

## 2. Εισαγωγή στη γλώσσα HTML

---

Το πρώτο από δύο κεφάλαια για την HTML ξεκινά με μια σύντομη εισαγωγή στη γλώσσα αυτή και στην ιστορία της καθώς και με αναφορές στο πρότυπο που την καθορίζει.

Η πρώτη επαφή με τη γλώσσα γίνεται μέσω μια απλής ιστοσελίδας που είναι και η αφετηρία για να εισαχθούν οι βασικές έννοιες: **στοιχεία (elements)**, **ετικέτες (tags)**, **γνωρίσματα (attributes)**, και δομή της σελίδας. Επίσης παρουσιάζονται βασικά στοιχεία της γλώσσας (`<head>`, `<body>`, `<p>`, `<h1>` ... `<h6>`, `<hr>`, `<br>`, `<em>`, `<li>`, `<ol>`, `<ul>`, `<a>`, κλπ.). Επίσης γίνεται εισαγωγή στο Document Object Model (DOM), ενώ η τελευταία ενότητα παρουσιάζει το στοιχείο πίνακας (`<table>`).

### 2.1 Εισαγωγή

Όπως έχει ήδη αναφερθεί, τρεις ήταν οι βασικές τεχνολογίες που συνέθεσαν αρχικά την πρόταση δημιουργίας του παγκόσμιου ιστού. Αφενός το πρωτόκολλο HTTP επικοινωνίας πελάτη/εξυπηρετητή, και ο μηχανισμός παγκόσμιου εντοπισμού πόρων (Universal Resource Locator, URL), που είδαμε στο προηγούμενο κεφάλαιο, αφετέρου η γλώσσα **HTML (Hyper Text Markup Language)**, η γλώσσα δημιουργίας και σύνταξης περιεχομένου ιστοσελίδων. Η HTML θα μας απασχολήσει εκτενώς στη συνέχεια.

#### 2.1.1 Ιστορική αναφορά

Αρχικά η HTML ήταν μία γλώσσα που καθόριζε την εμφάνιση του περιεχομένου της ιστοσελίδας. Σήμερα η HTML5 (έκδοση 5) παίζει το ρόλο της βάσης μιας ιστοσελίδας, και καθορίζει το περιεχόμενό της και τη δομή της, ενώ συμπληρώνεται ουσιαστικά από δύο ακόμη τεχνολογίες, τις οποίες θα δούμε σε επόμενα κεφάλαια, την CSS3, γλώσσα που ορίζει μέσω κανόνων την παρουσίαση του περιεχομένου, και τη γλώσσα JavaScript, μία γλώσσα προγραμματισμού γενικού σκοπού, που επιτρέπει σε μια ιστοσελίδα να παρουσιάζει δυναμική συμπεριφορά, δηλαδή να αλλάζει μετά από ενέργειες του χρήστη.

Η HTML είναι μία γλώσσα *επισημείωσης (markup)* που καθορίζει τη δομή και το περιεχόμενο μιας ιστοσελίδας. Ορίζει για παράδειγμα τα κείμενα της σελίδας, αλλά και τη δομή τους, τις παραγράφους στις οποίες αυτά χωρίζονται, τις κεφαλίδες τους, τις εικόνες, βίντεο κλπ που αυτή περιέχει, κλπ. Η CSS έκδοση 3 είναι μία γλώσσα που ορίζει τον τρόπο παρουσίασης του περιεχομένου της ιστοσελίδας. Καθορίζει το στυλ παρουσίασης των διαφόρων στοιχείων, το χρώμα τους, το μέγεθος τους, τη γραμματοσειρά με την οποία παρουσιάζεται το κείμενο, κλπ. Τέλος η JavaScript ή JS είναι μία διερμηνευόμενη (interpreted) γλώσσα προγραμματισμού, η οποία εκτελείται μέσα στο περιβάλλον του browser και η εκτέλεσή της επηρεάζει την εμφάνιση της ιστοσελίδας. Επιτρέπει να ανταποκρίνεται η ιστοσελίδα σε ενέργειες του χρήστη, δηλαδή να μην είναι απλά μια στατική σελίδα αλλά να προσφέρει μια αλληλεπιδραστική εμπειρία.

Η HTML ορίστηκε αρχικά ως μία απλοποιημένη έκδοση της SGML, η οποία είναι ένα πρότυπο (ISO 8879:1986) για επισημείωση (markup) κειμένων. Η γλώσσα πέρασε μια σειρά από εκδόσεις, με τις πιο σημαντικές την έκδοση HTML 3.2 που έγινε πρότυπο από το W3C το 1997 και την έκδοση HTML4 που ακολούθησε την ίδια εποχή. Η τρέχουσα έκδοση 5 είναι η πρώτη που πια δεν συνδέεται με την SGML. Η ομάδα εργασίας Web Hypertext Application Technology Working Group (WHATWG) σε συνεργασία με το W3C World Wide Web Consortium καθορίζει σήμερα την προτυποποίηση της HTML, εξασφαλίζοντας ότι σε όλο το διαδίκτυο υλοποιείται η ίδια γλώσσα. Αξίζει να επισκεφθεί κανείς την ιστοσελίδα <https://html.spec.whatwg.org/> όπου συντηρείται η τρέχουσα έκδοση του προτύπου από την ομάδα εργασίας WHATWG η οποία σε συμφωνία με την W3C έχει αναλάβει να συντηρεί και να αναπτύσσει το πρότυπο της

γλώσσας HTML. Μια ιστορική σημείωση που θα πρέπει να γίνει εδώ είναι ότι η ομάδα εργασίας WHATWG δημιουργήθηκε το 2004 όταν σε συνάντηση της επιτροπής του W3C δεν ψηφίστηκε η πρόταση από τη Mozilla και Opera για ανάπτυξη της HTML5 προς την κατεύθυνση που τελικά πήρε, αλλά αντίθετα αποφασίστηκε να προχωρήσει το W3C στην ανάπτυξη της XHTML, δηλαδή της HTML σε κωδικοποίηση XML. Εκπρόσωποι μεγάλων εταιριών του διαδικτύου (Mozilla, Opera, Apple) προχώρησαν στην ίδρυση της WHATWG που έκτοτε αναπτύσσει το πρότυπο και μάλιστα από το 2019 με συμφωνία και της W3C αποτελεί το μοναδικό σημείο αναφοράς του προτύπου. Στην ομάδα εργασίας σήμερα συμμετέχουν και άλλες εταιρίες διαδικτύου, όπως η Google και η Microsoft.

Θα πρέπει να σημειωθεί ότι η HTML5 υπάρχει ως W3C Recommendation από το 2014, ενώ αρχική πρόταση εκδόθηκε το 2008. Χρησιμοποιείται ήδη αρκετά χρόνια. Παρόλα αυτά υπάρχουν ακόμη πολλές ιστοσελίδες και ιστότοποι που είναι υλοποιημένες σε προγενέστερες εκδόσεις της HTML, όπως στην HTML4, η οποία υπάρχει ήδη από το 1997, και την προγενέστερη ακόμα HTML3.2. Ακόμη υπάρχουν φυλλομετρητές (κύρια παλαιότερες εκδόσεις του Internet Explorer), που δεν υποστηρίζουν την HTML5. Αυτά θα πρέπει να τα λάβουμε υπόψη μας όταν αναπτύσσουμε εφαρμογές, όπως θα συζητηθεί στη συνέχεια.

Τα κύρια χαρακτηριστικά της HTML5 σε σύγκριση με προηγούμενες εκδόσεις της HTML είναι:

- Έχει απλοποιήσει συντακτικά τη γλώσσα, και έχει βελτιώσει τα δομικά στοιχεία της, όπως τα στοιχεία `<header>`, `<footer>`, `<nav>`, `<article>`, `<section>`, κλπ.
- Έχει ακόμη βελτιώσει τις δυνατότητες που έχουν οι φόρμες για υποστήριξη του χρήστη στην εισαγωγή πληροφορίας, έχουμε για παράδειγμα τα καινούργια στοιχεία `<input type='number'>`, ή τύπου `date`, `time`, `calendar`, κλπ.
- Έχει εισάγει δυνατότητες για δημιουργία γραφικών στοιχείων κάτι που χρησιμοποιείται ιδιαίτερα στα παιχνίδια και σε άλλες εφαρμογές όπως είναι το στοιχείο `<canvas>`.
- Έχει βελτιώσει και έχει προτυποποιήσει την παρουσίαση πολυμεσικού περιεχομένου με το στοιχείο `<audio>` και το στοιχείο `<video>`, που είχαν ως συνέπεια ουσιαστικά να εκλείψουν άλλες τεχνολογίες όπως ήταν η τεχνολογία flash που χρησιμοποιούνταν παλαιότερα
- Έχει εισάγει API, διεπαφές της γλώσσας, που της δίνουν ιδιαίτερα ισχυρά χαρακτηριστικά όπως παραδείγματος χάρι δυνατότητα γεωεντοπισμού, δυνατότητες για σύρε-άφησε κλπ.

Αυτά είναι τα κύρια σημεία που θα τα δούμε στη συνέχεια.

## 2.1.2 Η πρώτη ιστοσελίδα

Ας δούμε τώρα πως θα κατασκευάσουμε την πρώτη μας ιστοσελίδα. Έστω σε ένα περιβάλλον ανάπτυξης όπως το **VS Code** δημιουργούμε ένα καινούργιο αρχείο. Το οποίο ονομάζουμε `ex00.html`, στο οποίο θα αποθηκεύσουμε ένα έγγραφο HTML.

```
<!DOCTYPE html>
<html lang="el">
  <head>
    <meta charset="utf-8">
    <title>τίτλος</title>
  </head>
  <body>
    <h1>Επικεφαλίδα</h1>
    <p> Πρώτη παράγραφος.</p>
```

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

Το έγγραφο αυτό όταν το ανοίξουμε σε ένα φυλλομετρητή έχει την εξής εμφάνιση:

# Επικεφαλίδα

Πρώτη παράγραφος.

{#fig:fig1 width=50%}

Όπως παρατηρούμε ένα έγγραφο HTML περιλαμβάνει και το περιεχόμενο που εμφανίζεται στην ιστοσελίδα (το κείμενο "Επικεφαλίδα" και το κείμενο "Πρώτη παράγραφος.") αλλά και ετικέτες που οργανώνονται ιεραρχικά και ορίζουν τα **στοιχεία** του εγγράφου.

Η πρώτη εντολή `<!DOCTYPE html>` είναι η δήλωση του τύπου εγγράφου που ακολουθεί. Δεν είναι υποχρεωτική, όμως είναι καλή πρακτική να μπαίνει ως πρώτη εντολή ενός εγγράφου HTML5. Σε προηγούμενες εκδόσεις της HTML ήταν η εντολή που συνέδεε το έγγραφο με τους κανόνες που έλεγχαν τη συντακτική του εγκυρότητα, για παράδειγμα ενός αρχείου DTD (Document Type Definition).

Η ρίζα της ιεραρχίας από στοιχεία του εγγράφου HTML5 είναι το στοιχείο `<html>`. Η αρχική ετικέτα `<html>` συνοδεύεται από την τερματική ετικέτα `</html>` ορίζοντας το στοιχείο αυτό. Το έγγραφο περιλαμβάνει μία ιεραρχία από στοιχεία. Η ρίζα έχει δύο παιδιά, το στοιχείο `<head>` και το στοιχείο `<body>`. Το μεν `<body>` περιέχει το περιεχόμενο του εγγράφου που θα δει ο χρήστης, πριν από αυτό όμως ορίζουμε ένα στοιχείο `<head>`.

Μέσα στο στοιχείο `<head>` περιέχονται μετα-δεδομένα του εγγράφου. Πολλά μετα-δεδομένα ορίζονται από την ετικέτα `<meta>`. Παρατηρούμε ότι η ετικέτα αυτή δεν έχει αντίστοιχη ετικέτα τερματισμού, ορίζει όπως θα αναφερθεί στη συνέχεια ένα **κενό στοιχείο** (void element). Μάλιστα η συγκεκριμένη ετικέτα περιλαμβάνει ένα γνώρισμα με την τιμή του `charset="utf-8"`. Με αυτό τον τρόπο ορίζουμε ότι η κωδικοποίηση των χαρακτήρων του εγγράφου είναι utf-8, ώστε ο φυλλομετρητής να εμφανίσει σωστά χαρακτήρες εκτός του λατινικού αλφαβήτου, όπως είναι παραδείγματος χάρη τα ελληνικά. Ένα άλλο στοιχείο που περιέχεται στο `<head>` είναι το `<title>` που μάς επιτρέπει να ορίσουμε τον τίτλο που θα εμφανιστεί στο παράθυρο του φυλλομετρητή.

Ας δούμε στη συνέχεια το κυρίως σώμα της ιστοσελίδας, το οποίο περιλαμβάνει το περιεχόμενο μέσα σε νέα στοιχεία. Η συγκεκριμένη ιστοσελίδα περιλαμβάνει το στοιχείο `<h1>` που σημαίνει επικεφαλίδα (header) πρώτου επιπέδου, και στη συνέχεια το στοιχείο `<p>` το οποίο αφορά μια παράγραφο (paragraph) κειμένου. Το κείμενο που περιλαμβάνεται ανάμεσα στις ετικέτες `<h1>` και `</h1>` καθώς και αυτό ανάμεσα

στις ετικέτες `<p>` και `</p>` έχει διαφορετική εμφάνιση στον φυλλομετρητή. Η επικεφαλίδα επιπέδου 1 εμφανίζεται με πιο έντονα και μεγάλα γράμματα. Όμως η ακριβής εμφάνιση του περιεχομένου των στοιχείων αυτών, όπως το χρώμα των χαρακτήρων και η γραμματοσειρά, καθορίζεται σε αυτή την περίπτωση από τον φυλλομετρητή, αφού δεν έχουμε ορίσει με κατάλληλες εντολές την εμφάνιση τους. Αν επιθυμούμε να ορίσουμε την εμφάνιση, θα πρέπει να εισάγουμε εντολές της CSS, όπως θα δούμε στη συνέχεια.

