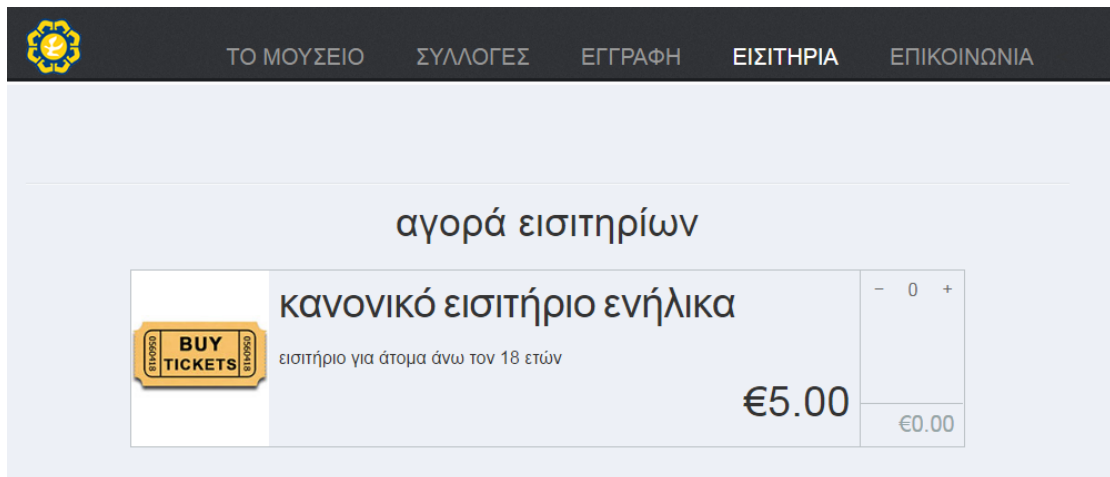
$(function() {
  $(".btn").click(function() {
    $(".form-signin").toggleClass("form-signin-left");
    $(".form-signup").toggleClass("form-signup-left");
    $(".frame").toggleClass("frame-long");
    $(".signup-inactive").toggleClass("signup-active");
    $(".signin-active").toggleClass("signin-inactive");
    $(".forgot").toggleClass("forgot-left");
    $(this).removeClass("idle").addClass("active");
  });
});

$(function() {
  $(".btn-signup").click(function() {
    $(".nav").toggleClass("nav-up");
    $(".form-signup-left").toggleClass("form-signup-down");
    $(".success").toggleClass("success-left");
    $(".frame").toggleClass("frame-short");
  });
});

$(function() {
  $(".btn-signin").click(function() {
    $(".btn-animate").toggleClass("btn-animate-grow");
    $(".welcome").toggleClass("welcome-left");
    $(".cover-photo").toggleClass("cover-photo-down");
    $(".frame").toggleClass("frame-short");
    $(".profile-photo").toggleClass("profile-photo-down");
    $(".btn-goback").toggleClass("btn-goback-up");
    $(".forgot").toggleClass("forgot-fade");
  });
});

```

Στη 4^η σελίδα “εισιτήρια” έχουμε το καλάθι όπου ο χρήστης μπορεί να αγοράσει εισιτήρια.



```
<div class="main">
  <h1>αγορά εισιτηρίων</h1>
  <section class="shopping-cart">
    <ol class="ui-list shopping-cart--list" id="shopping-cart--list">
      <script id="shopping-cart--list-item-template" type="text/template">
        <li class="_grid shopping-cart--list-item">
          <div class="_column product-image">
            
          </div>
          <div class="_column product-info">
            <h4 class="product-name">{{=name}}</h4>
            <p class="product-desc">{{=desc}}</p>
            <div class="price product-single-price">€{{=price}}</div>
          </div>
          <div class="_column product-modifiers" data-product-price="{={price}}">
            <div class="_grid">
              <button class="_btn _column product-subtract">minus</button>
              <div class="_column product-qty">0</div>
              <button class="_btn _column product-plus">plus</button>
            </div>
            <button class="_btn _column trash product-remove">Remove</button>
            <div class="price product-total-price">€0.00</div>
          </div>
        </li>
      </script>
    </ol>
```

```
<footer class="_grid cart-totals">
  <div class="_column total" id="totalCtr" align="center">
    <div class="cart-totals-key">Total</div>
    <div class="cart-totals-value">€0.00</div>
  </div>
  <div class="_column checkout">
    <button class="_btn checkout-btn _type-forward">Checkout</button>
  </div>
</footer>
```



