

3. Προχωρημένα θέματα της HTML

Το δεύτερο αυτό κεφάλαιο για την HTML παρουσιάζει το χειρισμό και την ενσωμάτωση πολυμέσων και άλλων αντικειμένων (εικόνες, βίντεο, ήχος, iframe, object) σε μια ιστοσελίδα, συγκεκριμένα, θα εμβαθύνουμε στις δυνατότητες που έχει η HTML5, η οποία υποστηρίζει πολυμέσα σε πολύ μεγαλύτερο βαθμό από ότι οι προηγούμενες εκδόσεις. Θα δούμε επίσης στοιχεία με σημασιολογικό περιεχόμενο που επιτρέπουν τη διάταξη και κατηγοριοποίηση του περιεχομένου μιας σελίδας. Επίσης θα γίνει αναφορά στις φόρμες HTML, και θα παρουσιαστούν οι πλούσιες επιλογές που παρέχει η HTML5 για δημιουργία φορμών σήμερα. Η παράθεση οδηγών σχεδιασμού φορμών και η συζήτηση για τον ορθό σχεδιασμό τους είναι η τελευταία ενότητα αυτού του κεφαλαίου.

3.1 Διαχείριση εικόνων

Όπως γνωρίζουμε μια ιστοσελίδα εκτός από κείμενο, που ήταν το αντικείμενο του προηγούμενου κεφαλαίου, περιλαμβάνει και άλλα μέσα, όπως εικόνες, βίντεο και ήχους. Θα ξεκινήσουμε από τις εικόνες.

Μια εικόνα μπορεί να ενσωματωθεί σε μια σελίδα ως στοιχείο `` image δηλαδή, που μας επιτρέπει να ενσωματώσουμε ένα εξωτερικό αρχείο που περιέχει εικόνα. Επίσης θα δούμε το στοιχείο `<picture>` που χρησιμοποιείται λιγότερο, όμως μας επιτρέπει να ενσωματώσουμε πολλαπλές εικόνες διαφορετικών διαστάσεων ούτως ώστε να ανταποκρινόμαστε σε διαφορετικά μεγέθη συσκευών.

Ας δούμε πρώτα το στοιχείο ``. Το στοιχείο αυτό είναι κενό στοιχείο (δεν απαιτείται ετικέτα τερματισμού του).

Το κύριο γνώρισμα του στοιχείου αυτού είναι το `src` (source) δηλαδή η πηγή, η διεύθυνση του διαδικτύου που υπάρχει η εικόνα. Αν η εικόνα είναι στον ίδιο φάκελο που είναι η ιστοσελίδα αρκεί το όνομά του αρχείου. Διαφορετικά θα πρέπει να δοθεί το URL, ή το σχετικό μονοπάτι.

Ένα άλλο γνώρισμα το οποίο είναι ιδιαίτερα σημαντικό, όπως θα εξηγηθεί στη συνέχεια, και θεωρείται υποχρεωτικό, είναι το `alt` (alternate text). Το γνώρισμα αυτό αφορά μία εναλλακτική λεκτική περιγραφή μέσω κειμένου, της εικόνας. Είναι πολύ σημαντικό γνώρισμα για πληρότητα παρουσίασης και ιδιαίτερα για εξυπηρέτηση ατόμων με ειδικές ανάγκες.

Εδώ βλέπουμε ένα παράδειγμα εισαγωγής ενός στοιχείου ``, δεν απαιτεί ετικέτα τερματισμού ``.

```

```

3.1.1 Μορφότυποι εικόνων

Ας δούμε τώρα τι τύπους εικόνων μπορούμε να εισάγουμε σε μια ιστοσελίδα. Αυτοί είναι οι τέσσερις πιο βασικοί τύποι:

- **GIF** (Graphic Interchange Format), επιτρέπει μόνο 256 χρώματα, επιτρέπει κίνηση (animated gif).
- **JPEG** (Joint Photographic Experts Group) υποστηρίζει υψηλή συμπίεση εικόνων, είναι συνεπώς η πιο συνήθης επιλογή. Όμως δεν υποστηρίζει κίνηση και διαφάνεια.

- **PNG** (Portable Network Graphics) υποστηρίζει διαφάνεια, συνεπώς είναι χρήσιμο και για μη ορθογώνιες εικόνες.
- **SVG** (Scalable Vector Graphics) μαθηματικά ορισμένο περιεχόμενο σε XML, υποστηρίζει κίνηση, είναι χρήσιμο για μεγάλες κλίμακες.

Υπάρχουν κι άλλοι τύποι, όπως PDF, TIFF, κλπ., αλλά οι παραπάνω είναι αυτοί που χρησιμοποιούνται περισσότερο.

Ο μορφότυπος GIF είναι ο πιο παλιός, επιτρέπει animation, με τη μορφή animated GIF, όμως είναι πολύ περιορισμένος ως προς τον αριθμό χρωμάτων. Ο JPEG είναι ο πιο συνηθισμένος μορφότυπος εικόνων στο διαδίκτυο. Επιτρέπει εκατομμύρια χρώματα και αποχρώσεις, υψηλή συμπίεση εικόνων και είναι η πιο συνηθισμένη επιλογή. Όμως δεν υποστηρίζει κίνηση και επίσης - κάτι σημαντικό - δεν υποστηρίζει διαφάνεια. Δηλαδή δεν μας επιτρέπει να έχουμε τμήματα της εικόνας διαφανή για να φαίνεται το υπόβαθρο που αυτό σημαίνει ότι είμαστε υποχρεωμένοι μέσω JPEG να έχουμε μόνο ορθογώνιες εικόνες. Ο PNG είναι μια εναλλακτική επιλογή του JPEG. Ο PNG είναι μορφότυπος ο οποίος έχει τα ίδια χαρακτηριστικά, όσον αφορά το πλήθος των χρωμάτων, υποστηρίζει όμως διαφάνεια και συνεπώς είναι χρήσιμος για μη ορθογώνιες εικόνες. Τέλος το SVG επιτρέπει τον ορισμό μέσω γραφικών εξισώσεων των εικονικών στοιχείων. Στην ουσία το περιεχόμενο της εικόνας αποτυπώνεται ως διανύσματα, είναι σε μορφή xml και υποστηρίζει κίνηση. Έχει ενδιαφέρον, αφού υποστηρίζει μεγέθυνση πολύ μεγάλης κλίμακας εικόνων γιατί διατηρεί τα βασικά στοιχεία αφού πρόκειται για διανυσματική αναπαράσταση, ανεξαρτήτως κλίμακας. Οι μορφότυποι αυτοί υποστηρίζονται από όλους τους σημαντικούς φυλλομετρητές σήμερα, βλέπε τον [σχετικό πίνακα από το λήμμα της wikipedia](#).

3.1.2 Η σημασία του γνωρίσματος **alt**

Το γνώρισμα **alt** του στοιχείου **** όπως ήδη αναφέρθηκε, περιέχει ένα εναλλακτικό κείμενο αντί για την εικόνα. Είναι πολύ σημαντικό γιατί για ένα άτομο με ειδικές ανάγκες, όπως για παράδειγμα με δυσκολία στην όραση, ο screen reader, ο αυτόματος αναγνώστης του περιεχομένου της ιστοσελίδας, στη θέση της εικόνας θα διαβάσει το περιεχόμενο του γνωρίσματος **alt**. Αν δεν υπάρχει το γνώρισμα αυτό, το άτομο με προβλήματα όρασης δεν μπορεί να έχει καμία αίσθηση για το περιεχόμενο της εικόνας και συνεπώς της ιστοσελίδας.

Επίσης εάν η εικόνα αργεί να κατέβει, αφού οι εικόνες είναι πολύ πιο μεγάλα αρχεία από το έγγραφο HTML που είναι αρχείο κειμένου, το γνώρισμα **alt** βοηθάει τον χρήστη να αντιληφθεί το περιεχόμενο της εικόνας που δεν έχει κατέβει.

Ακόμη, το γνώρισμα **alt** είναι πολύ σημαντικό για τις μηχανές αναζήτησης. Οι μηχανές αναζήτησης κυρίως επεξεργάζονται το λεκτικό περιεχόμενο της ιστοσελίδας, οπότε το κείμενο του γνωρίσματος **alt** προσφέρει μια πιο πλήρη περιγραφή του περιεχομένου της σελίδας.

Για όλους αυτούς τους λόγους το **alt** είναι υποχρεωτικό να το ορίζουμε και να του δίνουμε τιμή που αντιστοιχεί στην περιγραφή της εικόνας, κάθε φορά που εισάγουμε μία εικόνα στο έγγραφο HTML.

3.1.3 Άλλα γνωρίσματα του ****

Ας δούμε μερικά ακόμα γνωρίσματα του στοιχείου **** που είναι χρήσιμα. Δύο γνωρίσματα που έχουν ιδιαίτερη χρήση είναι το **width** και το **height**. Το **width** και το **height** δίνουν είτε σε pixel είτε επί τοις εκατό ή σε άλλες μονάδες μέτρησης (θα μιλήσουμε για μονάδες μέτρησης σε επόμενο κεφάλαιο) το μέγεθος που θα πάρει η εικόνα, το πλάτος δηλαδή και το ύψος της.

Εάν το μέγεθος της εικόνας που πρόκειται να τοποθετήσουμε σε μία θέση είναι διαφορετικό τότε η εικόνα μεγεθύνεται ή μικραίνει ανάλογα με την τιμή που θα πάρει το **width** ή το **height**. Θα πρέπει να προσέξουμε ότι δεν πρέπει να τα ορίζουμε και τα δύο. Πρέπει να ορίζουμε αυτό που έχει σημασία για τη διάταξη περιεχομένου της σελίδας μας, συνήθως το πλάτος γιατί αν ορίσουμε και τα δύο και ο λόγος **width** προς **height** δεν είναι ο λόγος των διαστάσεων της εικόνας, τότε η εικόνα παραμορφώνεται.

Ένα άλλο γνώρισμα που έχει ενδιαφέρον είναι το γνώρισμα **title**. Αυτό το γνώρισμα, όπως και το **alt**, παίρνει ως τιμή ένα κείμενο που περιέχει πρόσθετα στοιχεία για την εικόνα. Θα πρέπει να προσέξουμε ότι δεν είναι σωστό να θέτουμε την ίδια ακριβώς πληροφορία στο γνώρισμα **alt** και **title**. Τα δύο αυτά γνωρίσματα παίζουν διαφορετικό ρόλο. Το μεν **title** προσθέτει πληροφορία συμπληρωματική για κάποιον που βλέπει μεν την εικόνα αλλά θέλει κάτι περισσότερο, μία επεξήγηση, και συνήθως εμφανίζεται σαν **tooltip** πάνω από την εικόνα όταν το ποντίκι υπερίπταται της εικόνας. Ενώ το **alt** όπως είπαμε, είναι η εναλλακτική περιγραφή της εικόνας, άρα δίνει την ουσία της εικόνας για την περίπτωση που η εικόνα δεν φαίνεται ή δεν μπορεί να τη δει ο χρήστης.

3.1.4 Θέματα πνευματικών δικαιωμάτων

Κάτι που πρέπει να προσέξουμε ιδιαίτερα και αξίζει να το αναφέρουμε είναι ότι πρέπει να κάνουμε πολύ προσεκτική χρήση των εικόνων που βρίσκονται στο διαδίκτυο.

Η ενσωμάτωση εικόνας σε ένα έγγραφο HTML μέσω της ετικέτας **img**, όπως είδαμε είναι μια απλή διαδικασία. Μπορούμε να χρησιμοποιήσουμε τη διεύθυνση URL μιας εικόνας που βρίσκεται σε οποιονδήποτε ιστότοπο. Κάτι τέτοιο όμως θα πρέπει να γίνεται αφού ακολουθηθούν κάποια απαραίτητα βήματα, που σχετίζονται με τα πνευματικά δικαιώματα της.

Πρέπει να βεβαιωθούμε ότι ο κάτοχος της εικόνας έχει παραχωρήσει ρητή γραπτή άδεια ή το έχει αναφέρει στην ιστοσελίδα του ότι το υλικό είναι ελεύθερο και με κάποια άδεια χρήσης creative commons που επιτρέπει τη χρήση. Εάν αυτό δεν υπάρχει, ή δεν είναι σαφές, πρέπει να αποφεύγουμε τη χρήση του υλικού. Υπάρχει αρκετό υλικό ελεύθερα διαθέσιμο στο διαδίκτυο, όπως για παράδειγμα οι περισσότερες εικόνες στη wikipedia, όμως ακόμη και σε αυτή την περίπτωση, θα πρέπει να ακολουθούμε τις οδηγίες για αναφορά στο δημιουργό που επισυνάπτονται στο έργο.