Επίσης θα πρέπει να προσέξουμε ότι το περιεχόμενο του στοιχείου **h1** καθώς και του στοιχείου **p** ξεκινάνε από την αρχή της γραμμής, αυτό γιατί και τα δύο αυτά στοιχεία είναι **στοιχεία μπλοκ (block elements)**, καθένα από τα οποία ορίζει ένα ορθογώνιο χώρο (block) της σελίδας. Αντίθετα υπάρχουν άλλα στοιχεία, όπως το στοιχείο `<strong>`, τα οποία εισάγονται μέσα στην παράγραφο και επηρεάζουν μόνο τμήματα του ενός στοιχείου block, τα οποία λέγονται **στοιχεία γραμμής (inline elements)**. Για παράδειγμα, αν τροποποιήσουμε το στοιχείο `<p>` ως ακολούθως:

```
<p> <strong>Πρώτη</strong> παράγραφος.</p>
```

το αποτέλεσμα θα είναι το παρακάτω, (παρατηρούμε ότι η λέξη *Πρώτη* εμφανίζεται πιο έντονα):

# Επικεφαλίδα

Πρώτη παράγραφος.

{#fig:fig1b width=50%}

Υπάρχουν και άλλα στοιχεία γραμμής (inline elements), στοιχεία δηλαδή που εντάσσονται στο πλαίσιο μιας γραμμής και συνήθως αφορούν ένα μικρό τμήμα του περιεχομένου. Τέτοια είναι ακόμη το στοιχείο `<em>` (emphasis) και το `<a>` για anchor, δηλαδή υπερσυνδέσμους, και ούτω καθεξής. Θα πρέπει να προσέξουμε ότι μέσα σε ένα στοιχείο γραμμής δεν μπορούμε να εισάγουμε ένα στοιχείο block. Ο κανόνας συνεπώς είναι: Ένα block δεν χωράει μέσα σε ένα στοιχείο inline, ενώ αντίθετα ένα στοιχείο inline χωράει σε ένα block.

Επίσης μπορούμε να παρατηρήσουμε στην αρχική ετικέτα `<html>` ότι έχουμε εισάγει το γνώρισμα `lang` με τιμή "el". Το γνώρισμα αυτό δηλώνει τη γλώσσα του εγγράφου, και είναι πολύ χρήσιμο σε μηχανές αναζήτησης. Το γνώρισμα lang μπορεί να εισαχθεί σε διαφορετικά στοιχεία που περιέχουν κείμενο. Για παράδειγμα:

```
<p lang="fr">Ceci est un paragraphe.</p>
```

η παράγραφος αυτή δηλώνεται ότι είναι στη γαλλική γλώσσα.

Θα πρέπει να σημειωθεί ότι τυπικοί κώδικες γλωσσών είναι: "en" (english), "es", (spanish), "pt" (portuguese), "de" (german), "fr" (french), ενώ μια γενικότερη καταγραφή των κωδικών αυτών μπορεί κανείς να βρει στο language codes iso 639-1 [https://en.wikipedia.org/wiki/ISO\\_639-1](https://en.wikipedia.org/wiki/ISO_639-1).

### 2.1.3 Μορφοποίηση της πρώτης ιστοσελίδας

Έστω τώρα, ότι θέλουμε να καθορίσουμε εμείς σε κάποιο βαθμό την εμφάνιση του εγγράφου. Έστω ότι επιθυμούμε να εμφανίζονται τα γράμματα με κίτρινο χρώμα και το υπόβαθρο της σελίδας μπλε.

Αρκεί να προσθέσουμε στην περιοχή `<head>` του εγγράφου το εξής στοιχείο:

```
<style>
  body { background: navy; color: yellow;}
</style>
```

Η ετικέτα `<style>` επιτρέπει την εισαγωγή εντολών CSS που σχετίζονται με τα στοιχεία του εγγράφου, εδώ για το στοιχείο `<body>` (δηλαδή για όλο το περιεχόμενο του εγγράφου), ορίζουμε ότι το υπόβαθρό του να είναι χρώματος `navy` ενώ το χρώμα του κειμένου να είναι `yellow`.

Το αποτέλεσμα αν ξανα-ανοίξουμε το έγγραφο HTML στον φυλλομετρητή είναι το εξής:



Επικεφαλίδα

Πρώτη παράγραφος.

{#fig:fig2

width=50%}

### 2.1.4 Σύνταξη της HTML

Θα δούμε στην ενότητα αυτή τα κύρια στοιχεία της σύνταξης ενός εγγράφου HTML. Όπως ήδη συζητήσαμε, σε ένα έγγραφο HTML περιέχονται κείμενα που θα εμφανιστούν στον χρήστη, καθώς και **στοιχεία HTML (HTML elements)**.

#### Στοιχεία

Τα στοιχεία ενός εγγράφου HTML έχουν εν γένει την εξής μορφή:

```
<tag> περιεχόμενο </tag>
```

όπου το `<tag>` και `</tag>` είναι οι ετικέτες οριοθέτησης της αρχής και τέλους του στοιχείου. Να σημειωθεί ότι στα ονόματα των στοιχείων δεν υπάρχει διάκριση μεταξύ κεφαλαίων μικρών, άρα η ετικέτα `<head>` μπορεί επίσης να γραφτεί ως `<HEAD>` ή `<Head>`.

Το περιεχόμενο ενός στοιχείου μπορεί να είναι κείμενο που θα εμφανιστεί στον χρήστη, ή άλλα στοιχεία που εμπεριέχονται στο συγκεκριμένο στοιχείο, για παράδειγμα είδαμε ότι στο στοιχείο `<strong>` περιέχεται στο στοιχείο `<h1>`.

Η ιεραρχική παράθεση των στοιχείων θα πρέπει να είναι αυστηρή. Για παράδειγμα, ο παρακάτω κώδικας δεν είναι συντακτικά ορθός αφού η ιεράρχηση του στοιχείου `<strong>` και του στοιχείου `<p>` δεν είναι αυστηρή, και θα δημιουργήσει σύγχυση στον συντακτικό αναλυτή της HTML:

```
<p>Το χωρίο μου είναι <strong>πολύ όμορφο.</p></strong>
```

Θα πρέπει να σημειωθεί εν γένει ότι ο συντακτικός αναλυτής της HTML δεν είναι αυστηρός όσον αφορά συντακτικά σφάλματα, και χειρίζεται τα σφάλματα τέτοιου τύπου με τον καλύτερο δυνατό τρόπο, ώστε το έγγραφο να εμφανιστεί στον χρήστη.

## Γνωρίσματα

Η ετικέτα αρχής ενός στοιχείου μπορεί να περιέχει και γνωρίσματα (attributes), όπως στο παράδειγμα:

```
<tag class="myclass"> περιεχόμενο </tag>
```

Τα γνωρίσματα χωρίζονται μεταξύ τους με κενά, και ακολουθούνται από το `=` και την τιμή, συνήθως μέσα σε εισαγωγικά. Η τιμή μπορεί να περιέχεται σε εισαγωγικά ή όχι, αν όμως η τιμή περιλαμβάνει τον κενό χαρακτήρα, τότε είναι υποχρεωτικά τα εισαγωγικά, αφού ο κενός χαρακτήρας είναι διαχωριστικός χαρακτήρας μεταξύ των γνωρισμάτων. Εν γένει είναι καλή πρακτική τις τιμές των γνωρισμάτων να τις θέτουμε εντός εισαγωγικών αν και αυτό, όπως αναφέρθηκε δεν είναι υποχρεωτικό πάντα.

Υπάρχουν γνωρίσματα που είναι λογικές μεταβλητές, συνεπώς δεν παίρνουν τιμή, για παράδειγμα:

```
<input type="text" disabled>
```

Το στοιχείο `<input>` έχει δύο γνωρίσματα, εκ των οποίων το δεύτερο είναι λογική μεταβλητή που παίρνει την τιμή `true`.

Τα γνωρίσματα αφορούν ιδιότητες του συγκεκριμένου στοιχείου, όπως για παράδειγμα η μοναδική ταυτότητα του στοιχείου, η κλάση στοιχείων στην οποία ανήκει, κλπ.



## Τυπικά γνωρίσματα στοιχείων

Στην συνέχεια παρατίθενται κάποια τυπικά γνωρίσματα στοιχείων της HTML

- **id** Ορίζει μοναδική ταυτότητα του στοιχείου
- **class** Ορίζει την κλάση στυλ που μοιράζεται το στοιχείο με άλλα στοιχεία
- **hidden** Ορίζει αν το στοιχείο εμφανίζεται στο χρήστη (λογικό γνώρισμα)
- **lang** Ορίζει την γλώσσα περιεχομένου του στοιχείου
- **style** Ορίζει ένα inline στυλ CSS για το στοιχείο
- **contenteditable** Αποφασίζει αν επιτρέπεται η τροποποίηση του περιεχομένου του στοιχείου
- **data-....** Αποθήκευση κρυφών δεδομένων, πχ `<li data-animal-type="bird">Owl</li>`
- **dir** (direction) Κατεύθυνση κειμένου σε στοιχείο που περιέχει κείμενο
- **draggable** Ορίζει αν στο στοιχείο μπορεί να εφαρμοστεί ενέργει σύρε-και-άφησε
- **dropzone** Ορίζει τι γίνεται αν τα δεδομένα που αφήνονται στο στοιχείο (σύρε-άφησε): αντιγράφονται, μετακινούνται ή συνδέονται
- **spellcheck** Ορίζει αν το στοιχείο μπορεί να ελεγχθεί για γλωσσική ορθότητα
- **tabindex** Ορίζει τη σειρά επίσκεψης των στοιχείων μέσω του πλήκτρου tab
- **title** Πρόσθετη πληροφορία, πχ `<abbr title="World Health Organization">WHO</abbr>`
- **translate** Ορίζει αν το περιεχόμενο πρέπει να μεταφραστεί

Εκ των γνωρισμάτων αυτών, τα **id** και **class** είναι ιδιαίτερα σημαντικά και χρησιμοποιούνται το πρώτο ως μοναδική ταυτότητα του στοιχείου, για παράδειγμα για αναφορά σε αυτό σε κώδικα JavaScript, ενώ το δεύτερο δηλώνει την κλάση στην οποία ανήκει το στοιχείο, και συνεπώς ποιοι κανόνες εμφάνισής του (εντολές CSS) ισχύουν.

## Κενά στοιχεία

Υπάρχουν κάποια στοιχεία τα οποία εισάγονται με μόνο μια ετικέτα αρχής, χωρίς να περιλαμβάνουν περιεχόμενο. Δηλαδή είναι αυτάρκη, τα ονομάζουμε **κενά στοιχεία (void elements)** γιατί ακριβώς δεν έχουν περιεχόμενο. Τέτοια στοιχεία είναι το `<hr>` (horizontal rule), δηλαδή μια διαχωριστική οριζόντια γραμμή, `<br>` (break), στοιχείο αλλαγής γραμμής, `<img>` για εισαγωγή εικόνας, κλπ. Όπως είδαμε στο παράδειγμα της προηγούμενης ενότητας, το στοιχείο `<meta>` ήταν ένα κενό στοιχείο.

Σε κάποιες προηγούμενες εκδόσεις της HTML οι ετικέτες των κενών στοιχείων έπρεπε να αυτοτερματιστούν ως `<tag/>`, υποδεικνύοντας ότι η ετικέτα τερματίζει στον εαυτό της. Αυτό πια δεν απαιτείται στην HTML5.

## Ειδικοί χαρακτήρες - διαχείριση κενών

Ένα άλλο θέμα που θα πρέπει να προσέξουμε είναι το πώς να χειριστούμε τους ειδικούς χαρακτήρες που χρησιμοποιούνται στη σύνταξη των ετικετών HTML όταν αυτοί πρέπει να εισαχθούν ως απλό κείμενο μέσα στο έγγραφο.

Ας πάρουμε ένα παράδειγμα. Έστω ότι επιθυμούμε να εμφανίσουμε στον χρήστη το εξής κείμενο:

Στην HTML για παράγραφο χρησιμοποιούμε την ετικέτα `<p>`.

Αν θέλουμε αυτό το κείμενο να το εμφανίσουμε στο χρήστη αυτό δεν είναι δυνατόν γιατί η ακολουθία χαρακτήρων `<p>` συνιστά η ίδια μια ετικέτα της HTML. Συνεπώς ο συντακτικός αναλυτής θα αντιληφθεί την ακολουθία αυτή ως εντολή HTML και όχι ως κείμενο.

Το πρόβλημα αυτό αντιμετωπίζεται με τη χρήση ειδικών ακολουθιών χαρακτήρων (character reference) που χρησιμοποιούνται για την εμφάνιση των ειδικών αυτών χαρακτήρων. Οι ακολουθίες αυτές περιλαμβάνουν το σύμβολο ampersand `'&'` και εν συνεχεία μία λεκτική περιγραφή του συμβόλου που ακολουθείται από το σύμβολο `;`

[σημείωση; το σύμβολο αυτό χρησιμοποιείται και στα Ελληνικά ως συντομογραφία της λέξης *και*, προέρχεται δε από παράφραση της λατινικής φράσης *'& per se and'*, δηλαδή *'& μόνο του σημαίνει και'*].

Η παρακάτω λίστα περιλαμβάνει τους πιο βασικούς χαρακτήρες που εισάγονται με αυτόν τον τρόπο.

