

---

# Γλώσσες, Εργαλεία και Αρχές Ανάπτυξης για τον Παγκόσμιο Ιστό

---



Τμήμα Αρχειονομίας, Βιβλιοθηκονομίας και Συστημάτων Πληροφόρησης

Μάρτιος 2023

## Αναθεωρήσεις Εγγράφου

Έκδοση	Ημερομηνία	Παρατηρήσεις
<b>0.9.9.3</b>	5/4/2020	Εργαστηριακές σημειώσεις HTML5
<b>0.9.9.4</b>	5/4/2021	Αλλαγές στα παραδείγματα (url)
<b>0.9.9.5</b>	19/3/2023	Αλλαγή στις ασκήσεις
<b>0.9.9.6</b>	31/3/2023	Επικαιροποίηση λίστας link [attribute]
<b>0.9.9.7</b>	28/4/2023	Επικαιροποίηση ασκήσεων 10,11,12,13

## Περιεχόμενα

1. Εισαγωγή.....	5
1.1 Προαπαιτούμενα.....	6
1.2 Εργαλεία Συγγραφής.....	7
1.3 Αρχές για Προσβάσιμο και Ενιαίο Σχεδιασμό .....	8
2. HTML (Hypertext Markup Language).....	10
2.1 Τυπική Δομή ενός HTML αρχείου .....	10
2.2 Στοιχεία (Elements) & Ετικέτες (Tags).....	12
2.3 Χαρακτηριστικά (attributes).....	13
2.4 <!DOCTYPE> .....	14
2.5 <head> : Μεταδεδομένα Εγγράφου .....	15
2.5.1 Το στοιχείο <title> .....	15
2.5.2 Το στοιχείο <meta> .....	15
2.5.3 Το στοιχείο <link>.....	16
2.5.4 Το στοιχείο <base> .....	17
2.6 Μοντέλα Περιεχομένου (Content Models).....	18
2.7 <body> : Το πραγματικό περιεχόμενο του εγγράφου .....	21
2.7.1 Επικεφαλίδες <h1>...<h6>.....	22
2.7.2 Παράγραφος <p> .....	23
2.7.3 Αλλαγή Γραμμής   .....	25
2.7.4 Οριζόντια Γραμμή <hr> .....	25
2.7.5 Προκαθορισμένη μορφοποίηση περιεχομένου <pre> .....	26
2.7.6 Απόσπασμα/Παράθεση <q> ή <blockquote> .....	27
2.7.7 Σχόλια .....	27
2.7.8 Διαμόρφωση Χαρακτήρων.....	28
2.7.9 Τα στοιχεία <span> και <div>.....	34
2.7.10 Χαρακτήρες Διαφυγής (Named Character Entities).....	36
2.7.11 Λίστες <ul>, <ol>.....	37
2.7.12 Μονοπάτια-Διαδρομές (Paths) & URLs.....	40
2.7.13 Υπερσύνδεσμοι <a>.....	42
2.7.14 Εικόνες – Γραφικά <img> .....	44
2.7.15 Πίνακες <table>.....	47

2.8	Δομή Ιστοσελίδας.....	58
2.8.1	Η δομή μιας τυπικής ιστοσελίδας.....	59
2.8.2	Δομή με Πίνακες .....	59
2.8.3	Δομή με <div>...</div> .....	61
2.8.4	Δομή με Σημασιολογικά Στοιχεία της HTML5.....	63
3.	Παράρτημα: Συστήματα Ελέγχου Εκδόσεων Κώδικα (VCS) .....	73
3.1	Εισαγωγή .....	73
3.2	Γενικά για το Git .....	74
3.3	Μοντέλο (Συν)εργασίας .....	74
3.3.1	Σενάριο 1: Τοπικά χωρίς τη χρήση λογαριασμού Cloud.....	75
3.3.2	Σενάριο 2: Συνεργασία σε ομάδα με χρήση GitHub.com .....	75
3.4	Λογισμικά για τη Διαχείριση Αποθετηρίων .....	76
3.4.1	Δημιουργία Λογαριασμού στο GitHub .....	76
3.4.2	Δημιουργία Project – Τοπικό Αποθετήριο .....	79
3.4.3	GitHub Desktop .....	80
3.4.4	Git Tools .....	84
3.4.5	Brackets + Git plugin.....	85

## 1. Εισαγωγή

Στις 12.3.2019 γιορτάσαμε τα 30<sup>α</sup> γενέθλια του WWW (World Wide Web). Πολλοί συγχέουν το Web με το Internet, με το δεύτερο να αποτελεί το μέσο, όπως για παράδειγμα η τηλεφωνική γραμμή αποτελεί το μέσο για την πραγματοποίηση τηλεφωνικών συνομιλιών.



Ο εφευρέτης του WWW, Sir Tim Berners-Lee, ξεκίνησε ένα ταξίδι προτείνοντας ένα πρωτότυπο ([ENQUIRE](#)) το οποίο θα επέτρεπε σε επιστήμονες του CERN (European Lab for Particle Physics, Geneva-Switzerland) να συνεργάζονται μεταξύ τους όπου και αν βρίσκονταν. Η αρχική ιδέα ήταν να επιτρέπει στους ερευνητές, από απομακρυσμένες τοποθεσίες, να δημιουργούν και να οργανώνουν περιεχόμενο σχετικά με την έρευνά τους, χωρίς αυτό να σημαίνει αποκλειστικά την τοπική καταγραφή, δημιουργία και αποστολή μεγάλων εγγράφων. Με άλλα λόγια προσπάθησε να επιλύσει ζητήματα που αντιμετώπιζε ένας ερευνητής όπως:

- Αναγκαίο κατέβασμα μεγάλων αρχείων για πρόσβαση μόνο σε ένα μέρος αυτών με επιπτώσεις που σχετίζονται με την ταχύτητα του δικτύου και τη διαθεσιμότητα αποθηκευτικού χώρου.
- Οργάνωση και διασύνδεση περιεχομένου (cross-reference) ώστε να μπορούν να συσχετίζονται μέρη της έρευνας που έχουν καταγραφεί ανεξάρτητα από το σημείο εκκίνησής τους, δίνοντας δυνατότητα για εξερεύνηση νέων σεναρίων άρα περισσότερη έρευνα.
- Αποτελεσματική συνεργασία μεταξύ των ερευνητών που χρησιμοποιούν διαφορετικές τεχνολογίες δικτύων, μορφότυπους δεδομένων, κωδικοποιήσεις δεδομένων κτλ.

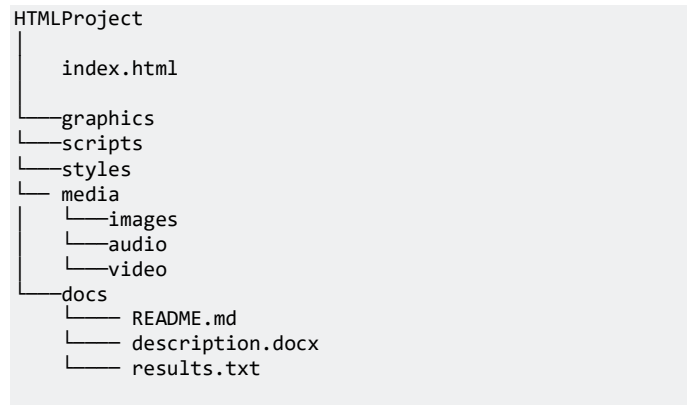
Ξεκινώντας με ένα μικρό σύνολο «σημάνσεων» (markup tags) και βασιζόμενη στην SGML (Standard Generalized Markup Language), η HTML αποτέλεσε μια μέθοδο επισήμειωσης/σήμανσης κειμένου με *σημασιολογικές μονάδες* (structural units) όπως π.χ. τίτλους, παραγράφους, επικεφαλίδες, λίστες κτλ. Τελικά κατέληξε να έχει ξεπεράσει κατά πολύ τις αρχικές προσδοκίες για την κλίμακα αποδοχής και εφαρμογής, έχοντας δημιουργήσει ένα πολύπλευρο πλέγμα σήμανσης που καλύπτει όλες τις δραστηριότητες και μέσα που δημιουργούν περιεχόμενο όπως π.χ. εικόνα, ήχο, κίνηση, μαθηματικές παραστάσεις, καμβά κτλ.

Εν τέλει, το πιο σημαντικό από τις μονάδες - και η πραγματική εφεύρεση του Sir Tim Berners-Lee - είναι η δημιουργία της οντότητας “*href*” η οποία επιτρέπει τη διασύνδεση αντικειμένων και τη δημιουργία ενός “*Web of Information*” (<https://www.w3.org/People/Raggett/book4/ch02.html>).

## 1.1 Προαπαιτούμενα

Η διαδικασία δημιουργίας ενός site (ιστοσελίδας) πέρα από το «δημιουργικό κομμάτι» περιλαμβάνει τη διαχείριση πολλαπλών αρχείων, όπως τα αρχεία HTML, αρχεία μορφοποίησης στυλ (stylesheets), αρχεία κώδικα (JavaScript) και αρχεία πολυμέσων (εικόνες, ήχοι, video κτλ.). Η διαχείριση των αρχείων αυτών απαιτεί την **οργάνωση τους σε μια δομή** η οποία θα προσφέρει ευκολία στον συγγραφέα να αναφέρεται σε αυτά μέσα από τον κώδικα. Στα επόμενα σημεία περιγράφονται καλές πρακτικές για την οργάνωση του υλικού και των πόρων που θα χρησιμοποιηθούν στην κατασκευή μιας ιστοσελίδας ([https://developer.mozilla.org/en-US/docs/Learn/Getting\\_started\\_with\\_the\\_web/Dealing\\_with\\_files](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/Dealing_with_files)):

1. Πριν ξεκινήσετε τη συγγραφή επιλέξτε έναν κατάλογο (π.χ. MyWeb) για να αποθηκεύσετε τις εργασίες σας ή τις προσωπικές σας σελίδες π.χ. το online βιογραφικό σας διατηρώντας την πρακτική των projects (κάθε project σε ξεχωριστό κατάλογο).
2. Τα ονόματα των αρχείων που θα δημιουργήσετε θα πρέπει να ακολουθούν την ίδια «πολιτική» ονοματοδοσίας και κυρίως σε ότι αφορά τη χρήση **κεφαλαίων/πεζών** μιας και αυτό διευκολύνει τη διασύνδεσή τους ελαχιστοποιώντας τα σφάλματα. Προτιμήστε τα ονόματα με πεζά γράμματα αποφεύγοντας τη χρήση κενών. Στην περίπτωση που χρειάζεται να διαχωρίσετε λέξεις στο όνομα του αρχείου, επιλέξτε την άνω παύλα “-” (dash) και όχι την κάτω “\_” (underscore) διότι η μηχανή αναζήτηση της Google χρησιμοποιεί την άνω παύλα ως διαχωριστή λέξεων ενώ την κάτω όχι.
3. Σχετικά με την οργάνωση των αρχείων σε καταλόγους μια καλή πρακτική αποτελεί η δημιουργία ενός ευρετηρίου `index.html` στον κεντρικό κατάλογο του ιστότοπου (website) και επιμέρους καταλόγους για κάθε τύπο αρχείων/πόρων που θα χρησιμοποιηθούν. Πιο συγκεκριμένα:
  - a. Το αρχείο **index.html** αποτελεί την πρώτη σελίδα του ιστότοπου (website) και διασυνδέει όλο το υπόλοιπο περιεχόμενο. Συμπληρωματικά, σε ένα κατάλογο /docs, μπορείτε να δημιουργήσετε αρχεία που προσφέρουν μια περιγραφή του project, των αναγκών βιβλιοθηκών, των παραμέτρων εγκατάστασης και των αποτελεσμάτων που έχουν παραχθεί.
  - b. Τα **αρχεία γραφικών-εικόνων** πρέπει να βρίσκονται σε ξεχωριστό κατάλογο με τυπικά ονόματα καταλόγων `images` ή `graphics`. Στην περίπτωση που η ιστοσελίδα διαχειρίζεται περιεχόμενο στην μορφή εικόνων/γραφικών θα πρέπει να διαχωρίζονται από τα διακοσμητικά γραφικά της σχεδίασης. Έτσι προτιμάμε τα γραφικά που ορίζουν τη σχεδίαση της σελίδας να βρίσκονται σε έναν κατάλογο “graphics”, ενώ όλα τα αρχεία πολυμέσων (εικόνες, ήχοι, βίντεο) σε έναν κεντρικό κατάλογο “media” με υποκαταλόγους “images”, “audio”, “video” κτλ.
  - c. Τα **αρχεία CSS** που περιέχουν τους κανόνες και στυλ μορφοποίησης θα πρέπει να αποθηκεύονται σε ξεχωριστό κατάλογο “styles”.
  - d. Τα **αρχεία JavaScript** που αφορούν προγραμματιστικές βιβλιοθήκες θα πρέπει να αποθηκεύονται σε ξεχωριστό κατάλογο “scripts”.



Εικόνα 1.1. Τυπική οργάνωση αρχείων ενός project

4. Όσο η ανάπτυξη της ιστοσελίδας εξελίσσεται και το περιεχόμενο εκτείνεται σε διαφορετικά αρχεία, ενδέχεται αυτά να καταλήγουν σε καταλόγους με διαφορετική οργάνωση και βάθος. Καθένα από τα αρχεία περιεχομένου (HTML) θα πρέπει να έχει «πρόσβαση» στα κοινόχρηστα αρχεία πόρων (CSS/JavaScript), άρα να γνωρίζει τη **διαδρομή (path)** με την οποία θα αναφέρεται σε αυτά. Το ίδιο ισχύει και στην περίπτωση που ένα αρχείο HTML πρέπει να αναφερθεί σε ένα άλλο (μέσω link) όταν αυτό βρίσκεται σε διαφορετική δομή. Η διαδικασία δεν είναι διαφορετική από αυτή που ακολουθούμε στον προσωπικό μας υπολογιστή για να εντοπίσουμε τα προσωπικά μας αρχεία.

Αρχικά ξεκινάμε από ένα σημείο αναφοράς (π.χ. προφίλ του χρήστη των Windows ή τον ριζικό κατάλογο του δίσκου C:\) και εν συνεχεία εφαρμόζουμε “/” ή “..” για να πλοηγηθούμε στο κατάλληλο βάθος της δομής/οργάνωσης των αρχείων. Έτσι όταν χρειάζεται να αναφερθούμε στο αρχείο “my-file.html” που βρίσκεται:

- Στο **ίδιο επίπεδο** με το αρχείο που επεξεργαζόμαστε, τότε αναφερόμαστε σε αυτό απλά με το όνομά του δηλαδή “my-file.html”
- Σε έναν υποκατάλογο** “subdirectory” του τρέχοντος καταλόγου, τότε προσαρτούμε το όνομα του καταλόγου και την κάθετο “/” στο όνομα του αρχείου δηλαδή “subdirectory/my-file.html”
- Σε ένα επίπεδο καταλόγου πάνω** από τον τρέχοντα κατάλογο, τότε θα πρέπει να αναρριχηθούμε χρησιμοποιώντας τα “..” δηλαδή “../my-file.html”
- Σε ένα άλλο υποδένδρο του συστήματος οργάνωσης αρχείων** και σε διαφορετικό βάθος, τότε θα πρέπει να αναρριχηθούμε και να κατέβουμε ανάλογα χρησιμοποιώντας συνδυασμούς των (b) και (c) π.χ. “../subdirectory/another-subdirectory/my-file.html”.

## 1.2 Εργαλεία Συγγραφής

Αποτελούν έναν κρίσιμο παράγοντα στην διαδικασία συγγραφής περιεχομένου με στόχο τη διευκόλυνση του συντάκτη στην δημιουργία ενός παγκόσμιου ιστού σχεδιασμένο για/και προσβάσιμο από όλους. Οι βασικές κατηγορίες (<https://www.w3.org/standards/agents/authoring>) στις οποίες εντάσσονται τα εργαλεία είναι:

- Εργαλεία συγγραφής WYSIWYG (π.χ. PineGrow, Dreamweaver κλπ.)
- Εργαλεία επεξεργασίας πηγαίου κώδικα (π.χ. Notepad++, VS Code, Atom, Brackets)
- Εφαρμογές Επεξεργασίας κειμένου με δυνατότητα αποθήκευσης HTML (π.χ. Microsoft Word/LibreOffice/OpenOffice → Αποθήκευση ως HTML)
- Εξειδικευμένα περιβάλλοντα Ανάπτυξης Εφαρμογών Web (π.χ. Dreamweaver, Visual Studio, JetBrains PhpStorm/WebStorm, NetBeans κλπ.)
- Λογισμικά που δημιουργούν περιεχόμενο Web μέσω «οδηγών», ή μαζικών εντολών (scripts) π.χ. <https://mobirise.com/>, <http://layoutit.com/>
- Λογισμικά που παράγουν και διαχειρίζονται ιστότοπους με δυναμικό περιεχόμενο από βάσεις δεδομένων - CMS (Content Management Systems), LMS (Learning Management Systems) π.χ. Joomla/Drupal/Wordpress και Moodle/E-Class αντίστοιχα.
- Εργαλεία ανάπτυξης πολυμεσικών εφαρμογών (π.χ. Adobe Captivate)
- Εφαρμογές διαχείρισης ηλεκτρονικού ταχυδρομείου (E-mail) που ενσωματώνουν δυνατότητα δημιουργίας και επεξεργασίας μηνυμάτων σε γλώσσα HTML.

### 1.3 Αρχές για Προσβάσιμο και Ενιαίο Σχεδιασμό

Σύμφωνα με την ιστοσελίδα “WHO data in 2017” (<http://www.who.int/en/news-room/fact-sheets/detail/blindness-and-visual-impairment>) ο αριθμός χρηστών με δυσκολίες όρασης ανέρχεται σε 253 εκατομμύρια. Οι έννοιες του *προσβάσιμου* και *ενιαίου σχεδιασμού* συναντιούνται όλο και πιο συχνά τα τελευταία χρόνια και αφορούν σχεδιαστικές προσεγγίσεις οι οποίες αποσκοπούν στην παραγωγή αγαθών τα οποία είναι **εύκολα στη χρήση, για όλους**. Το περιεχόμενο που παράγεται καθημερινά και προσφέρεται μέσω του παγκόσμιου ιστού δεν εξαιρείται και θα πρέπει να ακολουθεί τις αρχές αυτές. Έτσι για κάθε σελίδα που κατασκευάζουμε θα πρέπει να αξιοποιούμε όλα τα εργαλεία – με έμφαση αυτά που ενισχύουν τη *σημασιολογική διάσταση του περιεχομένου προς όφελος του τελικού χρήστη*.

Η έννοια της **προσβασιμότητας**, συχνά αναφέρεται ως “*Web Accessibility*”, περιγράφει πρακτικές με στόχο την κατάργηση περιορισμών στην πρόσβαση/αλληλεπίδραση σε σελίδες από ανθρώπους με ειδικές δεξιότητες. Η μεθοδολογία προσβλέπει στην αξιοποίηση όλων των μέσων με σκοπό να αντιμετωπιστούν προβλήματα ([https://en.wikipedia.org/wiki/Web\\_accessibility](https://en.wikipedia.org/wiki/Web_accessibility)):

- Οπτικά (Visual): δυσκολία ανάγνωσης, χαμηλή όραση, αχρωματοψία ή ακόμα και τύφλωση.
- Κινητικά (Mobility): δυσκολία ή ανικανότητα χρήσης χεριών.
- Ακουστικά (Auditory): δυσκολία ακοής.
- Κρίσεων (Seizures): φώτο-επιληπτικές κρίσεις που προκαλούνται από μεγάλης συχνότητας φωτισμό.
- Διανοητικά (Cognitive/Intellectual): Δυσκολίες μάθησης και ανάπτυξης που σχετίζονται με μνήμη, προσοχή και ικανότητα επίλυσης προβλημάτων.

Για κάθε ένα από τα παραπάνω έχουν αναπτυχθεί και εξελίσσονται **συσκευές** που βοηθούν όσους έχουν δυσκολία να αποκτήσουν πρόσβαση στο διαθέσιμο υλικό:



- *Αναγνώστες κειμένου/Screen Readers*: Συσκευές Ανάγνωσης οι οποίες χρησιμοποιούν τεχνολογίες text-to-speech για να αναπαράγουν το περιεχόμενο σε μορφή κατάλληλη ώστε να καταναλωθεί από όσους έχουν θέματα όρασης και όχι μόνο.
- *Τερματικά Braille*: Ηλεκτρονικές συσκευές που διαθέτουν «οθόνες» οι οποίες μπορούν να δημιουργήσουν χαρακτήρες “Braille” μέσω ενός μηχανισμού δημιουργίας ανάγλυφων σχημάτων τα οποία μπορούν να ενεργοποιηθούν από ένα αντίστοιχο πληκτρολόγιο.
- *Screen Magnification*: Λογισμικό που μεγεθύνει ό, τι προβάλλεται στην οθόνη κάνοντάς το πιο εύκολα αντιληπτό.
- *Speech Recognition*: Λογισμικό που επιτρέπει τη λήψη φωνητικών εντολών, τη μετατροπή υπαγόρευσης σε γραμματικά ορθό κείμενο – χρήσιμο για όσους αντιμετωπίζουν θέματα κίνησης.
- Υποτιτλισμός ή άλλο εποπτικό υλικό (π.χ. sign language).

Οι παραπάνω τεχνολογίες εξελίσσονται και νέες εφευρέσεις βρίσκουν τρόπο να βοηθήσουν περισσότερους χρήστες που δεν έχουν πρόσβαση στο περιεχόμενο. Για να μεγιστοποιηθεί όμως το όφελος με την εξέλιξη της τεχνολογίας θα πρέπει πάντα να έχουμε κατά νου πως **η πραγματική αξία βρίσκεται στο περιεχόμενο** και γι’ αυτό θα πρέπει να μεριμνήσουμε να είναι **έτοιμο** – εκμεταλλευόμενοι τη σύνταξη και τις σημασιολογικές δυνατότητες της HTML – και **προσβάσιμο από όλους!** Για κάθε μια από τις παραμέτρους του “Web Accessibility” έχουν δημιουργηθεί οδηγοί με σημαντικότερο αυτών να αποτελεί ο **“Web Content Accessibility Guidelines (WCAG)”** (<https://www.w3.org/WAI/standards-guidelines/wcag/>).

Τέλος, η έννοια του **universal design** προσβλέπει στο να επεκτείνει τις βασικές αρχές σχεδιασμού ώστε να επιτρέψει σε όλους, ανεξαρτήτως ηλικίας και ικανότητας να κάνουν χρήση χωρίς όμως να αλλάζει δραματικά ο βαθμός γενίκευσης (*level of generality*). Η υιοθέτηση των 7 αρχών (<http://universaldesign.ie/What-is-Universal-Design/The-7-Principles/>) έχει δείξει πως βελτιώνει την ταχύτητα, μειώνει τα σφάλματα και τον χρόνο εκμάθησης για όλους τους χρήστες κάνοντας τα πράγματα ευκολότερα για όλους.