3.1.5 Εικόνες με το στοιχείο **<figure>**

Στοιχεία που σχετίζονται με την παρουσίαση εικόνων είναι τα **<figure>** και **<figcaption>**. Θεωρούνται σημασιολογικά στοιχεία (περισσότερα τέτοια στοιχεία θα δούμε σε επόμενη ενότητα). Το στοιχείο **<figure>** επιτρέπει να ομαδοποιήσουμε μία εικόνα ****, καθώς και τη λεζάντα της μέσω του στοιχείου **<figcaption>**. Ας δούμε ένα παράδειγμα:

```
<figure>
  
  <figcaption>Το σπίτι μου στην οδό Φιλικών. </figcaption>
</figure>
```

3.1.6 Πολλαπλές εικόνες με το στοιχείο **<picture>**

Το στοιχείο <picture> χρησιμοποιείται για να υποστηρίξει πολλαπλές εικόνες. Ας δούμε ένα παράδειγμα.

```
<picture>
  <source media="(max-width: 650px)" srcset="img-600.jpg">
  <source media="(max-width: 900px)" srcset="img-900.jpg">
  
</picture>
```

Το στοιχείο <picture> περιέχει ένα πλήθος από στοιχεία <source> το κάθε ένα από τα οποία ορίζει μια διαφορετική εικόνα ανάλογα με τις συνθήκες που ορίζονται από το γνώρισμα **media** το οποίο περιέχει ένα media-query (καθορίζει τη συνθήκη με βάση την οποία η εικόνα εμφανίζεται). Στο παράδειγμα αν το μέγιστο πλάτος της οθόνης είναι 650px ή μικρότερο, τότε επιλέγεται η εικόνα **img-600.jpg**. Αν είναι μέχρι 900px, η εικόνα **img-900.jpg**, αν καμιά από τις δύο συνθήκες δεν ικανοποιείται τότε υπάρχει η προκαθορισμένη εικόνα που περιέχεται στο στοιχείο που είναι η εικόνα **img3.jpg**. Αυτή η εικόνα θα επιλεγεί επίσης, αν ο φυλλομετρητής δεν υποστηρίζει το στοιχείο <picture>, οπότε θα αγνοηθούν τα στοιχεία <source>.

Να σημειωθεί ότι τα στοιχεία <source> χρησιμοποιούνται εκτός από το πλαίσιο του στοιχείου <picture> που είδαμε εδώ, και στα στοιχεία <audio> και <video> που θα δούμε στη συνέχεια.

3.2 Βίντεο και ήχος, τα στοιχεία <video>, <audio>

Στην ενότητα αυτή θα δούμε δύο σημαντικά στοιχεία που αφορούν τα πολυμέσα που χρησιμοποιούνται ολοένα και περισσότερο σήμερα στο διαδίκτυο. Όπως βλέπουμε στο διαδίκτυο σήμερα, ενώ τείνει να λιγοστέψει η χρήση του κειμένου, η εικόνα και το βίντεο χρησιμοποιούνται όλο και περισσότερο.

Η HTML5 έχει βοηθήσει πάρα πολύ αυτή την άνοδο χρήσης αυτών των μέσων, υποκαθιστώντας διάφορες άλλες τεχνολογίες που δεν ήταν ενσωματωμένες σε προηγούμενες εκδόσεις της HTML και χρησιμοποιούνταν για τα πολυμέσα παλαιότερα. Σήμερα η HTML5 με τα στοιχεία **audio** και **video** μας επιτρέπει πολύ εύκολα να εισάγουμε πολυμέσα στη σελίδα μας. Και εδώ όπως με το στοιχείο **img** χρησιμοποιούμε το γνώρισμα **src=** με τιμή τη διεύθυνση που είναι το **audio** ή το **video** αντίστοιχα.

```
<video src="myvideo.mp4" controls>
  <p> Δεν υποστηρίζεται βίντεο </p>
</video>
<audio src="myaudio.mp3" autoplay controls loop>
```

Όπως φαίνεται στο παράδειγμα, τα στοιχεία αυτά υποστηρίζουν διάφορα γνωρίσματα, πολλά από τα οποία είναι κοινά και στα δύο:

- Το γνώρισμα **autoplay** σημαίνει ότι θα αρχίσει να εκτελείται το αρχείο βίντεο ή το αρχείο ήχου όταν έχει κατέβει επαρκές περιεχόμενο, ακόμα πριν ολοκληρωθεί το φόρτωμα.

- Το γνώρισμα **controls** σημαίνει ότι θα εμφανιστούν τα γραφικά στοιχεία ελέγχου της ροής του πολυμεσικού περιεχομένου, όπως stop/start/play.
- Το γνώρισμα **loop** σημαίνει ότι όταν τελειώσει να παίζει ο ήχος ή το βίντεο, θα ξαναρχίσουν από την αρχή.

Μέσα στο στοιχείο **<audio>** καθώς και στο **<video>**, μπορούμε να εισάγουμε κάποιο κείμενο το οποίο θα το δείξει ο φυλλομετρητής στην περίπτωση που δεν υποστηρίζει το στοιχείο αυτό (αν για παράδειγμα δεν υποστηρίζει την HTML5). Σε αυτή την περίπτωση εδώ συνήθως εισάγουμε ένα link για να κατεβάσει το αρχείο αφού ο ίδιος ο φυλλομετρητής δεν μπορεί να το παίξει.

3.2.1 Το στοιχείο **<video>**

Πριν από την HTML5 η ενσωμάτωση βίντεο σε ιστοσελίδες γινόταν με διάφορες τεχνολογίες εξωτερικές της HTML, όπως το Flash, Silverlight κλπ. Η HTML5 εισήγαγε το στοιχείο **video**. Σήμερα θα πρέπει να λάβουμε υπόψη μας ότι ένα μεγάλο ποσοστό των δεδομένων που διακινούνται στο διαδίκτυο είναι τύπου video, μια τάση που αναπτύσσεται περαιτέρω λόγω της αύξησης της ταχύτητας των δικτύων, της σύνδεσης συσκευών TV στο δίκτυο, της ευρείας χρήσης βίντεο σε κοινωνικά δίκτυα, της ανάγκης επικοινωνίας και συνεργασίας μέσω τηλεδιασκέψεων, κλπ., ενώ υπάρχει μεγαλύτερη ζήτηση να χρησιμοποιήσουμε το βίντεο σε παρουσίαση προϊόντων, στη διδασκαλία, σε παρουσίαση ερευνητικών αποτελεσμάτων, κλπ.

Όπως είδαμε το στοιχείο **<video>** της HTML5 μάς επιτρέπει να παρέχουμε πρόσβαση σε διαφορετικούς μορφότυπους βίντεο, ανάλογα με τις δυνατότητες που υποστηρίζει ο φυλλομετρητής του χρήστη.

Έστω το παρακάτω παράδειγμα:

```
<video controls>
  <source src="myvideo.mp4" type="video/mp4">
  <source src="myvideo.webm" type="video/webm">
  <p>Ο browser δεν υποστηρίζει video HTML5. Συνδεθείτε
    <a href="myvideo.mp4 ">με το βίντεο</a></p>
</video>
```

Παρατηρούμε ότι μέσω του στοιχείου **<source>** που είδαμε σε προηγούμενη ενότητα να παρέχει αναλλακτικές επιλογές εικόνων, δίνεται η δυνατότητα στον φυλλομετρητή να εμφανίσει διαφορετικούς μορφότυπους βίντεο, για παράδειγμα να διαθέτουμε εναλλακτικές εκδόσεις του ίδιου βίντεο για να υποστηρίζουμε περισσότερους φυλλομετρητές. Η πρώτη πηγή βίντεο είναι μορφότυπου **mp4** που είναι ο πιο συνηθισμένος μορφότυπος στο διαδίκτυο, ενώ η δεύτερη πηγή είναι μορφότυπου **webm**. Οι διαφορές μεταξύ **webm** και **mp4** σχετίζονται με τον τρόπο που γίνεται το πακετάρισμα της ροής **audio** και ροής **video** μέσα σε μία ροή. Ο μορφότυπος **webm** έχει τον ήχο Ogg Vorbis με codec βίντεο VP8/VP9 και υποστηρίζεται από Firefox και Chrome ενώ ο **mp4** που είναι πιο διαδεδομένος, έχει ήχο AAC mp3 και codec video H.264. που είναι πρότυπο. Υποστηρίζεται και από τους προηγούμενους αλλά και από τον Explorer και Safari.

Τα γνωρίσματα που μπορούμε να ορίσουμε σε ένα στοιχείο **<video>** είναι παρόμοια με αυτά που είδαμε προηγουμένως στο στοιχείο **** για τις εικόνες. Παραδείγματος χάρη όπως είπαμε το γνώρισμα **controls** μας επιτρέπει να εμφανίσουμε τα χειριστήρια του βίντεο (play/stop, κλπ.), το γνώρισμα **width** μας παρέχει τη δυνατότητα να ορίσουμε το πλάτος του παραθύρου του video, **height** το ύψος, κλπ. Αν

ορίσουμε και τα δύο αυτά γνωρίσματα, θα αγνοηθούν γιατί το video συντηρεί τον λόγο προβολής, δεν παραμορφώνεται όπως η εικόνα. Επίσης υπάρχει το γνώρισμα **autoplay** (άμεση έναρξη του βίντεο ενώ το αρχείο κατεβαίνει), **loop** (επανεκκίνηση του βίντεο μετά το τέλος του) και **muted** (ο ήχος σε σίγαση). Επίσης το γνώρισμα **poster** μας επιτρέπει να ορίσουμε ένα αρχείο img που μπαίνει στη θέση του παραθύρου του video όταν το video δεν παίζει. Όπως είπαμε ακόμη ως περιεχόμενο του στοιχείου <video> μπορούσε να βάλουμε εκτός από διάφορα στοιχεία <source> και κείμενο το οποίο θα φαίνεται αν ο φυλλομετρητής δεν δείχνει αυτό το video, το οποίο μπορεί να περιλαμβάνει υπερσύνδεσμο σε αρχείο βίντεο.

3.2.1 Το στοιχείο <audio>

Αντίστοιχα με το στοιχείο video είναι η λειτουργία του στοιχείου <audio> που αφορά τον ήχο. Θα πρέπει να είμαστε ιδιαίτερα προσεκτικοί με τη χρήση του ήχου (μουσική, μηνύματα, κλπ.), όπως βεβαίως και του βίντεο, χωρίς τη συγκατάθεση του χρήστη. Αν ο χρήστης συνδεθεί σε μια ιωτοσελίδα σε ένα δημόσιο χώρο (μέσο μεταφοράς, ή μια δημόσια βιβλιοθήκη) και απρόσμενα η σελίδα που επισκέφθηκε άρχισε να παράγει ήχους, αυτό μπορεί να ενοχλήσει τον χρήστη και τους γύρω του. Και σε αυτή την περίπτωση μπορούμε να χρησιμοποιήσουμε το γνώρισμα **controls**, ενώ όπως και στην προηγούμενη περίπτωση με το βίντεο και την εικόνα, μπορούμε να ορίσουμε πολλαπλές εναλλακτικές πηγές ήχου, καλύπτοντας την περίπτωση που ο φυλλομετρητής δεν υποστηρίζει κάποιο από αυτά.

Ένα παράδειγμα:

```
<audio controls>
  <source src="mysong.mp3" type="audio/mp3">
  <source src="mysong.ogg" type="audio/ogg">
  <p>Your browser δεν επιτρέπει ήχο HTML5. Ακολουθήστε
  a <a href="mysong.mp3">σύνδεσμο στο αρχείο ήχου</a></p>
</audio>
```

Ο μορφότυπος **MP3** είναι ο πιο συνηθισμένες, ενώ εναλλακτικά χρησιμοποιείται ο μορφότυπος **ogg**. Και στο στοιχείο <audio> μπορούμε να εισάγουμε κάποιο κείμενο που εμφανίζεται εναλλακτικά για την περίπτωση που ο φυλλομετρητής δεν υποστηρίζει το στοιχείο αυτό.

3.3 Ενσωμάτωση στοιχείων <iframe> και <object>

Σε αυτήν την ενότητα θα δούμε κάποιες πολύ σημαντικές δυνατότητες που μας δίνει η HTML5, η οποία μάς επιτρέπει να ενσωματώσουμε άλλα αντικείμενα, όπως μία άλλη ιστοσελίδα ή αρχεία διαφορετικών τύπων στη σελίδα μας. Αυτές είναι δυνατότητες πολύ ισχυρές τις οποίες όμως θα πρέπει να χρησιμοποιούμε με ιδιαίτερη προσοχή και μέτρο.

3.3.1 Ενσωμάτωση στοιχείων <iframe>

Η πρώτη δυνατότητα που μας δίνεται είναι η ενσωμάτωση στοιχείων <iframe> (inline frame). Το στοιχείο αυτό μας επιτρέπει να εισάγουμε ένα frame το οποίο ορίζουμε να λαμβάνει περιεχόμενο από ένα εξωτερικό πόρο, και να ενσωματώνεται σε μία δική μας σελίδα.