- `&lt;`; [`<` less than, μικρότερο]
- `&gt;`; [`>` greater than, μεγαλύτερο]
- `&quot;`; [`"` quote, διπλό εισαγωγικό]
- `&apos;`; [`'` apostrophe, μονό εισαγωγικό]
- `&amp;`; [`&` ampersand, χαρακτήρας `&`]
- `&nbsp;`; [ non breaking space, κενό]

Ιδιαίτερη μνεία πρέπει να γίνει για την τελευταία ακολουθία χαρακτήρων `&nbsp;` που εκπροσωπεί τον κενό χαρακτήρα μη-αλλαγής γραμμής. Η ακολουθία αυτή μπορεί να χρησιμοποιηθεί για την εισαγωγή κενών στο περιεχόμενο ενός στοιχείου της HTML. Ο λόγος ύπαρξης αυτής της αναπαράστασης του κενού είναι ο εξής: Η HTML εμφανίζει τους χαρακτήρες που δεν τυπώνονται οι οποίοι μπορεί να είναι το κενό, ο χαρακτήρας new line, η χαρακτήρας tab, κλπ., όταν περιέχονται στο έγγραφο ως ακολουθία χαρακτήρων, μετατρέπονται σε ένα μόνο κενό χαρακτήρα μη-αλλαγής γραμμής (non-breaking space).

Για παράδειγμα αν εισάγουμε σε ένα έγγραφο HTML τα εξής στοιχεία:

```
<p>το καλοκαίρι κάνει ζέστη.</p>

<p>το καλοκαίρι           κάνει
    ζέστη.</p>
```

Το αποτέλεσμα στον φυλλομετρητή θα είναι το εξής:

το καλοκαίρι κάνει ζέστη.

το καλοκαίρι κάνει ζέστη.

{#fig:fig3 width=70%}



Όπως βλέπουμε στην εικόνα @fig:fig3 τα δύο στοιχεία εμφανίζονται με ακριβώς τον ίδιο τρόπο. Αυτό γιατί την ακολουθία των κενών ή μη εκτυπώσιμων χαρακτήρων του δεύτερου στοιχείου, ακόμη και τον χαρακτήρα αλλαγής γραμμής (new line), θα την αντικαταστήσει ο αναλυτής της HTML με ένα μόνο κενό.

Όμως υπάρχουν περιπτώσεις που επιθυμούμε να υπάρχουν πολλά κενά. Τότε μπορούμε να χρησιμοποιήσουμε τον ειδικό χαρακτήρα `&nbsp;`, ενώ με το στοιχείο `<br>` μπορούμε να επιβάλουμε αλλαγή γραμμής.

## Σχόλια στον κώδικα HTML

Σε ένα έγγραφο HTML μπορούμε να εισάγουμε σχόλια με τον εξής τρόπο:

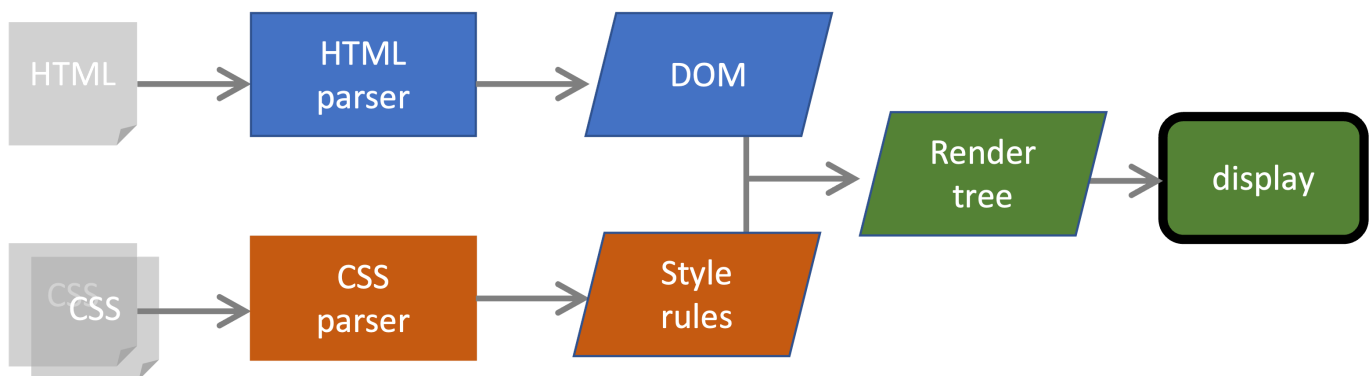
```
<!-- αυτό είναι ένα σχόλιο  
που μπορεί να εκτείνεται  
σε πολλές γραμμές ...  
-->
```

Παρατηρούμε ότι και τα σχόλια περιλαμβάνονται μεταξύ των συμβόλων `<` και `>` όπως όλα τα στοιχεία HTML, και μετά ακολουθεί η ακολουθία θαυμαστικό -- παύλα-παύλα, δηλώνοντας ότι αυτό που ακολουθεί είναι ένα σχόλιο, το οποίο τερματίζεται με την ακολουθία παύλα-παύλα.

### 2.1.5 Μοντέλο αντικειμένων του εγγράφου HTML (DOM)

Στην ενότητα αυτή θα δοθεί μια σύντομη περιγραφή της λειτουργίας του συντακτικού αναλυτή της HTML στο περιβάλλον του φυλλομετρητή και της διαδικασίας δημιουργίας του **Μοντέλου Αντικειμένων του εγγράφου (Document Object Model, DOM)**.

Στο παρακάτω διάγραμμα φαίνεται η διαδικασία συντακτικής ανάλυσης ενός εγγράφου HTML που έχει ως ενδιαμέσο βήμα τη δημιουργία μιας αναπαράστασης της ιεραρχίας των στοιχείων του εγγράφου στη μνήμη του υπολογιστή, ως Μοντέλο Αντικειμένων του Εγγράφου (DOM). Στη συνέχεια τα αντικείμενα αυτά σε συνδυασμό με φύλλα στυλ που συνοδεύουν το έγγραφο, παράγουν την εμφάνιση του εγγράφου στο παράθυρο του φυλλομετρητή.



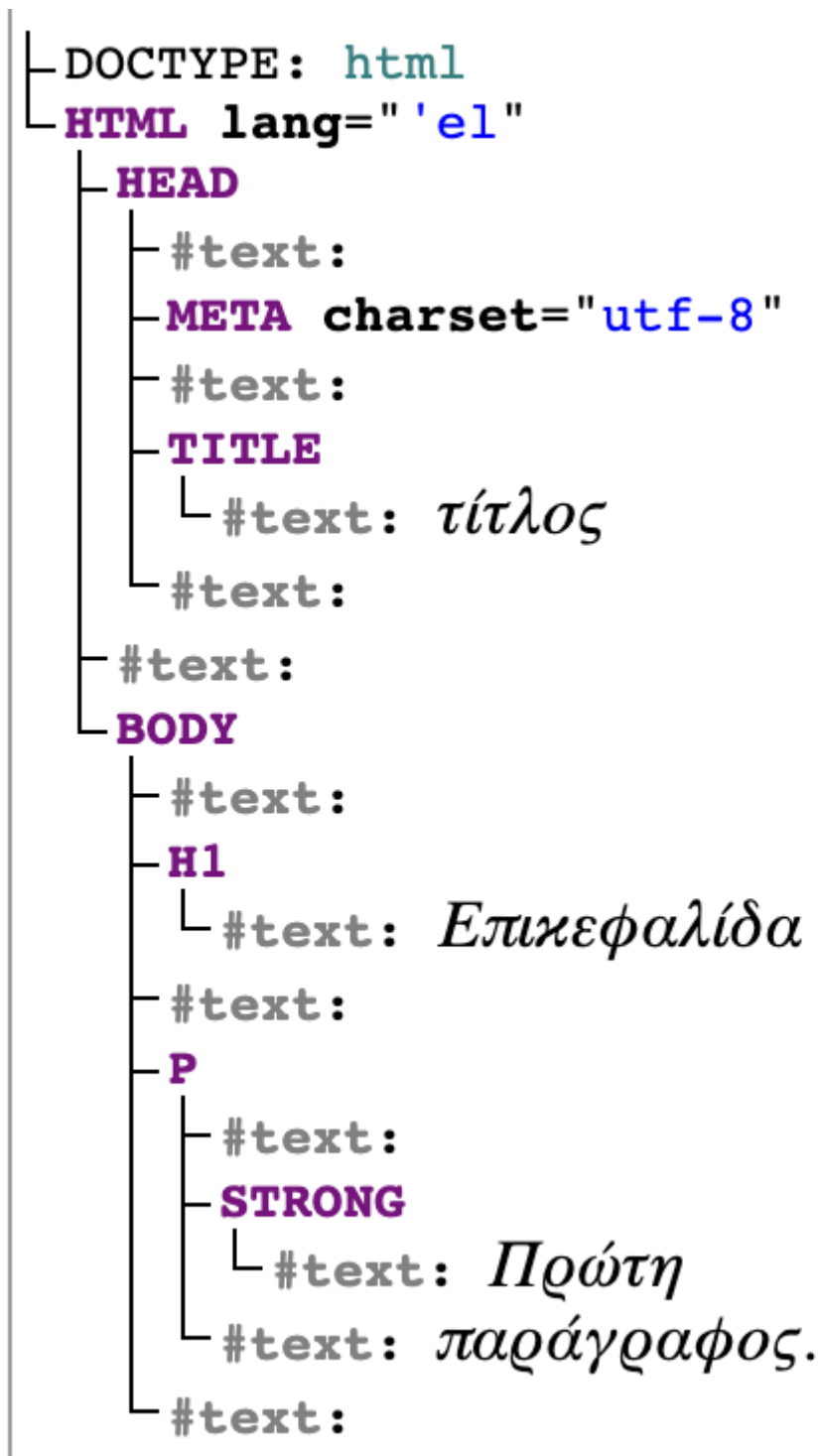
{#fig:fig4 width=70%}

Το DOM δημιουργείται κατά τη συντακτική ανάλυση του εγγράφου HTML. Στην ουσία είναι μία ιεραρχική δομή δεδομένων που παράγει μία νέου τύπου αναπαράσταση ενός εγγράφου HTML. Κάθε κόμβος του DOM είναι ένα στοιχείο της HTML ή δεδομένα προς παρουσίαση. Το έγγραφο HTML φορτώνεται στον φυλλομετρητή, με κύριο στόχο να παρουσιαστεί προς το χρήστη, ενώ γίνεται η συντακτική του ανάλυση,

παράγεται αυτή η ιεραρχική δομή αντικειμένων που έχει ως ρίζα το αντικείμενο `document` του φυλλομετρητή. Αυτή η ιεραρχική δομή δεδομένων ορίζεται ως το Document Object Model και αμέσως μόλις φορτωθεί και ολοκληρωθεί η συντακτική ανάλυση, γίνεται διαθέσιμο μέσω κάποιων διεπαφών στα προγραμματιστικά περιβάλλοντα, δηλαδή στην JavaScript. Μέσω της διεπαφής η Javascript βλέπει αυτά τα αντικείμενα που είναι τα στοιχεία της ιστοσελίδας και μπορεί να τα επεξεργαστεί, να το τροποποιήσει, να τα μεταφέρει από μία θέση του δέντρου αυτού σε μία άλλη, να τους αλλάξει τις ιδιότητες, το στυλ, να τα εμφανίσει, να τα αποκρύψει, και ούτω καθεξής.

Άρα η ύπαρξη του Document Object Model και των διεπαφών του, είναι πολύ σημαντικά γιατί επιτρέπουν σε μια γλώσσα προγραμματισμού όπως η Javascript, να αλληλοεπιδρά και να επηρεάζει τα στοιχεία της ιστοσελίδας και συνεπώς, να της προσδώσει δυναμική συμπεριφορά.

Υπάρχουν διάφορα εργαλεία που μάς επιτρέπουν να δούμε το DOM ενός εγγράφου. Για παράδειγμα στην ιστοσελίδα <https://software.hixie.ch/utilities/js/live-dom-viewer/> αν φορτώσουμε τον κώδικα της προηγούμενης ενότητας (αρχείο ex00.html), παράγεται η εξής αναπαράσταση του DOM:



{#fig:fig5

```
width=50%}
```

Παρατηρούμε ότι στο DOM περιέχονται με δομή δένδρου, κόμβοι, οι οποίοι είναι είτε περιεχόμενο (εμφανίζονται ως `#text`) είτε στοιχεία HTML.

Επίσης μπορούμε να δούμε ένα παράδειγμα χρήσης της προγραμματιστικής διεπαφής του DOM.

Ανοίγουμε το παραπάνω αρχείο στο φυλλομετρητή και ενεργοποιούμε τα εργαλεία ανάπτυξης (στο Chrome, επιλογή inspection), από τα οποία επιλέγουμε την κονσόλα JavaScript (console)

Αν δώσουμε την εντολή:

```
document.querySelector('body')
```

Θα πάρουμε ως απόκριση το υποσύνολο της ιεραρχίας του DOM που αφορά το στοιχείο `<body>` και τα παιδιά του.

Αν στη συνέχεια θέλουμε να βρούμε τα παιδιά του στοιχείου `body` μπορούμε να ζητήσουμε την τιμή της ιδιότητας `children` του αντίστοιχου κόμβου:

```
document.querySelector('body').children
```

Η απόκριση σε αυτή την περίπτωση είναι ένας πίνακας με δύο στοιχεία: `[h1, p]`.

Αν τέλος θέλουμε να βρούμε το πλήθος των παιδιών του `body`, μπορούμε να ζητήσουμε την ιδιότητα `length` του πίνακα αυτού:

```
document.querySelector('body').children.length
```

και η απάντηση βεβαίως στην περίπτωση αυτή είναι 2.

Αυτά είναι παραδείγματα της προγραμματιστικής διεπαφής του DOM, η οποία περιλαμβάνει πολλές δυνατότητες. Θα επανέλθουμε στη διεπαφή του DOM στο κεφάλαιο που συζητάμε την JavaScript.