```

addProduct: function () {
    "use strict";

    var qtyCtr = $(this).prev(".product-qty"),
        quantity = parseInt(qtyCtr.html(), 10) + 1;

    app.updateProductSubtotal(this, quantity);
},

subtractProduct: function () {
    "use strict";

    var qtyCtr = $(this).next(".product-qty"),
        num = parseInt(qtyCtr.html(), 10) - 1,
        quantity = num <= 0 ? 0 : num;

    app.updateProductSubtotal(this, quantity);
},

updateProductSubtotal: function (context, quantity) {
    "use strict";

    var ctr = $(context).closest(".product-modifiers"),
        productQtyCtr = ctr.find(".product-qty"),
        productPrice = parseFloat(ctr.data("product-price")),
        subtotalCtr = ctr.find(".product-total-price"),
        subtotalPrice = quantity * productPrice;

    productQtyCtr.html(quantity);
    subtotalCtr.html( subtotalPrice.to_$( ) );

    app.updateTotals();
},

```

```

updateTotals: function () {
    "use strict";

    var products = $(".shopping-cart--list-item"),
        subtotal = 0,
        shipping;

    for (var i = 0; i < products.length; i += 1) {
        subtotal += parseFloat( $(products[i]).find(".product-total-price").html().strip$( ) );
    }

    shipping = (subtotal > 0 && subtotal < (100 / 1.00)) ? app.shipping : 0;

    $("#subtotalCtr").find(".cart-totals-value").html( subtotal.to_$( ) );
    $("#taxesCtr").find(".cart-totals-value").html( (subtotal * 0.00).to_$( ) );
    $("#totalCtr").find(".cart-totals-value").html( (subtotal * 1.00 + shipping).to_$( ) );
    $("#shippingCtr").find(".cart-totals-value").html( shipping.to_$( ) );
},

attachEvents: function () {
    "use strict";

    $(".product-remove").on("click", app.removeProduct);
    $(".product-plus").on("click", app.addProduct);
    $(".product-subtract").on("click", app.subtractProduct);
},

```

```

] setProductImages: function () {
    "use strict";

    var images = $(".product-image"),
        ctr,
        img;

] for (var i = 0; i < images.length; i += 1) {
    ctr = $(images[i]),
    img = ctr.find(".product-image--img");

    ctr.css("background-image", "url(" + img.attr("src") + ")");
    img.remove();
}
},

] renderTemplates: function () {
    "use strict";

    var products = app.products,
        content = [],
        template = new t( $("#shopping-cart--list-item-template").html() );

] for (var i = 0; i < products.length; i += 1) {
    content[i] = template.render(products[i]);
}


    $("#shopping-cart--list").html(content.join(""));
}

];

app.renderTemplates();
app.setProductImages();
app.attachEvents();

```

Στη 5^η και τελευταία σελίδα “επικοινωνία” έχουμε πληροφορίες σχετικά με το που βρίσκετε το μουσείο και τρόπους με τους οποίους μπορεί ο χρήστης να επικοινωνήσει μαζί μας.


ΤΟ ΜΟΥΣΕΙΟ
ΣΥΛΛΟΓΕΣ
ΕΓΓΡΑΦΗ
ΕΙΣΙΤΗΡΙΑ
ΕΠΙΚΟΙΝΩΝΙΑ

Πληροφορίες
Παράπονα-Εισηγήσεις

Ώρες λειτουργίας και τηλέφωνα επικοινωνίας

Μουσείου 1 (Λευκωσία),
22865854 (είσοδος/πώληση
εισητηρίων)
22303112 (πληροφορίες)
22865805 (Κατάστημα)
E-mail: mail@demolink.org

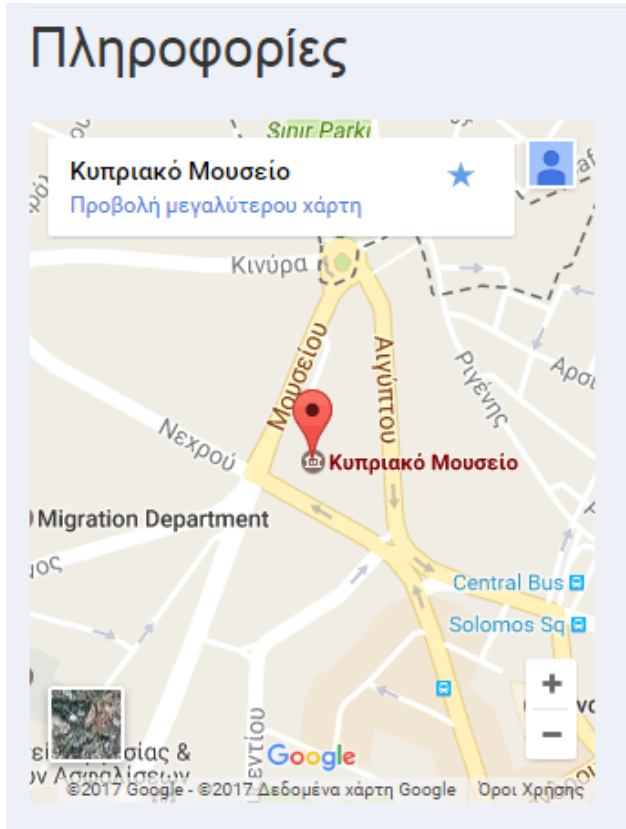
Ώρες λειτουργίας
Τρίτη - Παρασκευή: 8.00
-18.00
Σάββατο: 9.00 - 17.00
Κυριακή: 10.00 - 13.00
Δευτέρα: κλειστό
Κάθε πρώτη Τετάρτη του
μήνα: 8.00 - 20.00

Παράπονα-Εισηγήσεις