Ας δούμε ένα παράδειγμα:

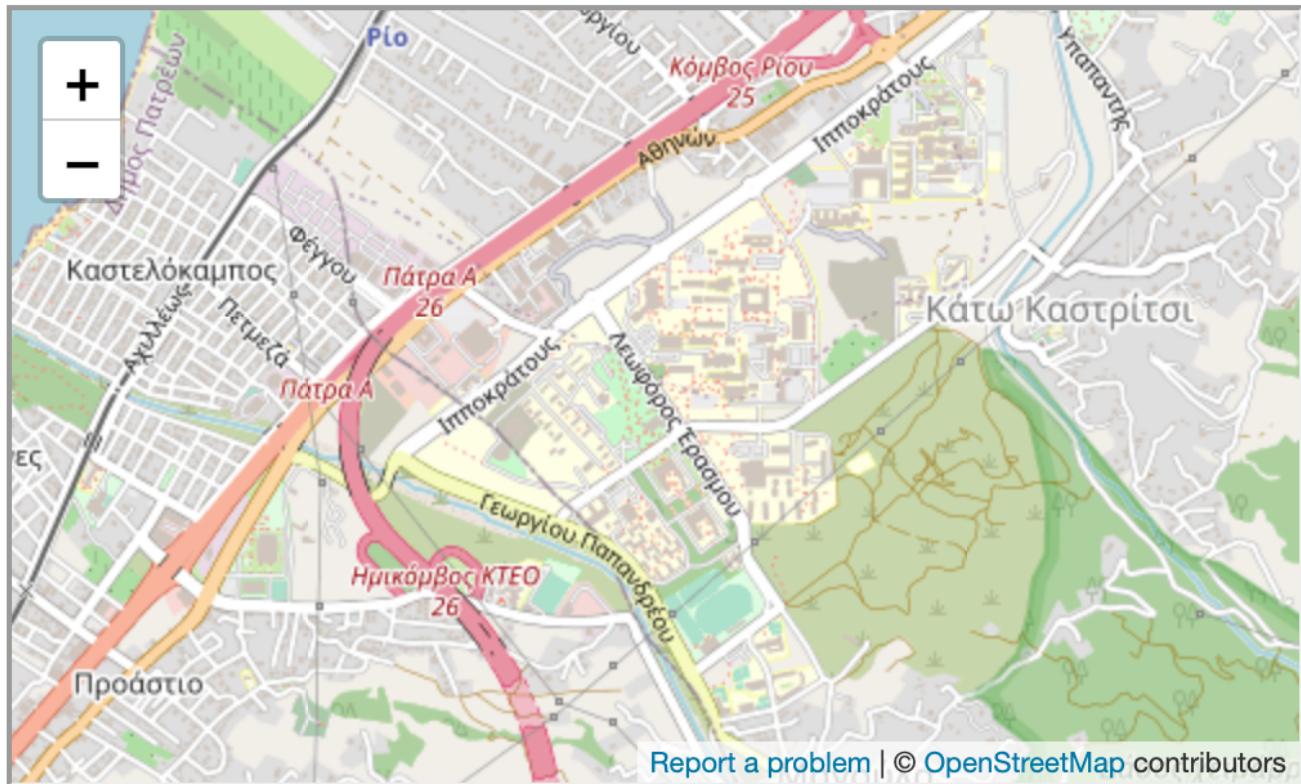
```

<body>
  <h1> Ο χάρτης του Πανεπιστημίου Πατρών</h1>
  <iframe id="inlineFrameExample"
    title="Το Πανεπιστήμιο Πατρών"
    width="500"
    height="300"
    frameborder="2"
    src="https://www.openstreetmap.org/export/embed.html?
bbox=21.7742%2C38.2814%2C21.7987%2C38.2943&layer=mapnik">
  </iframe>
</body>

```

Το αποτέλεσμα του παραπάνω εγγράφου είναι το εξής:

Ο χάρτης του Πανεπιστημίου Πατρών



{#fig:fig1 width=70%}

Εδώ παρατηρούμε ότι έχουμε ενσωματώσει ένα στοιχείο `<iframe>` το οποίο έχει τις δικές του διαστάσεις (`width`, `height`) και ως πηγή του περιεχομένου του ένα τμήμα του χάρτη openstreetmap. (σημείωση: το απόσπασμα του χάρτη είναι εύκολο να ανακτηθεί καθώς και ο κώδικας από την ιστοσελίδα των [openStreetMap](#), με την επιλογή `share` και ακολουθούμε τις οδηγίες για ανάκτηση κώδικα `iframe` για το περιεχόμενο του χάρτη που έχουμε επιλέξει.)

Ας δούμε τα κύρια χαρακτηριστικά του στοιχείου `<iframe>`. Αυτό παρουσιάζει αρκετές ομοιότητες με άλλα στοιχεία που είδαμε προηγούμενα. Δέχεται ως γνώρισμα ένα `source`, μια διεύθυνση ενός πόρου `url` από το οποίο θα αντλήσουμε το περιεχόμενο. Επίσης πρέπει να ορίσουμε τις διαστάσεις του `frame`, δηλαδή

του τμήματος της δικής μας ιστοσελίδας όπου θα τοποθετηθεί το περιεχόμενο από αυτόν τον εξωτερικό πόρο. Μπορούμε επίσης να ορίσουμε αν θα υπάρχει πλαίσιο, με το γνώρισμα **frameborder**, να επιτρέπεται ή όχι η πλήρης κάλυψη της οθόνης με το λογικό γνώρισμα **allowfullscreen**. Ακόμη ένα γνώρισμα με ιδιαίτερο ενδιαφέρον είναι το **sandbox** που επιτρέπει αυξημένη ασφάλεια του περιβάλλοντος του φυλλομετρητή μας από τον εξωτερικό πόρο. Επίσης μέσα στο στοιχείο **<iframe>** είναι καλό να εισάγουμε περιεχόμενο το οποίο θα δει ο χρήστης σε περίπτωση που ο φυλλομετρητής του δεν υποστηρίζει το στοιχείο **iframe**.

Για παράδειγμα αν στον παραπάνω κώδικα, προσθέσουμε το γνώρισμα **sandbox** ο φυλλομετρητής Chrome δεν επιτρέπει την εμφάνιση του χάρτη και παράγει το εξής διαγνωστικό: Blocked script execution in '[https://...](#)' because the document's frame is sandboxed and the 'allow-scripts' permission is not set .

Θέλει πολύ προσοχή εδώ βεβαίως γιατί θα πρέπει να εξασφαλίσουμε, όπως γίνεται και με άλλα πολυμεσικά υλικά με βίντεο, ήχους και εικόνες, ότι έχουμε δικαίωμα να χρησιμοποιήσουμε το περιεχόμενο και να το προβάλλουμε στη δική μας ιστοσελίδα. Θα πρέπει επίσης να προσέξουμε ότι πολλές ιστοσελίδες δεν επιτρέπουν να εισάγουμε σαν **iframe** το περιεχόμενό τους ενώ κάποιες άλλες το επιτρέπουν. Τέλος θα πρέπει να λάβουμε υπόψη μας ότι κάθε **iframe** που ενσωματώνουμε στη σελίδα μας, θα έχει τη δική του ιστορία, και **document** (DOM), επιβαρύνοντας τη μνήμη του φυλλομετρητή μας. Άρα θα πρέπει να γίνεται με φειδώ η ενσωμάτωση περιεχομένου με τη μέθοδο αυτή.

3.3.2 Ενσωμάτωση στοιχείων **<object>**

Μια εναλλακτική επιλογή ενσωμάτωσης περιεχομένου εξωτερικών πόρων είναι η χρήση στοιχείων **<object>**. Είναι ένα στοιχείο γενικού σκοπού το οποίο μάς επιτρέπει ενσωμάτωση του εξωτερικού πόρου, είτε ως εικόνας, είτε ως πόρου που θα τον χειριστεί ένα plugin.

Ας δούμε ένα παράδειγμα.

```
<object data="mypdf.pdf" type="application/pdf" width="800"
height="1200" >
<p>Μπορείτε να κατεβάσετε το αρχείο από
<a href="mypdf.pdf">τον σύνδεσμο (800KB).</a></p>
</object>
```

Στο παράδειγμα αυτό ενσωματώνουμε σε μια περιοχή 800 x 1200px της σελίδας μας ένα αρχείο pdf. Για τον σκοπό αυτό ορίζουμε ένα στοιχείο **<object>**, στο οποίο περνάμε τον πόρο μέσω του γνωρίσματος **data**. Είναι απαραίτητο μέσω του γνωρίσματος **type** να ορίσουμε τον τύπο του πόρου, χρησιμοποιώντας κατηγοριοποίηση MIME. Αυτό ώστε ο φυλλομετρητής να γνωρίζει πώς θα χειριστεί τον πόρο, εφόσον υπάρχει αυτή η δυνατότητα.

Άλλα γνωρίσματα που μπορούμε να χρησιμοποιήσουμε στο στοιχείο αυτό αφορούν το πλάτος του πλαισίου (**border**), και οι διαστάσεις του **width** και **height**.

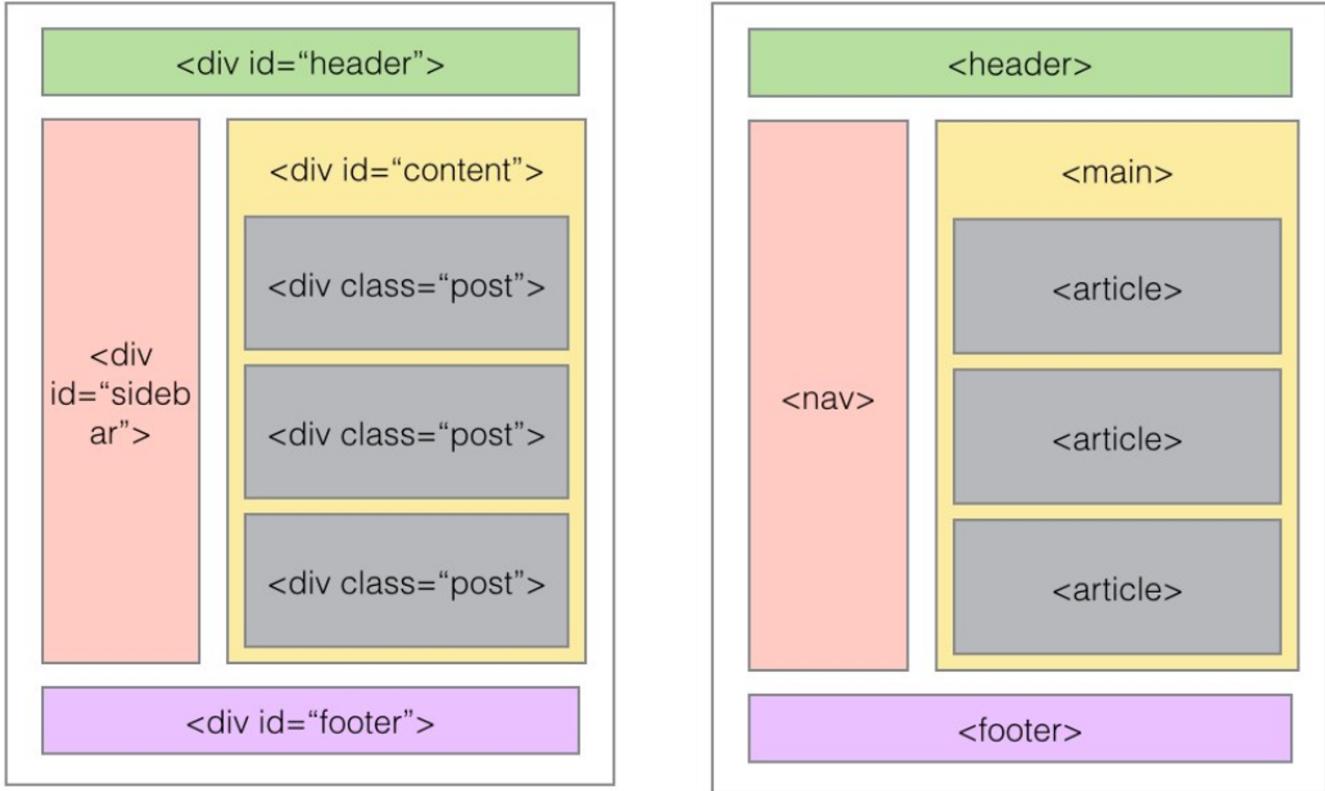
Άσκηση

Να πάτε στο youtube, και να ελέγξετε τον κώδικα που παράγεται αν επιλέξετε για κάποιο βίντεο την επιλογή "share" και στη συνέχεια "Ενσωμάτωση βίντεο". Ποιο στοιχείο HTML παράγεται; Πώς μπορείτε να

τροποποιήσετε τον κώδικα που σάς προτείνει ώστε να μην ξεκινάει το βίντεο αυτόματα, μελετήστε τις παραμέτρους του στοιχείου που προτείνεται.