## 2.2 Στοιχεία της HTML

### 2.2.1 Τα στοιχεία του `<head>`

Όπως έχουμε ήδη συζητήσει ένα έγγραφο HTML περιέχει δύο παιδιά της ρίζας του, το `<head>` και το `<body>`. Το στοιχείο `<head>` είναι το πρώτο από τα παιδιά της ρίζας `html` και περιέχει μετα-δεδομένα για το έγγραφο. Είναι στοιχεία αντίστοιχα με τις κεφαλίδες ενός μηνύματος HTTP, που είδαμε στο πρώτο κεφάλαιο. Περιλαμβάνει πληροφορίες οι οποίες δεν θα εμφανιστούν στον χρήστη αλλά είναι χρήσιμες για τον φυλλομετρητή, για να ξέρει πώς να χειριστεί το συγκεκριμένο έγγραφο. Επίσης πολλά μετα-δεδομένα βοηθάνε τις μηχανές αναζήτησης να κατανοήσουν το αντικείμενο της σελίδας.

Τα στοιχεία που μπορεί να περιέχονται στο τμήμα αυτό του εγγράφου HTML είναι:

`<base>`, `<link>`, `<script>`, `<style>`, `<title>`, `<meta>`

Ας δούμε καθένα από αυτά ξεχωριστά.

`<base>`, ορίζει την URL, με βάση την οποία δημιουργούνται τα σχετικά URL που συναντώνται στο έγγραφο. Δέχεται ως γνωρίσματα το `href` με τη διεύθυνση (υποχρεωτικό) και το `target` που καθορίζει, όταν επιλεγεί το URL, αν θα ανοίξει σε νέο `target="_blank"` ή στο ίδιο `target="_self"` παράθυρο του φυλλομετρητή. Παράδειγμα.

```
<base href="https://www.upatras.gr/" target="_blank">
```

`<link>` Με το στοιχείο αυτό γίνεται σύνδεση του εγγράφου HTML με ένα εξωτερικό πόρο. Η πιο συνηθισμένη χρήση του είναι για σύνδεση με εξωτερικά αρχεία στυλ. Παράδειγμα:

```
<link href="main.css" rel="stylesheet">
```

Μια άλλη χρήση του είναι για σύνδεση με το εικονίδιο της ιστοσελίδας μεγέθους 16px, που εμφανίζεται στην πάνω αριστερή γωνία του αντίστοιχου παράθυρου, ή καρτέλας, το **favicon (favorite icon)**.

```
<link rel="icon" href="favicon.ico" type="image/x-icon">
```

`<script>` Το στοιχείο αυτό χρησιμοποιείται για ενσωμάτωση κώδικα JavaScript στο έγγραφο, καθώς επίσης και δεδομένων. Ο κώδικας μπορεί να περιέχεται εντός του στοιχείου `<script>`, ή ο κώδικας να βρίσκεται σε ένα εξωτερικό αρχείο και μέσω του γνωρίσματος `src` να γίνεται αναφορά σε αυτόν.

Αν ο κώδικας JavaScript βρίσκεται σε εξωτερικό αρχείο η ενσωμάτωσή του γίνεται ως εξής:

```
<script src="javascript.js" defer></script>
```

Θα πρέπει να σημειωθεί ότι το γνώρισμα `defer` (λογική μεταβλητή) δηλώνει ότι ο κώδικας θα φορτωθεί αφού ολοκληρωθεί το φόρτωμα του εγγράφου HTML. Θα συζητήσουμε αυτή την επιλογή στο κεφάλαιο της JavaScript.

Αν τώρα επιθυμούμε να ενσωματώσουμε τον κώδικα μέσα στο ίδιο το έγγραφο HTML, αυτό μπορεί να γίνει ως εξής:

```
<script>
  alert("Καλώς ήρθατε!")
</script>
```

Θα πρέπει να σημειωθεί εδώ ότι το στοιχείο αυτό μπορεί να ενσωματωθεί και στο `<body>` του εγγράφου, ενώ είναι σημαντικό να ορίσουμε πότε θα εκτελεστεί ο κώδικας, δηλαδή πριν ή μετά την ολοκλήρωση δημιουργίας του DOM, όπως θα συζητήσουμε σε επόμενο κεφάλαιο που αφορά την εκτέλεση του κώδικα JavaScript.

`<style>` Το στοιχείο αυτό επιτρέπει την εισαγωγή εντολών CSS για το έγγραφο. Όπως είδαμε εξωτερικά αρχεία CSS μπορούν να συνδεθούν με χρήση του στοιχείου `link`, όμως κάποιες φορές κάποιες πιο ειδικές εντολές μορφοποίησης του περιεχομένου μπορεί να εισαχθούν μέσω του στοιχείου `<style>` στο τμήμα `<head>`.

Σε προηγούμενη ενότητα είδαμε πώς χρησιμοποιήθηκε το στοιχείο αυτό για να αλλάξουμε το χρώμα της σελίδας και του κειμένου στο εισαγωγικό μας παράδειγμα.

Εδώ θα προχωρήσουμε σε ένα πιο προχωρημένο παράδειγμα, όπου θα ορίσουμε ότι αν το παράθυρο του φυλλομετρητή έχει πλάτος μικρότερο ή ίσο με 600px να αλλάζει το χρώμα του παραθύρου σε μπλέ και τα γράμματα να γίνονται λευκά.

Η εντολή είναι η εξής:

```
<style>
  @media (max-width: 600px) {body {
    background-color: navy; color: white}}
</style>
```

Αυτό είναι ένα παράδειγμα **media query** που ορίζει ότι όταν ισχύει η πρώτη συνθήκη, τότε εκτελείται η εντολή. Θα γίνει εκτενής αναφορά στη σύνταξη των εντολών CSS στο αντίστοιχο κεφάλαιο.

**<title>** Το στοιχείο αυτό ορίζει το κείμενο στην άνω μπάρα του φυλλομετρητή ή της καρτέλας στην οποία παρουσιάζεται το έγγραφο HTML. Ένα παράδειγμα:

```
<title> Η πρώτη μου ιστοσελίδα </title>
```

Θα πρέπει να σημειωθεί ότι ο τίτλος της σελίδας βοηθάει τις μηχανές αναζήτησης, ιδιαίτερα αν είναι περιγραφικός του περιεχομένου της.

**<meta>** Το στοιχείο αυτό παρέχει τη δυνατότητα να εισάγουμε μετα-δεδομένα στην ιστοσελίδα. Το στοιχείο αυτό είναι ένα κενό στοιχείο, δηλαδή δεν χρειάζεται ετικέτα τερματισμού του, όπως ήδη είδαμε στο εισαγωγικό παράδειγμα. Στο παράδειγμα εκείνο, χρησιμοποιήσαμε το στοιχείο αυτό για περιγραφή της κωδικοποίησης των χαρακτήρων της σελίδας. (**<meta charset="utf-8">**). Αυτή είναι η πιο κλασσική χρήση του στοιχείου αυτού.

Υπάρχουν όμως ακόμη δύο χρήσεις με ευρεία διάδοση. Το στοιχείο όταν περιλαμβάνει τα γνωρίσματα **name**, **content** τότε περιέχει μετα-δεδομένα τα οποία επιτρέπουν σε μηχανές αναζήτησης να ευρετηριάσουν τη σελίδα πιο αποδοτικά. Ένα παράδειγμα:

```
<meta name="Description" content="author: N.Aavouris -C.Sintoris, title:
Introduction to web programming">
<meta name="robots" content="noindex, nofollow">
```

Στο παράδειγμα το πρώτο στοιχείο **<meta>** παρέχει μια περιγραφή του περιεχομένου της σελίδας, ενώ το δεύτερο ορίζει στις μηχανές αναζήτησης να μην ευρετηριάσουν τη σελίδα. Στις οδηγίες προς τους προγραμματιστές η Google κάνει ιδιαίτερη **αναφορά** στις δύο αυτές κατηγορίες μετα-δεδομένων.

Ένα άλλο παράδειγμα μεταδεδομένων, αφορά την εμφάνιση της σελίδας σε διαφορετικούς φυλλομετρητές, όπως για παράδειγμα στην περίπτωση κινητών με περιορισμένες διαστάσεις οθόνης και αυτόματη σμίκρυνση της σελίδας ώστε να χωράει το περιεχόμενο σε αυτή. Σε αυτή την περίπτωση θα πρέπει να χρησιμοποιηθεί το στοιχείο meta με γνώρισμα **name = "viewport"**. Για παράδειγμα αν επιθυμούμε η



σελίδα να ακολουθεί τις διαστάσεις της συσκευής, και να έχουμε ως αρχική κλίμακα 100% (δηλαδή να μην γίνεται σμίκρυνση της σελίδας), αυτό ορίζεται ως εξής:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Άλλες παράμετροι που μπορούν να προσδιοριστούν πέραν των width, initial-scale του παραδείγματος είναι: height, minimum-scale, maximum-scale, user-scalable.

### 2.2.2 Τα στοιχεία του περιεχομένου: **<body>**

Είδαμε στην προηγούμενη ενότητα το περιεχόμενο του **<head>** ενός εγγράφου HTML. Το δεύτερο τμήμα του εγγράφου, το **<body>** έχει ακόμη περισσότερο ενδιαφέρον βέβαια γιατί εδώ θα βρούμε το περιεχόμενο της σελίδας μας, Θα δούμε τι τρόπους μας παρέχει η HTML για την οργάνωσή του περιεχομένου.

#### **Στοιχεία κειμένου με σημασιολογία **h1**, ... **h6**, **p****

Ένα οποιοδήποτε κείμενο οργανώνεται σε παραγράφους, οι οποίες έχουν επικεφαλίδες διαφορετικών επιπέδων, όπως γίνεται για παράδειγμα με τα κεφάλαια και τις ενότητες αυτού του βιβλίου. Τα στοιχεία που περιέχουν τις επικεφαλίδες διαφόρων επιπέδων φέρουν τα ονόματα **h1**, **h2**, ..., **h6** (headings 1... 6). Η επικεφαλίδα του υψηλότερου επιπέδου είναι η **<h1>** που την είδαμε και στο εισαγωγικό μας παράδειγμα. Ενώ το απλό κείμενο εκτός των κεφαλίδων, περιέχεται σε στοιχεία **<p>** (paragraphs).

Ας δούμε ένα παράδειγμα. Έστω το παρακάτω χειρόγραφου του ποιητή μας Κώστα Καβάφη, που περιέχεται στο ψηφιακό αρχείο του Καβάφη του Ιδρύματος Ωνάση (<https://cavafy.onassis.org/el/object/39eh-xggr-mk3e/>).

## Τείχη

Χωρίς περίσκεψιν, χωρίς λύπην, χωρίς αιδῶ  
μεγάλα κ' ὑψηλά τριγύρω μου ἐκτίσαν τείχη.

Καὶ κάθομαι καὶ ἀπελπίζομαι τώρα ἐδῶ.  
Ἄλλο δὲν σκέπτομαι: τὸν νοῦν μου τρώγει αὐτὴ ἡ τύχη.

Διότι πράγματα πολλὰ ἔξω νὰ κάμω εἶχον.  
Ἢ ὅταν ἐκτίζαν τὰ τείχη πῶς νὰ μὴν προσέξω.

Ἀλλὰ δὲν ἄκουσα ποτέ κρότον κτιστῶν ἢ ἤχον.  
Ἀνεπαισθήτως μ' ἐκλείσαν ἀπὸ τὸν κόσμον ἔξω.

Κ. Π. Καβάφης

{#fig:kavafis width=60%}

Ο ποιητής έχει δώσει τον τίτλο στο ποίημα, τον οποίο έχει υπογραμμίσει για να τού δώσει βαρύτητα, έχει αφήσει κενό και έχει στη συνέχεια δώσει ξεχωριστά τις τέσσερις στροφές του ποιήματος, τέλος υπογράφει ιδιόχειρα.

Αν το κείμενο αυτού του ποιήματος δεν το είχαμε εισάγει σε στοιχεία HTML θα ήταν ως εξής:

<body>

Τείχη

Χωρίς περίσκεψιν, χωρίς λύπην, χωρίς αιδῶ  
μεγάλα κ' ὑψηλά τριγύρω μου ἐκτίσαν τείχη.

Και κάθομαι και ἀπελπίζομαι τώρα ἐδῶ.

Ἄλλο δὲν σκέπτομαι: τὸν νοῦν μου τρώγει αὐτὴ ἡ τύχη·

διότι πράγματα πολλὰ ἔξω νὰ κάμω εἶχον.

Ἢ ὅταν ἐκτίζαν τὰ τείχη πῶς νὰ μὴν προσέξω.

Ἀλλὰ δὲν ἄκουσα ποτέ κρότον κτιστῶν ἢ ἤχον.

Ἀνεπαισθήτως μ' ἐκλείσαν ἀπὸ τὸν κόσμον ἔξω.

</body>

Το αποτέλεσμα στον φυλλομετρητή είναι:

Τείχη Χωρίς περίσκεψιν, χωρίς λύπην, χωρίς αιδώ μεγάλα κ' υψηλά  
τριγύρω μου έκτισαν τείχη. Και κάθομαι και απελπίζομαι τώρα  
εδώ. Άλλο δεν σκέπτομαι: τον νουν μου τρώγει αυτή η τύχη· διότι  
πράγματα πολλά έξω να κάμω είχαν. Α όταν έκτιζαν τα τείχη πώς  
να μην προσέξω. Αλλά δεν άκουσα ποτέ κρότον κτιστών ή ήχον.  
Ανεπαισθήτως μ' έκλεισαν από τον κόσμο έξω.