## 2. HTML (Hypertext Markup Language)

Είναι η γλώσσα επισημείωσης/σήμανσης (markup) με την οποία έχουν δημιουργηθεί σχεδόν όλες οι σελίδες (περιεχόμενα) που μπορούμε να συναντήσουμε στον παγκόσμιο ιστό. Στα 30 χρόνια της έχει περάσει από διαφορετικές φάσεις/εκδόσεις (<https://en.wikipedia.org/wiki/HTML>), εμπλουτίζοντας τις σημασιολογικές της δυνατότητες (semantics) και το συντακτικό (syntax) της. Οι περισσότερες ιστοσελίδες που βρίσκονται αναρτημένες ακολουθούν τα πρότυπα HTML 4.0, XHTML 1.0 και HTML5.

Για κάποιον που ξεκινά τώρα να μαθαίνει Web Design/Development η δική μας πρόταση θα ήταν να ξεκινήσει με το συντακτικό της HTML5, με στόχο την εξοικείωση για *καθαρό (clean)* και *σαφώς ορισμένο (well structured)* HTML κώδικα. Εν συντομία, οι δυνατότητες που προσφέρει η γλώσσα HTML, σε συνδυασμό με τις συνεργαζόμενες τεχνολογίες (CSS/JavaScript) στους διαχειριστές/συγγραφείς περιεχομένου είναι:

1. Η δημιουργία/δημοσίευση περιεχομένου, το οποίο είναι σαφώς ορισμένο με χρήση κατάλληλων σημάτων - όπως επικεφαλίδων, παραγράφων, γραφικών και πολυμεσικών αντικειμένων, έτοιμο να καταναλωθεί τόσο από χρήστες όσο και μηχανικά από agents (π.χ. μηχανές αναζήτησης, screen readers/assistive devices).
2. Η ανάκτηση και διασύνδεση πληροφορίας μέσω υπερμεσικών συνδέσμων (hyperlinks) με τους οποίους υλοποιείται αυτό που ονομάζουμε “*web of information*”.
3. Η καταχώρηση/συλλογή δεδομένων από τους επισκέπτες μέσω ειδικών φορμών (π.χ. αναζήτηση πληροφοριών, κράτηση/παραγγελία προϊόντων ή υπηρεσιών κτλ.).
4. Η διασύνδεση/διαλειτουργικότητα με εξωτερικά συστήματα με ενσωμάτωση λειτουργικότητας που θεωρείται αναγκαία π.χ. εμφάνιση πρόγνωσης καιρού κτλ.

Για τη δημιουργία/επεξεργασία αρχείων HTML αρκεί ένας απλός επεξεργαστής κειμένου καθώς μιλάμε για αρχεία τύπου ASCII. Το αποτέλεσμα που εισπράττει ο επισκέπτης μιας ιστοσελίδας σπάνια προέρχεται μόνο από τη χρήση HTML, αλλά είναι ένας συνδυασμός τεχνολογιών όπως CSS, JavaScript κτλ. Τα αρχεία CSS/JavaScript είναι αρχεία κειμένου, άρα είναι σημαντικό να επιλέξουμε έναν επεξεργαστή ο οποίος να προσφέρει εγγενή υποστήριξη σε όλους τους τύπους αρχείων HTML/CSS/JavaScript.

### 2.1 Τυπική Δομή ενός HTML αρχείου

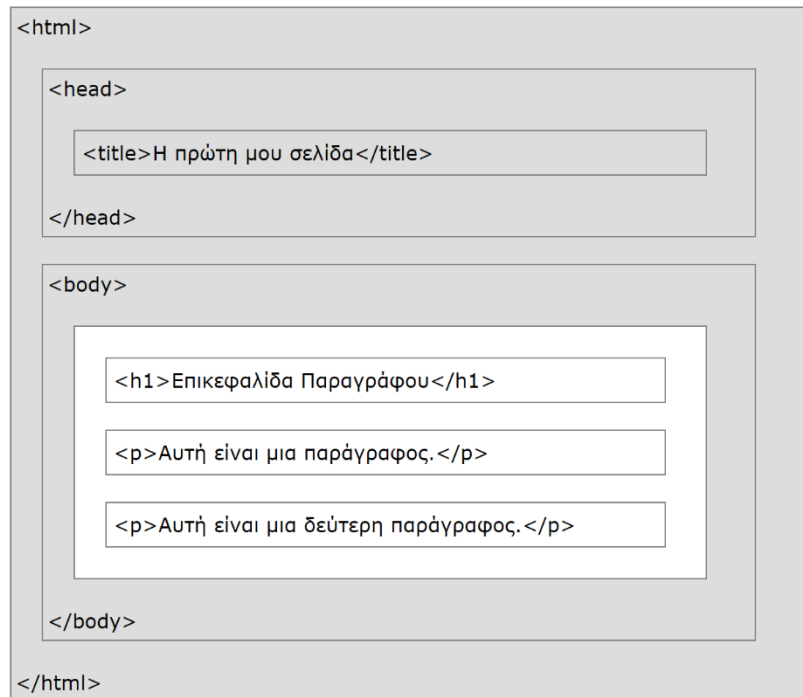
Τα στοιχεία της HTML γίνονται ευκολότερα κατανοητά αν «ακτινογραφήσουμε» μια υπάρχουσα σελίδα, αποκαλύπτοντας τον κώδικά της (View Page Source/Inspect). Μια βασική μορφή είναι η εξής:

*Παράδειγμα 2.1. Απλό παράδειγμα HTML σελίδας*

```
1. <!DOCTYPE html>
2. <html lang="el">
3. <head>
4.   <title> Η πρώτη μου σελίδα </title>
5.   <meta charset="utf-8">
6. </head>
```

7. `<body>`
8. `<h1>Επικεφαλίδα Παραγράφου</h1>`
9. `<p title="Σύντομη επεξήγηση παραγράφου">` Αυτή είναι μια παράγραφος.`</p>`
10. `<p>`Αυτή είναι μια δεύτερη παράγραφος. `</p>`
11. `</body>`
12. `</html >`

Αν αναπαραστήσουμε με γραφικό τρόπο το παραπάνω έγγραφο θα έχουμε το απλοποιημένο σχήμα που ακολουθεί:



Εικόνα 2.1. Γραφική αναπαράσταση κώδικα σελίδας

Από το παραπάνω παράδειγμα, μπορούμε να δούμε πως υπάρχουν στοιχεία HTML τα οποία δημιουργούν το αποτέλεσμα που εμφανίζεται στην οθόνη και άλλα που δεν αφορούν το ορατό περιεχόμενο:

1. `<!DOCTYPE html>`: Σηματοδοτεί τη δημιουργία ενός HTML εγγράφου και υποδεικνύει το πρότυπο (HTML/XHTML/HTML5 - <https://www.w3.org/QA/2002/04/valid-dtd-list.html>) βάσει του οποίου θα ελεγχθεί η εγκυρότητά του.
2. `<html>...</html>`: Αφορά την κεφαλή του εγγράφου HTML, και περιλαμβάνει όλα τα στοιχεία του εγγράφου – συχνά αναφέρεται ως ριζικό στοιχείο (root element).
3. `<head>...</head>`: Καταγράφει μεταδεδομένα (metadata) του εγγράφου όπως την κωδικοποίηση του, links σε εξωτερικά αρχεία μορφοποίησης (styles π.χ. CSS) ή προγραμματιστικές βιβλιοθήκες (scripts π.χ. jQuery/Bootstrap). Περιλαμβάνει οδηγίες που αξιοποιούνται από μηχανές αναζήτησης.
4. `<title>...</title>`: Ορίζει τον τίτλο του εγγράφου (αυτό που εμφανίζεται στην επικεφαλίδα του φυλλομετρητή) ή το όνομα με το οποίο αποθηκεύεται η σελίδα στους σελιδοδείκτες.

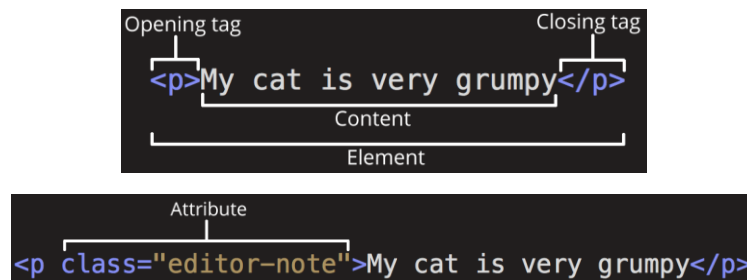
5. `<meta charset="utf-8">`: Ορίζει την κωδικοποίηση των χαρακτήρων του εγγράφου. Με την υποστήριξη Unicode λύνονται προβλήματα του παρελθόντος και θεωρείται καλή πρακτική η δήλωση του αν και προαιρετική.
6. `<body>...</body>`: Περιλαμβάνει όλο το ορατό περιεχόμενο της σελίδας είτε αυτό είναι κείμενο, εικόνες, videos, ήχοι κτλ.

Στις επόμενες ενότητες αναλύονται τα συστατικά με τα οποία δημιουργείται το περιεχόμενο των σελίδων.

## 2.2 Στοιχεία (Elements) & Ετικέτες (Tags)

Το βασικό συστατικό ενός εγγράφου HTML είναι το **στοιχείο** (element) το οποίο αποτελείται από:

- Την **ετικέτα ανοίγματος** (opening tag) π.χ. `<p>`
- Την **ετικέτα κλεισίματος** (closing tag) π.χ. `</p>`. Σε ορισμένες περιπτώσεις δεν απαιτείται π.χ. `<br>`, `<hr>`, ενώ σε άλλες μπορεί να είναι προαιρετικό (π.χ. `</p>`). Καλό είναι να μην παραλείπεται!
- Τα **περιεχόμενο** (content) που περικλείεται μεταξύ των δύο (2) ετικετών.
- Άλλα **χαρακτηριστικά** (attributes) τα οποία συμπληρώνουν το περιεχόμενο πληροφοριακά ή/και λειτουργικά.



Τα στοιχεία που συνθέτουν το περιεχόμενο μπορούν να συνδυάζονται, δημιουργώντας σύνθετες σημασιολογικές παραστάσεις, ενσωματώνοντας στοιχεία HTML μέσα σε άλλα. Αυτή η διαδικασία ονομάζεται **ενθυλάκωση** (nesting) και απαιτεί τη σωστή δήλωση των ετικετών κλεισίματος (αντίστροφα από τη σειρά εμφάνισης των ετικετών ανοίγματος). Στο επόμενο παράδειγμα η πρώτη παράγραφος εμφανίζει σωστά το αποτέλεσμα γιατί ακολουθείται ορθά η σειρά κλεισίματος των ετικετών. Αντίθετα στη δεύτερη, η σειρά έχει αντιστραφεί με αποτέλεσμα το περιεχόμενο να μην εμφανίζεται σωστά.

Παράδειγμα 2.2. Παράδειγμα ενθυλάκωσης ετικετών

1. `<p>My cat is <strong>very</strong> grumpy.</p>`
2. `<p>My cat is <strong>very grumpy.</p></strong>`

Το κεντρικό element που σηματοδοτεί την έναρξη ενός HTML εγγράφου είναι το `<html>` το οποίο αποτελείται από 2 παιδιά-children – το `<head>` και το `<body>` (μεταξύ τους είναι αδέρφια-siblings). Η HTML δεν κάνει διαχωρισμό πεζών/κεφαλαίων (π.χ. `<body>` ή `<BODY>` ή ακόμα και `<BoDy>`),

αποτελεί όμως καλή πρακτική να οριστεί εξαρχής μια πολιτική (π.χ. πεζοί χαρακτήρες) που θα εφαρμόζεται συνολικά στις σελίδες για λόγους *βελτίωσης της ανάγνωσης και αποσφαλμάτωσης*.

## 2.3 Χαρακτηριστικά (attributes)

Τα *χαρακτηριστικά (attributes)* που συνοδεύουν τα στοιχεία μπορούν να κατηγοριοποιηθούν σε:

- **Πληροφοριακά** (informative) που δεν επηρεάζουν τη συμπεριφορά του element, αλλά εμπλουτίζουν το περιεχόμενό του με πληροφορία που μπορούν προγράμματα/agents να αξιοποιούν κατάλληλα. Για παράδειγμα, δηλώνοντας τη γλώσσα (π.χ. lang="el") σε μέρος κειμένου που διαφοροποιείται από το υπόλοιπο κείμενο βοηθά τα προγράμματα να υποστηρίξουν καλύτερα το περιεχόμενο π.χ. αναγνώστες κειμένου, μηχανές αναζήτησης, μεταφραστές κτλ.
- **Λειτουργικά** (functional) που επηρεάζουν άμεσα τη λειτουργία του στοιχείου π.χ. το "href" χαρακτηριστικό του <a> καθορίζει τη διεύθυνση στην οποία θα μεταβεί ο χρήστης όταν πατήσει το σύνδεσμο.

*Παράδειγμα 2.3. Χρήση χαρακτηριστικών (title, src, alt, href)*

```
1. <!DOCTYPE html>
2. <html>
3.   <head>
4.     <title> Σπουδές </title>
5.     <meta charset="utf-8">
6.   </head>
7.   <body>
8.     <p title="Αντικείμενο του τμήματος"> Το αντικείμενο του τμήματος ...</p>
9.     
10.    <a href="http://www.alis.uniwa.gr"> Το τμήμα μου</a>
11.  </body>
12. </html>
```

Σε κάθε περίπτωση θα πρέπει:

1. Να υπάρχει κενό μεταξύ του ονόματος του χαρακτηριστικού και του ονόματος του στοιχείου (ετικέτας) ή άλλου χαρακτηριστικού.
2. Το όνομά του χαρακτηριστικού να ακολουθείται από το "="
3. Η τιμή του χαρακτηριστικού να περικλείεται σε διπλά εισαγωγικά "", εκτός και αν το περιεχόμενό του περιέχει τον χαρακτήρα "", οπότε πρέπει η τιμή να συνδυάζεται με τη χρήση των ' (single quotes).
4. Τα ονόματα των χαρακτηριστικών να εισάγονται με πεζούς χαρακτήρες.

Τα πιο σημαντικά από τα χαρακτηριστικά που μπορεί κανείς να χρησιμοποιήσει περιλαμβάνονται στην παρακάτω λίστα και κατά κάποιο τρόπο θεωρούνται *καθολικά* ([https://developer.mozilla.org/en-US/docs/Web/HTML/Global\\_attributes](https://developer.mozilla.org/en-US/docs/Web/HTML/Global_attributes)):

Πίνακας 2.1. Γενικά Χαρακτηριστικά (Global Attributes)

Χαρακτηριστικό	Περιγραφή
accesskey	Επιτρέπει τη δημιουργία συντομεύσεων πληκτρολογίου για την πρόσβαση σε ένα στοιχείο.
autocapitalize	Στα στοιχεία που δέχονται δεδομένα από το χρήστη μπορεί να αυτοματοποιήσει τη μετατροπή των δεδομένων σε κεφαλαία γραφή – σε επίπεδο πρότασης (on/sentences), λέξεων (words) ή χαρακτήρων (characters).
class	Περιέχει μια λίστα κλάσεων οι οποίες βοηθούν στην στόχευση και επιλογή στοιχείων από κανόνες CSS ή κώδικα JavaScript.
contenteditable	Δηλώνει αν το στοιχείο πρέπει να επιτρέπει (true) ή όχι (false) την επεξεργασία δεδομένων από τον χρήστη.
hidden	Επιτρέπει την απόκρυψη στοιχείων που για κάποια δεδομένη στιγμή δε θεωρούνται σχετικά αλλά θα εμφανιστούν στην συνέχεια υπό συνθήκη.
id	Ορίζει ένα μοναδικό αναγνωριστικό (id) – μοναδικό σε επίπεδο εγγράφου. Τυπικές χρήσεις είναι ο ορισμός «σελιδοδεικτών» (fragments) ή ο χειρισμός στοιχείων από κανόνες CSS ή κώδικα JavaScript.
lang	Χρησιμοποιείται για να δηλώσουμε τη γλώσσα ενός στοιχείου.
Spellcheck	Ορίζει την ενεργοποίηση ορθογραφικού ελέγχου στα περιεχόμενα (true/false).
style	Περιλαμβάνει δηλώσεις styling (CSS).
title	Περιλαμβάνει κείμενο το οποίο λειτουργεί συμπληρωματικά/επεξηγηματικά με το περιεχόμενο του στοιχείου – συνήθως εμφανίζεται με μορφή tooltip.

## 2.4 <!DOCTYPE>

Το στοιχείο <!DOCTYPE> ορίζει μια αναφορά σε ένα πρότυπο (σύνολο από κανόνες) βάσει του οποίου θα ελεγχθεί η εγκυρότητα του εγγράφου. Αυτό βοηθά τους περιηγητές να εμφανίσουν τα περιεχόμενα του αρχείου σωστά. Στον επόμενο πίνακα εμφανίζονται οι δηλώσεις που μπορεί να αντιμετωπίσετε για τις πιο διαδεδομένες εκδόσεις της HTML (προτείνεται αυτή της HTML5 - [https://www.w3schools.com/tags/tag\\_doctype.asp](https://www.w3schools.com/tags/tag_doctype.asp)).

Πίνακας 2.2. Χαρακτηριστικά ετικέτας <!DOCTYPE>

Έκδοση	DOCTYPE
HTML5	<!DOCTYPE html>
HTML4.01 (transitional)	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

Έκδοση	DOCTYPE
XHTML (transitional)	<code>&lt;!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"&gt;</code>

## 2.5 <head> : Μεταδεδομένα Εγγράφου

Τα μεταδεδομένα του εγγράφου βρίσκονται στο στοιχείο <head> και προσδιορίζουν στοιχεία που επηρεάζουν:

- Τον τίτλο και την κωδικοποίηση του εγγράφου.
- Τη μορφοποίηση της δομής και του περιεχομένων μέσω κανόνων μορφοποίησης styles/CSS .
- Την αλληλεπίδραση του χρήστη με το περιεχόμενο και τη δυνατότητα εμφάνισης δυναμικού περιεχομένου μέσω JavaScript.
- Οδηγίες που σχετίζονται με τον τρόπο που οι agents (browsers ή άλλα προγράμματα) θα διαχειριστούν το περιεχόμενο και τη δομή στη σελίδα.
- Οδηγίες που αφορούν τον τρόπο που θα εμφανιστούν τα περιεχόμενα της σελίδας ανάλογα με το μέγεθος της οθόνης της συσκευής.

Παράδειγμα 2.4. Παράδειγμα ενός <head> στοιχείου

```

1. <head>
2. <title> Η πρώτη μου σελίδα </title>
3. <meta charset="utf-8">
4. <meta name="description" content="My first HTML page">
5. <meta name="author" content="Nikolaos Lazaridis">
6. <link rel="stylesheet" href="style.css">
7. </head>
```

### 2.5.1 Το στοιχείο <title>

Το στοιχείο <title> μοναδικά προσδιορίζει το περιεχόμενο του εγγράφου μέσα από μια σύντομη περιγραφή που είναι απαραίτητη να υπάρχει στο στοιχείο <head>. Είναι αυτό που καθορίζει το λεκτικό που βλέπουμε στο tab του φυλλομετρητή ή όταν αποθηκεύουμε ένα σελιδοδείκτη.

### 2.5.2 Το στοιχείο <meta>

Το στοιχείο <meta> προσδιορίζει μεταδεδομένα του εγγράφου όπως ο συγγραφέας, τα δικαιώματα δημιουργού, η περιγραφή, λέξεις κλειδιά, οδηγίες προς robots, βαθμό καταλληλότητας κτλ. Από το Παράδειγμα 2.4 βλέπουμε πως ο τρόπος σύνταξης είναι της μορφής:

```
<meta name = "παράμετρος" content = "τιμή">
```

Οι επιλογές για τις μεταβλητές της εμφανίζονται συνοπτικά στον παρακάτω πίνακα:

Πίνακας 2.3. Χαρακτηριστικά ετικέτας <meta>

Παράμετρος	Χρήση	Παράδειγμα
author	Συγγραφέας/συντάκτης	<meta name="author" content="Nikolaos Lazaridis">
copyright	Δικαιώματα δημιουργού	<meta name="copyright" content="2020, ALIS">
description	Μια σύντομη περιγραφή του περιεχομένου	<meta name="description" content="A short HTML tutorial">
keywords	Λέξεις κλειδιά (διαχωρισμένες με κόμμα)	<meta name="keywords" content="html, tutorial">
robots	Θέτει περιορισμούς στις μηχανές αναζήτησης για τον τρόπο που θα προσπελάσουν τις σελίδες (index, noindex, nofollow, none)	<meta name="robots" content="noindex, nofollow">
rating	Μέθοδος αξιολόγησης της καταλληλότητας του περιεχομένου της σελίδας για παιδιά	<meta name="rating" content="general">
expires	Η ημερομηνία μετά την οποία το περιεχόμενο του εγγράφου θεωρείται εκπρόθεσμο ώστε να αξιοποιηθεί από τις μηχανές αναζήτησης για τη διαγραφή ληγμένων εγγράφων	<meta name="expires" content="Apr 20, 2020 00:00:00 EST">
refresh	Προσφέρει ένα μηχανισμό ανανέωσης και εμφάνισης ενός εγγράφου με δυνατότητα χρονικής παραμετροποίησης	<meta http-equiv="refresh" content="5"; url="file2.html">
charset	Χρησιμοποιείται για να οριστεί η κωδικοποίηση του εγγράφου	<meta charset="UTF-8"> ή <meta http-equiv="content-type" content="text/html; charset=UTF-8">

### 2.5.3 Το στοιχείο <link>

Το στοιχείο <link> χρησιμοποιείται για να ορίσει τη σχέση του τρέχοντος εγγράφου με ένα εξωτερικό αρχείο/πόρο. Μας επιτρέπει να ορίσουμε μια σειρά σχέσεων του τρέχοντος εγγράφου με άλλα έγγραφα του ιστότοπου, με σκοπό τη βελτίωση της πλοήγησης (first, last, next, prev). Η συχνότερη χρήση του είναι για να οριστούν συσχετίσεις με έγγραφα που ορίζουν το αισθητικό μέρος της σελίδας (CSS), με αρχεία γραφικών (icons) κτλ.

```
<link rel="stylesheet|alternate|author|bookmark|external|first|help|icon|last|license|next|nofollow|noreferrer|pingback|prefetch|prev|search|sidebar|tag|up" href="file or url">
```

Στον παρακάτω πίνακα φαίνονται συνοπτικά οι κυριότερες παράμετροι του στοιχείου <link> και οι τρόποι χρήσης τους:



Πίνακας 2.4. Χαρακτηριστικά ετικέτας <link>

Χαρακτηριστικό	Χρήση	Παράδειγμα
stylesheet	Ορίζει το εξωτερικό αρχείο που περιλαμβάνει τους κανόνες μορφοποίησης (CSS).	<link rel="stylesheet" href="default.css">
icon	Ορίζει ένα εικονίδιο με το οποίο θα εκπροσωπείται η σελίδα στους σελιδοδείκτες (bookmarks) και στις κεφαλίδες των browser tabs.	<link rel="icon" href="demo_icon.gif" type="image/gif">
author	Ορίζει σύνδεσμο με το έγγραφο που περιέχει πληροφορίες για τον συγγραφέα.	