3.4 Δομικά στοιχεία HTML

Σε αυτή την ενότητα θα δούμε κάποια ακόμα στοιχεία που έχουν εισαχθεί στην HTML5, τα οποία μας επιτρέπουν να δομήσουμε σημασιολογικά το περιεχόμενο μιας ιστοσελίδας.



{#fig:fig2 width=70%}

Έστω ότι το περιεχόμενο μιας σελίδας επιθυμούμε να το δομήσουμε όπως φαίνεται στο σχήμα @fig:fig2. Δηλαδή μια κεφαλίδα, ένα υποσέλιδο, μια πλαινή μπάρα που χρησιμεύει ως μενού πλοήγησης, και κάποιο κυρίως περιεχόμενο το οποίο δομείται σε επιμέρους τμήματα. Αυτή μπορεί να είναι μια ιστοσελίδα με ειδήσεις ή ένα ηλεκτρονικό κατάστημα, που μπορεί να δομηθεί με αυτόν τον τρόπο.

Ο κλασσικός τρόπος για να οργανώσουμε τους υποδοχείς περιεχομένου, σε προηγούμενες εκδόσεις της HTML είναι να ορίσουμε στοιχεία `<div>` καθένα από τα οποία έχει μια ταυτότητα `id` ή ανήκει σε μια κλάση `class` στην περίπτωση που το στοιχείο αυτό επαναλαμβάνεται. Αυτός ο τρόπος φαίνεται στο αριστερό σχήμα της εικόνας. Για παράδειγμα η πλαινή μπάρα πλοήγησης ορίζεται ως `<div id="sidebar">` ενώ κάθε ένα από τα επί μέρους άρθρα ορίζεται ως `<div class="post">`. Το στοιχείο `<div>` όμως δεν φέρει ιδιαίτερη σημασιολογία, είναι απλά ένας υποδοχέας, ένα block element. Με αυτή τη μέθοδο, όλα τα στοιχεία της σελίδας είναι `div` και αυτός παραμένει ένας τρόπος που πολλοί προγραμματιστές χρησιμοποιούν ακόμα και σήμερα.

Ας δούμε την καινούργια δυνατότητα που μας παρέχει η HTML5, η οποία εισάγει νέα στοιχεία που επιτρέπουν να προσδιορίζουμε τη σημασιολογία καθενός από αυτά τα δομικά στοιχεία.

Τέτοια στοιχεία είναι το στοιχείο `<header>` για την κεφαλίδα της σελίδας, `<nav>` για τη μπάρα πλοήγησης (Navigation bar) η οποία μπορεί να είναι στο πάνω μέρος ή στο πλάι, ανάλογα με τη διάταξη της σελίδας, ένα στοιχείο `<main>` το οποίο πρέπει να είναι ένα και μοναδικό και μέσα σε αυτό στοιχεία `<article>`

καθώς και `<section>`. Ακόμη μπορεί να υπάρχει μια πλαϊνή μπάρα που περιγράφεται από το στοιχείο `<aside>` και συνήθως περιέχεται στο στοιχείο `<main>`. Τέλος έχουμε ένα στοιχείο `<footer>` για το υποσέλιδο. Αυτά λοιπόν είναι τα κύρια δομικά στοιχεία που μας προσφέρονται, ένα παράδειγμα χρήσης τους φαίνεται στα δεξιά της εικόνας @fig:fig2

Μία συζήτηση που γίνεται αφορά για το πώς δομούνται τα στοιχεία `section` και `article`. Για κάποιους σχεδιαστές ένα `section` περιέχει πολλά στοιχεία `article`. Αυτή η άποψη προτείνει ένα `section` ως θεματική ομαδοποίηση περιεχομένου μεσα στο οποίο περιέχονται αυτόνομα τμήματα πληροφορίας που είναι τα `articles`. Κάθε ένα από αυτά τα `articles` θα πρέπει να έχει μέσα του `headers h1 h2` κλπ. με οργάνωση τέτοια ώστε το κάθε `article` να έχει μόνο ένα `header h1` μέσα στο οποίο να υπάρχουν πολλά `h2` και ούτω καθεξής. Αυτός ο τρόπος οργάνωσης έχει υιοθετηθεί στο παραπάνω σχήμα.

Για κάποιους άλλους σχεδιαστές υπάρχει η αντίστροφη δομή ότι το `article` είναι το κυρίως στοιχείο και οι επιμέρους περιοχές του είναι τα `sections`. Θα πρέπει όμως να λάβουμε υπόψη ότι για τα επί μέρους στοιχεία ενός `article` έχουμε τα άλλα δομικά στοιχεία, τα οποία υπήρχαν και παλαιότερα που είναι τα `h1, h2` κλπ., που μας επιτρέπουν να δομήσουμε ιεραρχικά το περιεχόμενο ενός `article`.

Τα `div` και `span` όπως είπαμε είναι στοιχεία που υπάρχουν ήδη και χρησιμοποιούνται εκτεταμένα. Μάλιστα μερικοί σχεδιαστές τα προτιμούν, είναι πιο συνηθισμένοι στο να τα χρησιμοποιούν. Όμως η νέα πρόταση της HTML5 είναι να χρησιμοποιήσουμε τα `div` και `span` μόνο για την περίπτωση που δεν μας ταιριάζει να χρησιμοποιήσουμε κάποιο από τα άλλα δομικά στοιχεία που ήδη αναφέραμε. Υπενθυμίζεται ότι το `span` είναι ένα `inline element` που αφορά τμήμα του κειμένου, ενώ το `div` είναι ένα `block element`. Τα στοιχεία αυτά όμως δεν φέρουν σημασιολογία, αλλά πρέπει εμείς να τους δώσουμε σημασιολογία συνήθως με ένα `id` και το `div` ορίζει ένα `block`. Υπάρχουν περιπτώσεις που ένα στοιχείο έχει ειδική σημασιολογία η οποία δεν καλύπτεται από τα δομικά στοιχεία που αναφέρθηκαν, τότε χρησιμοποιούνται τα `div` ή `span`. Ας δούμε ένα παράδειγμα.

Ας πούμε ότι έχουμε αυτή την παράγραφο `<p>` μέσα στην οποία θέλουμε να εισάγουμε μία σημείωση του επιμελητή της έκδοσης, η σημείωση εισάγεται μέσω ενός στοιχείου `span` στο οποίο επισυνάπτουμε την κλάση "editor-note".

```
<p> Η βασίλισσα μπήκε ακολουθούμενη από την πομπή της στο δωμάτιο <span
class="editor-note">[Σημείωση: Τα φώτα χαμηλώνουν καθώς εισέρχεται η
πομπή]</span>.</p>
```

Στην κλάση αυτή θα επισυνάψουμε κάποιο στυλ ειδικό για αυτού του τύπου τις σημειώσεις την οποία βάζουμε και εντάσσουμε μέσα σε αυτήν την παράγραφο.

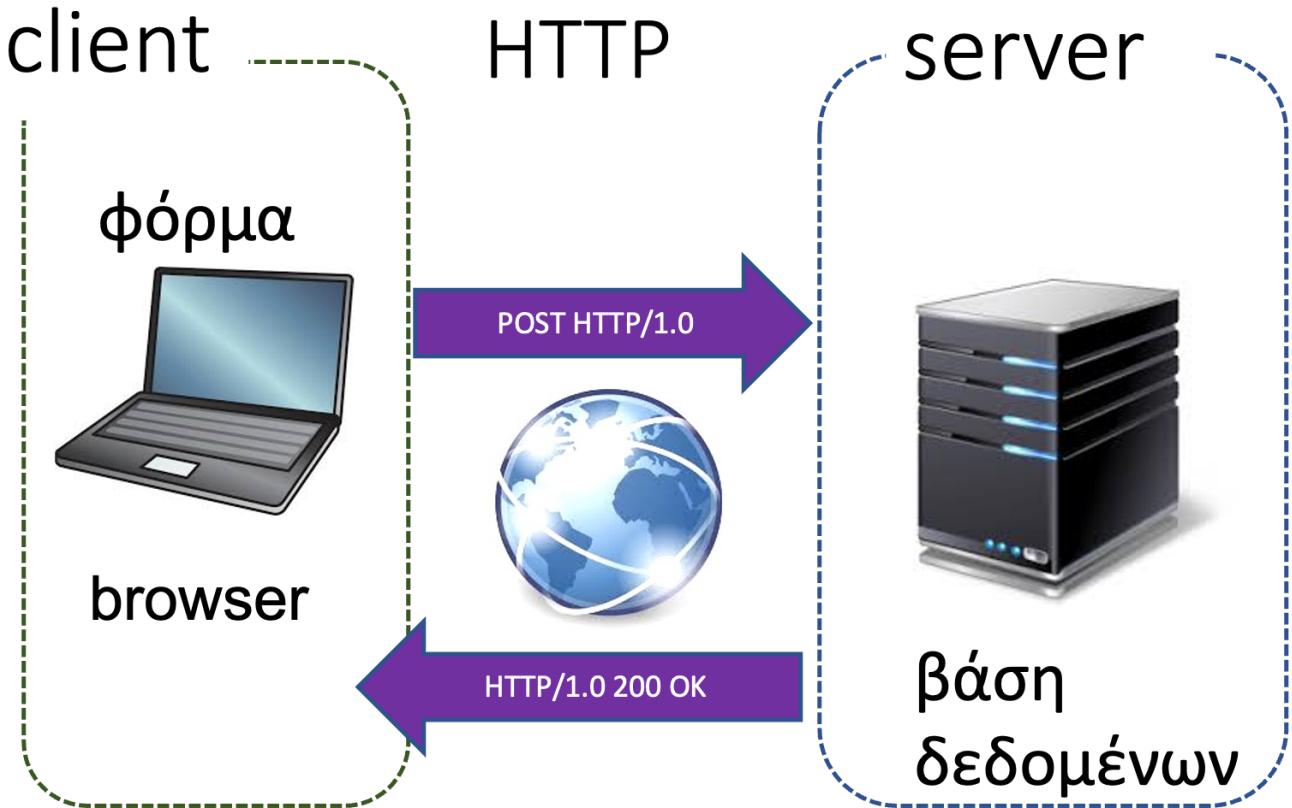
3.5 Φόρμες: το στοιχείο `<form>`

Αυτή είναι η τελευταία ενότητα του κεφαλαίου, στην οποία θα συζητήσουμε ένα πολύ σημαντικό στοιχείο της HTML που είναι το στοιχείο `<form>`. Είναι το στοιχείο που μας επιτρέπει να δημιουργήσουμε φόρμες μέσω των οποίων ο χρήστης εισάγει δεδομένα σε ένα σύστημα. Συμπληρώνουμε συνεχώς φόρμες στη ζωή μας, είτε διαδικτυακά, όταν συμπληρώνουμε ένα `password`, ένα όνομα χρήστη, ή τα δημογραφικά μας στοιχεία όταν ζητείται για να κάνουμε εγγραφή σε μία καινούργια υπηρεσία ή ακόμη συμπληρώνουμε έντυπες φόρμες όταν συναλλασσόμαστε με το δημόσιο ή με άλλους φορείς (συμπληρώνουμε ένα

ερωτηματολόγιο, υπογράφουμε μια υπεύθυνη δήλωση, κλπ.). Η σχεδίαση μίας καλής φόρμας διέπεται από κάποιες βασικές αρχές, τις οποίες θα συζητήσουμε στο πλαίσιο αυτής της ενότητας.

3.5.1 Η διαδικασία υποβολής φόρμας

Ας αρχίσουμε όμως με την αρχιτεκτονική μιας διαδικτυακής εφαρμογής που περιλαμβάνει μια φόρμα. Αυτή φαίνεται στο σχήμα.



{#fig:fig3 width=60%}

Οι εφαρμογές του διαδικτύου όπως έχει ήδη αναφερθεί, ακολουθούν το μοντέλο πελάτη-εξυπηρετητή. Ο πελάτης (ο φυλλομετρητής) είναι εκεί που παρουσιάζεται η φόρμα στον χρήστη. Όταν ο χρήστης συμπληρώσει τα στοιχεία της φόρμας τότε πατάει ένα πλήκτρο **υποβολή φόρμας (submit)**. Όταν ο χρήστης εισάγει δεδομένα ή υποβάλει τη φόρμα, τα στοιχεία ελέγχονται στον ίδιο τον φυλλομετρητή ως προς την εγκυρότητά τους.