{#fig:fig7 width=50%}

As οργανώσουμε τώρα το κείμενο με χρήση στοιχείων HTML, ως ακολούθως:

```
<body>
  <h1>Τείχη</h1>
  <p>Χωρίς περίσκεψιν, χωρίς λύπην, χωρίς αιδώ<br>
    μεγάλα κ' υψηλά τριγύρω μου έκτισαν τείχη.</p>

  <p>Και κάθομαι και απελπίζομαι τώρα εδώ.<br>
  Άλλο δεν σκέπτομαι: τον νουν μου τρώγει αυτή η τύχη.</p>

  <p>διότι πράγματα πολλά έξω να κάμω είχαν.<br>
  Α όταν έκτιζαν τα τείχη πώς να μην προσέξω.</p>

  <p>Αλλά δεν άκουσα ποτέ κρότον κτιστών ή ήχον.<br>
  Ανεπαισθήτως μ' έκλεισαν από τον κόσμο έξω. </p>
</body>
```

Το αποτέλεσμα στον φυλλομετρητή μοιάζει περισσότερο με το αρχικό ποίημα.

# Τείχη

Χωρίς περίσκεψιν, χωρίς λύπην, χωρίς αιδώ  
μεγάλα κ' υψηλά τριγύρω μου έκτισαν τείχη.

Και κάθομαι και απελπίζομαι τώρα εδώ.  
Άλλο δεν σκέπτομαι: τον νουν μου τρώγει αυτή η τύχη·

διότι πράγματα πολλά έξω να κάμω είχαν.  
Α όταν έκτιζαν τα τείχη πώς να μην προσέξω.

Αλλά δεν άκουσα ποτέ κρότον κτιστών ή ήχον.  
Ανεπαισθήτως μ' έκλεισαν από τον κόσμον έξω.

{#fig:fig8 width=50%}

Παραμένει θέμα υπό εξέταση, αφού μάλιστα δούμε σε επόμενα κεφάλαια τη χρήση CSS, πώς θα μπορούσαμε να προχωρήσουμε σε τροποποίηση της εμφάνισης των στοιχείων `<h1>` και `<p>` ώστε το ποίημα να μοιάζει σε μεγαλύτερο βαθμό το αρχικό χειρόγραφο του ποιητή.

Η χρήση των στοιχείων αυτών που φέρουν σημασιολογικό περιεχόμενο θα πρέπει να γίνεται με προσοχή. Τα στοιχεία `<h1>` έως `<h6>`, όπως αναφέρθηκε, αναπαριστούν επικεφαλίδες με φθίνουσα σπουδαιότητα. Η `<h1>` είναι η υψηλότερη επικεφαλίδα. Για μια ιστοσελίδα θα πρέπει να υπάρχει μόνο μία `<h1>` ή αν η ιστοσελίδα περιέχει πολλαπλά άρθρα, το καθένα να έχει μόνο ένα `<h1>`, όπως ο τίτλος ενός βιβλίου είναι ένας. Η `<h2>` είναι το δεύτερο επίπεδο, που θα μπορούσε να αντιστοιχεί στα κεφάλαια του βιβλίου. Η `<h3>` σε ενότητες του κάθε κεφαλαίου, και ούτω καθεξής μέχρι το επίπεδο `<h6>`. Κάθε μία από αυτές θα πρέπει να αντιστοιχεί σε διαφορετικού μεγέθους γραμματοσειρά.

Θα πρέπει να προσέξουμε ώστε να χρησιμοποιούνται τα στοιχεία αυτά όχι για την εμφάνιση του κειμένου, αλλά για να το παρουσιάσουμε ως μια ιεραρχία στοιχείων με νόημα. Αυτό έχει ιδιαίτερη σημασία για τα άτομα με ειδικές ανάγκες, αφού επηρεάζει την προσβασιμότητα της σελίδας. Κάποιοι σχεδιαστές χρησιμοποιούν την `<h1>` ή την `<h2>`, απλά για μορφοποίηση για να τονίσουν ένα τμήμα του κειμένου. Αυτός είναι εσφαλμένος τρόπος χρήσης των στοιχείων αυτών. Τα στοιχεία αυτά έχουν σημασιολογία. Σημαίνει ότι το στοιχείο `<h1>` είναι η επικεφαλίδα υψηλότερου επιπέδου, και συνήθως πρέπει να είναι μια μόνο σε όλη τη σελίδα. Οι κεφαλίδες `<h2>` είναι οι επικεφαλίδες του δεύτερου επιπέδου και στο περιεχόμενό τους, υπάρχουν κεφαλίδες τρίτου επιπέδου, κλπ. Κάποιοι χρήστες με δυσκολίες όρασης μπορεί να πλοηγούνται γρήγορα σε κάποιο επίπεδο για να τις εξετάσουν όπως κάνουμε όταν ξεφυλλίζουμε ένα βιβλίο, ώστε να ελέγξουν αν υπάρχει ένα κεφάλαιο που τους ενδιαφέρει. Αυτό γίνεται για παράδειγμα με τους screen readers, δηλαδή τους αναγνώστες της οθόνης που είναι συσκευές που χρησιμοποιούνται για άτομα με ειδικές ανάγκες με δυσκολίες όρασης. Αν χρησιμοποιήσουμε τα στοιχεία h1 έως h6 καταχρηστικά απλώς για παρουσίαση μεγέθους γραμματοσειρών, ουσιαστικά καταργούμε αυτή την αρχή

της ιεράρχησης του κειμένου και ο αναγνώστης της οθόνης δεν μπορεί να εξυπηρετήσει το άτομο με ειδικές ανάγκες.

## Στοιχεία κειμένου χωρίς σημασιολογία `div`, `span`

Δύο στοιχεία με ευρεία χρήση είναι τα `<div>` και `<span>`. Το `<div>` ορίζει ένα μπλοκ κειμένου, αντίστοιχα με το

, ενώ το `<span>` ορίζει μια ακολουθία χαρακτήρων μέσα σε ένα μπλοκ, είναι όπως λέγεται *στοιχείο γραμμής* (*inline element*). Τα δύο αυτά στοιχεία δεν φέρουν σημασιολογία αλλά ορίζουν στοιχεία στα οποία μπορούμε να επιβάλουμε ορισμένο στυλ εμφάνισης. Η χρήση τους είναι ιδιαίτερα εκτεταμένη από τότε που χρησιμοποιήθηκε εκτενώς η CSS για καθορισμό της μορφής του περιεχομένου της σελίδας.

Θα πρέπει να σημειωθεί ότι παλαιότερα η HTML είχε πολλές ετικέτες, οι οποίες τείνουν να καταργηθούν, και οι οποίες αφορούσαν τη μορφοποίηση. Όριζαν για παράδειγμα στοιχεία `<b>` για έντονο κείμενο (bold), `<i>` για πλάγια γραφή (italics) και `<u>` για υπογραμμισμένο κείμενο (underline). Αυτές τείνουν να καταργηθούν γιατί αντικαθίστανται από τα στυλ όπου εκεί ορίζουμε τρόπο εμφάνισης. Τα στυλ επιβάλλονται σε στοιχεία `<div>` και `<span>`, τα οποία ορίζουν περιοχές στις οποίες εφαρμόζουμε ένα στυλ εμφάνισης, με κατάλληλες εντολές CSS. Και αυτό γίνεται με το να δώσουμε ταυτότητα σε κάθε τέτοιο στοιχείο, με το γνώρισμα `id` ή να ορίσουμε μια κλάση που επιβάλλεται σε ένα σύνολο από στοιχεία με το γνώρισμα `class`.

Η παράγραφος `<p>` μοιάζει αρκετά με το `<div>`. Υπάρχουν κάποιες διαφορές. Η παράγραφος ορίζει μια παράγραφο κειμένου, ένα τμήμα ένα μπλοκ κειμένου που ξεχωρίζει από το υπόλοιπο κείμενο. Ο φυλλομετρητής προσθέτει ένα περιθώριο από το κείμενο που υπάρχει πριν ή μετά την παράγραφο. Μία ακόμη διαφορά είναι ότι **σε ένα στοιχείο `<p>` δεν μπορούμε να εισάγουμε άλλα στοιχεία block**, ενώ αυτό μπορεί να γίνει σε ένα `<div>`. Άρα ένα `div` μπορεί να περιέχει παραγράφους ή άλλα στοιχεία `div`, ενώ μία παράγραφος δεν μπορεί να περιέχει `div` ή άλλες παραγράφους.

Ας δούμε ένα παράδειγμα. Έστω ότι επιθυμούμε να διαφοροποιήσουμε τον πρώτο χαρακτήρα κάθε στροφής του ποιήματος, αυτό γίνεται ως εξής: Σε κάθε πρώτη σειρά εισάγουμε τον πρώτο χαρακτήρα σε ένα στοιχείο `span`. Ως παράδειγμα η πρώτη στροφή του ποιήματος γίνεται:

```
<p><span class="first-letter">Χ</span>ωρίς περίσκεψιν, χωρίς λύπην, χωρίς  
αιδώς<br>  
μεγάλα κ' υψηλά τριγύρω μου έκτισαν τείχη.</p>
```

Αυτό επαναλαμβάνεται για όλες τις στροφές. Επίσης στο στοιχείο `<head>` εισάγουμε έναν κανόνα μορφοποίησης για την κλάση `"first-letter"`, με βάση τον οποίο το περιεχόμενο του είναι διπλάσιο σε μέγεθος από το υπόλοιπο κείμενο (να σημειωθεί ότι το όνομα μιας κλάσης προσδιορίζεται με τον χαρακτήρα `"."` στις εντολές CSS, όπως θα δούμε στο αντίστοιχο κεφάλαιο).

```
<style>  
  .first-letter {font-size: 2em;}  
</style>
```

Το αποτέλεσμα είναι το εξής:

# Τείχη

Χωρίς περίσκεψιν, χωρίς λύπην, χωρίς αιδώ  
μεγάλα κ' υψηλά τριγύρω μου έκτισαν τείχη.

Και κάθομαι και απελπίζομαι τώρα εδώ.  
Άλλο δεν σκέπτομαι: τον νουν μου τρώγει αυτή η τύχη·

Διότι πράγματα πολλά έξω να κάμω είχαν.  
Α όταν έκτιζαν τα τείχη πώς να μην προσέξω.

Αλλά δεν άκουσα ποτέ κρότον κτιστών ή ήχον.  
Ανεπαισθήτως μ' έκλεισαν από τον κόσμο έξω.

{#fig:fig9 width=50%}

## Έμφαση σε κείμενο `<strong>`, `<em>`

Δύο στοιχεία που μπορούμε να χρησιμοποιήσουμε για να δώσουμε ιδιαίτερη έμφαση σε τμήματα ενός κειμένου είναι τα στοιχεία `<em>` και `<strong>`.

Και τα δύο αυτά στοιχεία χρησιμοποιούνται για να δώσουμε μία μεγαλύτερη έμφαση σε ένα κείμενο και είναι στοιχεία γραμμής (inline elements). Δηλαδή δεν δημιουργούν ένα νέο block κειμένου αλλά αλλάζουν την έμφαση σε κάποια τμήματα ενός κειμένου.

Ας δούμε ένα παράδειγμα. Ας πούμε ότι στην παρακάτω παράγραφο «χαίρομαι ιδιαίτερα που ήρθες στην ώρα σου» επιθυμούμε να τονίσουμε τη λέξη **ιδιαίτερα** και τη φράση **στην ώρα σου** και για αυτό τα βάζουμε μέσα σε ένα στοιχείο `<em>` (έμφαση).

```
<p>χαίρομαι <em>ιδιαίτερα</em> που ήρθες <em>στην ώρα σου</em></p>
```



Όταν παρουσιαστεί αυτό το κείμενο στο φυλλομετρητή θα δούμε ότι η λέξη ιδιαίτερα και η φράση στην ώρα σου εμφανίζονται με πλάγιους χαρακτήρες (*italics*).

## χαίρομαι *ιδιαίτερα* που ήρθες στην ώρα σου

{#fig:fig10 width=50%}

Το στοιχείο `<strong>` χρησιμοποιείται για να δοθεί ακόμα πιο μεγάλη έμφαση. Αν θέλουμε να συγκρίνουμε το `em` με το `strong` το `strong` δίνει με τον ίδιο τρόπο που το κάνει το `em` ακόμα μεγαλύτερη έμφαση στο τμήμα που αναφέρεται, μάλιστα αυτό λαμβάνεται υπόψη και από τους αναγνώστες για άτομα με ειδικές ανάγκες. Ένα παράδειγμα είναι, έστω ότι θέλουμε να πούμε "αυτό το υγρό είναι πολύ τοξικό". Επειδή επιθυμούμε να δώσουμε πολύ μεγάλη έμφαση, ώστε να προσέχουν όσοι χειρίζονται αυτό το υγρό. το διαμορφώνουμε ως εξής:

```
<p>αυτό το υγρό είναι <strong> πολύ τοξικό </strong></p>
```

Αυτή η παράγραφος θα εμφανιστεί στον φυλλομετρητή συνήθως με έντονη γραφή (**bold**).

## αυτό το υγρό είναι **πολύ τοξικό**

{#fig:fig11

width=50%}

Η έντονη γραφή (**bold**) λοιπόν έχει πιο μεγάλη βαρύτητα από ότι η πλάγια γραφή *italics*. Ένα ερώτημα που τίθεται είναι αν τα στοιχεία `<b>` (**bold**) και `<i>` (*italics*) που είναι παλαιότερα και οι αντίστοιχες ετικέτες πιο σύντομες αν συνεχίζουν να έχουν χρήση. Η απάντηση είναι ότι τα στοιχεία `<strong>` και `<em>` επειδή έχουν σημασιολογικό περιεχόμενο, εκφράζοντας έμφαση διαφορετικής βαρύτητας, υποστηρίζονται από τους αναγνώστες της οθόνης (*screenreaders*), οι οποίοι τονίζουν με πιο μεγάλη έμφαση τη φωνή του ομιλητή και έτσι και το άτομο με ειδικές ανάγκες μπορεί να αντιληφθεί καλύτερα το νόημα. Άρα συστήνονται πολύ περισσότερο από τα `<b>` και `<i>` που είναι μεν συντομογραφίες ως ετικέτες του ενός χαρακτήρα, όμως δεν φέρουν σημασιολογικό περιεχόμενο, αλλά απλά αλλάζουν την εμφάνιση.