#### 2.5.4 Το στοιχείο <base>

Το στοιχείο <base> χρησιμοποιείται για να διευκολύνει τους συντάκτες με τη δημιουργία και ορισμό των μονοπατιών (paths) που “οδηγούν” τους υπερσυνδέσμους εντός του ιστότοπου. Τα χαρακτηριστικά που διαχειρίζεται είναι τα παρακάτω:

1. href: Ορίζει τη βάση (πρόθεμα) με την οποία συμπληρώνονται όλα τα URL ώστε να προκύπτει το πραγματικό μονοπάτι των πόρων.
2. target: Ορίζει ενιαία πολιτική για το άνοιγμα των υπερσυνδέσμων με δυνατές τιμές (\_blank, \_parent, \_self, \_top)

Παράδειγμα 2.5. Παράδειγμα χρήσης base (metadata)

```

1. <!DOCTYPE html>
2. <html lang="el">
3. <head>
4.   <meta charset="utf-8">
5.   <base href="http://alis.uniwa.gr" target="_blank">
6.   <title>Base (head metadata) </title>
7. </head>
8. <body>
9. <h1>Τμήμα Αρχειονομίας, Βιβλιοθηκονομίας και Συστημάτων Πληροφόρησης</h1>
10. <h2>Προπτυχιακό Πρόγραμμα Σπουδών</h2>
11. <p>
12.   <a href="/undergraduate-curriculum">
13.     
15.   </a>
16. </p>
17. </body>
18. </html>

```

## 2.6 Μοντέλα Περιεχομένου (Content Models)

Στο προηγούμενο παράδειγμα, είδαμε τη σύνταξη ενός απλού σχετικά εγγράφου όπου εμφανίζονται τα στοιχεία που σηματοδοτούν τη δημιουργία ενός HTML όπως και τα μεταδεδομένα που επηρεάζουν την εμφάνιση και αντιμετώπιση των μηχανών αναζήτησης προς αυτό. Πριν προχωρήσουμε σε ετικέτες που επηρεάζουν τη μορφοποίηση του κειμένου, πρέπει να αναφερθούμε στα **μοντέλα περιεχομένου (content models)**.

Τα μοντέλα περιεχομένου περιγράφουν τον τρόπο και τους κανόνες με τους οποίους μπορούν τα στοιχεία να συνδυάζονται – ρυθμίζοντας το είδος απογόνων που το καθένα μπορεί να υποδεχτεί. Οι ετικέτες (tags) που μπορούμε να χρησιμοποιήσουμε για να δημιουργήσουμε **δομή** και **ροή περιεχομένου (document flow)** σε ένα έγγραφο, σε προηγούμενες εκδόσεις της HTML, ανήκαν σε μια από τις παρακάτω δύο κατηγορίες:

1. **Block-level** ([https://developer.mozilla.org/en-US/docs/Web/HTML/Block-level\\_elements](https://developer.mozilla.org/en-US/docs/Web/HTML/Block-level_elements)) όπως δείχνει το όνομα δημιουργούν ένα «πλαίσιο» καταλαμβάνοντας όλο το πλάτος της σελίδας. Έτσι η εμφάνιση του περιεχομένου που περικλείουν πάντα ξεκινά σε μια νέα γραμμή και μπορεί να περιλαμβάνει άλλα *block-level* ή *inline* αντικείμενα π.χ.:

```
<address>, <article>, <aside>, <blockquote>, <canvas>, <details>,
<dialog>, <dd>, <div>, <dl>, <dt>, <fieldset>, <figcaption>, <figure>,
<footer>, <form>, <h1>...<h6>, <header>, <hgroup>, <hr>, <li>, <main>,
<nav>, <ol>, <output>, <p>, <pre>, <section>, <table>, <ul>
```

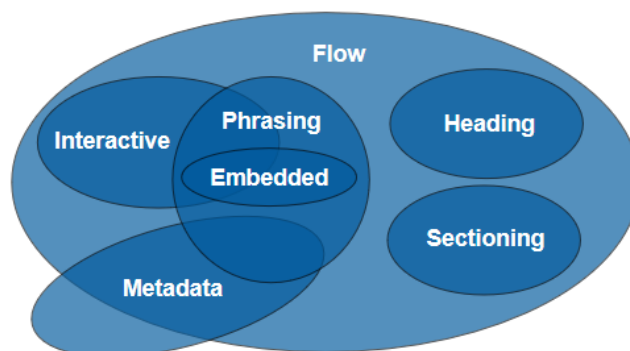
2. **Inline** ([https://developer.mozilla.org/en-US/docs/Web/HTML/Inline\\_elements](https://developer.mozilla.org/en-US/docs/Web/HTML/Inline_elements)): Σε αντίθεση με τα block-level, τα inline στοιχεία μπορούν να εμφανίζονται σε οποιοδήποτε σημείο μιας παραγράφου και μπορούν να περιλαμβάνουν κείμενο ή άλλο inline αντικείμενο, π.χ.:

```
<a>, <abbr>, <b>, <bdi>, <bdo>, <br>, <button>, <cite>, <code>, <data>,
<datalist>, <dfn>, <em>, <i>, <img>, <input>, <ins>, <kbd>, <label>,
<map>, <mark>, <object>, <picture>, <q>, <s>, <samp>, <script>,
<select>, <small>, <span>, <strong>, <sub>, <sup>, <textarea>, <time>,
<u>, <tt>, <var>, <video>
```

Στην HTML5, τα μοντέλα μετεξελίχθηκαν για να επεκτείνουν τη δομή και τις **σημασιολογικές δυνατότητες** της HTML καταλήγοντας σε 7 (<https://html.spec.whatwg.org/#content-models>) Καθένα από τα διαθέσιμα στοιχεία μπορεί να ανήκει σε μια ή περισσότερες κατηγορίες όπως φαίνεται στην Εικόνα 2.2:

- **Metadata:** Αφορά παραμέτρους που ρυθμίζουν θέματα παρουσίασης ή/και συμπεριφοράς του εγγράφου σε σχέση με τους agents (χρήστες/μηχανές), ακόμα και τη σχέση του εγγράφου με άλλα έγγραφα.
- **Embedded:** Περιεχόμενο που εισάγεται από άλλο πόρο (resource) μέσα στο έγγραφο π.χ. canvas, images, math
- **Interactive:** Αφορά περιεχόμενο που είναι προσανατολισμένο στην αλληλεπίδραση με το χρήστη π.χ. anchors <a>,

- **Phrasing:** Αποτελούν το κείμενο του εγγράφου, όπως επίσης και στοιχεία που μπορούν να το «χαρακτηρίσουν». Είναι σχεδόν ισοδύναμο με αυτό που οι προηγούμενες HTML εκδόσεις χαρακτήριζαν ως *inline*.
- **Flow:** Τα περισσότερα από τα στοιχεία που βρίσκονται στο <body> εντάσσονται σε αυτή την κατηγορία.
- **Sectioning:** Τα στοιχεία που οριοθετούν ενότητες περιεχομένου και ορίζουν την εμβέλεια (εύρος) των (επικεφαλίδων).
- **Heading:** Ορίζουν τις επικεφαλίδες ενότητων ή και ακόμα δημιουργούν ενότητες που επηρεάζουν το περίγραμμα (outline) της σελίδας.



Εικόνα 2.2. Βασικές Κατηγορίες Ετικετών (Content Models)

Στον παρακάτω πίνακα φαίνονται τα στοιχεία της HTML σε μια εναλλακτική ομαδοποίηση: (<https://developer.mozilla.org/en-US/docs/Web/HTML/Element>):

Πίνακας 2.5. HTML5 εναλλακτική ομαδοποίηση ετικετών

Κατηγορία Στοιχείων	Περιγραφή
Ρίζα Εγγράφου (Document Root)	Αποτελεί το κεντρικό στοιχείο (top level element) ενός HTML εγγράφου και συνήθως αναφερόμαστε σε αυτό ως “root element”. <a href="#">html</a>
Ρίζα Περιεχομένου (Sectioning Root)	Αποτελεί το περιεχόμενο της σελίδας (ορατό περιεχόμενο) και εμφανίζεται μόνο μια φορά στο έγγραφο. <a href="#">body</a>
Μεταδεδομένα Εγγράφου (Document Metadata)	Περιλαμβάνει μεταδεδομένα για τη σελίδα σχετικά με styles, scripts ή άλλη πληροφορία που θα βοηθήσει λογισμικά (π.χ. μηχανές αναζήτησης, αναγνώστες κειμένου, φυλλομετρητές) να μορφοποιήσουν κατάλληλα και να εμφανίσουν τα περιεχόμενα της σελίδας. <a href="#">base</a> , <a href="#">head</a> , <a href="#">link</a> , <a href="#">meta</a> , <a href="#">style</a> , <a href="#">title</a>

Κατηγορία Στοιχείων	Περιγραφή
<p>Διαχωρισμός Περιεχομένου (Content Sectioning)</p>	<p>Επιτρέπουν την ομαδοποίηση και οργάνωση του περιεχομένου της σελίδας σε λογικές ενότητες. Χρησιμοποιούνται για να δημιουργηθεί το περίγραμμα (outline) ενσωματώνοντας στοιχεία επικεφαλίδων και περιήγησης ως μέσα αναγνώρισης ενότητων περιεχομένου.</p> <p><a href="#">address</a>, <a href="#">article</a>, <a href="#">aside</a>, <a href="#">footer</a>, <a href="#">header</a>, <a href="#">h1</a>, h2, h3, h4, h5, h6, <a href="#">hgroup</a>, <a href="#">main</a>, <a href="#">nav</a>, <a href="#">section</a></p>
<p>Περιεχόμενο Ενότητων (Text content)</p>	<p>Η χρήση τους επιτρέπει την οργάνωση του περιεχομένου σε στοιχεία που προσδιορίζουν το ρόλο ή/και τη δομή του περιεχομένου. Μέσω αυτών υλοποιούνται και βελτιώνονται οι οδηγίες για προσβασιμότητα (accessibility <a href="https://developer.mozilla.org/en-US/docs/Glossary/accessibility">https://developer.mozilla.org/en-US/docs/Glossary/accessibility</a>) και βελτιστοποίηση των μηχανών αναζήτησης (SEO <a href="https://developer.mozilla.org/en-US/docs/Glossary/SEO">https://developer.mozilla.org/en-US/docs/Glossary/SEO</a>).</p> <p><a href="#">blockquote</a>, <a href="#">dd</a>, <a href="#">div</a>, <a href="#">dl</a>, <a href="#">dt</a>, <a href="#">figcaption</a>, <a href="#">figure</a>, <a href="#">hr</a>, <a href="#">li</a>, <a href="#">main</a>, <a href="#">ol</a>, <a href="#">p</a>, <a href="#">pre</a>, <a href="#">ul</a></p>
<p>Σημασιολογία Κειμένου (Inline/text-level Semantics)</p>	<p>Χρησιμοποιούνται για να ορίσουμε νόημα, δομή ή στυλ σε μια λέξη, γραμμή ή άλλο μέρος κειμένου.</p> <p><a href="#">a</a>, <a href="#">abbr</a>, <a href="#">b</a>, <a href="#">bdi</a>, <a href="#">bdo</a>, <a href="#">br</a>, <a href="#">cite</a>, <a href="#">code</a>, <a href="#">data</a>, <a href="#">dfn</a>, <a href="#">em</a>, <a href="#">i</a>, <a href="#">kbd</a>, <a href="#">mark</a>, <a href="#">q</a>, <a href="#">s</a>, <a href="#">samp</a>, <a href="#">small</a>, <a href="#">span</a>, <a href="#">strong</a>, <a href="#">sub</a>, <a href="#">sup</a>, <a href="#">time</a>, <a href="#">u</a>, <a href="#">var</a>, <a href="#">wbr</a></p>
<p>Εικόνες και Πολυμέσα (Image and Multimedia)</p>	<p>Η HTML υποστηρίζει περιεχόμενο που δέχεται αλληλεπίδραση και αντιστοιχεί σε πόρους όπως εικόνες, αρχεία ήχου και video.</p> <p><a href="#">area</a>, <a href="#">audio</a>, <a href="#">img</a>, <a href="#">map</a>, <a href="#">track</a>, <a href="#">video</a></p>
<p>Ενσωματωμένο Περιεχόμενο (Embedded content)</p>	<p>Συμπληρωματικά με τα παραπάνω πολυμέσα, η HTML μπορεί να εισάγει αντίστοιχο περιεχόμενο ακόμα και αν αυτό δεν υποστηρίζει αλληλεπίδραση με τον χρήστη.</p> <p><a href="#">embed</a>, <a href="#">iframe</a>, <a href="#">object</a>, <a href="#">param</a>, <a href="#">picture</a>, <a href="#">source</a></p>
<p>Προγραμματισμός (Scripting)</p>	<p>Η δυνατότητα δημιουργίας δυναμικού περιεχομένου ή εφαρμογές web, προσφέρεται με τη χρήση scripting γλωσσών με επικρατέστερη την JavaScript.</p> <p><a href="#">canvas</a>, <a href="#">noscript</a>, <a href="#">script</a></p>
<p>Οριοθετημένες αλλαγές (Demarcating edits)</p>	<p>Η χρήση τους προτείνεται για την προβολή ενδείξεων αλλαγών/τροποποιήσεων στο περιεχόμενο (σε αντιστοιχία με την παρακολούθηση αλλαγών)</p> <p><a href="#">del</a>, <a href="#">ins</a></p>
<p>Δεδομένα Πίνακα (Table Data)</p>	<p>Χρησιμοποιούνται για τη δημιουργία και διαχείριση δεδομένων σε μορφή πίνακα</p> <p><a href="#">caption</a>, <a href="#">col</a>, <a href="#">colgroup</a>, <a href="#">table</a>, <a href="#">tbody</a>, <a href="#">td</a>, <a href="#">tfoot</a>, <a href="#">th</a>, <a href="#">thead</a>, <a href="#">tr</a></p>

Κατηγορία Στοιχείων	Περιγραφή
Φόρμες (Forms)	<p>Η HTML διαθέτει ένα μεγάλο αριθμό στοιχείων με τα οποία μπορούν να δημιουργηθούν φόρμες αλληλεπίδρασης με τον χρήστη, με σκοπό να καταχωρήσει δεδομένα στην εφαρμογή. Υπάρχει ξεχωριστός οδηγός (<a href="https://developer.mozilla.org/en-US/docs/Learn/HTML/Forms">https://developer.mozilla.org/en-US/docs/Learn/HTML/Forms</a>) με προχωρημένα θέματα που αφορούν τις φόρμες όπως δόμηση και μορφοποίηση φορμών όπως και πραγματοποίηση ελέγχων εγκυρότητας των στοιχείων που καταχωρούνται.</p> <p><a href="#">button</a>, <a href="#">datalist</a>, <a href="#">fieldlist</a>, <a href="#">form</a>, <a href="#">input</a>, <a href="#">label</a>, <a href="#">legend</a>, <a href="#">meter</a>, <a href="#">optgroup</a>, <a href="#">option</a>, <a href="#">output</a>, <a href="#">progress</a>, <a href="#">select</a>, <a href="#">textarea</a></p>

Αξιοποιώντας τα στοιχεία της HTML5, δίνουμε έμφαση στην σημασιολογική διάσταση του περιεχομένου και επιτυγχάνουμε τη δημιουργία εγγράφων με περισσότερο νόημα τόσο για τους τελικούς χρήστες όσο και για τους agents (μηχανές αναζήτησης, συσκευές ανάγνωσης κτλ.).

Θα πρέπει να γνωρίζετε πως **δεν είναι πρακτικό να προσπαθήσετε να αποστηθίσετε τα παραπάνω στοιχεία**, αλλά το να γνωρίζετε **τη σημασία τους** και το πώς μπορείτε να τα αξιοποιήσετε ανατρέχοντας στις πηγές τεκμηρίωσης. Σε αυτές μπορείτε να δείτε τους περιορισμούς των ετικετών σε ό, τι αφορά τον τρόπο που αυτά μπορούν σωστά να αξιοποιηθούν (π.χ. τι είδους απογόνους μπορεί ένα στοιχείο να υποδεχτεί ή τον βέλτιστο τρόπο συνδυασμού τους).

## 2.7 <body> : Το πραγματικό περιεχόμενο του εγγράφου

Η HTML σαν γλώσσα, σε αντίθεση με άλλες, ακολουθεί πιο χαλαρούς κανόνες σύνταξης με σκοπό να αποτρέπει την αποτυχία εμφάνισης περιεχομένου. Για παράδειγμα, αν σε ένα μέρος κειμένου δεν ορίσουμε σήμανση το πιο πιθανό είναι να αντιμετωπιστεί και να εμφανιστεί ως παράγραφος.

Επιπλέον, στα αρχεία HTML οι χαρακτήρες επιστροφής (return/enter) αγνοούνται, όπως και όλοι οι χαρακτήρες κενού πλην του πρώτου. Μια καλή πρακτική κατά τη συγγραφή είναι αυτή όπου **η ετικέτα ανοίγματος ακολουθείται αμέσως από την ετικέτα κλεισίματος** (με τους νέους επεξεργαστές κειμένου αυτό επιτυγχάνεται αυτόματα).

Στις επόμενες ενότητες θα αναλύσουμε, μέσα από απλά παραδείγματα, τα στοιχεία με τα οποία μπορούμε να ορίσουμε το περιεχόμενο και τη δομή των εγγράφων. Γενικά, τα στοιχεία αυτά μπορούν να ομαδοποιηθούν σε κατηγορίες όπως:

- μορφοποίησης κειμένου
- ομαδοποίησης κειμένου
- φυσικής ή λογικής διαμόρφωσης χαρακτήρων
- εισαγωγής χαρακτήρων διαφυγής
- εισαγωγής λιστών, συνδέσμων, γραφικών, πινάκων, φορμών και
- σημασιολογικής σχεδίασης δομής του εγγράφου.

Αρχικά θα δούμε ποια είναι η συμπεριφορά ενός browser (default rendering) σε ένα έγγραφο που δεν έχει καμία δομή (σήμανση) παρά μόνο περιεχόμενο.

## Άσκηση 1

Δημιουργήστε ένα κείμενο χρησιμοποιώντας έναν editor (π.χ. Notepad) γράφοντας τις προτάσεις που βρίσκονται στα εισαγωγικά ακριβώς όπως φαίνονται - διατηρώντας τις κενές γραμμές και τα κενά μεταξύ τους. Αποθηκεύστε το έγγραφο (δίνοντάς του την επέκταση .html) και ανοίξτε το με έναν browser της επιλογής σας. Τι παρατηρείτε ;

“Εισαγωγή στην HTML.

Η HTML χρησιμοποιεί tags για να καθορίσει το είδος του περιεχομένου και τη δομή των σελίδων. Για παράδειγμα, αυτό το κείμενο αποτελεί μια παράγραφο!

Και αυτή είναι μια δεύτερη παράγραφος”.

### 2.7.1 Επικεφαλίδες <h1>...<h6>

Οι επικεφαλίδες <h1>...<h6> ανήκουν στην κατηγορία «μορφοποίησης κειμένου». Υπάρχουν 6 επίπεδα επικεφαλίδων αριθμημένα από το 1 ως 6, με το **1<sup>ο</sup> να είναι το πιο σημαντικό** και αυτό που εμφανίζεται με τη μεγαλύτερη γραμματοσειρά. Είναι σημαντικό να θυμάστε πως η χρήση των επικεφαλίδων δίνει δομή (ιεράρχηση/περίγραμμα) στο περιεχόμενο καθιστώντας το πιο «κατανοητό» από ανθρώπους και μηχανές.

## Άσκηση 2

Δημιουργήστε ένα έγγραφο όπως παρακάτω και ανοίξτε το στον φυλλομετρητή. Τι παρατηρείτε σχετικά με την εμφάνιση επικεφαλίδων (μέγεθος και στυλ γραμματοσειράς); Στη συνέχεια, δημιουργήστε κατάλληλο περιεχόμενο που να περιλαμβάνει επικεφαλίδες και παραγράφους αξιοποιώντας επικεφαλίδες επιπέδου 1 και 2. Ελέγξτε το περίγραμμα της σελίδας χρησιμοποιώντας το πρόσθετο Document Outliner.

```
1. <!DOCTYPE html>
2. <html lang="el">
3.   <head>
4.     <meta charset="utf-8">
5.     <title>HTML</title>
6.   </head>
7.   <body>
8.     <h1>Επικεφαλίδα 1</h1>
9.     <h2>Επικεφαλίδα 2</h2>
10.    <h3>Επικεφαλίδα 3</h3>
11.    <h4>Επικεφαλίδα 4</h4>
12.    <h5>Επικεφαλίδα 5</h5>
13.    <h6>Επικεφαλίδα 6</h6>
14.    Αυτή είναι περιεχόμενο (παράγραφος) χωρίς σήμανση!
15.    <p>Ενώ αυτή είναι μια παράγραφος μέσα σε κατάλληλο στοιχείο παραγράφου</p>
16.  </body>
17. </html>
```

Από το πρότυπο της HTML δεν υπάρχει περιορισμός στο πλήθος των <h1> που μπορεί να περιέχει ένα έγγραφο. Επίσης, καλή πρακτική θεωρείται η χρήση συνεχόμενης αρίθμησης – το να μην παραλείπονται επίπεδα επικεφαλίδων π.χ. από το 1<sup>ο</sup> στο 3<sup>ο</sup> και μετά στο 6<sup>ο</sup>.

**Προσοχή!** Θα πρέπει να έχετε πάντα υπόψη πως η επιλογή των επικεφαλίδων θα πρέπει να σχετίζεται με τη σημαντικότητα της ενότητας (π.χ. κεντρικό κεφάλαιο, υποενότητα κτλ.) και όχι με θέματα μορφοποίησης του κειμένου, μιας και αυτά μπορούν να ρυθμιστούν με κανόνες/στυλ CSS.

### 2.7.2 Παράγραφος <p>

Η ετικέτα <p> αποτελεί το πιο βασικό στοιχείο ομαδοποίησης περιεχομένου καθώς σηματοδοτεί την έναρξη νέας παραγράφου. Χωρίς τη χρήση της ετικέτας <p> όλο το κείμενο μιας σελίδας θα αντιμετωπιστεί ως μια τεράστια παράγραφος, όπου οι περιττοί χαρακτήρες κενού (spaces) και οι χαρακτήρες αλλαγών γραμμής αγνοούνται δημιουργώντας ένα όχι και τόσο ελκυστικό οπτικό αποτέλεσμα.