Για παράδειγμα αν υπήρχε ένα υποχρεωτικό στοιχείο της φόρμας που ζητάει την ηλικία ελέγχεται αν συμπληρώθηκε και αν η ηλικία που έδωσε ο χρήστης είναι αποδεκτή. Αφού γίνουν οι έλεγχοι εάν κάποιος από τους ελέγχους δεν ικανοποιηθεί ένα μήνυμα στέλνεται στο χρήστη και ζητείται να συμπληρώσει και να ξανά-υποβάλει τη φόρμα. Αυτοί οι έλεγχοι μπορεί να γίνουν από τα ίδια τα στοιχεία της HTML5, ενώ συχνά χρησιμοποιείται κώδικας JavaScript.

Αν υποθέσουμε τώρα ότι η φόρμα συμπληρώθηκε σωστά, τότε τα στοιχεία της φόρμας στέλνονται μέσα από το πρωτόκολλο http προς τον εξυπηρετητή είτε με τη μέθοδο GET ή την μέθοδο POST. Ο εξυπηρετητής λαμβάνει τα στοιχεία, ελέγχει την εγκυρότητά τους, και τα συσχετίζει με στοιχεία που διαθέτει ήδη. Ο εξυπηρετητής για παράδειγμα μπορεί να ελέγξει αν το όνομα που δίνει κάποιος ως Login name έχει ήδη χρησιμοποιηθεί από άλλους χρήστες. Εάν τα στοιχεία είναι έγκυρα, τότε ο εξυπηρετητής

παράγει μία απόκριση η οποία επιστρέφει προς το χρήστη. Η απόκριση αυτή μπορεί να είναι: εντάξει, ο κωδικός που έδωσες ήταν σωστός, μπορείς να μπεις στην υπηρεσία μας, πχ στην ιστοσελίδα του κοινωνικού δικτύου, ή μία ανάδραση μπορεί να είναι λάθος, τα στοιχεία που έδωσες δεν είναι σωστά. Επίσης ο εξυπηρετητής μπορεί να αποθηκεύσει τα στοιχεία που έδωσε ο χρήστης σε μια βάση δεδομένων.

Η σύνταξη του στοιχείου `<form>` φαίνεται στη συνέχεια:

```
<form action="/my-handling-form-page" method="post">
    ... τα στοιχεία της φόρμας ...
</form>
```

Δύο γνωρίσματα που καθορίζουν τη συμπεριφορά του στοιχείου `form` είναι το γνώρισμα `action` και `method`.

- Το γνώρισμα `action` παίρνει ως τιμή τη διεύθυνση url του πόρου του διαδικτύου στην οποία θα αποσταλεί η φόρμα για επεξεργασία όταν την υποβάλλει ο χρήστης. Εάν δεν έχει δοθεί τιμή, το γνώρισμα `action`, αυτό σημαίνει ότι η φόρμα θα σταλεί στην ίδια διεύθυνση url της ιστοσελίδας στην οποία υπάρχει η ίδια η φόρμα.
- Το γνώρισμα `method` περιγράφει τη μέθοδο της http που θα χρησιμοποιηθεί για αποστολή της φόρμας στον εξυπηρετητή, και αυτή είναι είτε `post` ή `get`. Αν δεν δοθεί τιμή στο γνώρισμα `method` τότε θεωρείται ότι η μέθοδος είναι η `get`.

Ένα από τα ερωτήματα που τίθενται είναι ποια από τις μεθόδους `get/post` να χρησιμοποιηθεί για υποβολή της φόρμας.

Η διαφορά είναι ότι η `get` περιλαμβάνει τα δεδομένα που εισάγαγε ο χρήστης της φόρμας στο URL που βρίσκεται στην κεφαλίδα του μηνύματος ως query string. Είδαμε στο πρώτο κεφάλαιο πώς συντάσσεται μια διεύθυνση URL στην οποία εισάγουμε τα δεδομένα της φόρμας. Για παράδειγμα η διεύθυνση <https://www.google.gr/search?q=python> έχει προκύψει από αποστολή μιας φόρμας με ένα πεδίο με όνομα `q` στο οποίο ο χρήστης έδωσε την τιμή `python`. Το μέγεθος του URL είναι περιορισμένο (2048 χαρακτήρες) άρα δεν μπορούμε να στείλουμε πολλά δεδομένα με αυτόν τον τρόπο. Ένα θετικό όμως αυτής της μεθόδου είναι ότι μπορούμε να αποθηκεύσουμε το αποτέλεσμα της συμπλήρωσης της φόρμας ως διεύθυνση URL.

Η `post` από την άλλη μεριά, είναι η πιο συνηθισμένη μέθοδος για αποστολή στον εξυπηρετητή των δεδομένων της φόρμας. Τα δεδομένα σε αυτή την περίπτωση στέλνονται στο σώμα του μηνύματος. Θα πρέπει να σημειωθεί ότι δεν υπάρχουν περιορισμοί στο μέγεθος δεδομένων που στέλνονται με τη μέθοδο `post`, όμως δεν αποθηκεύονται στο URL τα δεδομένα, όπως στην περίπτωση της μεθόδου `get`.

3.5.2 Περιεχόμενο στοιχείου `<form>`

Στην ενότητα αυτή θα συζητήσουμε τα στοιχεία που μπορεί να περιέχονται μέσα σε ένα στοιχείο `<form>`.

Όπως ξέρουμε από την εμπειρία μας, μια φόρμα μπορεί να περιέχει πεδία εισαγωγής δεδομένων, που αντιστοιχούν όπως θα δούμε σε διάφορους τύπους στοιχείων `<input>` καθώς και πεδία με δυνατότητα επιλογής από εναλλακτικές τιμές, κλπ. Επίσης η φόρμα εμπεριέχει επεξηγηματικά κείμενα τα οποία είναι απαραίτητα ως υπόδειξη στο χρήστη, είναι τα στοιχεία `<label>`.

Το πιο συνηθισμένο στοιχείο <input> είναι ένα πεδίο κειμένου μέσα στο οποίο καλείται να εισάγει ο χρήστης δεδομένα, σύμφωνα με μια υπόδειξη. Εναλλακτικά μπορεί η φόρμα να περιέχει ένα μεγαλύτερο πλαίσιο πολλών γραμμών, στο οποίο ο χρήστης μπορεί να εισάγει κείμενο, για παράδειγμα να γράψει τα σχόλιά του. Αυτό είναι το στοιχείο <textarea> της HTML.

Επίσης σε μια φόρμα θα δούμε στοιχεία επιλογής ναι/όχι (checkbox), ή αμοιβαία αποκλειόμενες μεταξύ τους επιλογές (radio buttons), στοιχεία επιλογής τιμής από εύρος τιμών (range). Επίσης η φόρμα μπορεί να περιέχει πιο σύνθετα στοιχεία, πεδία επιλογής ημερομηνίας, χρώματος, αρχείου, κλπ. Τέλος περιέχει πλήκτρα τα οποία έχουν σχέση με τη φόρμα, όπως το πλήκτρο Υποβολή που όταν πατηθεί η φόρμα υποβάλλεται στον εξυπηρετητή, ή το πλήκτρο Εκκαθάριση για επανεκκίνηση συμπλήρωσης της φόρμας.

3.5.3 Παράδειγμα φόρμας

Ακολουθεί το πρώτο παράδειγμα φόρμας που θα συζητήσουμε.

Παρακαλώ δώστε τα στοιχεία σας:

Όνομα:

Email:

{#fig:fig4 width=50%}

Η φόρμα αυτή περιέχει δύο πεδία εισαγωγής κειμένου, ένα για το όνομα, και ένα για το email. Επίσης περιέχει ένα τίτλο, υποδείξεις και ένα πλήκτρο υποβολής.

Ο κώδικας που παράγει αυτή τη φόρμα είναι ο εξής:

```
<form>
    <h1>Παρακαλώ δώστε τα στοιχεία σας:</h1>
    <div>
        <label for="name">Όνομα:</label>
        <input type="text" id="name" name="user_name">
    </div>
    <div>
        <label for="mail">Email:</label>
        <input type="email" id="mail" name="user_mail">
    </div>
    <div>
        <input type="submit" value="Υποβολή">
    </div>
</form>
```

Το στοιχείο `<form>` περιέχει ένα στοιχείο `<h1>` και τρία στοιχεία `<div>`, το καθένα από τα οποία περιέχει ένα στοιχείο `<input>`.

Αν εξετάσουμε το πρώτο στοιχείο `<div>`, περιέχει ένα `<label>`, που αφορά στην υπόδειξη προς τον χρήστη, και ένα στοιχείο `<input type="text">` που είναι το πεδίο στο οποίο εισάγει ο χρήστης τα δεδομένα. Τα δύο αυτά στοιχεία συνδέονται μεταξύ τους μέσω του γνωρίσματος `for` του στοιχείου `<label>`, το οποίο έχει τιμή την ταυτότητα `id` του στοιχείου `<input>`.

Όπως θα δούμε υπάρχουν πολλοί διαφορετικοί τύποι στοιχείων `<input>`, το πιο συνηθισμένο είναι το στοιχείο τύπου `text`. Το στοιχείο `<input>` έχει ως γνώρισμα `name="user_name"`. Το γνώρισμα αυτό είναι το όνομα της μεταβλητής στην οποία εκχωρείται η τιμή που θα δώσει ο χρήστης στο αντίστοιχο πεδίο. Το γνώρισμα `name` είναι υποχρεωτικό αν θέλουμε να έχουμε πρόσβαση στα δεδομένα του στοιχείου, ή μεταβλητή αυτή χρησιμοποιείται από την JavaScript και στέλνεται ως ζεύγος `μεταβλητή=τιμή` στον εξυπηρετητή.

Το ίδιο θα παρατηρήσουμε να γίνεται και στο επόμενο πεδίο `<input type="email">`, στο οποίο δίνεται το όνομα `name="user_mail"`.

Όταν υποβάλουμε τη φόρμα, αν υποθέσουμε ότι ο χρήστης έχει συμπληρώσει στο πρώτο πεδίο το όνομα "Nikos" και στο δεύτερο πεδίο "nikos@mail.com", θα παρατηρήσουμε ότι η φόρμα θα υποβληθεί (method=get) στο URL :

```
...?user_name=Nikos&user_mail=nikos%40mail.com
```

Παρατηρούμε ότι τα ζεύγη τιμών που στέλνονται στον εξυπηρετητή μέσω του URL απαρτίζονται από το όνομα του κάθε πεδίου και την τιμή που εισήγαγε ο χρήστης.

Ένα ερώτημα που τίθεται είναι σε τι διαφέρει το στοιχείο `<input type="email">` από το πρώτο στοιχείο `<input type="text">` αφού και τα δύο έχουν ακριβώς την ίδια εμφάνιση, και τα δύο επιτρέπουν την εισαγωγή κειμένου στον χρήστη. Η διαφορά φαίνεται αν βάλουμε ένα κείμενο για παράδειγμα "abcd" στο δεύτερο πεδίο και προσπαθήσουμε να υποβάλουμε τη φόρμα, τότε θα πάρουμε ένα μήνυμα σαν αυτό που φαίνεται στην εικόνα:

Παρακαλώ δώστε τα στοιχεία σας:

Όνομα:

Email:

! Please include an '@' in the email address. 'abcd' is missing an '@'.

{#fig:fig4b width=50%}

Αυτό το μήνυμα προκύπτει από τον έλεγχο που γίνεται κατά την υποβολή της φόρμας στο πεδίο τύπου "email", από τον οποίο προκύπτει ότι η συμβολοσειρά που έδωσε ο χρήστης δεν είναι αποδεκτή διεύθυνση

email αφού δεν περιέχει το σύμβολο "@".

Το τρίτο στοιχείο input της φόρμας είναι τύπου "submit". Το στοιχείο αυτό έχει τη μορφή ενός πλήκτρου και έχει την ειδική χρήση, ότι όταν πατηθεί από τον χρήστη γίνονται έλεγχοι και αν δεν προκύψει κάποιο πρόβλημα όπως αυτό που συζητήσαμε παραπάνω, η φόρμα υποβάλλεται στο URL που ορίζει το γνώρισμα **action** του στοιχείου <form>, με τη μέθοδο που ορίζει το γνώρισμα **method**. Αν για παράδειγμα έχουμε:

```
<form action="/form-handler.html" method="post">
```

τότε η φόρμα θα υποβληθεί στο σχετικό URL /form-handler.html με τη μέθοδο POST.