### Δείκτης και κάτω-δείκτης κειμένου `<sup>`, `<sub>`

Μία ανάγκη που έχουμε είναι να παρουσιάσουμε κάποιους χαρακτήρες σαν δείκτες πάνω ή κάτω από την κανονική σειρά του κειμένου.

Για παράδειγμα έστω ότι επιθυμούμε να παρουσιάσουμε τις εξής φράσεις:

Γιορτάζουμε την 25<sup>η</sup> Μαρτίου.

Η χημική σύνθεση της καφεΐνης είναι : C<sub>8</sub>H<sub>10</sub>N<sub>4</sub>O<sub>2</sub>.

Αν  $x^2$  είναι 9, τότε το  $x$  είναι είτε 3 είτε -3.

{#fig:fig12 width=50%}

Για τη λέξη 25<sup>η</sup> Μαρτίου ο χαρακτήρας η είναι δείκτης με μικρότερη γραμματοσειρά. Το ίδιο για τη χημική σύνθεση όπου χρησιμοποιούμε κάτω δείκτες. Πώς μπορούμε να κάνουμε αυτό στην HTML;

Ας δούμε τον κώδικα που παράγει αυτή την ημερομηνία.

```
<p>Γιορτάζουμε την 25<sup>η</sup> Μαρτίου.</p>
```

Παρατηρούμε ότι σε ένα στοιχείο `<p>` paragraph έχουμε το κείμενο «Γιορτάζουμε την 25<sup>η</sup> Μαρτίου» όπου ορίζουμε το superscript ως στοιχείο `<sup>` τον χαρακτήρα δείκτη. Έτσι ο χαρακτήρας αυτός εμφανίζεται ως άνω δείκτης για να μπορέσουμε να εκφράσουμε την ημερομηνία με αυτό τον τρόπο.

Ας δούμε τώρα το δεύτερο παράδειγμα της χημικής ένωσης.

```
<p>Η χημική σύνθεση της καφεΐνης είναι :  
C<sub>8</sub>H<sub>10</sub>N<sub>4</sub>O<sub>2</sub>.</p>
```

Εδώ παρατηρώ ότι τα στοιχεία που επιθυμώ να θέσω ως κάτω δείκτες τα εντάσσω σε στοιχεία `<sub>` με αντίστοιχο τρόπο.

Το ίδιο και για την περίπτωση της μαθηματικής έκφρασης:

```
<p>Αν  $x^{2}$  είναι 9, τότε το  $x$  είναι είτε 3 είτε -3.</p>
```

### Οριζόντια γραμμή και αλλαγή γραμμής `<hr>`, `<br>`

Επίσης σημαντικά στοιχεία είναι αυτά που μας επιτρέπουν είτε να ορίσουμε μία οριζόντια γραμμή που διαχωρίζει περιοχές του κειμένου είτε να αλλάξουμε γραμμή. Και τα δύο αυτά στοιχεία είναι κενά στοιχεία.

Η ετικέτα με την οποία δημιουργούμε ένα στοιχείο οριζόντιας γραμμής είναι η `<hr>`, (Horizontal rule) που διαχωρίζει περιοχές. Μάλιστα αξίζει να αναφερθεί ότι μπορούμε να ορίσουμε με κάποιες παραμέτρους το πλάτος αυτής της γραμμής το χρώμα της, κλπ.

Η ετικέτα αλλαγής γραμμής είναι η `<br>` (break), που είναι όπως έχει ήδη αναφερθεί, μια κενή ετικέτα. Το στοιχείο αυτό μπορεί να μας επιτρέψει να αλλάξουμε γραμμή μέσα σε ένα στοιχείο μπλοκ. Επίσης να

σημειωθεί ότι ο χαρακτήρας ASCII αλλαγής γραμμής, δεν λαμβάνεται υπόψη από τον συντακτικό αναλυτή της HTML αφού αντικαθίσταται από ένα κενό χαρακτήρα.

Θυμίζουμε ότι ένα στοιχείο μπλοκ όπως είναι ένα `<div>` ή `<p>` ξεκινάει σε μία καινούργια γραμμή.

Όπως είδαμε όμως και στο παράδειγμα του ποιήματος, αν επιθυμούμε να αλλάξουμε γραμμή σε ένα μπλοκ στοιχείο, τότε πρέπει να εισάγουμε ένα στοιχείο

.

### Στοιχεία μορφοποίησης κειμένου `<pre>`, `<blockquote>`

Το στοιχείο `<pre>` που σημαίνει `pre formatted` (προ-μορφοποιημένο κείμενο) ορίζει μία περιοχή κειμένου στην οποία μπορούμε να διατηρήσουμε τους κενούς χαρακτήρες, χαρακτήρες αλλαγής γραμμής, κλπ., καθώς και να εμφανίζεται το κείμενο με χαρακτήρες με σταθερό διάστημα ανά χαρακτήρα (monospace).

Ας δούμε ένα παράδειγμα. Αυτό το τμήμα κειμένου είναι κώδικας Python που πρέπει να διατηρήσουμε τη στοίχιση:

```
<h1>Python code</h1>
<pre>
def add_mult(*x):
    a, m = 0, 1
    for i in x:
        a += i
        m *= i
    return (a, m)
</pre>
```

Το αποτέλεσμα είναι:

# Python code

```
def add_mult(*x):
    a, m = 0, 1
    for i in x:
        a += i
        m *= i
    return (a, m)
```

{#fig:fig13

width=50%}

Το επόμενο ενδιαφέρον στοιχείο είναι το `<blockquote>` όπου στην ουσία είναι μία παράγραφος με αριστερή στοίχιση, η οποία όμως είναι στοιχισμένη πιο δεξιά σε σχέση με το υπόλοιπο κείμενο. Αυτό το στυλ εμφάνισης κειμένου προέρχεται από επιστημονικές δημοσιεύσεις όπου γίνεται verbatim αναφορά ενός άλλου κειμένου, το οποίο βάζουμε σε εισαγωγικά και το παρουσιάζουμε με ένα διαφορετικό τρόπο ώστε να φαίνεται ότι δεν είναι δικό μας, αλλά αναφορά την οποία έχουμε περιλάβει στο κείμενο, και αναφερόμαστε σε αυτό. Αυτό λοιπόν το quoting όπως λέμε την αναφορά σε κάποιον άλλον, χρησιμεύει εδώ για παρουσίαση κειμένου με αυτό τον τρόπο.

Βλέπουμε εδώ ένα παράδειγμα. Ένα απόσπασμα από τον λόγο του Κολοκοτρώνη προς τους μαθητές του Γυμνασίου, το 1838. Επειδή τα λόγια είναι όπως καταγράφηκαν εισάγονται ως στοιχείο `<blockquote>`.

```
<body>
<h1>Ομιλία του Κολοκοτρώνη στην Πνύκα</h1>
<p>Ο γέρος του Μωριά επισκέφθηκε το Βασιλικό Γυμνάσιο της Αθήνας,
    ανέβηκε με τους μαθητές στην Πνύκα και τους είπε τα εξής
    </p>
<blockquote>
    "Όταν αποφασίσαμε να κάμωμε την Έπανάσταση, δεν έσυλλογισθήκαμε
    ούτε πόσοι είμεθα ούτε πώς δεν έχουμε ἄρματα ούτε ότι οι
    Τούρκοι
    έβαστοῦσαν τὰ κάστρα καὶ τὰς πόλεις ούτε κανέναν φρόνιμος μᾶς
    εἶπε «ποῦ πᾶτε ἐδῶ νὰ πολεμήσετε μὲ σιταροκάραβα βατσέλα»,
```

```
        αλλά ως μία βροχή έπεσε εις όλους μας ή επιθυμία της  
        ελευθερίας μας  
    </blockquote>  
</body>
```

Το έγγραφο αυτό στον φυλλομετρητή εμφανίζεται ως εξής:

## Ομιλία του Κολοκτρώνη στην Πνύκα

Ο γέρος του Μωριά επισκέφθηκε το Βασιλικό Γυμνάσιο της Αθήνας, ανέβηκε με τους μαθητές στην Πνύκα και τους είπε τα εξής λόγια:

“Όταν αποφασίσαμε να κάμωμε την Έπανάσταση, δεν έσυλλογισθήκαμε ούτε πόσοι είμεθα ούτε πως δεν έχουμε άρματα ούτε ότι οι Τούρκοι έβαστούσαν τα κάστρα και τας πόλεις ούτε κανέναν φρόνιμος μάς είπε «ποῦ πᾶτε ἐδῶ να πολεμήσετε με σιταροκάραβα βατσέλα», αλλά ως μία βροχή έπεσε εις όλους μας ή επιθυμία της ελευθερίας μας

{#fig:fig14 width=50%}

### Λίστες <ol>, <ul>

Ένα άλλο στοιχείο που έχει ενδιαφέρον και χρειάζεται συχνά να παρουσιάσουμε είναι λίστες από αντικείμενα, λίστες από οδηγίες, λίστες από εικόνες, κλπ.

Για παράδειγμα πριν πάμε στο μανάβικο έχουμε σημειώσει ότι θέλουμε να αγοράσουμε τα εξής φρούτα:

- Μήλα
- Πορτοκάλια
- Μανταρίνια
- Μπανάνες

Σε αυτή τη λίστα έχουμε συμπεριλάβει τέσσερα φρούτα, το ένα κάτω από το άλλο, χωρίς να υπάρχει συγκεκριμένη σειρά (μη ταξινομημένη λίστα - unordered list). Σε αυτού του τύπου τις λίστες χρησιμοποιούμε κάποιο σύμβολο, όπως είναι εδώ η τέλεια, που ξεχωρίζει τα στοιχεία της λίστας.

Πώς ορίζουμε λοιπόν μία λίστα στην HTML; Αν δούμε κώδικα που έχει παράγει αυτήν εδώ την εικόνα, θα διαπιστώσουμε ότι η λίστα ορίζεται ως ένα στοιχείο <ul> (unordered list, μη ταξινομημένη λίστα), δηλαδή μια λίστα χωρίς σειρά. Το στοιχείο <ul> περιέχει τα επί μέρους στοιχεία της λίστας, τα φρούτα, τα οποία ορίζονται ως στοιχεία list items <li> και εδώ φαίνονται τα στοιχεία αυτά.

```
<ul>  
  <li>Μήλα</li>  
  <li>Πορτοκάλια</li>  
  <li>Μανταρίνια</li>  
  <li>Μπανάνες</li>  
</ul>
```

Θα πρέπει να σημειώσουμε ότι ως επί μέρους στοιχείο μιας λίστας μπορεί να είναι και μία άλλη λίστα.

Ας δούμε ένα παράδειγμα. Ας πούμε ότι κάτω από τα πορτοκάλια θέτουμε ένα καινούργιο στοιχείο, το οποίο είναι μία καινούργια λίστα unordered list η οποία περιέχει τα δικά της στοιχεία (συγκεκριμένα είδη πορτοκαλιών, list items).

- Μήλα
- Πορτοκάλια
  - Σαγκουίνια
  - Μέρλιν
  - Βαλένθια
- Μανταρίνια
- Μπανάνες

Αυτό το νέο στοιχείο λοιπόν είναι ένα αντικείμενο που υπάρχει στην αρχική λίστα και περιέχει με τη σειρά του δικά του στοιχεία. Αν θέλουμε να δούμε την καινούργια εικόνα βλέπουμε τώρα ότι έχουμε τα αρχικά στοιχεία και ένα από τα στοιχεία είναι μία υπο-λίστα κάτω από το αντικείμενο αυτό.

Ο κώδικας που παράγει αυτή την εικόνα είναι ο εξής:

```
<ul>
  <li>Μήλα</li>
  <li>Πορτοκάλια
    <ul>
      <li>Σαγκουίνια</li>
      <li>Μέρλιν</li>
      <li>Βαλένθια</li>
    </ul>
  </li>
  <li>Μανταρίνια</li>
  <li>Μπανάνες</li>
</ul>
```

Μια δεύτερη περίπτωση λίστας είναι εκείνη που περιέχει αντικείμενα που έχουν μία ορισμένη σειρά. Έστω ότι θέλουμε να δώσουμε οδηγίες μαγειρικής. Σε αυτή την περίπτωση δημιουργούμε ένα στοιχείο `<ol>`, που σημαίνει ordered list (ταξινομημένη λίστα).

1. Βάζουμε το αυγό στο νερό
2. Βράζουμε για 5'
3. Απομακρύνουμε από τη φωτιά
4. Καθαρίζουμε το αυγό
5. Σερβίρουμε με αλάτι, πιπέρι

Ο κώδικας που παράγει αυτή την αναπαράσταση είναι ο εξής:

```
<ol>
  <li>Βάζουμε το αυγό στο νερό</li>
  <li>Βράζουμε για 5'</li>
  <li>Απομακρύνουμε από τη φωτιά</li>
  <li>Καθαρίζουμε το αυγό</li>
```



```
<li>Σερβίρουμε με αλάτι, πιπέρι</li>
</ol>
```

Αν για κάποιο λόγο επιθυμούμε η αρίθμηση των στοιχείων της λίστας να μην αρχίσει από το 1 αλλά από άλλον αριθμό, το ορίζουμε με το γνώρισμα `start` του στοιχείου `<ol>`.

```
<ol start="5">
```

Επίσης μπορούμε να ορίσουμε αν η φορά αρίθμησης θα είναι αύξουσα ή φθίνουσα. Η τελευταία επιλογή ορίζεται με το γνώρισμα `reversed`.

Για παράδειγμα η εντολή:

```
<ol start="5" reversed>
```

Θα αριθμήσει τα στοιχεία της ταξινομημένης λίστας από το 5 με φθίνουσα σειρά, 4, 3, 2 και τα λοιπά.

## Υπερσύνδεσμοι `<a>`

Σε αυτή την ενότητα θα δούμε πως σε ένα έγγραφο HTML μπορούμε να εισάγουμε υπερσυνδέσμους.