```

<div class="grid_4 prefix_1">
<h5>Παράθυρο Εισαγωγής</h5>
<form id="contact-form" action="#">
<fieldset>
<label>
<input type="text" value="Name" onFocus="if(this.value=='Name')(this.value='')" onBlur="if(this.value=='')(this.value='Name')">
</label>
<label>
<input type="text" value="Email" onFocus="if(this.value=='Email')(this.value='')" onBlur="if(this.value=='')(this.value='Email')">
</label>
<label>
<input type="text" value="Phone" onFocus="if(this.value=='Phone')(this.value='')" onBlur="if(this.value=='')(this.value='Phone')">
</label>
<textarea onFocus="if(this.value=='Message')(this.value='')" onBlur="if(this.value=='')(this.value='Message')">Message</textarea>
<a href="#" class="button1" onClick="document.getElementById('contact-form').reset()">clear</a> <a href="#" class="button1"
onClick="document.getElementById('contact-form').submit()">send</a>
</fieldset>
</form>
</div>

```



```

<div class="grid_4 alpha">
<iframe width="310" height="375" frameborder="0" scrolling="no" marginheight="0" marginwidth="0"
src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d3261.3942437444457!2d33.35352751476688!3d35.1
style="border:0" allowfullscreen></iframe>
</div>

```

Με το Google Maps API καταφέραμε να τοποθετήσετε ένα διακρατικό χάρτη με μια απλή αίτηση HTTP.

Forming the URL

The URL for an Google Maps Embed API request is as follows:

```

https://www.google.com/maps/embed/v1/MODE?key=YOUR_API_KEY&parameters

```

Where:

- {MODE} is one of place, directions, search, view, or streetview.
- {YOUR_API_KEY} is your free API key.
- parameters include optional parameters, as well as mode-specific parameters.

Ευχαριστίες

Θα θέλαμε να ευχαριστήσουμε τον καθηγητή κ. Ν. Παπαδάκη για την πολύτιμη βοήθεια που μας προσέφερε κατά τη διάρκεια της εκπόνησης την πτυχιακής εργασίας.

Βιβλιογραφία

1. Luke Welling Laura Thomson, "Ανάπτυξη Web εφαρμογών με Php και MySQL", Γκιούρδας, 2011
2. <http://el.wikipedia.org>
3. <http://www.techteam.gr/wiki/PHP>
4. <http://www.php.net>
5. <http://www.mysql.com/>
6. <http://www.w3schools.com/>
7. <http://www.sqlschool.gr/>
8. <http://www.w3.org/>
9. <https://developers.google.com/maps/documentation/embed/guide>
10. <http://www.theacropolismuseum.gr/en>
11. <http://www.cssflow.com>
12. <http://www.mcw.gov.cy/mcw/DA/DA.nsf/All/4D3CBED9569C1B9BC22573B5003B867B?OpenDocument>
13. <http://cyprusmuseumguide.com/wp-content/uploads/2014/05/onlineguidewebsite.pdf>
14. <http://www.nicosia.org.cy/el-GR/discover/museums/cyprus-museum>
15. <http://codepen.io/>
16. <http://html-tuts.com/back-to-top-button-jquery/>
17. http://www.alt-codes.net/euro_alt_code.php