3.5.4 Τύποι στοιχείων μιας φόρμας

Στην ενότητα αυτή θα κάνουμε μια σύντομη επισκόπηση των κυριότερων τύπων στοιχείων <input>, αλλά και άλλων στοιχείων. Στο παράδειγμα της προηγούμενης ενότητας είδαμε στοιχεία <input> τύπου **text**, **email**, και **submit**. Παρακάτω έχουμε μια πιο πλήρη εικόνα:

Πεδία εισαγωγής κειμένου από τον χρήστη

- <input type="text"> πεδίο εισαγωγής κειμένου
- <input type="email"> εισαγωγή email
- <input type="number"> εισαγωγή αριθμητικής τιμής
- <input type="password"> εισαγωγή κωδικού (****), οι χαρακτήρες δεν φαίνονται
- <input type="search"> πεδίο αναζήτησης
- <input type="url"> εισαγωγή διεύθυνσης URL
- <input type="tel"> εισαγωγή αριθμ. τηλεφώνου
- <textarea> ορίζει πεδίο εισαγωγής κειμένου πολλών γραμμών

Στοιχεία επιλογής από λίστα ή γραφική διεπαφή

- <select> ορίζει μια λίστα από drop-down επιλογές οι οποίες περιέχοντα σε στοιχεία <option> ή σε ομάδα επιλογών <optgroup>
- <datalist> ορίζει μια προκαθορισμένη λίστα από στοιχεία για ένα στοιχείο <input>
- <input type="checkbox"> επιλογή ναι/όχι
- <input type="radio"> επιλογή μιας από εναλλακτικές περιπτώσεις
- <input type="range"> επιλογή με κύλιση σε ράβδο τιμών
- <input type="date"> εισαγωγή/επιλογή ημερομηνίας από ημερολόγιο
- <input type="datetime-local"> εισαγωγή/επιλογή τοπικής ημερομηνίας από ημερολόγιο
- <input type="month"> εισαγωγή/επιλογή μήνα από ημερολόγιο
- <input type="time"> εισαγωγή/επιλογή χρονικής στιγμής
- <input type="week"> εισαγωγή/επιλογή εβδομάδας
- <input type="color"> επιλογή χρώματος από παλέτα
- <input type="file"> επιλογή αρχείου από το σύστημα αρχείων

Πλήκτρα

- <input type="button"> γενικού σκοπού πλήκτρο
- <input type="image"> εικόνα ως πλήκτρο υποβολής
- <input type="reset"> πλήκτρο επαναφοράς της φόρμας στην αρχική κατάσταση

- `<input type="submit">` πλήκτρο υποβολής της φόρμας
- `<button>` ορίζει ένα πλήκτρο, εναλλακτικά του στοιχείου `<input type="button">`.

Για να συμπληρωθεί η εικόνα των στοιχείων που μπορεί να βρούμε σε μια φόρμα εκτός των παραπάνω, στη φόρμα μπορεί να υπάρχουν τα εξής στοιχεία:

Άλλα στοιχεία

- `<input type="hidden">` κρυφό πεδίο για αποθήκευση και υποβολή δεδομένων
- `<label>` επεξηγηματικό κείμενο για κάποιο στοιχείο
- `<fieldset>` και `<legend>` προσδιορίζουν μια ομάδα στοιχείων
- `<output>` στοιχείο που περιέχει το αποτέλεσμα υπολογισμού

Όπως βλέπουμε η HTML5 διαθέτει ένα πλούσιο σύνολο από στοιχεία φόρμας, πολλά από τα οποία, (όπως για παράδειγμα το `<input type="date">`), βοηθούν τον χρήστη στην εισαγωγή δεδομένων, ενώ άλλα (όπως το `<input type="email">`) κάνουν έλεγχο της εισόδου του χρήστη σύμφωνα με κάποιους βασικούς κανόνες.

Στην επόμενη ενότητα θα δούμε κάποια από τα γνωρίσματα που μπορεί να πάρουν τα στοιχεία αυτά.

3.5.5 Κύρια γνωρίσματα των στοιχείων φόρμας

Ήδη έχουμε δει κάποια βασικά **γνωρίσματα** των στοιχείων της φόρμας, όπως το `name` που ορίζει την μεταβλητή στην οποία θα εκχωρηθεί η τιμή του χρήστη. Επίσης είδαμε το γνώρισμα `value` στο πλήκτρο υποβολής. Στη συνέχεια κάνουμε μια σύντομη αναφορά στα κυριότερα γνωρίσματα.

- `name` Η μεταβλητή που παίρνει την τιμή του στοιχείου στην JavaScript ή στον εξυπηρετητή
- `value` Η αρχική τιμή του στοιχείου ή για τα πλήκτρα, το κείμενό τους
- `autocomplete` "on"/"off" επιτρέπει/απαγορεύει στον φυλλομετρητή να βοηθήσει με την συμπλήρωση της εισόδου του χρήστη
- `autofocus` Υποδηλώνει το στοιχείο που αποκτάει την εστίαση όταν φορτωθεί η φόρμα
- `max, min` Μέγιστη και ελάχιστη επιτρεπόμενη τιμή αριθμητικού στοιχείου
- `maxlength, minlength` Μέγιστος και ελάχιστος αριθμός χαρακτήρων που επιτρέπονται στα δεδομένα που εισάγονται σε στοιχείο `<input>`
- `required` Υποχρεωτικό πεδίο που πρέπει να συμπληρωθεί πριν υποβληθεί η φόρμα
- `placeholder` Σύντομη περιγραφή της τιμής που αναμένεται να εισάγει ο χρήστης, εντός του πεδίου.
- `size` Ορατό μήκος σε αριθμό χαρακτήρων του στοιχείου `<input>`
- `disabled` Ανενεργό στοιχείο, δεν αποστέλλεται όταν γίνεται υποβολή της φόρμας
- `readonly` Μη τροποποιήσιμο πεδίο, πεδίο μόνο για ανάγνωση.

Ασκήσεις αυτοαξιολόγησης

Άσκηση 1. Το γνώρισμα `value` το είδαμε ήδη ως γνώρισμα του πλήκτρου υποβολής, όπου στην περίπτωση αυτή ορίζει το κείμενο που εμφανίζεται στο πλήκτρο. Όμως θα πρέπει να προσέξουμε ότι η σημασιολογία του γνωρίσματος είναι αρκετά διαφορετική για στοιχεία `input`, όπου το γνώρισμα προσδιορίζει την αρχική τιμή του πεδίου όταν φορτωθεί η φόρμα.

Ποια η εμφάνιση της φόρμας όταν φορτωθεί και τι θα γίνει αν ο χρήστης εισάγει την τιμή 15.

```
<label for="age">Ηλικία:</label>
<input type="number" id="age" name="age" min=18, max=99, value=20><br>
```

Απάντηση: Στην περίπτωση αυτή όταν φορτωθεί η φόρμα στο πεδίο της ηλικίας το πεδίο θα έχει την τιμή 20. Τα γνωρίσματα `min` και `max` ορίζουν την ελάχιστη και μέγιστη επιτρεπόμενη τιμή, συνεπώς αν ο χρήστης δώσει την τιμή 15, η τιμή δεν θα επιτραπεί και θα πάρει ένα μήνυμα λάθους.

Άσκηση 2. Το γνώρισμα `placeholder` ορίζει την υπόδειξη προς τον χρήστη που θα εμφανιστεί στο πεδίο. Έστω το παρακάτω στοιχείο της φόρμας:

```
<label for="fname">Όνομα:</label><br>
<input type="text" size="10" id="fname" name="fname" placeholder='γράψτε
το όνομά σας '><br>
```

Περιγράψτε τι θα δει ο χρήστης όταν φορτωθεί η φόρμα.

Απάντηση: Ο χρήστης θα δει το πεδίο Όνομα, Το κείμενο που θα εμφανιστεί εντός του πεδίου είναι αυτό που ορίζει το γνώρισμα `placeholder`, συνεπώς το κείμενο-υπόδειξη: γράψτε το όνομά σας . Όμως έχει οριστεί ότι το μήκος του πεδίου είναι 10 χαρακτήρες (γνώρισμα `size`). Συνεπώς ο χρήστης θα δει μόνο τους 10 πρώτους χαρακτήρες της υπόδειξης: γράψτε το ό.

3.5.6 Στοιχεία επιλογής

Ως τώρα έχουμε δει στοιχεία εισαγωγής δεδομένων από τον χρήστη, όπως τα στοιχεία τύπου `text`, `password`, `email`, `number`, κλπ.

Όμως όταν η τιμή που περιμένουμε από τον χρήστη ανήκει σε ένα κλειστό σύνολο, είναι προτιμότερο να ζητήσουμε από τον χρήστη να επιλέξει μια από τις τιμές, αντί για να την εισάγει.

Ας δούμε τους διαφορετικούς τρόπους που έχουμε για να δώσουμε αυτή τη δυνατότητα στον χρήστη.

Λίστα επιλογών `select/option`

Ο πιο κλασικός τρόπος επιλογής είναι από αναδυόμενη λίστα επιλογών. Ας υποθέσουμε ότι ένας ορειβατικός σύλλογος ζητάει από τα μέλη του να επιλέξουν τον προορισμό της επόμενης εξόρμησης, ανάμεσα σε 4 υποψήφιους προορισμούς.

Η εμφάνιση της ιστοσελίδας είναι:

Διαλέξτε προορισμό:

Άγραφα

Τζουμέρκα

Πήλιο

Ζαγόρια

Submit

{#fig:fig5 width=50%}

Ακολουθεί ο σχετικός κώδικας:

```
<form action="/action_page.html">
  <label for="places"> Διαλέξτε προορισμό:</label>
  <select id="places" name="place">
    <option value="agrafa" selected> Άγραφα</option>
    <option value="tzoumerka"> Τζουμέρκα</option>
    <option value="pilio"> Πήλιο</option>
    <option value="zagoria"> Ζαγόρια</option>
  </select><br>
  <input type="submit">
</form>
```

Παρατηρούμε ότι το στοιχείο που διαμορφώνει το πεδίο επιλογής είναι το στοιχείο `<select>`. Το όνομα του πεδίου είναι `name=place`.

Εντός του στοιχείου `<select>`, ορίζουμε τις επιλογές ως στοιχεία `<option>`. Καθένα από αυτά τα στοιχεία έχει ως περιεχόμενο το κείμενο που θα δει ο χρήστης. Το γνώρισμα `value` της αντίστοιχης επιλογής είναι η τιμή που θα πάρει το στοιχείο αν την επιλέξει ο χρήστης. Έτσι αν ο χρήστης επιλέξει το `Πήλιο`, η μεταβλητή θα λάβει την τιμή `place = pilio`.

Επίσης θα πρέπει να προσέξουμε ότι η επιλογή `Άγραφα` είναι προεπιλεγμένη όταν φορτώνεται η φόρμα, και αν ο χρήστης δεν κάνει καμιά ενέργεια επιλογής κατά την υποβολή της φόρμας θα είναι η τιμή που θα επιστραφεί. Συνήθως θέτουμε ως προεπιλεγμένη τιμή εκείνη που είναι πιο πιθανόν να επιλέξουν οι περισσότεροι χρήστες.

Κύλιση σε πεδίο τιμών `range`

Ένας εναλλακτικός τρόπος να επιλέξει ο χρήστης μια τιμή είναι μέσω μιας ράβδου κύλισης. Το στοιχείο φόρμας που επιτρέπει κάτι τέτοιο είναι το `<input type="range">`. Ας δούμε ένα παράδειγμα.

Έστω ότι ζητάμε από τον χρήστη να επιλέξει την τιμή της έντασης του ήχου μεταξύ 0 και 100 db, με κύλιση επί της ράβδου.

Ένταση ήχου [0..100db]

{#fig:fig6 width=50%}

Ο παρακάτω κώδικας υλοποιεί αυτό το στοιχείο.

```
<form>
    <label for="volume">Ένταση ήχου [0..100db]</label>
    <input type="range" id="volume" name="vol" min="0" max="100"
value="80">
</form>
```

Θα πρέπει να σημειωθεί ότι η αρχική τιμή και αντίστοιχα η θέση της ράβδου κύλισης είναι 80 (γνώρισμα **value**).

Περισσότερες από μια επιλογές **checkbox**

Υπάρχουν περιπτώσεις που θα επιθυμούσαμε στον χρήστη να επιλέξει περισσότερες από μια επιλογές. Αυτό μπορεί να γίνει με χρήση πολλαπλών στοιχείων **<input type="checkbox">** τα οποία επιτρέπουν το καθένα να επιλεγεί ή όχι.

Ας δούμε ένα παράδειγμα. Έστω σε μια ιστοσελίδα που ζητάει από τον χρήστη τα ενδιαφέροντα του και του επιτρέπει να επιλέξει κάποια από τα παρακάτω τρία ενδιαφέροντα:

Μού αρέσουν....