*Παράδειγμα 2.6. Παράδειγμα εγγράφου χωρίς δομή κειμένου*

```
1. <!DOCTYPE html>
2. <html lang="el">
3. <head>
4.   <meta charset="utf-8">
5.   <title>Παράγραφοι και Επικεφαλίδες</title>
6. </head>
7. <body>
8.   Εισαγωγή στην HTML
9.
10.  Η HTML χρησιμοποιεί tags για να καθορίσει το είδος του περιεχομένου και τη
    δομή των σελίδων. Αυτή η πρόταση είναι η δεύτερη αυτής της παραγράφου.
11.  Αυτή η παράγραφος αποτελείται από 2 προτάσεις που είναι στοιχισμένες στο κέντρο
    της οθόνης. Η παράγραφος είναι το πιο σημαντικό στοιχείο, χρησιμοποιήστε το μέχρι
    να σκεφτείτε κάποιο πιο κατάλληλο για το περιεχόμενό σας.
12. </body>
13. </html>
```

Αποτελεί **καλή πρακτική** η τοποθέτηση των επικεφαλίδων σε ξεχωριστή γραμμή, παραθέτοντας με μια κενή γραμμή τις παραγράφους του κειμένου <p> και αντίστοιχα αφήνοντας ένα κενό μεταξύ παραγράφων. Αυτό δημιουργεί ένα αποτέλεσμα που είναι **εύκολα αναγνώσιμο** και οδηγεί σε αυτό που ονομάζουμε «**καλογραμμένος κώδικας**».

Σε παλαιότερες εκδόσεις της HTML υπήρχε η δυνατότητα χρήσης ετικετών που μορφοποιούσαν παραγράφους ή κείμενο (π.χ. στοίχιση περιεχομένου ή αλλαγή γραμματοσειράς με τη χρήση των ετικετών <center> και <font> αντίστοιχα) οι οποίες έχουν καταργηθεί με την HTML5 ([https://developer.mozilla.org/en-US/docs/Web/HTML/Element#Obsolete\\_and\\_deprecated\\_elements](https://developer.mozilla.org/en-US/docs/Web/HTML/Element#Obsolete_and_deprecated_elements)). Η μορφή του περιεχομένου διαμορφώνεται πλέον αποκλειστικά μέσα από κανόνες μορφοποίησης (styles/CSS) όπως παρακάτω:

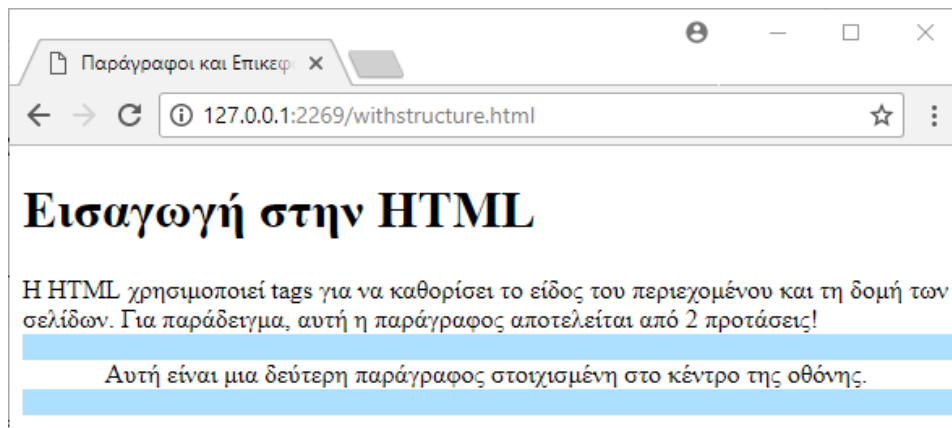
*Παράδειγμα 2.7. Παράδειγμα χρήσης style για στοίχιση κειμένου*

```
<p style="text-align: center;">Αυτή η παράγραφος είναι στοιχισμένη στο κέντρο της
οθόνης. </p>
```

### Άσκηση 3

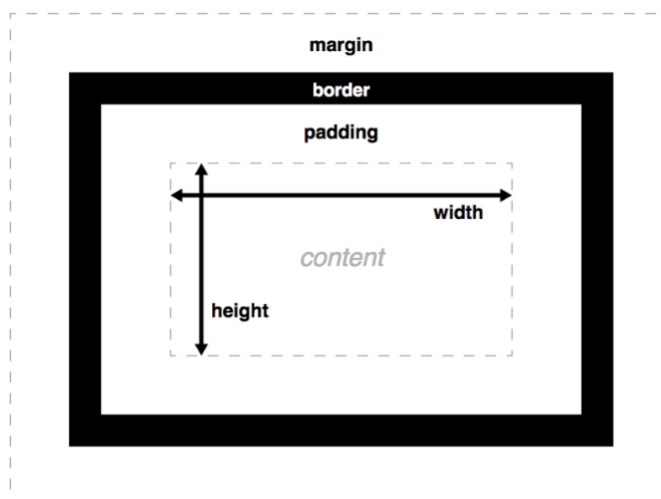
Στο παραπάνω παράδειγμα θα ορίσουμε δομή στο περιεχόμενο χρησιμοποιώντας κατάλληλες ετικέτες παραγράφου και επικεφαλίδας. Επιπλέον, για τη δεύτερη παράγραφο εφαρμόστε κατάλληλη μορφοποίηση στυλ για να στοιχίσετε το κείμενο στο κέντρο.

Θα παρατηρήσετε πως η χρήση παραγράφου συνοδεύεται από την ύπαρξη ενός κενού (margin) μεταξύ αυτής και των γειτονικών στοιχείων. Έτσι διαχωρίζεται το κείμενο της μιας παραγράφου από αυτό της επόμενης προσφέροντας ένα καλύτερο οπτικό αποτέλεσμα. Αυτό συμβαίνει γιατί όλα τα αντικείμενα στην HTML περιβάλλονται από ένα «πλαίσιο», όπως εμφανίζεται μεταξύ των μπλε γραμμών στην παρακάτω εικόνα. Στην Εικόνα 2.4 μπορούμε να δούμε τα χαρακτηριστικά του **box model** ([https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction\\_to\\_CSS/Box\\_model](https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction_to_CSS/Box_model)) τα οποία ρυθμίζουν παραμέτρους εμφάνισης όπως εξωτερικά και εσωτερικά περιθώρια, πλαίσια και διαστάσεις.



Εικόνα 2.3. Επίδειξη διαστημάτων (margins) παραγράφου

**Προσοχή!** Στην περίπτωση που επιθυμούμε να υπάρχει μεγαλύτερο κενό μεταξύ δύο παραγράφων θα πρέπει να αποφύγουμε τη χρήση δημιουργίας κενής παραγράφου <p>, καθώς αυτή δημιουργεί πρόβλημα σε χρήστες βοηθητικών συσκευών. Επιπλέον, αυτή η μορφοποίηση μπορεί εύκολα να επιτευχθεί με κανόνες CSS (margin property).



Εικόνα 2.4. Χαρακτηριστικά Box Model



**Η ετικέτα παραγράφου θα πρέπει να χρησιμοποιείται μόνο όταν αυτή πρόκειται να υποδεχτεί περιεχόμενο!**

### 2.7.3 Αλλαγή Γραμμής <br>

Το στοιχείο αλλαγής γραμμής <br> ανήκει στην κατηγορία «μορφοποίησης κειμένου» και προκαλεί την αλλαγή γραμμής σε περιπτώσεις όπου χρειάζεται τα στοιχεία να εμφανιστούν σε διαφορετικές γραμμές, αλλά αφορούν περιεχόμενο που ανήκει στην ίδια ενότητα (π.χ. παράγραφο).

#### Άσκηση 4

Δημιουργήστε το παρακάτω έγγραφο εισάγοντας κατάλληλα στοιχεία αλλαγών γραμμής <br> για να εμφανιστεί η διεύθυνση με την επιθυμητή μορφή. Ποια θα ήταν η διαφορά αν ορίζαμε κάθε γραμμή της διεύθυνσης ως ξεχωριστή παράγραφο;

```
1. <!DOCTYPE html>
2. <html lang="el">
3.   <head>
4.     <meta charset="utf-8">
5.     <title>Αλλαγή Γραμμής</title>
6.   </head>
7.   <body>
8.     <p>Τμήμα Αρχειονομίας, Βιβλιοθηκονομίας και Συστημάτων Πληροφόρησης
9.       Πανεπιστήμιο Δυτικής Αττικής
10.      Πανεπιστημιούπολη Άλσους Αιγιάλεω
11.      Αιγιάλεω
12.      12243
13.      Ελλάδα
14.   </p>
15. </body>
16. </html>
```

### 2.7.4 Οριζόντια Γραμμή <hr>

Το στοιχείο οριζόντιας γραμμής <hr> ανήκει στην κατηγορία «μορφοποίησης κειμένου» και αναπαριστά μια *θεματική διακοπή* μεταξύ των στοιχείων παραγράφου (π.χ. αλλαγή στο θέμα μέσα σε μια ενότητα ή αλλαγή σκηνικού σε μια ιστορία). Συνήθως, χρησιμοποιείται όταν θέλουμε να διαχωρίσουμε μεγάλα κομμάτια κειμένου μέσα στο έγγραφο (π.χ. οριοθετώντας τμήματα/ενότητες/κεφάλαια) και ως αποτέλεσμα έχει την εισαγωγή μιας οριζόντιας γραμμής η οποία μπορεί να μορφοποιηθεί με κανόνες CSS (π.χ. αλλαγή ύψους/πλάτους κτλ.).

#### Άσκηση 5

Δημιουργήστε το παρακάτω έγγραφο εισάγοντας το στοιχείο οριζόντιας γραμμής (...) για να διαχωρίσετε τη μια ενότητα από την επόμενη.

```
1. <!DOCTYPE html>
2. <html lang="el">
3.   <head>
4.     <meta charset="utf-8">
5.     <title>Οριζόντια Γραμμή</title>
```

```

6.  </head>
7.  <body>
8.    <h1>Ερευνητικό Εργαστήριο "Διαχείρισης Πληροφορίας" </h1>
9.
10.   <h2> Αντικείμενο</h2>
11.   <p>Αντικείμενό του είναι η εφαρμοσμένη έρευνα σε θέματα των τεχνολογιών
    πληροφορικής και των μεθόδων αποτελεσματικής διαχείρισης των πληροφοριών
    ανεξαρτήτως του υποστρώματος στο οποίο είναι καταχωρισμένες.</p>
12.
13.   <!-- οριζόντια γραμμή -->
14.
15.   <h2> Επικοινωνία</h2>
16.   <p>Τμήμα Αρχειονομίας, Βιβλιοθηκονομίας και Συστημάτων Πληροφόρησης
    Πανεπιστήμιο Δυτικής Αττικής
    Πανεπιστημιούπολη Άλσους Αιγάλεω
    Αιγάλεω
    12243
    Ελλάδα
22.   </p>
23. </body>
24. </html>

```

#### 2.7.5 Προκαθορισμένη μορφοποίηση περιεχομένου <pre>

Υπάρχουν περιπτώσεις που απαιτείται η εισαγωγή κειμένου με συγκεκριμένη μορφοποίηση (preformatted) διατηρώντας την ύπαρξη πολλαπλών χαρακτήρων κενού (space/tab) και αλλαγών γραμμής. Σε αυτή την περίπτωση μπορούμε να χρησιμοποιήσουμε την ετικέτα <pre>, η οποία χρησιμοποιεί εξ' ορισμού μια πιο τεχνική γραμματοσειρά (monospace).

#### Άσκηση 6

Τροποποιήστε παρακάτω την παράγραφο της διεύθυνσης κάνοντας χρήση του στοιχείου <pre>. Τι παρατηρείτε στο αποτέλεσμα σχετικά με τη μορφοποίηση του κειμένου και ειδικότερα σε σχέση με τη χρήση των <br>;

```

1.  <!DOCTYPE html>
2.  <html lang="el">
3.  <head>
4.    <meta charset="utf-8">
5.    <title>Οριζόντια Γραμμή</title>
6.  </head>
7.  <body>
8.    Ερευνητικό Εργαστήριο "Διαχείρισης Πληροφορίας"
9.    <p>Τμήμα Αρχειονομίας, Βιβλιοθηκονομίας και Συστημάτων Πληροφόρησης
10.     Πανεπιστήμιο Δυτικής Αττικής
11.     Πανεπιστημιούπολη Άλσους Αιγάλεω
12.     Αιγάλεω
13.     12243
14.     Ελλάδα
15.   </p>
16. </body>
17. </html>

```

**Προσοχή!** Εντός των ετικετών <pre> δε θα πρέπει χρησιμοποιούνται άλλα html tags με εξαίρεση το tag συνδέσμου (<a>) - οπότε το <pre> να προορίζεται για πολύ συγκεκριμένη χρήση π.χ. μικρό

μέγεθος κειμένου χωρίς ιδιαίτερη δομή ή τη δημιουργία πρόχειρων σχεδίων ή πινάκων με χρήση χαρακτήρων όπως |, - κ.λπ.

### 2.7.6 Απόσπασμα/Παράθεση <q> ή <blockquote>

Το στοιχείο <blockquote> είναι ένα από τα στοιχεία ομαδοποίησης περιεχομένου (block-level). Χρησιμοποιείται για να εισάγουμε παραθέσεις/αποσπάσματα κειμένου ιδιαίτερης έκτασης τα οποία εμφανίζονται με μια εσοχή σε σχέση με τη μορφοποίηση της παραγράφου. Το κυριότερο χαρακτηριστικό (attribute) είναι το cite (παραπομπή) το οποίο παραπέμπει στην πηγή προέλευσης (URL) του αποσπάσματος. Για μικρότερης έκτασης παραθέσεις, σε επίπεδο πρότασης, χρησιμοποιείται το στοιχείο q (inline), το οποίο προκαλεί την εμφάνιση του περιεχομένου με εισαγωγικά «».

Παράδειγμα 2.8. Παράδειγμα χρήσης blockquote

```
1. <!DOCTYPE html>
2. <html lang="el">
3. <head>
4. <meta charset="utf-8">
5. <title>Μορφοποίηση blockquote</title>
6. </head>
7. <body>
8. <h1>Albert Einstein</h1>
9. <p>Albert Einstein (14 March 1879 – 18 April 1955) was a German-
  born theoretical physicist who developed the theory of relativity, one of the two
  pillars of modern physics (alongside quantum mechanics). His work is also known
  for its influence on the philosophy of science. </p>
10. <blockquote cite="https://www.brainyquote.com/authors/albert_einstein">
11. <p>Anyone who has never made a mistake has never tried anything new.</p>
12. </blockquote>
13. <footer>-Albert Einstein</footer>
14. </body>
15. </html>
```

**Προσοχή!** Η χρήση του χαρακτηριστικού cite δεν διαφοροποιεί το οπτικό αποτέλεσμα αλλά χρήση του αποτελεί καλή πρακτική αφού μπορεί να αξιοποιηθεί από μηχανές αναζήτησης και άλλες βοηθητικές συσκευές (π.χ. screen readers).

### 2.7.7 Σχόλια

Η HTML υποστηρίζει την εισαγωγή σχολίων τα οποία μπορούν να περιορίζονται σε μία γραμμή ή να περικλείουν ένα σύνολο γραμμών. Το σχόλιο περικλείεται μεταξύ των <!-- και --> με αποτέλεσμα ανάλογα με το πρόγραμμα που χρησιμοποιείται να ξεχωρίζει από τα ενεργά στοιχεία (φαίνεται πιο αχνό ή γκρι). Η βασική σύνταξη φαίνεται παρακάτω:

Παράδειγμα 2.9. Παράδειγμα εισαγωγής σχολίου σε 1 γραμμή

```
<!-- Εδώ γράφετε τα σχόλιά σας -->
```

Τα σχόλια μπορούν να εκτείνονται σε περισσότερες γραμμές και για λόγους αποσφαλμάτωσης μπορεί ένα ολόκληρο μπλοκ γραμμών να αποκλειστεί, ώστε προσωρινά να μην εμφανίζεται.

#### Παράδειγμα 2.10. Παράδειγμα εισαγωγής σχολίου σε μπλοκ γραμμών

1. `<!--`
2. Μην εμφανίσεις την εικόνα και την παράγραφο προς το παρόν (σε αναμονή τελικού γραφικού και κειμένου)
3. `<p> Περιγραφή της εικόνας ... </p>`
4. ``
5. `-->`

**Προσοχή!** Σε αντίθεση με άλλες γλώσσες προγραμματισμού στις οποίες τα σχόλια είναι γνωστά μόνο στον συγγραφέα κατά τη σύνταξη του κώδικα, τα σχόλια της HTML εμφανίζονται όταν ζητείται η εμφάνιση του πηγαίου κώδικα της σελίδας. Απαιτείται ιδιαίτερη προσοχή για το είδος πληροφορίας που εισάγεται ως σχόλιο!

### 2.7.8 Διαμόρφωση Χαρακτήρων

Οι προηγούμενες ενότητες περιγράφουν ετικέτες που μορφοποιούν και ταυτόχρονα ομαδοποιούν περιεχόμενο δημιουργώντας περιοχές περιεχομένου (blocks). Το περιεχόμενο που περιλαμβάνεται σε αυτές μπορεί να διαμορφωθεί περαιτέρω κάνοντας χρήση ετικετών για **λογική** (logical) ή **φυσική** (presentational) μορφοποίηση.

#### 2.7.8.1 Λογική Μορφοποίηση (Logical tags)

Με την HTML5 έγινε περισσότερο επιτακτική η οργάνωση των σελίδων τόσο από πλευράς περιεχομένου όσο και από πλευράς δομής δίνοντας ακόμα μεγαλύτερη έμφαση στη σημασιολογική τους μορφοποίηση. Καλή πρακτική αποτελεί η χρήση ετικετών φυσικής μορφοποίησης όταν δεν υπάρχουν αντίστοιχες ετικέτες λογικής μορφοποίησης για την επίτευξη του τελικού αποτελέσματος. Στην συνέχεια περιγράφουμε τις πιο διαδεδομένες ετικέτες λογικής μορφοποίησης.

W3C Specification HTML5.1 – παραδείγματα για τα στοιχεία μορφοποίησης:

<https://www.w3.org/TR/html51/textlevel-semantic.html#textlevel-semantic>

- **<abbr>...</abbr>**

Χρησιμοποιείται για να δοθούν συντομογραφίες ή **ακρωνύμια** και δέχεται προαιρετικά το στοιχείο title για να αποδοθεί **αποκλειστικά** η πλήρης περιγραφή του.

#### Παράδειγμα 2.11. Παράδειγμα χρήσης <abbr>

1. Μπορείς να χρησιμοποιήσεις `<abbr title="Cascading Style Sheets">CSS</abbr>` για να επιμεληθείς σχεδιαστικά το `<abbr title="HyperText Markup Language">HTML</abbr>` περιεχόμενο.

- **<dfn>...</dfn>**

Χρησιμοποιείται όταν μια λέξη αντιστοιχεί σε **ορισμό** και τυπικά εμφανίζεται με πλάγια γραφή. Καλή πρακτική αποτελεί ο συνδυασμός του με το χαρακτηριστικό title ή ακόμα καλύτερα με το στοιχείο συντομογραφίας <abbr>. Η περιγραφή θα πρέπει να βρίσκεται στο κοντινότερο στοιχείο-γονέα όπως παράγραφος <p>, ενότητα <section> ή ζεύγος <dt>/<dd>.

#### Παράδειγμα 2.12. Παράδειγμα χρήσης <dfn>

1. Ορισμός - dfn
2. - Το `<dfn>dfn</dfn>` χρησιμοποιείται για να δώσουμε ορισμούς.
3. - Το `<dfn title="definitions tag">dfn</dfn>` χρησιμοποιείται για να δώσουμε ορισμούς. Με τη χρήση του title εμφανίζεται tooltip με το πέρασμα του κέρσορα
4. - Το `<dfn><abbr title="definitions tag">dfn</abbr></dfn>` χρησιμοποιείται για να δώσουμε ορισμούς. Με τη χρήση του abbr γίνεται διακριτή η περιοχή του ορισμού

- **<em>...</em>**

Αποτελεί το στοιχείο που δίνει **έμφαση** στο περιεχόμενο και συνήθως εμφανίζεται με πλάγια γραφή. Συνήθως χρησιμοποιείται για να αλλάξει το νόημα μιας πρότασης (π.χ. «Έχει περάσει μόνο μία ώρα, **καλά που δεν άργησες**») ή για να χρησιμοποιηθεί από συσκευές (π.χ. Screen readers) οι οποίες θα αλλάξουν τη χροιά του κειμένου διαβάζοντας το.

#### Παράδειγμα 2.13. Παράδειγμα χρήσης <em>

1. Έμφαση - Em
2. Το στοιχείο em χρησιμοποιείται για δώσουμε έμφαση και συνήθως εμφανίζεται σε `<em>πλάγια γραφή</em>`.

- **<cite>...</cite>**

Χρησιμοποιείται για την αναφορά σε **τίτλο ενός έργου** τέχνης (π.χ. βιβλίο, ταινία, τηλεοπτική σειρά, μουσική, γλυπτό, πίνακας ζωγραφικής κτλ.), περιλαμβάνοντας τον τίτλο του έργου, τον δημιουργό ή μια ηλεκτρονική διεύθυνση URL. Συνήθως εμφανίζεται με πλάγια γραφή.

#### Παράδειγμα 2.14. Παράδειγμα χρήσης <cite>

1. Αναφορά σε τίτλο/έργο - cite
2. Το στοιχείο cite χρησιμοποιείται για αναφερθούμε στον τίτλο ενός έργου (Βιβλίο, Ταινία, Μουσική, Πίνακας Ζωγραφικής, Γλυπτό ...).
3. Ο Bob Ross (Οκτώβριος 1942 - Ιούλιος 1995) ήταν ένας αμερικανός ζωγράφος που δημιούργησε και παρουσίασε τη σειρά `<cite>The Joy of Painting</cite>` - ένα εκπαιδευτικό πρόγραμμα που προβλήθηκε στις ΗΠΑ από το δίκτυο PBS από το 1983 ως το 1994.

- **<code>...</code>**

Χρησιμοποιείται για να παρουσιάσουμε **πηγαίο κώδικα** (source code) και συνήθως εμφανίζεται με γραφή σταθερού πλάτους (monospace).

#### Παράδειγμα 2.15. Παράδειγμα χρήσης <code>

1. Κώδικας υπολογιστών - code
2. Το στοιχείο code χρησιμοποιείται για εισάγουμε κώδικα υπολογιστών (π.χ. γλώσσα προγραμματισμού) και εμφανίζεται με γραμματοσειρά monospace.
3. Η συνάρτηση printCourse() εμφανίζει μιας μορφής αναφορά για το μάθημα

```

4. <code> public void printCourse() {<br>
5.         System.out.println("-----");<br>
6.         System.out.printf ("Course Title: %S", title)<br>
7.         ...<br>
8.     }<br>
9. </code>

```

- **<kbd>...</kbd>**

Χρησιμοποιείται για το περιεχόμενο που αναφέρεται σε **δεδομένα που εισάγονται από το πληκτρολόγιο** (δεδομένα εισόδου) και συνήθως εμφανίζεται με γραφή σταθερού πλάτους (monospace).