Οι υπερσύνδεσμοι (hyperlinks) είναι μία θεμελιώδης έννοια στο διαδίκτυο και στην HTML ιδιαίτερα. Να λάβουμε υπόψη μας ότι η λέξη *hyper* αποτελεί συστατικό του *hypertext* *hyperlink* που είναι συστατικό του ίδιου του ονόματος της γλώσσας HTML που σημαίνει *hypertext markup language*, επίσης αποτελεί συστατικό του πρωτοκόλλου που χρησιμοποιείται για τη μετάδοση του περιεχομένου HTML που είναι το *HTTP* *hypertext transfer protocol* και ούτω καθεξής. Επίσης να σημειώσουμε ότι *hyper* προέρχεται από την Ελληνική λέξη *υπερ* που ως δασυνόμενη λέξη αρχίζει στα Αγγλικά από γράμμα *h*. Ας δούμε λοιπόν πώς εισάγουμε έναν υπερσύνδεσμο μέσα σε μία ιστοσελίδα.

Ο υπερσύνδεσμος εισάγεται ως στοιχείο `<a>` (anchor, άγκυρα). Είναι μία αγκύρωση ενός στοιχείου του εγγράφου, συνήθως σε ένα άλλο έγγραφο HTML, σε μία άλλη ιστοσελίδα.

Με αυτό τον τρόπο με την εισαγωγή ενός στοιχείου `<a>`, το οποίο περιέχει ως γνώρισμα `href` τη διεύθυνση του πόρου-στόχου, η ιστοσελίδα μας μπορεί να συνδεθεί με εκείνη την ιστοσελίδα αρκεί ο χρήστης να διαλέξει το αντικείμενο που περιέχεται στο στοιχείο `<a>`.

Ας δούμε ένα παράδειγμα.

```
<a href="https://www.upatras.gr">The University of Patras</a>
```

Εδώ έχουμε το στοιχείο `<a>` το οποίο περιέχει το γνώρισμα `href` το οποίο έχει ως τιμή τη διεύθυνση ενός πόρου, μια ιστοσελίδα στόχο. Το περιεχόμενο του στοιχείου είναι το κείμενο "The University of Patras", είναι αυτό που θα δει ο χρήστης και όταν το επιλέξει θα ξεκινήσει ένα αίτημα HTTP για ανάκτηση του πόρου και αν όλα πάνε καλά, θα φορτωθεί η ιστοσελίδα-στόχος με την οποία θα αλληλοεπιδράσει ο

χρήστης. Οι φυλλομετρητές έχουν καθορισμένη μορφοποίηση των υπερσυνδέσμων (πχ χρώμα μπλε με υπογράμμιση), ώστε ο χρήστης να αντιληφθεί άμεσα ότι αυτό το αντικείμενο, αυτό το τμήμα του κειμένου ή αυτή η εικόνα αποτελεί υπερσύνδεσμο και αν επιλεγεί, αν πατηθεί θα οδηγηθεί σε μία άλλη σελίδα.

Ας δούμε μερικές βασικές αρχές που διέπουν τη σχεδίαση υπερσυνδέσμων όταν σχεδιάζουμε τις ιστοσελίδες μας.

*Κανόνας νούμερο 1:* Θα πρέπει να προσέξουμε ιδιαίτερα το κείμενο του υπερσυνδέσμου. Το κείμενο του υπερσυνδέσμου πρέπει να είναι περιγραφικό, δηλαδή να περιγράφει το στόχο γιατί είναι βασική πληροφορία που θα οδηγήσει τον χρήστη να επιλέξει έναν υπερσύνδεσμο έναντι κάποιου άλλου και να αποφεύγουμε κοινοτυπίες τύπου πατήστε εδώ εκεί κλπ. Ας δούμε ένα παράδειγμα. Εδώ έχω δύο υπερσυνδέσμους: (α) *Πανεπιστήμιο Πατρών* και (β) *Πατήστε εδώ για το Πανεπιστήμιο Πατρών*. Ποια είναι η καλύτερη επιλογή; Η πρώτη είναι πιο περιγραφική και σαφής, η υπόδειξη "πατήστε εδώ" δεν χρειάζεται, αφού αυτό προκύπτει από την εμφάνιση του υπερσυνδέσμου.

*Κανόνας νούμερο 2:* Θα πρέπει να παρέχονται πληροφορίες στον χρήστη για τη συνέπεια επιλογής υπερσυνδέσμου, για παράδειγμα αν πατώντας έναν υπερσύνδεσμο θα κατεβάσει ένα αρχείο, θα χρειάζεται ένα plug-in, θα πρόκειται να οδηγηθεί σε ένα νέο παράθυρο κλπ. Ακολουθούν σχετικά παραδείγματα. Αν ο υπερσύνδεσμος download Report συνεπάγεται κατέβασμα ενός αρχείου τύπου pdf μέγεθος 10 Mb, αυτή είναι μία χρήσιμη πληροφορία για να ξέρει ο χρήστης αν πρέπει να επιλέξει αυτό τον υπερσύνδεσμο. Το ίδιο αν ο υπερσύνδεσμος περιγράφεται με τις λέξεις "Δες το βίντεο" θα ενεργοποιήσει μια ροή βίντεο HD quality σε ένα καινούργιο παράθυρο.

*Κανόνας νούμερο 3:* Δεν είναι σωστό να χρησιμοποιούμε την ίδια τη διεύθυνση του πόρου ως υπερσύνδεσμο. Αυτή είναι μια κακή πρακτική, αφού η ανάγνωση του URL ακούγεται ιδιαίτερα άσχημα.

*Κανόνας νούμερο 4:* Το κείμενο του υπερσυνδέσμου δεν θα πρέπει να είναι ιδιαίτερα μεγάλο. Σκεφτείτε αν για ένα άτομο με ειδικές ανάγκες το οποίο θα ακούσει τον υπερσύνδεσμο πρέπει ο screenreader να διαβάσει ένα μακρύ κείμενο που αποτελεί το κείμενο του υπερσυνδέσμου. Τέλος σκεφτείτε ότι αν χρησιμοποιήσετε την ίδια κοινότυπη λέξη όπως "πατήστε εδώ" για μία σειρά από υπερσυνδέσμους σε μία σελίδα, τότε θα είναι δύσκολο για ένα άτομο με ειδικές ανάγκες είτε έναν απλό χρήστη να ξεχωρίσει τους υπερσυνδέσμους.

Γνωρίσματα της ετικέτας `<a>`. Το πρώτο γνώρισμα είναι το γνώρισμα `href` που σημαίνει *hyperlink reference* που ορίζει, τη διεύθυνση του στόχου. Έχουμε ήδη δει μερικά παραδείγματα. Ένα δεύτερο βασικό γνώρισμα που είναι χρήσιμο είναι το γνώρισμα `title`. Είναι προαιρετικό γνώρισμα το οποίο περιέχει ένα κείμενο περιγραφής του στόχου το οποίο εμφανίζεται ως tooltip, ένα μικρό αναδυόμενο κείμενο όταν με τη δεικτική συσκευή ο χρήστης βρεθεί πάνω από τον υπερσύνδεσμο.

Ένα παράδειγμα:

```
<a href="https://www.upatras.gr" title="Το τρίτο Ελληνικό  
Πανεπιστήμιο">Πανεπιστήμιο Πατρών</a>.
```

Όταν ο χρήστης περάσει με το ποντίκι πάνω από τον υπερσύνδεσμο βλέπει την εξής πρόσθετη πληροφορία:

# Πανεπιστήμιο Πατρών.

Το τρίτο Ελληνικό Πανεπιστήμιο

{#fig:fig15 width=50%}

Ένα άλλο γνώρισμα που είναι χρήσιμο είναι το γνώρισμα `target`. Αυτό ορίζει πώς θα συμπεριφερθεί ο φυλλομετρητής όταν κατέβει η νέα ιστοσελίδα. Για παράδειγμα όταν πάρει την τιμή `_blank` τότε ο πόρος στον οποίο απευθυνόμαστε θα ανοιχτεί σε ένα νέο παράθυρο ή σε νέα καρτέλα του φυλλομετρητή. Η τιμή `_self` που είναι η εξ ορισμού τιμή, φορτώνει τον πόρο, στο ίδιο παράθυρο.

Αλλα γνώρισμα πιο εξειδικευμένα είναι το γνώρισμα `download`. Αυτό ορίζει ότι η σύνδεση αυτή μας οδηγεί στο να κατεβάσουμε ένα πόρο, ένα αρχείο συνήθως.

Ένα παράδειγμα:

```
<a href="www.example.com" download="manual.pdf"> Κατεβάστε το εγχειρίδιο  
(pdf) </a>
```

Εδώ ορίζουμε ότι αν επιλεγεί αυτός ο υπερσύνδεσμος απλά θα κατέβει το αρχείο `manual.pdf`.

Μία άλλη περίπτωση υπερσυνδέσμου είναι υπερσύνδεσμος που δεν έχει σα στόχο μία άλλη ιστοσελίδα αλλά έχει ως στόχο τον τοπικό client αποστολής ηλεκτρονικού ταχυδρομείου. Όταν επιλέξει ο χρήστης αυτόν τον υπερσύνδεσμο τότε χρησιμοποιώντας το πρωτόκολλο `mailto`, αναζητάται ο τοπικός email client και αναδύεται ένα παράθυρο του e-mail client για να αποστείλουμε ένα ηλεκτρονικό μήνυμα στην διεύθυνση στο στόχο που ορίζεται από τον υπερσύνδεσμο.

Ένα παράδειγμα:

```
<a href="mailto:announces@ece.upatras.gr"> μήνυμα στο Τμήμα</a>
```

Μία άλλη ειδική περίπτωση που αξίζει να αναφέρουμε είναι ο υπερσύνδεσμος να αναφέρεται σε ένα σημείο όχι σε μία άλλη ιστοσελίδα αλλά σε κάποιο άλλο σημείο στην ίδια σελίδα.

Υπάρχουν διάφοροι τρόποι για να γίνει αυτό. Ας δούμε ένα παράδειγμα. Βλέπουμε εδώ ότι ορίζουμε μία αναφερόμαστε σε έναν υπερσύνδεσμο που ορίζεται ως `#locationA` (υπενθυμίζουμε ότι στο URL το σύμβολο `#` ορίζει ένα επί μέρους στοιχείο ενός πόρου, αν η διεύθυνση αρχίζει με `#` εννοείται ότι αποτελεί επί μέρους στοιχείου της ίδιας της σελίδας ).

```
<a href="#locationA ">Πηγαίνετε στο A</a>
```

Πρέπει όμως κάπου στην ιστοσελίδα να υπάρχει αυτός ο στόχος. Ένας τρόπος για να ορίσουμε αυτόν το στόχο είναι να υπάρχει ένα στοιχείο `<a>` που έχει ως όνομα, με το γνώρισμα `name`, ακριβώς αυτό το όνομα, το `locationA`.

```
<a name= "locationA"> το σημείο A είναι εδώ</a>
```

Μία άλλη μέθοδος είναι να διασυνδεθούμε με ένα στοιχείο το οποίο έχει το όνομα της αγκύρωσης ως ιδιότητα `id` του στοιχείου αυτού. Ας δούμε ένα παράδειγμα.

Εάν υπάρχει ένα στοιχείο που έχει ως ταυτότητα `id=mailing_address`, βλέπουμε εδώ ένα στοιχείο, είναι ένα στοιχείο `h2`.

```
<h2 id="Mailing_address">Ηλεκτρονικό ταχυδρομείο</h2>
```

Σε αυτό το στοιχείο μπορώ να αναφερθώ ως εξής:

```
<p>Η διεύθυνση <a href="#Mailing_address"> email της εταιρίας μας</a>  
βρίσκεται στο τέλος της σελίδας.</p>
```

Στη συνέχεια ας εξετάσουμε το URL που εμφανίζεται ως τιμή στο γνώρισμα `href` ενός στοιχείου `<a>`. Υπάρχουν διευθύνσεις που είναι απόλυτες διευθύνσεις, που αρχίζουν από το πρωτόκολλο `https://` και ορίζουν μία πλήρη διεύθυνση πόρου ή είναι σχετικές διευθύνσεις, τότε έχουμε μόνο το όνομα του αρχείου αλλά αυτό σημαίνει ότι το αρχείο βρίσκεται στον ίδιο φάκελο που βρίσκεται η ιστοσελίδα. Εάν ορίσουμε μόνο κάποιο υπο-φάκελο όπως για παράδειγμα `/a/b/` τότε ο web server θα αναζητήσει εκεί κάποια από τα προκαθορισμένα ονόματα αρχείων. Το πιο συνηθισμένο είναι το αρχείο `index.html`. Μπορεί όμως να υπάρχουν και άλλα ονόματα όπως `default.html`, κλπ. Εάν δεν υπάρχει το προκαθορισμένο αρχείο, δηλαδή ένα από τα παραπάνω, τότε μπορεί να μας επιστρέψει, αν αυτό επιτρέπεται, τον κατάλογο των αρχείων που βρίσκονται σε αυτόν τον υπο-φάκελο. Τέλος αν δεν υπάρχει κάποιο αρχείο `b` στον υπο-φάκελο `a` τότε μπορεί να μας επιστρέψει τον κατάλογο `a` εφόσον επιτρέπεται. Σε αυτή την περίπτωση το μήνυμα `http` που θα μας επιστρέψει τον κατάλογο θα έχει τον κωδικό 301, γιατί έχουμε την περίπτωση επανακατεύθυνσης σε διαφορετικό πόρο.

## Πίνακες (`<table>`)

Σε αυτή την ενότητα θα δούμε το στοιχείο `<table>` που μας επιτρέπει να δημιουργήσουμε πίνακες, διδιάστατους υποδοχείς δεδομένων, αντίστοιχους με ένα φύλλο Excel, που περιέχουν τα δεδομένα σε κελιά.