Η ζωή στη φύση

Το διάβασμα

Ο αθλητισμός

υποβολή

{#fig:fig7 width=50%}

Ο χρήστης μπορεί να επιλέξει κανένα, ένα δύο ή τρία από τα ενδιαφέροντα αυτά. Ο παρακάτω κώδικας υλοποιεί αυτή τη φόρμα:

```
<form>
<div id ='likes'>Μού αρέσουν.... </div>
  <input type="checkbox" id="item1" name="item1" value="nature">
  <label for="item1">Η ζωή στη φύση </label><br>
  <input type="checkbox" id="item2" name="item2" value="reading">
  <label for="item2"> Το διάβασμα </label><br>
  <input type="checkbox" id="item3" name="item3" value="sports">
  <label for="item3">Ο αθλητισμός </label><br>
  <input type="submit" value='υποβολή'>
</form>
```

Το καθένα από τα στοιχεία τύπου **checkbox** έχει γνωρίσματα όνομα (**name**) και τιμή (**value**). Αν ο χρήστης επιλέξει το συγκεκριμένο στοιχείο τότε η μεταβλητή που αντιστοιχεί στο όνομα του λαμβάνει την τιμή που ορίζει το γνώρισμα **value**.

Για παράδειγμα στην παραπάνω εικόνα έχουν επιλεγεί τα δύο πρώτα ενδιαφέροντα. Όταν υποβάλουμε τη φόρμα με αυτή την επιλογή το URL θα περιέχει το εξής query string: **?item1=nature&item2=reading**.

Αμοιβαία αποκλειόμενες επιλογές **radio**

Μια ακόμη περίπτωση στοιχείου φόρμας, είναι το **radio** που περιλαμβάνει 2 ή περισσότερες αμοιβαία αποκλειόμενες επιλογές. Το όνομα του στοιχείου αυτού προέρχεται από τα παλιά ραδιόφωνα που είχαν πλήκτρα επιλογής μπάντας συχνότητας, όπως στην εικόνα @fig:fig8.



{#fig:fig8 width=50%}

Άς δούμε ένα παράδειγμα. Έστω φόρμα που ζητάει από τον χρήστη να δηλώσει το φύλο του και επιτρέπει αυτές τις τρεις επιλογές.

- άνδρας
- γυναίκα
- άλλο

{#fig:fig9 width=50%}

Θα χρησιμοποιήσουμε τα στοιχεία τύπου **radio** τα οποία θα έχουν κοινό όνομα της μεταβλητής (γνώρισμα **name**). Μια ομάδα τέτοιων γνωρισμάτων η οποία τυπικά θα είναι κοντά το ένα με το άλλο θα επιτρέπουν στον χρήστη να επιλέξει ένα από αυτά.

Ο κώδικας που υλοποιεί αυτή τη φόρμα είναι:

```
<form>
  <input type="radio" name="gender" value="male" checked> άνδρας<br>
  <input type="radio" name="gender" value="female"> γυναίκα<br>
  <input type="radio" name="gender" value="other"> άλλο<br>
</form>
```

Όπως βλέπουμε σε αυτό το παράδειγμα, έχει προεπιλεγεί (λογικό γνώρισμα **checked**) το πρώτο από τα στοιχεία.

Τα στοιχεία συνδέονται μέσω του γνωρίσματος **name="gender"** το οποίο είναι κοινό και για τα τρία στοιχεία τύπου **radio**.

3.5.7 Επιλογή μέσω γραφικής διεπαφής

Υπάρχει μια ομάδα από στοιχεία φόρμας, τα οποία εισήχθησαν στην HTML5, και τα οποία βοηθάνε τον χρήστη να εισάγει τιμή μέσω γραφικού παράθυρου χρησιμοποιώντας το εγγενές γραφικό σύστημα του υπολογιστή. Επιλογή στοιχείων όπως ημερομηνίας από ένα ημερολόγιο, χρώματος από μια παλέτα χρωμάτων, αρχείο από μια διεπαφή του συστήματος αρχείων, κλπ. Τα στοιχεία αυτά είναι εξαιρετικά χρήσιμα και διευκολύνουν τον χρήστη να εισάγει τα αντίστοιχα δεδομένα, όπως μια ημερομηνία, με σωστή μορφοποίηση, χωρίς σφάλματα πληκτρολόγησης. Το πρόβλημα με τα στοιχεία αυτά, είναι ότι δεν υποστηρίζονται ακόμη από όλους τους φυλλομετρητές. Θα πρέπει να ελέγξουμε αν ο φυλλομετρητής υποστηρίζει το στοιχείο με κατάληλο κώδικα JavaScript, και να δώσουμε εναλλακτικές επιλογές, πχ να δημιουργήσουμε ένα στοιχείο select/option με τους μήνες του χρόνου, ή τις ημερομηνίες στην περίπτωση του στοιχείου τύπου **date**.

Ας δούμε ένα παράδειγμα, μέσω του οποίου θα εισάγουμε και ένα άλλο σημαντικό στοιχείο που είναι το **<fieldset>/<legend>** που χρησιμεύει για την ομαδοποίηση των στοιχείων.

Το παράδειγμα αφορά την εισαγωγή της ημερομηνίας και ώρας γέννησης ενός βρέφους.

Η ιστοσελίδα έχει την εξής εμφάνιση:

{#fig:fig10 width=50%}

Ας δούμε τον κώδικα της αντίστοιχης φόρμας

```
<fieldset>
  <legend>Στοιχεία γέννησης</legend>
  <label for="date">Ημερομηνία γέννησης</label>
  <input type="date" id="date" name="date-birth"><br>
  <label for="time">Ώρα γέννησης</label>
  <input type="time" id="time" name="time-birth">
</fieldset>
```

Το πρώτο στοιχείο που χρησιμοποιούμε σε αυτό το παράδειγμα είναι το στοιχείο **<fieldset>** το οποίο περικλείει μια ομάδα από συγγενή στοιχεία. Επίσης το στοιχείο αυτό συνοδεύεται από το στοιχείο **<legend>** το οποίο μάς επιτρέπει να δώσουμε μια επικεφαλίδα στην ομάδα. Ένα σχετικό σχόλιο είναι ότι η ομαδοποίηση των στοιχείων της φόρμας βοηθάει τον χρήστη να κατανοήσει καλύτερα τι του ζητάμε και είναι πολύ καλή πρακτική να χρησιμοποιήσουμε το στοιχείο ομαδοποίησης.

Παρατηρούμε ότι η φόρμα περιέχει δύο στοιχεία **<input>** τύπου **date** και **time**.

Παρατηρούμε ότι για τα στοιχεία αυτά υπάρχει η δυνατότητα είτε να εισάγει ο χρήστης μια ημερομηνία ή ώρα, πληκτρολογώντας την, είτε να επιλέξει το σχετικό εικονίδιο και να αλληλεπιδράσει από το σχετικό γραφικό στοιχείο του φυλλομετρητή.

Στην εικόνα @fig:fig11 φαίνεται η αλληλεπίδραση με το πρώτο στοιχείο στον φυλλομετρητή Chrome. Η εμφάνιση του στοιχείου επιλογής ημερομηνίας μπορεί να διαφέρει σε διάφορους φυλλομετρητές.

Στοιχεία γέννησης

Ημερομηνία γέννησης

Ωρα γέννησης

Υποβολή

March 2022 ▾ ↑ ↓

M	T	W	T	F	S	S
28	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3
4	5	6	7	8	9	10

Clear **Today**

{#fig:fig11 width=50%}

Κάτι που θα πρέπει να προσέξουμε είναι ότι ανεξάρτητα από τη εμφάνιση του στοιχείου στην διεπαφή χρήστη, η τιμή που θα αποθηκευτεί είναι πάντα στη μορφή **yyyy-mm-dd**. Έτσι για παράδειγμα αν ο χρήστης εισάγει 09/03/2022, θα αποθηκευτεί η τιμή **date-birth=2022-03-09**.

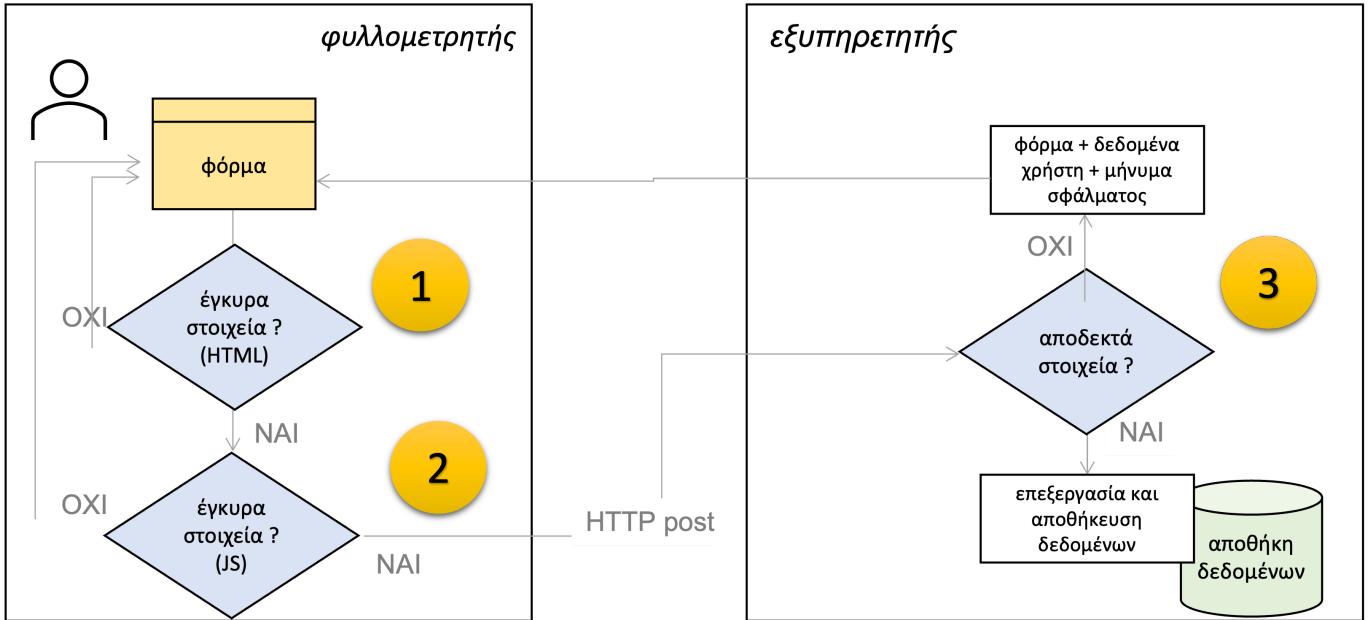
Επίσης θα πρέπει να σημειωθεί ότι φυλλομετρητές που δεν υποστηρίζουν αυτό το στοιχείο το αντιλαμβάνονται ως απλό στοιχείο **<input type="text">**. Βεβαίως σε τέτοια περίπτωση δεν γίνεται κανένας έλεγχος για τη συμβολοσειρά που εισάγει ο χρήστης, αν είναι αποδεκτή ημερομηνία και σύμφωνη με την προδιαγραφή μορφοποίησής της.

Αντίστοιχη είναι η συμπεριφορά του δεύτερου στοιχείου **<input>**. Να σημειωθεί και εδώ ότι η ώρα αποθηκεύεται ως συμβολοσειρά σε 24ωρο ρολόι, ανεξάρτητα από τη διεπαφή χρήστη. Για παράδειγμα αν ο χρήστης δώσει την τιμή **08:36 PM**, αυτή θα αποθηκευτεί ως συμβολοσειρά **20:36**, οπότε το σχετικό query string για το στοιχείο αυτό θα είναι: **time-birth=20%3A46**.

3.5.8 Έλεγχος εγκυρότητας της φόρμας

Ένα ιδιαίτερα σημαντικό θέμα είναι πώς θα γίνει ο έλεγχος εγκυρότητας μιας φόρμας. Αν ζητήσουμε από τον χρήστη την ηλικία του και απαντήσει 999, είναι μεν συντακτικά ορθό (number), όμως εννοιολογικά δεν

έχει νόημα. Υπάρχουν πολλαπλά επίπεδα ελέγχου εγκυρότητας της φόρμας, όπως φαίνεται στο σχήμα @fig:fig12.



{#fig:fig12 width=70%}

Οι φάσεις ελέγχου εγκυρότητας είναι:

1. έλεγχοι που γίνονται στον φυλλομετρητή από τα ίδια τα στοιχεία της φόρμας (HTML)
2. έλεγχοι που γίνονται στον φυλλομετρητή από κώδικα JavaScript
3. έλεγχοι που γίνονται στον εξυπηρετητή στη γλώσσα που είναι γραμμένος ο εξυπηρετητής της ιστοσελίδας

Βεβαίως μπορεί να γίνει συνδυασμός των παραπάνω.