*Παράδειγμα 2.16. Παράδειγμα χρήσης <kbd>*

1. Δεδομένα Εισόδου - kbd
2. Το στοιχείο kbd χρησιμοποιείται για δηλωθούν στοιχεία που εισάγει ο χρήστης από το πληκτρολόγιο και εμφανίζεται με γραμματοσειρά monospace.
3. Η συνάρτηση createCourse() δημιουργεί ένα στιγμιότυπο ζητώντας από το χρήστη να εισάγει δεδομένα για τις μεταβλητές (instance variables). Για παράδειγμα για να δηλώσουμε ένα μάθημα ως μεικτό, στην ερώτηση για το αν περιλαμβάνει εργαστηριακό μέρος, ο χρήστης πρέπει να εισάγει την τιμή
4. <code><code> kbd yes </code></code>

- **<samp>...</samp>**

Χρησιμοποιείται όταν θέλουμε να δηλώσουμε ένα μέρος κειμένου όπως θα εμφανίζονταν ως **έξοδος εκτέλεσης** προγράμματος Η/Υ ή δείγμα ενός συνόλου.

*Παράδειγμα 2.17. Παράδειγμα χρήσης <samp>*

1. Έξοδος κατά την εκτέλεση προγράμματος - samp
2. Το στοιχείο samp χρησιμοποιείται για δηλώσουμε ένα το αποτέλεσμα εξόδου από την εκτέλεση ενός προγράμματος και εμφανίζεται με γραμματοσειρά monospace.
3. Η συνάρτηση printCourse() εμφανίζει στην οθόνη την αναφορά του μαθήματος λαμβάνοντας υπόψη τις τιμές (instance variables) που δόθηκαν κατά την κλήση της createCourse()
4. <code><code> samp-----<br>
5. Title:.....Computer Networks<br>
6. Semester:.....2<br>
7. Lab:.....true<br>
8. Theory grade:.....7.88<br>
9. Lab grade:.....6.95<br>
10. Course grade:.....7.5080004<br>
11. -----<br>
12. </code></code>

- **<strong>...</strong>**

Αποτελεί το σημασιολογικό ισοδύναμο του <b> και χρησιμοποιείται για να δηλώσουμε **σημαντικότητα** σε ένα μέρος κειμένου το οποίο θέλουμε να τονίσουμε (π.χ. «Αυτό το υλικό είναι πολύ τοξικό!»). Συνήθως εμφανίζεται με έντονη γραφή (bold).

Παράδειγμα 2.18. Παράδειγμα χρήσης <strong>

1. Σημαντικό κείμενο - strong
2. <p> Χρησιμοποιείται όταν θέλουμε να δώσουμε την αίσθηση ότι το κείμενο που περικλείει είναι <strong> σημαντικό </strong> </p>.

- <var>...</var>

Χρησιμοποιείται για να δηλώσουμε την ύπαρξη μιας **μεταβλητής** (variable) και συνήθως εμφανίζεται με πλάγια γραφή.

Παράδειγμα 2.19. Παράδειγμα χρήσης <var>

1. Η μεταβλητή <var>tryAgain</var> είναι τύπου boolean και χρησιμοποιείται ως συνθήκη ελέγχου εξόδου από τους βρόχους λογικών ελέγχων των δεδομένων.

- <address>...</address>

Χρησιμοποιείται όταν θέλουμε να εμφανίσουμε στοιχεία επικοινωνίας (π.χ. του δημιουργού/διαχειριστή του περιεχομένου ή του οργανισμού/επιχείρησης του ιστότοπου) και συνήθως εμφανίζεται με πλάγια γράμματα. Η διεύθυνση μπορεί να είναι οποιασδήποτε μορφής π.χ. φυσική, ηλεκτρονική (URL), ηλεκτρονικού ταχυδρομείου ακόμα και σε μορφή γεωγραφικών συντεταγμένων.

Παράδειγμα 2.20. Παράδειγμα χρήσης <address>

1. Στοιχεία Επικοινωνίας - address
2. Χρησιμοποιείται όταν θέλουμε να εμφανίσουμε στοιχεία επικοινωνίας (π.χ. του διαχειριστή/δημιουργού του περιεχομένου) και συνήθως εμφανίζεται με πλάγια γράμματα.
3. <address>
4. Μπορείτε να επικοινωνήσετε με τη <a href="http://www.teiath.gr/sdo/lis/categories.php?id=6378&lang=el">Γραμματεία</a>.<br>
5. Για σημαντικά ζητήματα μπορείτε να απευθυνθείτε στον πρόεδρο του τμήματος <a href="http://www.teiath.gr/sdo/lis/articles.php?id=6379&lang=el">Καθ. XYZ</a>.<br>
6. Τέλος, μπορείτε να μας επισκεφθείτε:<br>
7. Πανεπιστήμιο Δυτικής Αττικής <br>
8. Πανεπιστημιούπολη 1 <br>
9. Αγ. Σπυρίδωνος <br>
10. 122 43 <br>
11. Αιγάλεω <br>
12. </address>

- <sub>...</sub> και <sup>...</sup>

Χρησιμοποιείται για να εμφανίσει το κείμενο με μορφή δείκτη (subscript) ή εκθέτη (superscript).

Παράδειγμα 2.21. Παράδειγμα χρήσης Δείκτη και Εκθέτη

1. Δείκτης και Εκθέτης - sub και sup
2. Οι συντεταγμένες των στοιχείων στον οριζόντιο άξονα X αναπαρίστανται με τις μεταβλητές  $x_{sub>1</sub>}$  ...  $x_{sub>n</sub>}$ .
- 3.

4. Ίσως η πιο διαδεδομένη εξίσωση στον κόσμο είναι η  $E=mc^2$ .

- **<mark>...</mark>**

Αποτελεί μια νέα ετικέτα της HTML5 και χρησιμοποιείται για τον τονισμό (highlight) ενός μέρους του κειμένου. Μαζί με τα <strong> και <em> είναι τα πιο συχνά χρησιμοποιούμενα tags για να δοθεί έμφαση/ένταση/τονισμός στο κείμενο.

*Παράδειγμα 2.22. Παράδειγμα χρήσης highlight κειμένου*

1. Τονισμός/highlight κειμένου - mark
2. Βασικές προϋποθέσεις για την επιτυχή ολοκλήρωση ενός μαθήματος είναι η τακτική παρακολούθηση, η εκπρόθεσμη υποβολή των εργασιών και η <mark>δήλωση του μαθήματος </mark>!

- **<del>...</del> και <ins>...</ins>**

Αποτελούν στοιχεία με τα οποία μπορεί κάποιος να επισημάνει **αλλαγές** (αναθεωρήσεις) που έχουν πραγματοποιηθεί στο κείμενο π.χ. **προσθήκες** (insertions) και **διαγραφές** (deletions). Δίνει την αίσθηση της παρακολούθησης αλλαγών όπως αυτές επισημαίνονται στο MSWord. Χαρακτηριστικά με τα οποία συνδυάζεται είναι τα cite (URI που παραπέμπει στην πηγή των αλλαγών π.χ. πρακτικά συνάντησης) και datetime (καταγράφει τη χρονική στιγμή πραγματοποίησης της αλλαγής).

*Παράδειγμα 2.23. Παράδειγμα επισήμανσης αλλαγών κειμένου*

1. <h2>Επισήμανση αλλαγών κειμένου - ins/del</h2>
2. <p> <ins>Αυτή η παράγραφος είναι νέα.</ins></p>
3. <p><del>Μόνο αυτό το αρχικό μέρος του κειμένου έχει διαγραφεί</del> Η συνέχεια της παραγράφου παραμένει ως έχει.</p>
4. <del><p>Τέλος, αυτή η παράγραφος έχει διαγραφεί!</p></del>

- **<time>...</time>**

Αποτελεί νέα προσθήκη της HTML5 και χρησιμοποιείται για να δηλωθούν χρονικά δεδομένα σε αναγνώσιμη μορφή. Επιπλέον, η χρήση του χαρακτηριστικού datetime επιτρέπει τη δήλωση μιας μηχανικά αναγνώσιμης μορφή η οποία μπορεί να αξιοποιηθεί από agents (π.χ. για να προτείνουν την εισαγωγή εγγραφών στο ημερολόγιο ή για να ενεργοποιήσουν reminders σε προγραμματισμένα γεγονότα).

*Παράδειγμα 2.24. Παράδειγμα επισήμανσης χρονολογικών δεδομένων*

1. <h2> Σήμανση χρονικών δεδομένων </h2>
2. <p> Η ετικέτα time αποτελεί νέα προσθήκη της HTML5 και χρησιμοποιείται για να οριστεί μια χρονική τιμή σε αναγνώσιμη μορφή.</p>
3. <p> Το εργαστήριο της Παρασκευής θα ξεκινά στις <time>16.00</time></p>
4. <p>Την Τετάρτη θα πραγματοποιηθεί <time datetime="2018-05-02 18:30">Εκδήλωση</time> στο Συνεδριακό Κέντρο.</p>



### 2.7.8.2 Φυσική Μορφοποίηση (Physical tags)

Οι ετικέτες που αναφέρθηκαν έχουν μια έντονη σημασιολογική συσχέτιση με το περιεχόμενο που περικλείουν. Αντίθετα, οι ετικέτες φυσικής μορφοποίησης περιορίζονται καθαρά για μορφοποιήσεις παρουσίασης (presentational elements) προσφέροντας:

- **έντονη** γραφή (<b>...</b>) – αντίστοιχη λογική ετικέτα <strong>...</strong>
- **πλάγια** γραφή (<i>...</i>) – αντίστοιχη λογική ετικέτα <em>...</em>
- **υπογραμμισμένη** γραφή (<u>...</u>)

Παράδειγμα 2.25. Παράδειγμα χρήσης ετικετών φυσικής μορφοποίησης

```
1. <!DOCTYPE html>
2. <html lang="el">
3. <head>
4. <meta charset="utf-8">
5. <title>Φυσική Μορφοποίηση Περιεχομένου</title>
6. </head>
7. <body>
8.
9. <h1> Ετικέτες Φυσικής Μορφοποίησης</h1>
10. <p>Σε αυτή την ενότητα θα εξετάσουμε τις ετικέτες φυσικής μορφοποίησης με τις οποίες μπορούμε να αλλάξουμε τη μορφή του κειμένου χωρίς να του προσδίδουμε σημασιολογική διάσταση. </p>
11.
12. Γραφή Έντονη/σημαντική - (b vs strong)
13. Χρησιμοποιείται για να μορφοποιηθεί το κείμενο με <b>έντονη γραφή</b>, σε αντίθεση αντίθεση με την ετικέτα strong η οποία έχει το ίδιο οπτικό αποτέλεσμα αλλά και τονίζει τη σημαντικότητα του κειμένου.
14.
15. Γραφή Πλάγια/έμφαση - (i vs em)
16. Χρησιμοποιείται για να μορφοποιηθεί το κείμενο με <i>πλάγια γραφή</i>, σε αντίθεση με την ετικέτα em η οποία έχει το ίδιο οπτικό αποτέλεσμα αλλά επιπλέον προσδίδει έμφαση στο κείμενο
17.
18. Γραφή υπογράμμισης - u
19. Χρησιμοποιείται για να μορφοποιηθεί το κείμενο με <u>υπογραμμισμένη γραφή </u>
20. </body>
21. </html>
```

Στο παρελθόν υπήρχε μια τάση απομάκρυνσης από τα φυσικά tags (π.χ. <b>, <i>) με αποτέλεσμα οι συγγραφείς να προτιμούν τα λογικά (π.χ. <strong>, <em>) ισοδύναμά τους. Αυτά επιτύγχαναν το ίδιο αισθητικό αποτέλεσμα, ενώ ταυτόχρονα ενίσχυαν τη σημασιολογική διάσταση του περιεχομένου επιτρέποντας στους ανάλογους agents (π.χ. Screen readers) να αξιοποιήσουν την πληροφορία *αλλάζοντας τη χροιά όπου ήταν απαραίτητο για να αποδώσουν τη σημαντικότητα/έμφαση του κειμένου*. Υπάρχουν όμως περιπτώσεις που τα στοιχεία φυσικής μορφοποίησης θεωρούνται **κατάλληλα** για:

- Αναφορά επιστημονικών ονομάτων <i>...</i>
- Αναφορά ξένων λέξεων <i lang="uk-latn">...</i>

- Αναφορά ορθογραφικών λαθών <u>...</u>
- Επισήμανση οδηγιών (π.χ. εκτέλεση μιας συνταγής) <b>...</b>

## Άσκηση 7

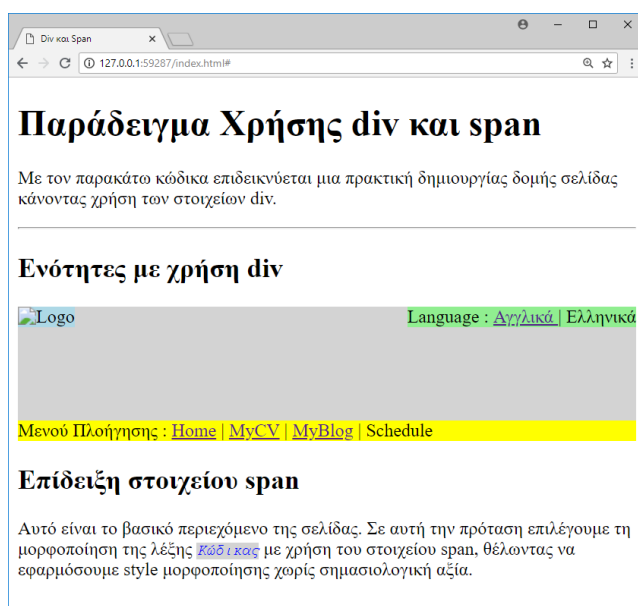
Με βάση τα παραπάνω παραδείγματα λογικής και φυσικής μορφοποίησης, δημιουργήστε για κάθε στοιχείο HTML5 το δικό σας παράδειγμα. Τα παραδείγματα θα πρέπει να αποθηκευτούν στο ίδιο αρχείο δημιουργώντας μια κατάλληλη δομή (με χρήση επικεφαλίδων που να περιλαμβάνουν το όνομα του στοιχείου π.χ. Παράδειγμα time) λαμβάνοντας υπόψη τη μορφή του κειμένου που δημιουργήσατε. Μπορείτε να διαχωρίσετε θεματικά τα περιεχόμενα κάνοντας χρήση του στοιχείου <hr>.

### 2.7.9 Τα στοιχεία <span> και <div>

Η συγγραφή ενός HTML εγγράφου πρέπει εκμεταλλεύεται πρωτίστως τα σημασιολογικά «στοιχεία» που δίνει η γλώσσα. Ενώ τα στοιχεία που έχουν αναφερθεί διαθέτουν σημασιολογικό φορτίο (π.χ. <p> = Paragraph, <h1> = Document Heading), πολλοί συνεχίζουν να χρησιμοποιούν εργαλεία που δε διαθέτουν σημασιολογική βάση, αλλά χρησιμοποιήθηκαν στο παρελθόν για να επηρεάσουν σχεδιαστικά - με τη χρήση κανόνων μορφοποίησης (styles) - το περιεχόμενο, όπως:

- **div**: Αποτελεί ένα block-level στοιχείο το οποίο χρησιμοποιείται για να ομαδοποιήσει περιεχόμενο, κυρίως για τη μορφοποίηση της δομής του εγγράφου. Με τη χρήση των χαρακτηριστικών id και class ορίζονται λογικές ενότητες μέσα στο κείμενο (π.χ. header, main, footer)
- **span**: Αποτελεί ένα inline-level στοιχείο το οποίο χρησιμοποιείται για να μορφοποιηθεί ένα μέρος του κειμένου με σκοπό να εφαρμόσουμε με ενιαίο τρόπο κάποιο στυλ μορφοποίησης.

Στο επόμενο παράδειγμα βλέπουμε τη χρήση των στοιχείων div και span, όπου για να γίνει διακριτή η σχετική τους θέση, χρησιμοποιήθηκαν κανόνες (styles) στο <head> του εγγράφου.



Εικόνα 2.5. Χρήση στοιχείων <div> και <span>

Παράδειγμα 2.26. Παράδειγμα χρήσης ετικετών <div> και <span>

```
1. <!DOCTYPE html>
2. <html lang="el">
3. <head>
4.   <meta charset="utf-8">
5.   <title>Div και Span</title>
6.   <style>
7.     #header {
8.       width: 100%;
9.       height: 100px;
10.      background-color: lightgray;
11.    }
12.
13.    #logo {
14.      float: left;
15.      background-color: lightblue;
16.    }
17.
18.    #languageBar {
19.      background-color: lightgreen;
20.      float: right;
21.    }
22.
23.    #navigationBar {
24.      background-color: yellow;
25.    }
26.
27.    .myCode {
28.      font-family: monospace;
29.      font-style: italic;
30.      color: blue;
31.      background-color: lightgray;
32.    }
33.  </style>
34. </head>
35. <body>
36.
37.   <h1> Παράδειγμα Χρήσης div και span </h1>
38.   <p> Με τον παρακάτω κώδικα επιδεικνύεται μια πρακτική δημιουργίας δομής σελίδας
    κάνοντας χρήση των στοιχείων div. </p>
39.   <hr>
40.   <h2> Ενότητες με χρήση div</h2>
41.   <div id="header">
42.     <div id="logo">
43.       
44.     </div>
45.     <div id="languageBar">
46.       Language : <a href="#">Αγγλικά </a> | Ελληνικά
47.     </div>
48.   </div>
49.
50.   <div id="navigationBar">
51.     Μενού Πλοήγησης : <a href="#">Home</a> | <a href="#">MyCV</a> | <a href="#">MyBlog</a> | Schedule
52.   </div>
53.
54.   <div id="main">
55.     <h2> Επίδειξη στοιχείου span</h2>
56.     <p>Αυτό είναι το βασικό περιεχόμενο της σελίδας. Σε αυτή την πρόταση επιλέγουμε τη μορφοποίηση της λέξης <span class="myCode">Κώδικας</span> με χρήση του στοιχείου span, θέλωντας να εφαρμόσουμε style μορφοποίησης χωρίς σημασιολογική αξία.
57.   </div>
58. </body>
59. </html>
```

### 2.7.10 Χαρακτήρες Διαφυγής (Named Character Entities)

Λόγω της φύσης της γλώσσας και του τρόπου που οι φυλλομετρητές δρουν προστατευτικά στην μορφοποίηση και εμφάνιση του περιεχομένου, υπάρχουν χαρακτήρες που δεν επιτρέπεται να χρησιμοποιηθούν ως μέρος του περιεχομένου. Για παράδειγμα, ορισμένες φορές χρειάζεται να εμφανίσουμε σύμβολα που αποτελούν μέρος του συντακτικού της HTML (" $<$ ", " $>$ ") και η χρήση τους θα προκαλούσε διαταραχή στο περιεχόμενο. Επίσης, υπάρχουν περιπτώσεις όπου χαρακτήρες δεν μπορούν να εισαχθούν με τη βοήθεια του πληκτρολογίου καθώς αποτελούν εξειδικευμένα σύμβολα ή γραφικά.

Για αυτές τις περιπτώσεις η HTML διαθέτει ακολουθίες χαρακτήρων γνωστούς ως *χαρακτήρες διαφυγής* (Named Character Entities), για να διαχειριστούν περιεχόμενο που περιέχουν τα παρακάτω σύμβολα:

- Το αριστερό άγκιστρο/μικρότερο ( $<$ )
- Το δεξί άγκιστρο/μεγαλύτερο ( $>$ )
- Το σύμβολο ampersand ( $&$ )

Ο φυλλομετρητής αναγνωρίζοντας τα σύμβολα της ακολουθίας τα αντικαθιστά με το αντίστοιχο τελικό σύμβολο. Η σύνταξη του ξεκινά πάντα με το ampersand (" $&$ ") και ολοκληρώνεται με το colon (" $;$ "), όπως στον παρακάτω πίνακα:

Πίνακας 2.6. Παραδείγματα NCE

Σύμβολο	Περιγραφή	Ακολουθία χαρακτήρων
	non-breaking space	$\&\text{nbsp};$ ( $\&\#160;$ )
$<$	μικρότερο από	$\&\text{lt};$ ( $\&\#60;$ )
$>$	μεγαλύτερο από	$\&\text{gt};$ ( $\&\#62;$ )
$&$	ampersand	$\&\text{amp};$ ( $\&\#38;$ )
$"$	διπλά εισαγωγικά	$\&\text{quot};$ ( $\&\#34;$ )
$'$	μονά εισαγωγικά	$\&\text{apos};$ ( $\&\#39;$ )
$\text{€}$	euro	$\&\text{euro};$ ( $\&\#8364;$ )
$\text{©}$	copyright	$\&\text{copy};$ ( $\&\#169;$ )
$\text{®}$	registered trademark	$\&\text{reg};$ ( $\&\#174;$ )

Δεν υπάρχει για όλα τα σύμβολα η ονομασία *entity* ( $\&\text{όνομα}$ ), γι' αυτό το λόγο μπορούν τα περισσότερα σύμβολα να κληθούν με τον **δεκαδικό κωδικό** τους όπως εμφανίζεται στην παρένθεση.

Από τον παραπάνω πίνακα θα ξεχωρίσουμε την ακολουθία  $\&\text{nbsp};$  που σχετίζεται με τη διαχείριση κενών (whitespace) και μπορεί να χρησιμοποιηθεί για:

- Να εισαχθούν πολλαπλά κενά τα οποία θα εμφανιστούν υποχρεωτικά, παρακάμπτοντας τον κανόνα που επιβάλλει τα κενά πέρα του πρώτου να αγνοούνται.
- Να συνδέσουμε μια φράση που αποτελείται από περισσότερες της μιας λέξης εξασφαλίζοντας πως αυτές δε θα διαχωριστούν σε διαφορετικές γραμμές, αλλά θα βρίσκονται πάντα η μια δίπλα στην άλλη.

Οι ακολουθίες διαφυγής μπορούν να προκαλέσουν τη δημιουργία εξειδικευμένων μαθηματικών, νομισματικών, σχεδιαστικών, γεωμετρικών συμβόλων ή γραφικών όπως παρακάτω ([https://www.w3schools.com/html/html\\_symbols.asp](https://www.w3schools.com/html/html_symbols.asp)) :

Πίνακας 2.7. Στοιχεία/Σύμβολα γραφικών

Σύμβολο	Περιγραφή	Ακολουθία χαρακτήρων
←	αριστερό βέλος	&larr; (&#8592;)
→	δεξί βέλος	&rarr; (&#8594;)
↑	πάνω βέλος	&uarr; (&#8593;)
↓	κάτω βέλος	&darr; (&#8595;)
∃	υπάρχει	&exist; (&#8707;)
∀	για κάθε	&forall; (&#8704;)
∈	ανήκει	&isin; (&#8712;)
∉	δεν ανήκει	&notin; (&#8713;)
Σ	άθροισμα	&sum; (&#8721;)
∞	άπειρο	&infin; (&#8734;)

Τέλος, μπορούν να χρησιμοποιηθούν για την εμφάνιση ειδικών χαρακτήρων όπως ορίζονται σε κάθε ξένη γλώσσα όπως :

- &ampuml - Οριζόντια διαλυτικά (Umlaut) π.χ. &ouml; → ö
- &tilde; - Περισπωμένες π.χ. &ntilde; → ñ
- &grave; - Τόνοι πάνω σε κεφαλαία π.χ. &Egrave; → È

**Προσοχή!** Σε αντίθεση με τη γραφή των ετικετών, οι ακολουθίες χαρακτήρων διαφυγής επηρεάζονται από τη χρήση πεζών/κεφαλαίων.