Για να ορίσουμε ένα πίνακα στην HTML χρειαζόμαστε ένα συνδυασμό από στοιχεία που είναι παιδιά του στοιχείου `<table>`. Ως παιδιά του πίνακα ορίζουμε τα στοιχεία `<tr>` (table rows), που αφορούν στις γραμμές του πίνακα. Κάθε γραμμή περιέχει με τη σειρά της κελιά. Τα κελιά ορίζονται ως στοιχεία `<td>` δηλαδή table data.

Ας δούμε ένα παράδειγμα. Έστω ότι επιθυμούμε να δημιουργήσουμε ένα πίνακα δύο γραμμών και δύο στηλών, (με 4 κελιά). Αυτό γίνεται με τον παρακάτω κώδικα.

```
<table>
  <tr>
    <td> κελί11 </td>
    <td> κελί12 </td>
  </tr>
  <tr>
    <td> κελί21 </td>
    <td> κελί22 </td>
  </tr>
</table>
```

Παρατηρούμε εδώ λοιπόν στην πρώτη γραμμή `<tr>` περιέχονται δύο κελιά `<td>`, με περιεχόμενο *κελί11* και *κελί12*. Στη συνέχεια έχουμε μία δεύτερη γραμμή που πάλι ορίζεται από ένα στοιχείο `<tr>` που περιέχει επίσης δύο κελιά με περιεχόμενο *κελί21* και *κελί22*. Το πλαίσιο γύρω από κάθε κελί και το συνολικό πλαίσιο γύρω από τον πίνακα, καθορίζονται από φύλλα στυλ, όπως θα δούμε στη συνέχεια.

Ένα στοιχείο `<table>` δέχεται επίσης έναν άλλο τύπο κελιών αντί για τα `<td>`, που είναι τα στοιχεία `<th>`, (table header). Είναι τα κελιά της κεφαλίδας του πίνακα, που συνήθως υπάρχουν στην πρώτη σειρά (κελιά που περιγράφουν το περιεχόμενο των αντίστοιχων στηλών), και έχουν πιο έντονη εμφάνιση. Επίσης ο πίνακας, δέχεται ένα ακόμη στοιχείο, το `<caption>` το οποίο αφορά τη λεζάντα περιγραφής του πίνακα.

Ας δούμε ένα παράδειγμα. Έστω ότι θέλουμε να δημιουργήσουμε ένα πίνακα με τα στοιχεία των τριών ψηλότερων Ελληνικών βουνών, που περιέχει τρεις πληροφορίες για κάθε βουνό, το όνομά του, το ύψος του, και την περιοχή που βρίσκεται.

Η τελική εμφάνιση του πίνακα φαίνεται στη συνέχεια

## Greek Mountains

Mountain	Height (m)	Region
Olympus	2,917	Pieria
Smolikas	2,637	Ioannina
Kaimaktsalan	2,524	Florina

{#fig:fig16 width=50%}

Ο πίνακας περιλαμβάνει 4 γραμμές, την πρώτη που είναι η κεφαλίδα με τα ονόματα των στηλών, και 3 γραμμές με δεδομένα των βουνών. Επίσης στο πάνω μέρος περιλαμβάνει τη λεζάντα.

Ο κώδικας που παράγει τον παραπάνω πίνακα είναι ο εξής:

```
<table>
  <caption>Greek Mountains</caption>
  <tr>
    <th>Mountain</th>
    <th>Height (m) </th>
    <th>Region</th>
  </tr>
  <tr>
    <td> Olympus </td>
    <td> 2,917 </td>
    <td> Pieria </td>
  </tr>
  <tr>
    <td> Smolikas </td>
    <td> 2,637 </td>
    <td> Ioannina </td>
  </tr>
  <tr>
    <td> Kaimaktsalan </td>
    <td> 2,524 </td>
    <td> Florina </td>
  </tr>
</table>
```

Παρατηρούμε ότι οι κεφαλίδες εμφανίζονται με έντονους χαρακτήρες και με κεντρική στοίχιση. Βλέπω εδώ δηλαδή τα *Mountain*, *Height*, *Region* είναι με έντονα γράμματα και στο κέντρο των αντίστοιχων στηλών.

Ένα άλλο ενδιαφέρον θέμα είναι, σε έναν πίνακα να πρέπει κάποια κελιά να εκτείνονται σε περισσότερες από μία στήλες ή επίσης να εκτείνονται σε περισσότερες από μία γραμμές.

Ας δούμε ένα παράδειγμα. Ας πούμε ότι θέλουμε να φτιάξουμε τον παρακάτω πίνακα με στοιχεία από τον πληθυσμό κάποιας χώρας.



Πληθυσμός		
Ηπειρωτική χώρα	Άνδρες	3000000
	Γυναίκες	3300000
Νησιωτική χώρα	Άνδρες	1200000
	Γυναίκες	1300000

{#fig:fig17 width=50%}

Παρατηρούμε ότι το πρώτο κελί εκτείνεται και στις τρεις στήλες του πίνακα. Στη συνέχεια υπάρχει στην πρώτη στήλη ένα κελί που περιέχει τις λέξεις "Ηπειρωτική χώρα" που εκτείνεται σε δύο γραμμές, καθώς και στη συνέχεια το κελί "Νησιωτική χώρα" που επίσης εκτείνεται σε δύο γραμμές. Συνολικά ο πίνακας περιλαμβάνει πέντε γραμμές, και τρεις στήλες αλλά μόνο 11 κελιά. Πώς γίνεται αυτό με χρήση των ετικετών της HTML και των γνωρισμάτων των ετικετών αυτών;

Όπως θα δούμε θα χρησιμοποιήσουμε το γνώρισμα `colspan` που επιτρέπει την έκταση του αντίστοιχου κελιού σε πολλές στήλες, και του γνωρίσματος `rowspan` που επιτρέπει την επέκταση του αντίστοιχου κελιού σε πολλές γραμμές.

Ο παρακάτω κώδικας παράγει αυτό το αποτέλεσμα.

```
<table>
  <tr >
    <td colspan="3">Πληθυσμός</td>
  </tr>
  <tr>
    <td rowspan="2">Ηπειρωτική χώρα</td>
    <td>Άνδρες</td>
    <td>3000000</td>
  </tr>
  <tr>
    <td>Γυναίκες</td>
    <td>3300000</td>
  </tr>
  <tr>
    <td rowspan="2">Νησιωτική χώρα</td>
    <td>Άνδρες</td>
    <td>1200000</td>
```

```
</tr>
<tr>
  <td>Γυναίκες</td>
  <td>1300000</td>
</tr>
</table>
```

Όπως βλέπουμε θέτοντας το γνώρισμα `colspan="3"` στο πρώτο στοιχείο, έχει ως συνέπεια το κελί να εκταθεί σε τρεις στήλες. Το ίδιο με το πρώτο στοιχείο της δεύτερης γραμμής που παίρνει το γνώρισμα `rowspan="2"` με αποτέλεσμα αυτό να καλύπτει δύο γραμμές. Με αυτόν τον τρόπο λοιπόν έχω κελιά που εκτείνονται είτε οριζόντια καλύπτοντας πολλές στήλες ή κατακόρυφα καλύπτοντας πολλές γραμμές κάτι που πολύ συχνά χρειαζομαι και το βλέπω και σε φύλλα Excel κλπ.

Κάποια ακόμη στοιχεία που χρησιμοποιούνται σε πίνακες είναι τα στοιχεία `<thead>` (table head) για την κεφαλίδα του πίνακα, `<tfoot>` (table foot) για το υποσέλιδο του πίνακα, και `<tbody>` (table body) για το σώμα του πίνακα. Τα στοιχεία αυτά είναι δομικά στοιχεία του πίνακα. Αυτά μπορεί να συνδυαστούν με στυλ έτσι ώστε αυτά τα τμήματα του πίνακα να έχουν ορισμένο στυλ εμφάνισης. Επίσης μπορεί να επαναλαμβάνονται. Παραδείγματος χάρη δομικά στοιχεία που μπορεί να επαναλαμβάνονται είναι ένας πίνακας που εκτείνεται σε πολλές σελίδες και σε κάθε σελίδα να έχει κεφαλίδα και υποσέλιδο που να μας υπενθυμίζει το περιεχόμενό του.

Θα κάνουμε στη συνέχεια μια σύντομη αναφορά στον τρόπο που ορίζουμε το στυλ εμφάνισης των στοιχείων ενός πίνακα. Περισσότερες λεπτομέρειες για την ακριβή σύνταξη των εντολών CSS αποτελούν αντικείμενο επόμενου κεφαλαίου.

παράδειγμα 1.

```
table, th, td {border: 2px solid black;}
```

Στην εντολή αυτή ορίζουμε τα στοιχεία `table`, `th` και `td`, που είναι στοιχεία που ορίζουν τον πίνακα και τα κελιά του. Για τα στοιχεία αυτά, ο κανόνας ορίζει ότι οι το πλαίσιό τους είναι πλάτους 2 Pixel, με συνεχή γραμμή μαύρου χρώματος, άρα κάθε κελί θα έχει το δικό του πλαίσιο.

παράδειγμα 2.

```
th, td {padding: 10px;}
```

Η εντολή αυτή ορίζει ότι τα στοιχεία `th` και `td` θα έχουν εσωτερικό περιθώριο μεταξύ των δεδομένων και του πλαισίου του αντίστοιχου κελιού, 10 pixel.

παράδειγμα 3.

```
table {border-spacing: 5px;}
```

Η απόσταση μεταξύ κελιών ορίζεται από την παράμετρο `border-spacing` που εδώ ορίζεται να είναι 5px  
παράδειγμα 4.

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

Εδώ ορίζουμε τα κελιά της κεφαλίδας να στοιχηθούν αριστερά, όπως και τα κελιά δεδομένων, η παράμετρος `text-align` μας επιτρέπει να στοιχίσουμε το περιεχόμενο των κελιών δεξιά, αριστερά ή στο κέντρο, ενώ υπάρχει και η `vertical-align` η οποία μάς επιτρέπει να στοιχίσουμε πάνω, κάτω ή στο κέντρο το περιεχόμενο των κελιών κατά την κατακόρυφη διάσταση.

Θα πρέπει να κλείσουμε αυτή την ενότητα με ένα σχόλιο για τη χρήση του στοιχείου `<table>`. Το στοιχείο αυτό χρησιμοποιείται από κάποιους σχεδιαστές εκτός από δημιουργία πινάκων, και για διάταξη του περιεχομένου της σελίδας. Η σελίδα μπορεί να οριστεί ως ένας πίνακας και τα κελιά να περιέχουν τα επί μέρους στοιχεία της σελίδας, εικόνες, κείμενα κλπ. Αυτή είναι όμως μία πολύ κακή πρακτική και θα πρέπει να αποφεύγεται. Υπάρχουν πολλοί σοβαροί λόγοι για τους οποίους το `table` δεν πρέπει να χρησιμοποιείται για διάταξη περιεχομένου. Διάταξη περιεχομένου όπως θα δούμε στα επόμενα κεφάλαια μπορεί να γίνει χρησιμοποιώντας μηχανισμούς της CSS. Μερικοί λόγοι για τους οποίους αυτή η πρακτική πρέπει να αποφεύγεται είναι οι εξής:

- Τα στοιχεία `<table>` μειώνουν την προσβασιμότητα για χρήστες με προβλήματα όρασης. Δεδομένου ότι ο ορισμός στυλ πινάκων είναι πιο περίπλοκος από ότι με τις τεχνικές διάταξης μέσω CSS, με αποτέλεσμα σύγχυση των screenreader.
- Οι πίνακες είναι πιο δύσκολο να συντηρηθούν, αφού οι διατάξεις των στοιχείων πίνακα είναι σύνθετες.
- Οι πίνακες δεν προσαρμόζονται σε διαφορετικές συσκευές, γιατί έχουν σταθερό μέγεθος που ορίζεται από το περιεχόμενό τους.

Για όλους αυτούς τους λόγους οι πίνακες θα πρέπει να χρησιμοποιούνται μόνο για αυτό το οποίο δημιουργήθηκαν, δηλαδή για την οργανωμένη παρουσίαση δεδομένων.

## 2.3 Ασκήσεις

### Άσκηση 1

Να δημιουργήσετε μια σελίδα που περιέχει το βιογραφικό σας.

Στο πάνω μέρος να υπάρχει ένας πίνακας περιεχομένων με συνδέσμους στις επί μέρους ενότητες του βιογραφικού, να περιλάβετε ενότητες για προσωπικά στοιχεία, σπουδές, εργασιακή εμπειρία, ενδιαφέροντα. Να περιλάβετε υπερσυνδέσμους σε σελίδες που σας αρέσουν.

### Άσκηση 2

Σπονδυλωτά	Πτηνά		Κότα
	Ερπετά		Σαύρα
	Αμφίβια		Βάτραχος
	Ψάρια		Καρχαρίας
	Θηλαστικά	Τρωκτικά	Ποντίκι
		Προβοσκιδωτά	Ελέφαντας
		Πρωτεύοντα	Πίθηκος
		Κητώδη	Δελφίνι
		Σαρκοφάγα	Λιοντάρι
		Οπληφόρα	Άλογο
		Χειρόπτερα	Νυχτερίδα
Ασπόνδυλα	Μαλάκια		Καλαμάρι
	Εχινόδερμα		Αχινός
	Αρθρόποδα		Καβούρι
	Σπόγγοι		Σφουγγάρι
	Κνιδόζωα		Τσούχτρα
	Σκώληκες		Σκουλήκι

{#fig:fig18 width=50%}

Έστω ο πίνακας της εικόνας @fig:fig18. Ζητείται:

1. Να δημιουργήσετε ιστοσελίδα με τον πίνακα
2. Να αλλάξετε την εμφάνιση αυξάνοντας το πλάτος του πλαισίου των κελίων σε 4px.
3. Να προσθέστε λεζάντα και υποσέλιδο.