Η πρώτη φάση θα πρέπει να μην θεωρηθεί δεδομένη, αφού όπως αναφέρθηκε τα νέα στοιχεία HTML, τα οποία περιλαμβάνουν έλεγχο εγκυρότητας δεν υποστηρίζονται από όλους τους φυλλομετρητές. Επίσης να σημειωθεί ότι τα στοιχεία αυτά δεν περιλαμβάνουν τη γνώση της εφαρμογής αλλά γενικούς κανόνες. Όμως παραμένει ένα πρώτο επίπεδο ελέγχου, το οποίο θα πρέπει να υποστηρίζεται, όπως για παράδειγμα η ύπαρξη του γνωρίσματος **required** σε ένα στοιχείο, επιβάλει την εισαγωγή μιας τιμής πριν την υποβολή της φόρμας.

Υπάρχει η δυνατότητα επιβολής πιο συγκεκριμένων κανόνων εγκυρότητας, μέσω του ορισμού ενός προτύπου (pattern) στο γνώρισμα **pattern** ενός στοιχείου <input>.

```
<input type=" " pattern ="RegEx" >
```

Μπορούμε να ορίσουμε ένα πρότυπο της συμβολοσειράς που θα εισάγει ο χρήστης μέσω του γνωρίσματος **pattern**, το οποίο δέχεται μια κανονική έκφραση (regular expression).

Ακολουθεί ένα παράδειγμα για ένα πρότυπο που αφορά τους Ελληνικούς ταχυδρομικούς κώδικες υποθέτοντας ότι οι ελληνικοί ταχυδρομικοί κώδικες είναι της μορφής XXX XX ή XXXXX όπου X αριθμητικό ψηφίο

```
<label for="postalCode">Ταχυδρομικός κώδικας:</label><br>
<input type="text" id="postalCode" name="PCODE"
       pattern="^[\d]{3} [\d]{2}$"><br><br>
```

Μια δεύτερη φάση ελέγχου εγκυρότητας είναι αυτή να γίνει στον φυλλομετρητή με χρήση κώδικα JavaScript. Συνήθως ορίζουμε μια συνάρτηση, η οποία καλείται όταν ο χρήστης πατήσει το πλήκτρο υποβολής της φόρμας. Η συνάρτηση η οποία καλείται πριν την υποβολή, μέσω του χειριστή συμβάντος **onsubmit** που ορίζεται στο στοιχείο **<form>**. Επίσης μπορεί να υπάρχουν επιμέρους συναρτήσεις που ελέγχουν την τιμή που εισάγεται σε επί μέρους πεδία.

```
<form id="myForm" onsubmit="return(validate());">
```

Η συνάρτηση **validate()** πρέπει να επιστρέψει **true** αν ο έλεγχος εγκυρότητας είναι επιτυχής και **false** αν δεν είναι επιτυχής.

Ένα παράδειγμα συνάρτησης **validate()** για μια φόρμα που ζητάει την ηλικία του χρήστη με τον περιορισμό ότι η είσοδος δεν επιτρέπεται για χρήστες κάτω των 18 ετών, είναι ο εξής:

```
function validate(){
    if (document.querySelector("#age").value<18){
        document.querySelector("#message").innerHTML=
        "Δεν είναι επιτρεπτή η είσοδος σε άτομο ηλικίας μικρότερης των 18
ετών";
        // διαμόρφωση του μηνύματος σφάλματος
        document.querySelector("#message").style.color="#ef0000";
        document.querySelector("#message").autofocus = "on";
        document.querySelector("#myForm").reset();
        document.querySelector("#age").style.backgroundColor="white";
        return false;
    }
    return true;
}
```

Να σημειωθεί εδώ ότι ο κώδικας αυτός θα γίνει περισσότερο κατανοητός όταν σε επόμενα κεφάλαια γίνει εισαγωγή στη γλώσσα προγραμματισμού JavaScript.

Το τελευταίο βήμα στον έλεγχο εγκυρότητας γίνεται στον εξυπηρετητή. Θα πρέπει να σημειωθεί ότι ο έλεγχος αυτός είναι απαραίτητος, γιατί υπάρχει πάντα το ενδεχόμενο κάποιος να προσπέρασε τους ελέγχους στην πλευρά του φυλλομετρητή και να έστειλε στον εξυπηρετητή μη-έγκυρα δεδομένα, με κίνδυνο στην ασφάλεια του συστήματος. Επίσης στο επιχείρημα γιατί δεν αρκούμεθα στον έλεγχο αυτής της φάσης και απαιτούμε να γίνουν έλεγχοι και στον φυλλομετρητή, θα πρέπει να σημειώσουμε ότι είναι καλό η ανάδραση στον χρήστη να παρέχεται άμεσα και όχι αφού σταλούν τα δεδομένα στον εξυπηρετητή, όπως περιγράφεται και στις οδηγίες καλής σχεδίασης φορμών της επόμενης ενότητας.

3.5.9 Οδηγίες σχεδίασης φόρμας

Αντικείμενο της ενότητας αυτής είναι να γίνει μια σύντομη αναφορά στις βασικές οδηγίες που πρέπει να εφαρμόζουμε για την καλή σχεδίαση φορμών. Δεδομένης της σημασίας των φορμών στις εφαρμογές διαδικτύου, οι οδηγίες αυτές θα πρέπει να αποτελούν οδηγό μας όταν σχεδιάζουμε φόρμες.

Μια σύντομη λίστα από κανόνες είναι η εξής:

1. Η φόρμα να έχει πάντα κατατοπιστικό τίτλο
2. Οι υποδείξεις να είναι λιτές και σαφείς
3. Να είναι σαφές στον χρήστη ποια πεδία είναι υποχρεωτικά και ποια προαιρετικά
4. Τα μηνύματα σφάλματος όταν προκύψουν να είναι υποστηρικτικά
5. Να γίνεται λογική ομαδοποίηση και διάταξη πεδίων (βλέπε στοιχείο <fieldset>)
6. Η διάταξη περιεχομένου να είναι ελκυστική (χρώματα, αποστάσεις, κλπ.)
7. Η πικνότητα πληροφορίας στη φόρμα να είναι ομοιόμορφη
8. Να παρέχεται εύκολη πρόσβαση σε πληροφορίες για νέους χρήστες
9. Να υπάρχει ομοιομορφία σε υποδείξεις, μηνύματα σφάλματος και ορολογία
10. Να παρέχεται ένδειξη για τα όρια, αν υπάρχουν, στο μέγεθος της πληροφορίας που θα εισαχθεί σε ένα πεδίο
11. Ο έλεγχος ορθότητας τιμών να γίνεται έγκαιρα, κατά προτίμηση πριν την υποβολή της συνολικής φόρμας, αλλά κατά την εισαγωγή της μη έγκυρης τιμής
12. Να παρέχονται συγκεκριμένες οδηγίες για την υποβολή της φόρμας.

Θα ολοκληρώσουμε τις οδηγίες για καλή σχεδίαση φορμών, με επιλογή 20 κανόνων για εύχρηστες φόρμες από τη [σελίδα του D. Travis](#) για την **ευχρηστία ιστοσελίδων**, ενότητα *Forms and data entry usability guidelines*. Όπως θα δούμε κάποιες από τις οδηγίες ευχρηστίας, επαναλαμβάνουν κανόνες καλής σχεδίασης που αναφέρθηκαν πιο πάνω, ενώ παρέχονται κανόνες που σχετίζονται με την ίδια τη διαδικασία συμπλήρωσης της φόρμας και πιθανά προβλήματα που ο χρήστης μπορεί να συναντήσει.

1. Τα πεδία της φόρμας να περιέχουν προεπιλεγμένες τιμές όπου είναι δυνατόν.
2. Όταν μια διαδικασία περιλαμβάνει έντυπα, η φόρμα πρέπει να είναι συμβατή με τη δομή τους.
3. Θα πρέπει να προσφέρεται αυτόματη μορφοποίηση πχ. δύο πεδία με "," μεταξύ τους για ακέραιο και δεκαδικό μέρος τιμής.
4. Οι υποδείξεις πρέπει να εξηγούν με σαφήνεια τι αναμένεται να εισαχθεί.
5. Τα πεδία κειμένου πρέπει να έχουν κατάλληλο μήκος για την αναμενόμενη απάντηση.
6. Να υπάρχει σαφής διάκριση μεταξύ "υποχρεωτικών" και "προαιρετικών" πεδίων.
7. Θα πρέπει να υπάρχει έγκαιρη προειδοποίηση εάν απαιτούνται εξωτερικές πληροφορίες για την ολοκλήρωση (π.χ. αριθμός διαβατηρίου), πριν ο χρήστης αρχίσει τη διαδικασία.
8. Τα πεδία στις φόρμες πρέπει να ομαδοποιούνται λογικά και κάθε ομάδα έχει μια επικεφαλίδα.
9. Τα πεδία στις φόρμες θα πρέπει να περιέχουν υποδείξεις, παραδείγματα ή τυπικές απαντήσεις που υποδεικνύουν την αναμενόμενη εισαγωγή.
10. Όταν οι υποδείξεις έχουν τη μορφή ερωτήσεων, οι ερωτήσεις αναφέρονται σε σαφή, απλή γλώσσα.
11. Τα αναπτυσσόμενα μενού, τα κουμπιά επιλογής και οι επιλογές ναι/όχι προτιμώνται από πεδία εισαγωγής κειμένου γιατί ελαχιστοποιούν το γνωστικό φορτίο του χρήστη.
12. Ο δρομέας τοποθετείται εκεί όπου απαιτείται η είσοδος (γνώρισμα **autofocus**).
13. Οι αναμενόμενες μορφές δεδομένων υποδεικνύονται σαφώς (π.χ. ημερομηνίες) και ορίζονται οι μονάδες τιμών.
14. Οι χρήστες να είναι σε θέση να ολοκληρώσουν απλές εργασίες εισάγοντας μόνο τις βασικές πληροφορίες (όχι άχρηστα υποχρεωτικά πεδία).

15. Η φόρμα να επιτρέπει στους χρήστες να παραμένουν στην ίδια μέθοδο αλληλεπίδρασης για όσο το δυνατόν περισσότερο (να μη χρειάζεται να κάνουν συχνά μετάβαση από πληκτρολόγιο σε ποντίκι και πάλι σε πληκτρολόγιο).
16. Τα πεδία εισαγωγής κειμένου να υποδεικνύουν το ποσό και τη μορφή των δεδομένων που πρέπει να εισαχθούν.
17. Να γίνεται έλεγχος των στοιχείων πριν από την υποβολή της φόρμας, και να γίνεται έλεγχος σε επίπεδο πεδίου και σε επίπεδο φόρμας την κατάλληλη στιγμή.
18. Να παρέχεται διευκόλυνση στη διόρθωση σφαλμάτων (π.χ. όταν μια φόρμα δεν είναι πλήρης, τοποθετώντας τον δρομέα στην τοποθεσία όπου απαιτείται διόρθωση).
19. Να υπάρχει συνέπεια μεταξύ της εισαγωγής και της εμφάνισης δεδομένων.
20. Οι υποδείξεις βρίσκονται κοντά στα πεδία εισαγωγής δεδομένων (π.χ. είναι στοιχισμένες δεξιά).

3.6 Ασκήσεις

Ασκηση 1

Να δημιουργήσετε μια σελίδα στην οποία θα περιέχονται πληροφορίες για σας και την περιοχή που ζείτε. Να ενσωματώσετε φωτογραφία και ένα χάρτη της περιοχής σας. Να χρησιμοποιήσετε σημασιολογικά δομικά στοιχεία.

Ασκηση 2

Να δημιουργήσετε μια ιστοσελίδα με πληροφορίες για τον πλανήτη Αφροδίτη. Να περιέχει εικόνες και βίντεο καθώς και πληροφορίες για τον πλανήτη σε μορφή κειμένου. Να χρησιμοποιήσετε σημασιολογικά δομικά στοιχεία.

Ασκηση 3

Να κατασκευάσετε μια ιστοσελίδα εισαγωγής στοιχείων για κράτηση δωματίου σε μια αλυσίδα ξενοδοχείων.

Η φόρμα να περιέχει τις εξής ομάδες στοιχείων

Προσωπικά στοιχεία όνομα, επώνυμο, διεύθυνση, τηλέφωνο, email

Επιλογή ξενοδοχείου από μια λίστα ξενοδοχείων να επιλέγεται ένα

Τύπος δωματίου να επιλέξτε ανάμεσα στους παρακάτω τύπους: μονόκλινο/δίκλινο/τρίκλινο/σουίτα

Ημερομηνίες κράτησης Να περιέχει ημερομηνία άφιξης και ημερομηνία αναχώρησης

Τρόπος πληρωμής να περιέχει μια από τις παρακάτω επιλογές:

- Κατάθεση σε τραπεζικό λογαριασμό
- Πιστωτική κάρτα

Η δεύτερη επιλογή να έχει ακόμη τις εξής επιλογές: Visa/Mastercard/American Express

Τέλος η φόρμα να έχει πλήκτρο υποβολής και καθαρίσματος της φόρμας.