#### Άσκηση 8

Στην προηγούμενη άσκηση δημιουργήθηκε με δικά σας παραδείγματα ένας οδηγός χρήσης των ετικετών λογικής και φυσικής μορφοποίησης της HTML5. Στην επικεφαλίδα του παραδείγματος εμφανίζεται το όνομα της ετικέτας χωρίς τις γωνιακές αγκύλες < . >. Διορθώστε το περιεχόμενο ώστε να μπορούν να εμφανιστούν οι ετικέτες με την πλήρη μορφή π.χ. <time> κάνοντας χρήση των κατάλληλων χαρακτήρων διαφυγής όπου χρειάζεται.

#### 2.7.11 Λίστες <ul>, <ol>

Οι λίστες αποτελούν έναν αποτελεσματικό τρόπο οργάνωσης και παρουσίασης της πληροφορίας με τα περισσότερα εργαλεία συγγραφής (κειμενογράφοι, προγράμματα διαχείρισης ηλεκτρονικής αλληλογραφίας κτλ.) να προσφέρουν υποστήριξη σε αυτές. Έτσι και η HTML προσφέρει τη δυνατότητα ορισμού λιστών όπως:

- **Λίστες ορισμών** (definition lists) – εισάγονται με την ετικέτα <dl>
- **Μη αριθμημένες λίστες** (unordered lists) – εισάγονται με την ετικέτα <ul>
- **Αριθμημένες λίστες** (ordered lists) – εισάγονται με την ετικέτα <ol>

Με συνδυασμό των παραπάνω στοιχείων και με εμφώλευση (nesting) μπορούμε να δημιουργήσουμε λίστες πολλαπλών επιπέδων.

#### 2.7.11.1 Λίστες Ορισμών

Η λίστα ορισμών αποτελεί μια δομή που συνδέει όρους με τις αντίστοιχες περιγραφές τους. Η δημιουργία λίστας ορισμών πραγματοποιείται με την ετικέτα `<dl>...</dl>`. Τα στοιχεία της λίστας (όροι) ορίζονται με την ετικέτα `<dt>...</dt>` (**term**) και η περιγραφή τους με την ετικέτα `<dd>...</dd>` (**description**). Οι λίστες ορισμών προσφέρουν τη δυνατότητα καταχώρησης τόσο πολλαπλών όρων (terms) όσο και πολλαπλών περιγραφών (descriptions).

Παράδειγμα 2.27. Παράδειγμα Λίστας Ορισμών

```
1. <dl>
2.   <dt> Λίστες Ορισμών </dt>
3.   <dd>Αποτελούν λίστες που αποτελούνται από 2 μέρη: τον όρο (term) και την
      περιγραφή του (description)</dd>
4.
5.   <dt>Μη αριθμημένες Λίστες</dt>
6.   <dd>Αποτελούν λίστες τα στοιχεία των οποίων δεν έχουν ορισμένη σειρά
      (αρίθμηση). Με CSS μπορεί να ρυθμιστεί το γραφικό που προηγείται του
      στοιχείου.</dd>
7.
8.   <dt>Αριθμημένες Λίστες </dt>
9.   <dd>Αποτελούν λίστες τα στοιχεία των οποίων έχουν ορισμένη σειρά (αρίθμηση).
      Με CSS μπορεί να ρυθμιστεί το γραφικό που προηγείται του στοιχείου καθώς και
      ο αριθμός έναρξης της αρίθμησης.</dd>
10. </dl>
```

#### Άσκηση 9

Δημιουργήστε τη δική σας λίστα ορισμών για 5 αντικείμενα του ενδιαφέροντός σας αξιοποιώντας τα στοιχεία `<dl>`, `<dt>` και `<dd>`. Εμπλουτίστε τη λίστα με συντομογραφίες `<abbr>` και άλλα στοιχεία λογικής μορφοποίησης για να ενισχύσετε σημασιολογικά το περιεχόμενο όπου χρειάζεται.

Τέλος, μια τυπική χρήση των λιστών ορισμών είναι για την εμφάνιση μεταδεδομένων ως λίστα ζευγών κλειδιού-τιμής (“key-value” pair) όπως:

Παράδειγμα 2.28. Παράδειγμα Λίστας Ορισμών εμφάνιση ζευγών «χαρακτηριστικού -τιμής»

```
1. <dl>
2.   <dt>Name</dt>
3.   <dd>Godzilla</dd>
4.
5.   <dt>Born</dt>
6.   <dd>1952</dd>
7.
8.   <dt>Birthplace</dt>
9.   <dd>Japan</dd>
10.
11.  <dt>Color</dt>
```

12. `<dd>Green</dd>`
13. `</dl>`

#### 2.7.11.2 Μη αριθμημένες λίστες

Η μη αριθμημένη λίστα (unordered list) ορίζεται με την ετικέτα `<ul>...</ul>`, ενώ τα στοιχεία της με τις ετικέτες `<li>...</li>` (list items). Η τελική ετικέτα `</li>` δεν είναι υποχρεωτική αλλά θεωρείται καλή πρακτική να την χρησιμοποιούμε γιατί βελτιώνει την αναγνωσιμότητα του κώδικα.

#### Παράδειγμα 2.29. Παράδειγμα ορισμού μη αριθμημένης λίστας

1. `<ul>`
2. `<li>Εισαγωγή στην Επιστήμη των Υπολογιστών (Α' Εξάμηνο)</li>`
3. `<li>Δίκτυα Υπολογιστών (Β' Εξάμηνο)</li>`
4. `<li>Βάσεις Δεδομένων (Δ' Εξάμηνο)</li>`
5. `<li>Διαχείριση Γνώσης (Ε' Εξάμηνο)</li>`
6. `<li>Ηλεκτρονική Εκδοτική (Ε' Εξάμηνο)</li>`
7. `<li>Ανάλυση και Σχεδιασμός Συστημάτων Πληροφόρησης (ΣΤ' Εξάμηνο)</li>`
8. `<li>Εφαρμογές στον Παγκόσμιο Ιστό (Ζ' Εξάμηνο)</li>`
9. `</ul>`

Εξ' ορισμού το γραφικό στοιχείο που προηγείται των δεδομένων της λίστας είναι ένα “bullet” (μικρή μαύρη συμπαγής κουκίδα), το οποίο με κανόνα/style CSS (list-style-type) μπορεί να οριστεί σε **disc**, **circle**, **square** ή **none**.

```
<ul style="list-style-type:circle">
```

#### 2.7.11.3 Αριθμημένες λίστες

Η αριθμημένη λίστα (ordered list) ορίζεται με την ετικέτα `<ol>...</ol>` και τα στοιχεία της (list items), όπως και στη μη αριθμημένη λίστα, προκύπτουν από τη χρήση ετικετών `<li>...</li>`. Εύκολα μπορούμε να μεταβούμε από μια μη αριθμημένη σε αριθμημένη λίστα αλλάζοντας την κεντρική ετικέτα σε `<ol>...</ol>`.

Η αριθμημένη λίστα προσφέρει επιπλέον χαρακτηριστικά μέσω των οποίων μπορεί να μορφοποιηθεί περαιτέρω:

- Ο αριθμός έναρξης της αρίθμησης (χαρακτηριστικό value)
- Ο τύπος αρίθμησης, χαρακτηριστικό type με αποδεκτές τιμές: {**1**, **A**, **a**, **I**, **i**} → {αριθμοί, γράμματα (πεζά ή κεφαλαία), ρωμαϊκή αρίθμηση (πεζά ή κεφαλαία)}.

#### Άσκηση 10

Ενσωματώστε το παραπάνω παράδειγμα λίστας σε ένα αρχείο HTML και μετατρέψτε τη μη αριθμημένη λίστα σε αριθμημένη. Τα στοιχεία της λίστας θα πρέπει να ακολουθούν την λατινική αρίθμηση (με πεζούς χαρακτήρες) π.χ. i, ii, ...

#### 2.7.11.4 Εμφώλευση Λιστών

Η HTML προσφέρει τη δυνατότητα εμφώλευσης λιστών, δηλαδή την ενσωμάτωση μιας λίστας μέσα σε άλλη, επιτρέποντας τη δημιουργία λιστών πολλαπλών επιπέδων (nested lists). Στο

επόμενο παράδειγμα, δημιουργούμε μια λίστα εξαμήνων (1<sup>ο</sup> επίπεδο) και για κάθε εξάμηνο μια λίστα με τα μαθήματα του εξαμήνου (2<sup>ο</sup> επίπεδο). Κάθε στοιχείο `<li>...</li>` μπορεί να περιέχει άλλες λίστες, ή άλλα στοιχεία όπως εικόνες και συνδέσμους.

*Παράδειγμα 2.30. Παράδειγμα ορισμού πολυεπίπεδης λίστας*

```
1. <ul>
2.   <li> A' Εξάμηνο
3.     <ol>
4.       <li>Εισαγωγή στην Επιστήμη των Υπολογιστών (A' Εξάμηνο)</li>
5.       ...
6.     </ol>
7.   </li>
8.   ...
9. </ul>
```

#### Άσκηση 11

Δημιουργήστε μια αριθμημένη λίστα εξαμήνων (A,C,D...) στην οποία θα εμφανίσετε τα μαθήματα της σχολής σε μορφή μη αριθμημένης λίστας (bullets). Για κάθε μάθημα θα εμφανίσετε σε μορφή μη αριθμημένης λίστας (bullets) τα λογισμικά που απαιτούνται για την εκπόνηση των ασκήσεων και την παρακολούθηση του μαθήματος γενικότερα.

#### 2.7.12 Μονοπάτια-Διαδρομές (Paths) & URLs

Η έννοια του **URL** (Uniform Resource Locator) είναι απλή και σχετίζεται με την τοποθεσία ενός πόρου στο Web (π.χ. η νέα κεντρική ιστοσελίδα του Πα.ΔΑ είναι η <http://www.uniwa.gr>). Τα URLs βασίζονται στην έννοια της **διαδρομής** για τον εντοπισμό αρχείων/πόρων, οι οποίοι μπορούν να είναι μεταξύ άλλων:

1. Άλλες σελίδες του site
2. Γραφικά-Εικόνες
3. Στοιχεία μορφοποίησης CSS
4. Προγραμματιστικά στοιχεία JavaScript

Έτσι μια **διαδρομή/μονοπάτι** (path) ενός αρχείου/πόρου σχετίζεται με τα «βήματα» που πρέπει να ακολουθήσουμε στην δομή αρχείων του site, ώστε το αρχείο που επεξεργαζόμαστε να μπορεί να «δει» και να αξιοποιήσει τους πόρους που χρειάζεται.

Κάθε ιστοσελίδα αποτελείται από ένα **ρίζικό κατάλογο** (root) κάτω από τον οποίο οργανώνεται όλη η δομή αρχείων που έχουμε αναφέρει σε αρχική ενότητα (1.1). Έτσι είναι σημαντικό να γνωρίζουμε πώς μπορούμε να πλοηγηθούμε στα επίπεδα της οργάνωσης μέσα από τον κώδικα των στοιχείων των σελίδων που συγγράφουμε.

«Πώς μπορούμε να οργανώσουμε αποτελεσματικά τις διαδρομές για διαφορετικούς τύπους πόρων; Πώς μπορούμε να κατευθύνουμε τον φυλλομετρητή ώστε να εντοπίσει τον πόρο που χρειαζόμαστε λαμβάνοντας υπόψη την τρέχουσα τοποθεσία του αρχείου που επεξεργαζόμαστε;»



Συχνά προτιμάμε τα αρχεία κειμένου (HTML) ενός site να βρίσκονται οργανωμένα είτε σε ένα κεντρικό κατάλογο είτε σε μια θεματικά οργανωμένη δομή καταλόγων. Στην περίπτωση διαχείρισης άλλων πόρων (π.χ. γραφικά, εικόνες, ήχοι, CSS, JavaScript) επιλέγουμε την οργάνωσή τους σε ξεχωριστούς καταλόγους. Έτσι, θα πρέπει να αναζητάμε τον πόρο στους οργανωμένους καταλόγους του site βάσει της κατηγορίας τους.

Τα σύγχρονα εργαλεία ανάπτυξης προσφέρουν βοήθεια στην αναζήτηση και δημιουργία των διαδρομών κατά τη συγγραφή των αρχείων HTML. Η διαδρομή (path) από το τρέχον αρχείο στον πόρο που θέλουμε να χρησιμοποιήσουμε μπορεί να οριστεί με 2 τρόπους:

- **Απόλυτη Αναφορά** στην θέση του πόρου η οποία θα μπορούσε να βασίζεται:
  - στον **ριζικό κατάλογο** του site (σύστημα αρχείων) π.χ.  
"site/\_images/logo.png" αν θεωρήσουμε πως ο ριζικός κατάλογος της ιστοσελίδας είναι ο site, κατευθύνουμε τον φυλλομετρητή στην τοποθεσία \site\\_images για να αναζητήσει ένα αρχείο με όνομα logo.png.
  - στην URL διεύθυνση π.χ.  
"http://www.teiath.gr/images/top\_logo\_teia\_el.jpg", ξεκινώντας από τη κεντρική διεύθυνση του site ο φυλλομετρητής ακολουθεί τη δομή καταλόγων και εντοπίζει μέσα σε αυτή (π.χ. κατάλογος εικόνων images) το ζητούμενο αρχείο.
- **Σχετική αναφορά** στην θέση του πόρου βάσει της θέσης του τρέχοντος αρχείου. Αυτό σημαίνει πως η θέση του αρχείου που επεξεργαζόμαστε θα καθορίσει την κατεύθυνση προς την οποία θα κινηθούμε για να εντοπίσουμε τον πόρο. Για παράδειγμα:
  - Ο πόρος βρίσκεται στον **ίδιο κατάλογο** με το υπό-επεξεργασία αρχείο, οπότε η καταγραφή του ονόματος του πόρου αρκεί για να εντοπιστεί.
  - Ο πόρος βρίσκεται σε **μεγαλύτερο βάθος**, οπότε θα ακολουθήσουμε τη διαδρομή συμπληρώνοντας επίπεδα με τη χρήση "/"
  - Ο πόρος βρίσκεται σε **μικρότερο βάθος**, οπότε θα χρειαστεί να αναρριχηθούμε σε κατάλληλο επίπεδο με τη χρήση "../" και στη συνέχεια να ακολουθήσουμε τη διαδρομή (χρησιμοποιώντας "/") για να εντοπίσουμε τον κατάλογο.

Παράδειγμα 2.31. Παράδειγμα μονοπατιών (paths) σε γραφικά/εικόνες

```
1. <!DOCTYPE html>
2. <html lang="el">
3. <head>
4.   <meta charset="utf-8">
5.   <title>Διαχείριση Πόρων & Μονοπάτια (paths)</title>
6. </head>
7. <body>
8.   <h1> Διαχείριση Πόρων & Μονοπάτια (paths) </h1>
9.
10.  <h2> Στον ίδιο φάκελο </h2>
11.  <p>Με σχετική αναφορά στον ίδιο κατάλογο με το υπο-επεξεργασία αρχείου </p>
12.  
```

```

13.
14. <h2> Σε εξωτερική διεύθυνση (URL) </h2>
15. <p> Με απόλυτη θέση σε εξωτερική σελίδα (url) </p>
16. 
17.
18. <h2> Σε υποκατάλογο </h2>
19. <p> Με σχετική αναφορά στον κατάλογο _images που βρίσκεται στο ίδιο επίπεδο μ
    ε το υπο-επεξεργασία html</p>
20. 
21.
22. <h2> Σε γονικό φάκελο</h2>
23. <p> Με σχετική αναφορά στον γονικό (../) κατάλογο του υπό επεξεργασία αρχείου
    </p>
24. 
25.
26. <h2> Σε υποκατάλογο του γονικού φακέλου</h2>
27. <p> Με σχετική αναφορά σε υποκατάλογο (generic_images) του γονικού (../) κατα
    λόγου του υπό-επεξεργασία αρχείου</p>
28. 
29.
30. </body>
31. </html>

```

Ο **εντοπισμός πόρων και σύνδεσή τους** με το τρέχον έγγραφο ισχύει για τη διασύνδεση/αναφορά συνδέσμων, γραφικών/εικόνων, αρχείων μορφοποίησης (CSS) και προγραμματιστικών βιβλιοθηκών (JavaScript). Όπως έχουμε περιγράψει σε προηγούμενη ενότητα, θα πρέπει να μπορούμε να κατευθύνουμε το σύνδεσμο στον πόρο χρησιμοποιώντας *σχετικές* ή *απόλυτες* διαδρομές. Η χρήση σχετικών διαδρομών προσφέρει το πλεονέκτημα της **μεταφερισιμότητας** καθώς τα αρχεία μπορούν να μετακινούνται χωρίς να επηρεάζεται η λειτουργία του site μιας και οι σχετικές θέσεις των αρχείων παραμένουν ίδιες.

### 2.7.13 Υπερσύνδεσμοι <a>

Οι **υπερ/σύνδεσμοι** (links) αποτελούν την πραγματική δύναμη της HTML και είναι το στοιχείο που κάνει πράξη την έννοια του “web” στον ορισμό του WWW. Μέσω συνδέσμων διασυνδέεται το έγγραφό μας με άλλα έγγραφα (ή πόρους) τα οποία μπορούν να βρίσκονται σε *τοπικά* ή σε *απομακρυσμένα* συστήματα. Επιπλέον, επιτρέπει σε *τμήματα των εγγράφων* μας να αποκτούν δική τους «*διεύθυνση*» ώστε να μπορεί κάποιος να κατευθύνεται (navigation functionality) σε αυτά ανεξάρτητα από τη θέση τους στο έγγραφο.

Οι φυλλομετρητές εμφανίζουν τους συνδέσμους μορφοποιημένους συνήθως με υπογραμμισμένη γραφή διαχωρίζοντας με χρώματα το αν είναι *ενεργοί*, τους *έχουμε ήδη επισκεφτεί* ή είναι *επιλεγμένοι*. Η ετικέτα που υλοποιεί έναν σύνδεσμο είναι η <a> (anchor) και στην πιο απλή μορφή είναι:

*Παράδειγμα 2.32. Παράδειγμα δημιουργίας συνδέσμου*

```

1. <p>Αυτός είναι ένας σύνδεσμος στο
2. <a href="http://www.uniwa.gr">Πανεπιστήμιο Δυτικής Αττικής</a>.
3. </p>

```

Θεωρείται καλή πρακτική η ενσωμάτωση συμπληρωματικής πληροφορίας μέσω του χαρακτηριστικού title το οποίο παρέχει λεπτομέρειες για τον σύνδεσμο όπως π.χ. το είδος της πληροφορίας που περιλαμβάνει η σελίδα, επιτρέποντας στο χρήστη να πάρει την πληροφορία τοποθετώντας τον κέρσορα του ποντικιού του πάνω στον σύνδεσμο (hover) π.χ.

Παράδειγμα 2.33. Παράδειγμα δημιουργίας συνδέσμου (με title)

1. <p>Δημιουργία συνδέσμου
2. <a href="http://www.uniwa.gr"
3. title="Το νέο πανεπιστήμιο της Δυτικής Αττικής">στο Πα.Δ.Α.</a>.
4. </p>

Τα κυριότερα χαρακτηριστικά της ετικέτας <a> φαίνονται στον παρακάτω πίνακα:

Πίνακας 2.8. Χαρακτηριστικά ετικέτας <a>

Χαρακτηριστικό	Χρήση	Παράδειγμα
download	Όταν ο σύνδεσμος αφορά τη μεταφόρτωση - ορίζει το όνομα του αρχείου	<a href="/documents/myCV.pdf" download="LazCV.pdf">
href	Ορίζεται η διεύθυνση της σελίδας (πόρου)	<a href="http://www.uniwa.gr">Πανεπιστήμιο Δυτικής Αττικής</a>
hreflang	Χρησιμοποιείται όταν θέλουμε να οριστεί η γλώσσα που χρησιμοποιεί το διασυνδεδεμένο έγγραφο	<a href="http://www.uniwa.gr" hreflang="el">Πανεπιστήμιο Δυτικής Αττικής</a>
media	Ορίζει το μέσο στο οποίο το διασυνδεδεμένο έγγραφο έχει τη βέλτιστη εμφάνιση	<a href="att_a_media.asp?output=print" media="print and (resolution:300dpi)">Open media attribute page for print.</a>
rel	Ορίζει τη σχέση του διασυνδεδεμένου εγγράφου με το τρέχον	<a rel="value">value = alternate, author, bookmark, external, help, license, next, follow, nofollow, prev, search, tag
target	Ορίζει την τοποθεσία που θα εμφανιστεί το έγγραφο του συνδέσμου (_blank, _parent, _self[default], _top)	<a href="http://www.uniwa.gr" target="_blank">Πανεπιστήμιο Δυτικής Αττικής</a>
type	Ορίζει τον τύπο του διασυνδεδεμένου μέσου	<a href="http://www.uniwa.gr" type="text/html">Πανεπιστήμιο Δυτικής Αττικής</a>

Μια επιπλέον δυνατότητα που μας παρέχει η ετικέτα <a> είναι η δημιουργία «σελιδοδεικτών» (fragments) σε τοποθεσίες του τρέχοντος εγγράφου. Αυτοί δημιουργούνται με το σύμβολο “#” ακολουθούμενο από το όνομα αναφοράς (anchor/fragment id).

Το **όνομα αναφοράς** ορίζεται σε μια HTML ετικέτα με τη χρήση του χαρακτηριστικού (id) και πρέπει να είναι μοναδικό στο εύρος της σελίδας.

#### Παράδειγμα 2.34. Παράδειγμα χρήσης fragments

```
1. <!DOCTYPE html>
2. <html lang="el">
3. <head>
4. <meta charset="utf-8">
5. <title>Σύνδεσμοι</title>
6. </head>
7. <body>
8. <h1>Σύνδεσμοι </h1>
9. <h2> Περιεχόμενα </h2>
10. <ul>
11.     <li> Περιεχόμενο <a href="#section1">Ενότητα 1</a></li>
12.     <li> Περιεχόμενο <a href="#section2">Ενότητα 2</a></li>
13. </ul>
14.
15. <h2 id="section1"> Ενότητα 1 </h2>
16. <p>Lorem ipsum ...</p>
17.
18. <h2 id="section2"> Ενότητα 2</h2>
19. <p>Lorem ipsum ... </p>
20. </body>
21. </html>
```


#### 2.7.14 Εικόνες – Γραφικά <img>

Η χρήση εικόνων και γραφικών αποτελεί σημαντικό μέρος της συγγραφής HTML περιεχομένου. Η εισαγωγή γραφικών πραγματοποιείται με τη χρήση της ετικέτας <img>. Στην πιο απλή της μορφή:

#### Παράδειγμα 2.35. Παράδειγμα χρήσης ετικέτας γραφικών <img>

```

```



Οι κυριότερες παράμετροι για την ενσωμάτωση γραφικών είναι οι εξής:

- **src:** Η πηγή του προς εμφάνιση αρχείου – αποτελείται από 2 μέρη, τη διαδρομή και το όνομα του αρχείου (εμφανίζονται παραπάνω με **κόκκινο** και **μπλε** βέλος αντίστοιχα).
- **alt:** Ένα περιγραφικό όνομα για την εικόνα το οποίο θα χρησιμοποιηθεί στην περίπτωση αδυναμίας εμφάνισης της εικόνας (ή ακόμα και από screen readers).
- **width & height:** Αποτελούν προαιρετικές παραμέτρους και ρυθμίζουν το μέγεθος εμφάνισης της εικόνας. Μπορούμε να ελέγχουμε το μέγεθος του αποτελέσματος (ανεξάρτητα από την πραγματική του διάσταση) αλλά θα πρέπει να φροντίσουμε να διατηρήσουμε την αναλογία για να αποφύγουμε την παραμόρφωση!

Η σημαντικότερη από τις παραπάνω παραμέτρους είναι η **src**, η οποία κατά ένα μέρος υποδεικνύει το **μονοπάτι (διαδρομή)** στο οποίο βρίσκεται η εικόνα. Ορίζει επίσης το αρχείο, το οποίο μπορεί βάσει επέκτασης (extension) να ανήκει σε έναν από τους πιο διαδεδομένους τύπους γραφικών όπως **jpg, png, gif, bmp, svg**.

Σε ότι αφορά τη χρήση των **width** και **height**, η HTML5 επιτρέπει τη δήλωση τους **μόνο** σε pixels, ενώ σε παλαιότερες εκδόσεις της HTML η τιμή μπορούσε να εμφανίζεται ως ποσοστό του αρχικού μεγέθους (π.χ. 50%).

Παράδειγμα 2.36. Παράδειγμα χρήσης ετικέτας γραφικών <img> με width

```

```

Εναλλακτικά, προτείνουμε τη χρήση του χαρακτηριστικού style για να οριστεί το πλάτος/ύψος της εικόνας:

Παράδειγμα 2.37. Παράδειγμα χρήσης ετικέτας γραφικών <img> με style

```

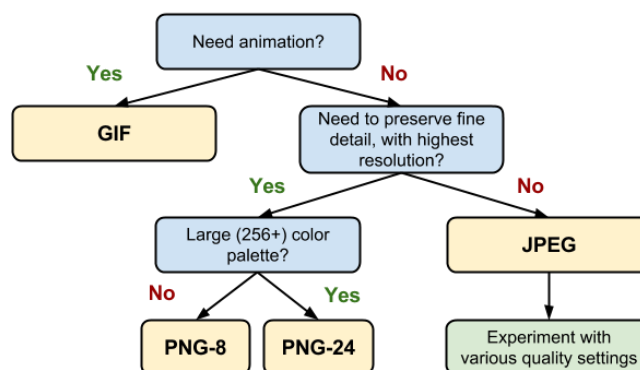
```

## Άσκηση 12

Δημιουργήστε ένα έγγραφο HTML μέσα στο οποίο θα εισάγετε 2 εικόνες της επιλογής σας. Η πρώτη από τις δύο εικόνες θα πρέπει να έχει μικρή ανάλυση (π.χ. 240px) ενώ η δεύτερη μεγάλη (π.χ. >1920px – προτιμήστε μια εικόνα με μεγάλο μέγεθος π.χ. >5MB). Με βάση τα παραπάνω παραδείγματα, πειραματιστείτε δημιουργώντας πολλαπλά αντίγραφα επηρεάζοντας τις διαστάσεις τους (width, height) και παρατηρήστε τις επιπτώσεις στην ποιότητα εμφάνισης της εικόνας. Επιμεληθείτε κατάλληλα τα περιεχόμενα του στοιχείου alt για κάθε μια από τις εικόνες.

**Προσοχή!** Θα πρέπει πάντα να λαμβάνουμε υπόψη πως η υπερβολική χρήση εικόνων προκαλεί καθυστερήσεις στην εμφάνισή τους και στην ολοκλήρωση φόρτωσης της σελίδας. Για αυτό το λόγο θα πρέπει να φροντίζουμε ώστε, **τόσο το πλήθος των εικόνων όσο και το μέγεθός τους**, να μην επιβαρύνουν το περιεχόμενο της σελίδας.

Προς αυτή την κατεύθυνση υπάρχει η διαδικασία *βελτιστοποίησης των γραφικών για χρήση τους στο Web*, η οποία αποτελείται από βήματα μέσω των οποίων τα γραφικά μειώνουν το μέγεθός τους χωρίς να χάνουν το επίπεδο λεπτομέρειας που απαιτείται. Τέλος, η επιλογή του τύπου του γραφικού (μεταξύ των gif, jpg, png) που μπορεί να χρησιμοποιηθεί σχετίζεται με τη χρήση που προορίζεται και προκύπτει από μια σειρά ερωτήσεων όπως φαίνεται στο παρακάτω σχήμα.



Εικόνα 2.6. Πηγή : <https://developers.google.com/web/fundamentals/performance/optimizing-content-efficiency/image-optimization>

#### 2.7.14.1 Εικόνες ως σύνδεσμοι

Μια άλλη πολύ διαδεδομένη χρήση των εικόνων, είναι η **χρήση τους ως συνδέσμους** όπου τη θέση του κειμένου καταλαμβάνει ένα γραφικό. Το πιο διαδεδομένο σενάριο είναι η δημιουργία ενός «λευκώματος» όπου μικρού μεγέθους εικόνες (thumbnails) προσφέρουν την επιλογή προεπισκόπησης, ενώ με ενεργοποίηση του συνδέσμου ο επισκέπτης λαμβάνει την πλήρη μορφή της εικόνας. Αυτό βελτιώνει δραματικά την απόδοση των σελίδων μιας και το μέγεθος των thumbnails είναι αρκετά μικρότερο σε σχέση με το κανονικό μέγεθος του γραφικού.

```
<a href="_images/1200px-HTML5_logo_and_wordmark.svg.png" alt="HTML5 Logo"></a>
```

Η δημιουργία γραφικών κατάλληλων για το διαδίκτυο μπορεί να πραγματοποιηθεί είτε με εξειδικευμένα – και ακριβά – προγράμματα επεξεργασίας εικόνας (π.χ. Adobe Photoshop) όσο και δωρεάν λύσεις όπως το XNView (διαθέσιμο στη διεύθυνση <https://www.xnview.com/en/>) το οποίο μας επιτρέπει να επεξεργαστούμε τόσο το περιεχόμενο όσο και τις διαστάσεις των γραφικών.

#### 2.7.14.2 Εικόνες για διαφορετικές συσκευές

Με την εξάπλωση των φορητών συσκευών (tablets, smartphones κτλ.) δημιουργήθηκε η ανάγκη εμφάνισης γραφικών/εικόνων οι οποίες θα προσαρμόζονται στο μέγεθος της οθόνης που χρησιμοποιείται. Σε αυτή την περίπτωση εισάγεται με την HTML5 η ετικέτα <picture> η οποία συντάσσεται όπως παρακάτω:

Παράδειγμα 2.38. Παράδειγμα χρήσης ετικέτας γραφικών <picture>

```
1. <picture>
2.   <source media="(min-width: 650px)" srcset="logo.png">
3.   <source media="(min-width: 465px)"
   srcset="http://www.teiath.gr/images/top_logo_teia_el.jpg">
4.   
5. </picture>
```

Αλλάζοντας το μέγεθος παραθύρου του φυλλομετρητή, πρακτικά αλλάζει το viewport σύμφωνα με το οποίο επιλέγεται η ανάλογου μεγέθους εικόνα. Η ετικέτα <img> απαιτείται για να εξασφαλιστεί συμβατότητα με φυλλομετρητές που δεν υποστηρίζουν την ετικέτα <picture> (π.χ. IE <= 12 και Safari <= 9.0) ή ως εναλλακτική λύση αν καμία από τις διαστάσεις του viewport δεν ταιριάζει.

#### 2.7.14.3 Σημασιολογικά διαχειρίσιμες εικόνες

Η ευρεία χρήση των εικόνων έφερε την ανάγκη της σημασιολογικής διαχείρισής τους σε σχέση με το υπόλοιπο περιεχόμενο. Έτσι οι εικόνες ως «πολίτες 1<sup>ης</sup> κατηγορίας του www» απαιτούν πλέον τον σημασιολογικό διαχωρισμό τους. Με την ετικέτα <figure> οι εικόνες αντιμετωπίζονται ως *περιεχόμενο αυτοδύναμο*, το οποίο συνοδευόμενο από λεζάντα <figcaption> μπορεί να χαρακτηριστεί ως «εννοιολογικά πλήρης οντότητα».

Παράδειγμα 2.39. Παράδειγμα χρήσης ετικέτας γραφικών <figure>

```
1. <figure>
```

```

2.   
5.   <figcaption>Technological Educational Institute of Athens (final
      logo)</figcaption>
6. </figure>

```

Πιο αναλυτικά, η ετικέτα <figure> ορίζει το πλαίσιο μέσα στο οποίο θα εμφανιστεί ένα γραφικό, το οποίο μπορεί να είναι εικόνα, γράφημα, διάγραμμα, κομμάτι κώδικα ή ακόμα και ένα ποίημα.

*Παράδειγμα 2.40. Παράδειγμα χρήσης ετικέτας <figure> για citation*

```

1. <figure>
2.   <figcaption><cite>Edsger Dijkstra :-</cite></figcaption>
3.   <p>"If debugging is the process of removing software bugs,
4.   <br />
5.   then programming must be the process of putting them in"</p>
6. </figure>

```

Ταυτόχρονα η ετικέτα <figcaption> δημιουργεί μια λεζάντα για το περιεχόμενο, τοποθετείται πριν ή μετά την εικόνα και μπορεί να αποτελείται από κείμενο με παραγράφους, συνδέσμους κτλ. Έτσι, το περιεχόμενο αυτό θεωρείται πλήρες και μπορεί εύκολα να μετακινηθεί ως μια οντότητα στο περιεχόμενο της σελίδας, χωρίς να επηρεάζει τη ροή του περιεχομένου της.

### ΆΣΚΗΣΗ 13

Δημιουργήστε ένα έγγραφο HTML στο οποίο θα εμφανίζονται 3 εικόνες της επιλογής σας η μία κάτω από την άλλη εισάγοντας και κατάλληλη λεζάντα. Οι εικόνες θα πρέπει να έχουν ικανοποιητικό μέγεθος, ώστε όταν εμφανίζονται σε πλήρη οθόνη να μην υπάρχει ορατή απώλεια στην ποιότητα τους. Για κάθε μια από τις εικόνες δημιουργήστε μικρογραφίες (thumbnails) με πλάτος (width) το πολύ 240px. Κάνοντας χρήση των υπερσυνδέσμων (a) οδηγήστε το πάτημα της μικρογραφίας στην εμφάνιση του πλήρους μεγέθους της εικόνας σε νέο παράθυρο. Τόσο οι εικόνες (σε πλήρες μέγεθος) όσο και οι μικρογραφίες τους θα πρέπει να δημιουργηθούν σε ξεχωριστό φάκελο με όνομα images. Τα ονόματα των αρχείων θα πρέπει να είναι όμοια με τη διαφορά ότι σε αυτά των μικρογραφιών στο τέλος υπάρχει το "-small" π.χ. eikona1.png και eikona1-small.png. Για τη δημιουργία των μικρογραφιών μπορείτε να χρησιμοποιήσετε λογισμικά επεξεργασίας εικόνας (π.χ. XNView, Ζωγραφική).

#### 2.7.14.4 Εικόνες ως γραφικό φόντου σελίδας

Οποιοδήποτε γραφικό μπορεί να χρησιμοποιηθεί ως φόντο (background) σε μια σελίδα θέτοντας απλά ως style στην ετικέτα του <body> την παράμετρο «background-image» όπως παρακάτω.

*Παράδειγμα 2.41. Παράδειγμα χρήσης εικόνας ως φόντο σελίδας*

```
<body style="background-image:url('filename.ext');">
```

#### 2.7.15 Πίνακες <table>

Η HTML προσφέρει ένα μεγάλο αριθμό στοιχείων με τα οποία μπορούμε να διαχειριστούμε και να μορφοποιήσουμε δεδομένα σε *πινακοειδή μορφή (tabular data)*. Με ταυτόχρονη χρήση CSS κανόνων το τελικό αποτέλεσμα μπορεί να είναι εξαιρετικό.



## Planets data

Data about the planets of our solar system (Planetary facts taken from <a href="#">NASA's Planetary Fact Sheet - Metric</a> )											
		Name	Mass (10 <sup>24</sup> kg)	Diameter (km)	Density (kg/m <sup>3</sup> )	Gravity (m/s <sup>2</sup> )	Length of day (hours)	Distance from Sun (10 <sup>6</sup> km)	Mean temperature (°C)	Number of moons	Notes
Terrestrial planets		Mercury	0.330	4,879	5427	3.7	4222.6	57.9	167	0	Closest to the Sun
		Venus	4.87	12,104	5243	8.9	2802.0	108.2	464	0	
		Earth	5.97	12,756	5514	9.8	24.0	149.6	15	1	Our world
		Mars	0.642	6,792	3933	3.7	24.7	227.9	-65	2	The red planet
Jovian planets	Gas giants	Jupiter	1898	142,984	1326	23.1	9.9	778.6	-110	67	The largest planet
		Saturn	568	120,536	687	9.0	10.7	1433.5	-140	62	
	Ice giants	Uranus	86.8	51,118	1271	8.7	17.2	2872.5	-195	27	
		Neptune	102	49,528	1638	11.0	16.1	4495.1	-200	14	
Dwarf planets		Pluto	0.0146	2,370	2095	0.7	153.3	5906.4	-225	5	Declassified as a planet in 2006, but this <a href="#">isn't</a>

Εικόνα 2.7. Παράδειγμα HTML πίνακα.

Στο παρελθόν οι πίνακες χρησιμοποιήθηκαν ευρέως, όχι μόνο για να παρουσιάσουν δεδομένα, αλλά για να δημιουργήσουν τη δομή μιας σελίδας ([https://developer.mozilla.org/en-US/docs/Learn/Accessibility/HTML#Page\\_layouts](https://developer.mozilla.org/en-US/docs/Learn/Accessibility/HTML#Page_layouts)). Αυτό συνέβη γιατί η υποστήριξη των φυλλομετρητών στην CSS δεν ήταν καθολική και οι πίνακες πρόσφεραν ενιαία αντιμετώπιση. Αυτή η τακτική έχει σχεδόν εξαλειφθεί μέσω της χρήσης CSS. Οι λόγοι που θα πρέπει να αποφεύγεται η χρήση πινάκων για τη δημιουργία δομής είναι οι εξής:

- **Μειώνεται δραματικά η προσβασιμότητα** χρηστών με προβλήματα όρασης – οι συσκευές (screen readers) αναγνωρίζουν μεν τα στοιχεία των πινάκων χωρίς να ξεχωρίζουν ότι το περιεχόμενό τους αφορά δομή – δημιουργώντας σύγχυση κατά την ανάγνωση του.
- Η χρήση πινάκων δημιουργεί πολύπλοκες δομές με τελικό αποτέλεσμα έναν κώδικα που είναι **δύσκολο να συντηρηθεί και να αποσφαλματωθεί**.
- Δεν οδηγούν σε περιεχόμενο που είναι **responsive**. Σε σχέση με άλλα στοιχεία sectioning (header, section, article, div) που καταλαμβάνουν το πλήρες πλάτος του γονικού στοιχείου, οι πίνακες προσαρμόζονται στο μέγεθος των περιεχομένων των κελιών τους και απαιτείται ειδική μέριμνα για να μπορεί ο πίνακας να εμφανιστεί σωστά σε συσκευές με διαφορετικά μεγέθη οθόνης.

Σκοπός είναι να δημιουργήσουμε πίνακες οι οποίοι θα επιτρέπουν την ανάγνωσή τους ακόμα και από χρήστες με προβλήματα όρασης π.χ. με χρήση βοηθητικών συσκευών.

### 2.7.15.1 Βασικά στοιχεία πινάκων <table>, <tr>, <td>

Αρχικά, θα εξετάσουμε ποια είναι τα συστατικά στοιχεία ενός πίνακα - γραμμές, στήλες, επικεφαλίδες, συγχωνευμένα κελιά που εκτείνονται σε περισσότερες στήλες/γραμμές κτλ. Το βασικό στοιχείο που σηματοδοτεί τη δημιουργία ενός πίνακα είναι το <table>...</table>, και η μικρότερη μονάδα που ορίζει ένα κελί είναι το <td>...</td> (table data).

Οι περισσότεροι φυλλομετρητές εξ' ορισμού εφαρμόζουν μορφοποίηση για την εμφάνιση στοιχείων πίνακα, όπου τα όρια των κελιών και το περίγραμμα του πίνακα δεν είναι ορατά. Έτσι, θα χρησιμοποιήσουμε το παρακάτω style (<style>...</style> στο <head>), ώστε να αποκαλυφθούν τα όρια (border) του πίνακα και των κελιών του.

Παράδειγμα 2.42. Βασικό στυλ περιθωρίων πίνακα (border)

1. <style>
2. table, th, td {
3.     border: 1px solid black;
4. }
5. </style>



Παράδειγμα 2.43. Παράδειγμα πίνακα με 1 γραμμή με χρήση <td>...</td>

```
1. <h2> Απλός πίνακας 1 γραμμής </h2>
2.   <table>
3.     <td> Είμαι το πρώτο κελί</td>
4.     <td> Είμαι το δεύτερο κελί</td>
5.     <td> Είμαι το τρίτο κελί</td>
6.     <td> Είμαι το τέταρτο κελί</td>
7.   </table>
```

Στην περίπτωση που χρειαζόμαστε περισσότερες γραμμές τότε θα πρέπει να εμφωλεύσουμε τα κελιά μιας γραμμής μεταξύ των στοιχείων <tr>...</tr> (table row).

Παράδειγμα 2.44. Παράδειγμα πίνακα με χρήση <tr>...</tr>

```
1. <h2> Πίνακας 2 γραμμών </h2>
2. <table>
3.   <tr>
4.     <td> Είμαι το πρώτο κελί</td>
5.     <td> Είμαι το δεύτερο κελί</td>
6.     <td> Είμαι το τρίτο κελί</td>
7.     <td> Είμαι το τέταρτο κελί</td>
8.   </tr>
9.   <tr>
10.    <td> 2η γραμμή - πρώτο κελί</td>
11.    <td> 2η γραμμή - δεύτερο κελί</td>
12.    <td> 2η γραμμή - τρίτο κελί</td>
13.    <td> 2η γραμμή - τέταρτο κελί</td>
14.  </tr>
15. </table>
```

#### 2.7.15.2 Επικεφαλίδες Πίνακα

Αν πίνακας περιέχει γραμμή που φιλοξενεί τις **επικεφαλίδες** των στοιχείων, απαιτείται η χρήση των στοιχείων <th>...</th> (table header). Το χαρακτηριστικό scope ορίζει αν ένα στοιχείο επικεφαλίδα <th> αναφέρεται σε στήλη, γραμμή ή ομάδα στηλών/γραμμών. Παρότι δεν έχει επίπτωση στην εμφάνιση των δεδομένων, αξιοποιείται από βοηθητικές συσκευές (screen readers). Στην περίπτωση που οι επικεφαλίδες βρίσκονται στην 1<sup>η</sup> στήλη τότε το 1<sup>ο</sup> στοιχείο κάθε γραμμής θα πρέπει να είναι <th>...</td> με scope="row".

Παράδειγμα 2.45. Παράδειγμα πίνακα με επικεφαλίδες

```
1. <h2> Πίνακας με επικεφαλίδες</h2>
2. <table>
3.   <caption> Αποταμίευση 2018</caption>
4.   <tr>
5.     <th></th>
6.     <th scope="col">Μήνας</th>
7.     <th scope="col">Αποταμίευση</th>
8.   </tr>
9.   <tr>
10.    <td>1</td>
11.    <td>Ιανουάριος</td>
12.    <td>€100</td>
13.  </tr>
```

```

14. <tr>
15.     <td>2</td>
16.     <td>Φεβρουάριος</td>
17.     <td>€80</td>
18. </tr>
19. </table>

```

Μια επιπλέον δυνατότητα για τους πίνακες αφορά την εισαγωγή λεζάντας <caption> η οποία τοποθετείται συνήθως μετά την ετικέτα ανοίγματος του πίνακα. Η λεζάντα είναι ένα στοιχείο ιδιαίτερα χρήσιμο για χρήστες με προβλήματα όρασης, μιας και αξιοποιείται από βοηθητικές συσκευές.

#### Άσκηση 14

Με βάση το προηγούμενο παράδειγμα, τροποποιήστε τη δομή του πίνακα μεταθέτοντας τις επικεφαλίδες στην 1<sup>η</sup> στήλη. Προσοχή στον ορισμό του score για συσχέτισμό των επικεφαλίδων με τα δεδομένα των γραμμών!

Αποταμίευση 2018

	Μήνας	Αποταμίευση
1	Ιανουάριος	€100
2	Φεβρουάριος	€80

#### 2.7.15.3 Στοιχεία Δομής <thead>, <tfoot>, <tbody>

Όταν οι πίνακες αρχίζουν να μεγαλώνουν σε μέγεθος και πολυπλοκότητα, καλό είναι να τους ενισχύσουμε τη σημασιολογική δομή χρησιμοποιώντας στοιχεία όπως <thead>, <tfoot> και <tbody>, τα οποία επιτρέπουν τον ορισμό περιοχών επικεφαλίδων (header), υποσέλιδου (footer) και σώματος δεδομένων (body).

- Το στοιχείο <thead> ενθυλακώνει το μέρος του πίνακα με τις επικεφαλίδες των στηλών (συνήθως την πρώτη γραμμή του πίνακα)
- Το στοιχείο <tfoot> ενθυλακώνει το τελικό μέρος του πίνακα, αυτό συνήθως συναντάται στην τελική γραμμή στοιχείων του πίνακα (π.χ. γραμμή συνόλων).
- Το στοιχείο <tbody> ενθυλακώνει το κύριο μέρος δεδομένων του πίνακα.

Αν και τα συγκεκριμένα στοιχεία δεν προσφέρουν βελτίωση στην προσβασιμότητα του πίνακα, καθιστούν ευκολότερες τις μορφοποιήσεις μέσω κανόνων styling.

#### 2.7.15.4 Συγχωνευμένα κελιά

Σε αρκετές περιπτώσεις απαιτείται η συγχώνευση κελιών τα οποία διατρέχουν πολλαπλές γραμμές ή στήλες. Στο επόμενο παράδειγμα το στοιχείο <th> της 2<sup>ης</sup> στήλης (γραμμή 5) είναι το αποτέλεσμα συγχώνευσης 2 κελιών γειτονικών στηλών (colspan="2").

Παράδειγμα 2.46. Παράδειγμα πίνακα με συγχωνευμένα κελιά (colspan)

```

1. <h2> Πίνακες με Συγχωνευμένα κελιά</h2>
2. <table>
3.     <tr>
4.         <th>Τμήμα/Σχολή</th>
5.         <th colspan="2">Τηλέφωνο</th>

```

```

6.         </tr>
7.         <tr>
8.             <td>Γραμματεία</td>
9.             <td>210-5385203</td>
10.            <td>6986123456</td>
11.        </tr>
12.        <tr>
13.            <td>Προϊστάμενος</td>
14.            <td>210-5385202</td>
15.            <td>6986654321</td>
16.        </tr>
17.    </table>

```

Η συγχώνευση μπορεί να πραγματοποιηθεί και σε επίπεδο γραμμών όπως στο επόμενο παράδειγμα. Παρατηρήστε πώς το συγχωνευμένο κελί διατρέχει 2 γραμμές (rowspan="2").

Παράδειγμα 2.47. Παράδειγμα πίνακα με συγχωνευμένα κελιά (rowspan)

```

1. <h2> Πίνακες με Συγχωνευμένα κελιά (rowspan)</h2>
2. <table>
3.     <tr>
4.         <th>Τμήμα/Σχολή</th>
5.         <td>Γραμματεία</td>
6.         <td>Προϊστάμενος</td>
7.     </tr>
8.     <tr>
9.         <th rowspan="2">Τηλέφωνο</th>
10.        <td>210-5385203</td>
11.        <td>6986123456</td>
12.    </tr>
13.    <tr>
14.        <td>210-5385202</td>
15.        <td>6986654321</td>
16.    </tr>
17.</table>

```

## Άσκηση 15

Τροποποιήστε τον παρακάτω κώδικα για να δημιουργήσετε το εβδομαδιαίο πρόγραμμα μαθημάτων της σχολής. Το πρόγραμμα αναφέρεται στο τυπικό εξάμηνο του μαθήματος (B) - προσαρμόστε το πρόγραμμα σύμφωνα με το εξάμηνό σας, περιλαμβάνοντας τα εργαστηριακά σας τμήματα.

Ωρολόγιο Πρόγραμμα 2017-2018 (Εαρινό)

	Δευτέρα	Τρίτη	Τετάρτη	Πέμπτη	Παρασκευή
8-9					
9-10					
10-11					
11-12					
12-13					
13-14					
14-15	Εισαγωγή στη Τεχνολογία των Πληροφοριών (Θ)	Υπηρεσίες Πληροφορικής (Θ)		Γαλλικά (Θ)	
15-16					
16-17	Ανάπτυξη και Διαχείριση Συλλογής (Θ)				
17-18					
18-19			Πηγές Πληροφορικής (Θ)		
19-20					

```

1. <h2> Ωρολόγιο Πρόγραμμα</h2>

```

```

2. <table>
3.   <caption> Ωρολόγιο Πρόγραμμα 2017-2018 (Εαρινό)</caption>
4.   <tr>
5.     <th></th>
6.     <th scope="col">Δευτέρα</th>
7.     <th scope="col">Τρίτη</th>
8.     <th scope="col">Τετάρτη</th>
9.     <th scope="col">Πέμπτη</th>
10.    <th scope="col">Παρασκευή</th>
11.  </tr>
12.  <tr>
13.    <th>8-9</th>
14.    <td></td>
15.    <td></td>
16.    <td></td>
17.    <td></td>
18.    <td></td>
19.  </tr>
20.  <tr>
21.    <th>9-10</th>
22.    <td></td>
23.    <td></td>
24.    <td></td>
25.    <td></td>
26.    <td></td>
27.  </tr>
28.  <tr>
29.    <th>10-11</th>
30.    <td></td>
31.    <td></td>
32.    <td></td>
33.    <td></td>
34.    <td></td>
35.  </tr>
36.  <tr>
37.    <th>11-12</th>
38.    <td></td>
39.    <td></td>
40.    <td></td>
41.    <td></td>
42.    <td></td>
43.  </tr>
44.  <tr>
45.    <th>12-13</th>
46.    <td></td>
47.    <td></td>
48.    <td></td>
49.    <td></td>
50.    <td></td>
51.  </tr>
52.  <tr>
53.    <th>13-14</th>
54.    <td></td>
55.    <td></td>
56.    <td></td>
57.    <td rowspan="3">Γαλλικά (θ)</td>
58.    <td></td>

```

```

59.     </tr>
60.     <tr>
61.         <th>14-15</th>
62.         <td rowspan="2">Εισαγωγή στη Τεχνολογία των Πληροφοριών (Θ)</td>
63.         <td></td>
64.         <td></td>
65.         <td></td>
66.     </tr>
67.     <tr>
68.         <th>15-16</th>
69.         <td rowspan="4">Υπηρεσίες Πληροφόρησης (Θ)</td>
70.         <td></td>
71.         <td></td>
72.     </tr>
73.     <tr>
74.         <th>16-17</th>
75.         <td rowspan="2">Ανάπτυξη και Διαχείριση Συλλογής (Θ)</td>
76.         <td></td>
77.         <td></td>
78.         <td></td>
79.     </tr>
80.     <tr>
81.         <th>17-18</th>
82.         <td rowspan="3">Πηγές Πληροφόρησης (Θ)</td>
83.         <td></td>
84.         <td></td>
85.     </tr>
86.     <tr>
87.         <th>18-19</th>
88.         <td></td>
89.         <td></td>
90.         <td></td>
91.     </tr>
92.     <tr>
93.         <th>19-20</th>
94.         <td></td>
95.         <td></td>
96.         <td></td>
97.         <td></td>
98.     </tr>
99. </table>

```

#### 2.7.15.5 Εμφωλευμένοι Πίνακες (nested tables)

Σε ιδιαίτερες περιπτώσεις που η πολυπλοκότητα του πίνακα είναι τέτοια που η συγχώνευση κελιών δε μπορεί να εξασφαλίσει το αποτέλεσμα που θέλουμε, υπάρχει η δυνατότητα να ενσωματώσουμε στα περιεχόμενα έναν άλλο πίνακα (table nesting).

title1			title2	title3
cell1	cell2	cell3	cell2	cell3
cell4			cell5	cell6

Εικόνα 2.8. Πίνακας μέσα σε Πίνακα

#### Παράδειγμα 2.48. Παράδειγμα εμφωλευμένου πίνακα

```
1. <h2> Εμφωλευμένοι πίνακες (nesting tables)</h2>
2.   <table id="table1">
3.     <tr>
4.       <th>title1</th>
5.       <th>title2</th>
6.       <th>title3</th>
7.     </tr>
8.     <tr>
9.       <td id="nested">
10.        <table id="table2">
11.          <tr>
12.            <td>cell1</td>
13.            <td>cell2</td>
14.            <td>cell3</td>
15.          </tr>
16.        </table>
17.      </td>
18.      <td>cell2</td>
19.      <td>cell3</td>
20.    </tr>
21.    <tr>
22.      <td>cell4</td>
23.      <td>cell5</td>
24.      <td>cell6</td>
25.    </tr>
26.  </table>
```

#### 2.7.15.6 Μορφοποίηση πίνακα

Στις προηγούμενες ενότητες αναλύσαμε τα δομικά στοιχεία για τον ορισμό πινάκων στην HTML. Η προκαθορισμένη μορφοποίηση των φυλλομετρητών είναι τέτοια που πρακτικά αναιρεί όλη την εκφραστικότητα της δομής με αποτέλεσμα να μη διαχωρίζονται τα δεδομένα μεταξύ τους και το τελικό οπτικό αποτέλεσμα να μην είναι το βέλτιστο. Στο επόμενο παράδειγμα, έχουμε την έξοδο (output) και τον HTML κώδικα για την παρακολούθηση εξόδων.

Αρχείο Εξόδων				
Παρακολούθηση εξόδων				
Αγορά	Κατάστημα	Ημερομηνία	Αξιολόγηση	Κόστος (€)
Κούρεμα	Δρ. Τρύφονας	15/04	Μέτριο	20
Pizza Special	Vicenzo	16/04	Εξαιρετικό	20
Παπούτσια Adidas	Intersport	20/04	Ακριβό	165
Συναυλία Scorpions	Καλλιμάρμαρο	07/05	Good	60
SUM				265

#### Παράδειγμα 2.49. Παράδειγμα μορφοποίησης πίνακα

```
1. <!DOCTYPE html>
2. <html>
3.   <head>
4.     <meta charset="utf-8">
5.     <title>Αρχείο Εξόδων</title>
6.     <style>
7.
```

```

8.     </style>
9. </head>
10. <body>
11.     <h1>Αρχείο Εξόδων</h1>
12.     <table>
13.         <caption>Παρακολούθηση εξόδων</caption>
14.         <tr>
15.             <th>Αγορά</th>
16.             <th>Κατάστημα</th>
17.             <th>Ημερομηνία</th>
18.             <th>Αξιολόγηση</th>
19.             <th>Κόστος (€)</th>
20.         </tr>
21.         <tr>
22.             <td>Κούρεμα</td>
23.             <td>Δρ. Τρύφωνας</td>
24.             <td>15/04</td>
25.             <td>Μέτριο</td>
26.             <td>20</td>
27.         </tr>
28.         <tr>
29.             <td>Pizza Special</td>
30.             <td>Vicenzo</td>
31.             <td>16/04</td>
32.             <td>Εξαιρετικό</td>
33.             <td>20</td>
34.         </tr>
35.         <tr>
36.             <td>Παπούτσια Adidas</td>
37.             <td>Intersport</td>
38.             <td>20/04</td>
39.             <td>Ακριβό</td>
40.             <td>165</td>
41.         </tr>
42.         <tr>
43.             <td>Συναυλία Scorpions</td>
44.             <td>Καλλιμάραρο</td>
45.             <td>07/05</td>
46.             <td>Απίστευτο!</td>
47.             <td>60</td>
48.         </tr>
49.         <tr>
50.             <td colspan="4">SUM</td>
51.             <td>265</td>
52.         </tr>
53.     </table>
54. </body>
55. </html>

```

Στη συνέχεια θα εξετάσουμε παρεμβάσεις με κανόνες μορφοποίησης (styling) που βελτιώσουν το παρουσιαστικό και την αναγνωσιμότητα του πίνακα:

1. Αρχικά έχουμε γενικές ρυθμίσεις που αφορούν τη μορφή του πίνακα σε σχέση με το χώρο που θα καταλάβει (layout, width), τα όρια των στοιχείων του (border-collapse) και τη μορφή που θα έχει το περίγραμμα του (border width/color).

```
table {
  table-layout: fixed;
  width: 100%;
  border-collapse: collapse;
  border: 3px solid purple;
}
```

Η επιλογή `table-layout: fixed` αποτελεί μια καλή πρακτική που βελτιώνει τη συμπεριφορά του πίνακα σαν δομή. Συνήθως επιθυμούμε τη ρύθμιση του πλάτους βάσει του πλάτους των επικεφαλίδων, ενώ εξ' ορισμού το πλάτος των στηλών καθορίζεται από το μέγεθος του περιεχομένου των κελιών τους.

2. Η γραμμή συνόλων (SUM) συνήθως εμφανίζεται με έντονη γραφή – για να γίνει αυτό πρέπει να αξιοποιήσουμε το στοιχείο `<tfoot>` με τον παρακάτω κανόνα μορφοποίησης.

```
tfoot {
  font-weight: bold;
}
```

3. Παρόμοιες μορφοποιήσεις μπορούν να εφαρμοστούν αν στοχεύσουμε στα στοιχεία επικεφαλίδων `<thead>`, όπου συνήθως επιθυμούμε το φόντο της γραμμής επικεφαλίδων να είναι πιο σκούρο ή διαφορετικό στυλ γραφής (εναλλακτικά μπορούμε να χρησιμοποιήσουμε το στοιχείο `<th>`).

```
thead {
  font-style: italic;
  background-color: rgb(235,235,235);
}
```

4. Η λεζάντα μπορεί περαιτέρω να μορφοποιηθεί ως προς την απόστασή της από το πλαίσιο του πίνακα, το στυλ γραμματοσειράς, τη θέση, το χρώμα, τη στοίχιση και την απόσταση των χαρακτήρων, με τον παρακάτω κανόνα:

```
caption {
  font-family: Verdana, sans-serif;
  padding: 10px;
  font-style: normal;
  caption-side: top;
  color: #000;
  text-align: center;
  letter-spacing: 2px;
}
```

5. Η διαχείριση των ορίων του πίνακα για τα κελιά δεδομένων και επικεφαλίδων μπορεί να περιλαμβάνει την μορφοποίηση των ορίων (πάχος και χρώμα γραμμής) καθώς και την απόσταση του περιεχομένου από τα όρια των κελιών (`padding`).

```
td, th {
  border: 1px solid rgb(190,190,190);
  padding: 10px 20px;
}
```

6. Η στοίχιση των στοιχείων των κελιών `<td>` του πίνακα (`text-align: center`) είναι μια μορφοποίηση που συνήθως βελτιώνει την τελική παρουσίαση των δεδομένων. Στην



περίπτωση που δεν απαιτείται συνολική στοίχιση, αλλά σε επιλεγμένες στήλες, μπορεί να δημιουργηθεί κανόνας με χρήση του χαρακτηριστικού `class`.

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

7. Τέλος, μια χρήσιμη μορφοποίηση αφορά την εφαρμογή διαφορετικού τόνου φόντου στις μονές (even) και ζυγές (odd) γραμμές `<tr>` του πίνακα:

```
tr:nth-child(even) td {  
    background-color: rgb(250,250,250);  
}  
tr:nth-child(odd) td {  
    background-color: rgb(245,245,245);  
}
```

#### 2.7.15.7 Μορφοποίηση Στηλών πίνακα

Στην περίπτωση που επιθυμούμε την εφαρμογή κανόνων styling σε μεμονωμένες στήλες ενός πίνακα, χρειάζεται να εφαρμόσουμε τους κανόνες σε κάθε `<td>` στοιχείο για κάθε γραμμή `<tr>` του πίνακα, όπως παρακάτω:

*Παράδειγμα 2.50. Παράδειγμα μορφοποίησης στηλών*

```
1. <table>  
2.   <tr>  
3.     <th>Data 1</th>  
4.     <th style="background-color: yellow">Data 2</th>  
5.   </tr>  
6.   <tr>  
7.     <td>Calcutta</td>  
8.     <td style="background-color: yellow">Orange</td>  
9.   </tr>  
10.  <tr>  
11.   <td>Robots</td>  
12.   <td style="background-color: yellow">Jazz</td>  
13. </tr>  
14. </table>
```

Για να αποφευχθεί η επαναληπτική μορφοποίηση μπορούμε να χρησιμοποιήσουμε στοιχεία όπως το `<col>` και `<colgroup>`, όπως παρακάτω:

*Παράδειγμα 2.51. Παράδειγμα μορφοποίησης στηλών με <colgroup>*

```
1. <table>  
2.   <colgroup>  
3.     <col>  
4.     <col style="background-color: yellow">  
5.   </colgroup>  
6.   <tr>  
7.     <th>Data 1</th>
```

```

8.         <th>Data 2</th>
9.     </tr>
10.    <tr>
11.        <td>Calcutta</td>
12.        <td>Orange</td>
13.    </tr>
14.    <tr>
15.        <td>Robots</td>
16.        <td>Jazz</td>
17.    </tr>
18. </table>

```

Παρατηρήστε ότι θέλουμε να μορφοποιήσουμε τη 2<sup>η</sup> στήλη, γι' αυτό πρέπει πρώτα να ορίσουμε ένα κενό στοιχείο `<col>` (αντιστοιχεί στην 1<sup>η</sup> στήλη του πίνακα που δεν απαιτεί μορφοποίηση).

## 2.8 Δομή Ιστοσελίδας

Αδιαμφισβήτητα το **περιεχόμενο** μιας ιστοσελίδας αποτελεί την κύρια αξία και το βασικό λόγο προσέλκυσης και διατήρησης της βάσης επισκεπτών. Εξίσου σημαντική όμως είναι και η **δομή** της, η οποία ορίζεται μέσω στοιχείων επικεφαλίδων (headings), ομαδοποίησης (grouping) και σημασιολογικής διάταξης (sectioning). Η απουσία κατάλληλης δομής έχει τα παρακάτω αποτελέσματα:

- Οι χρήστες συνήθως περιμένουν να εντοπίσουν με σχετική ευκολία το ζητούμενο περιεχόμενο διαβάζοντας αρχικά τις διαθέσιμες επικεφαλίδες. Αν δεν καταφέρουν να τις εντοπίσουν εγκαίρως, συνήθως εγκαταλείπουν.
- Οι μηχανές αναζήτησης δίνουν μεγαλύτερη βαρύτητα στο περιεχόμενο των επικεφαλίδων καθώς αυτές επηρεάζουν την τελική κατάταξη των σελίδων. Χωρίς τη χρήση επικεφαλίδων η κατάληξη ενός SEO (**S**earch **E**ngine **O**ptimization = βελτίωση της κατάταξης μιας σελίδας στις μηχανές αναζήτησης) εγχειρήματος δε θα είχε τα αναμενόμενα οφέλη.
- Οι χρήστες με ειδικές δεξιότητες (π.χ. δυσκολίες οπτικής αναγνώρισης) συχνά χρησιμοποιούν βοηθητικές συσκευές (screen readers) για να αποκτήσουν γρήγορη και εύκολη πρόσβαση στο περιεχόμενο. Η διαδικασία που συνήθως ακολουθείται είναι η ανάγνωση του περιγράμματος (outline) της σελίδας, όπως αυτό προκύπτει από τις επικεφαλίδες, ώστε να εντοπιστεί το πραγματικό περιεχόμενο που ενδιαφέρει τους χρήστες.
- Η εφαρμογή κανόνων styling (μέσω CSS) ή ακόμα και η ενσωμάτωση κώδικα αλληλεπίδρασης (μέσω JavaScript) απαιτεί την ύπαρξη στοιχείων ομαδοποίησης (sectioning) ώστε να μπορούν τα στοιχεία του περιεχομένου να ελέγχονται με αποτελεσματικό τρόπο.

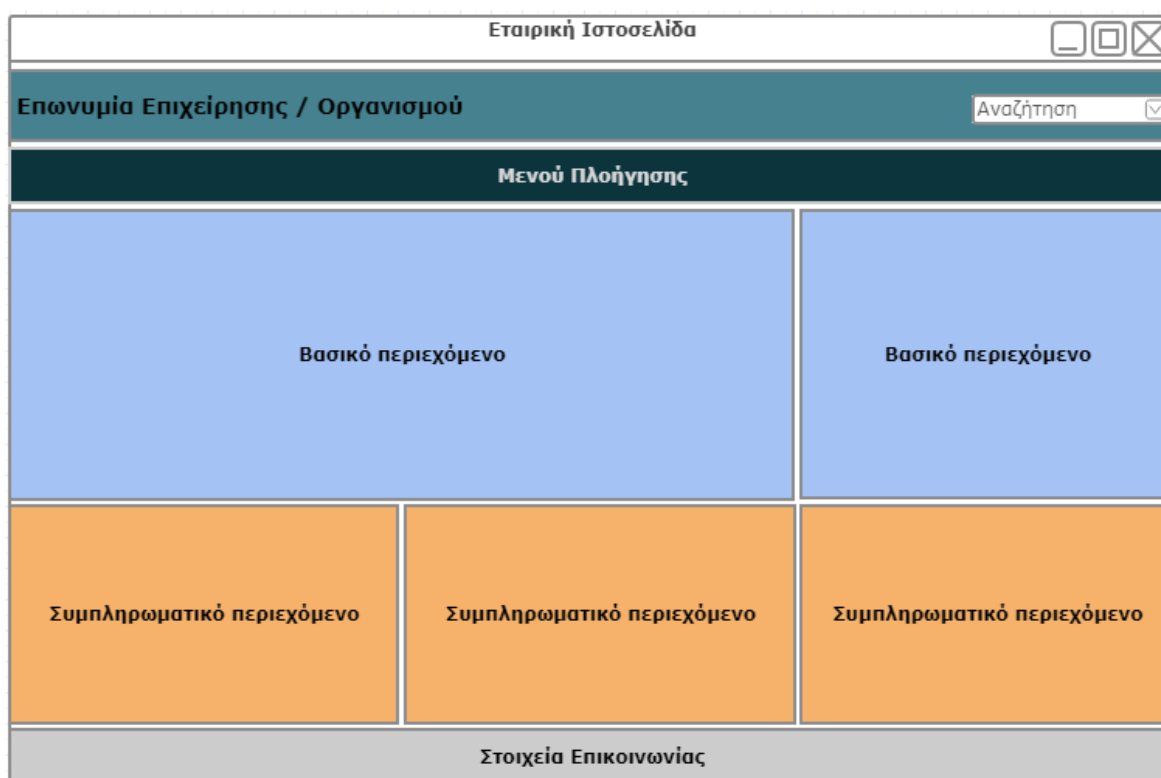
Για όλους τους παραπάνω λόγους απαιτείται η δημιουργία κατάλληλης δομής στο περιεχόμενο των σελίδων. Με την HTML5 προσφέρονται ακόμα περισσότερα *σημασιολογικά σχεδιαστικά εργαλεία* για τον σκοπό αυτό.

### 2.8.1 Η δομή μιας τυπικής ιστοσελίδας

Μια ιστοσελίδα συνήθως αποτελείται από διαφορετικές περιοχές, σε κάθε μια από τις οποίες ομαδοποιείται και εμφανίζεται διαφορετικού τύπου περιεχόμενο. Σύνηθες είναι στο πάνω μέρος της σελίδας (header) να εμφανίζεται ένα λογότυπο με πληροφοριακό κείμενο για την ιστοσελίδα.

Στη συνέχεια, μια περιοχή πλοήγησης (navigation) περιλαμβάνει συνδέσμους στις επιμέρους περιοχές του ιστότοπου, βοηθήματα όπως αυτό της αλλαγής γλώσσας και τη δυνατότητα αναζήτησης περιεχομένου στον ιστότοπο.

Στο κέντρο της σελίδας συναντάμε το βασικό περιεχόμενο-άρθρο (article) της σελίδας, ενώ σε κοντινή περιοχή μπορεί να εμφανίζεται συμπληρωματικό υλικό σε σχέση με το βασικό περιεχόμενο. Τέλος, στο κάτω μέρος (footer) εμφανίζονται στοιχεία επικοινωνίας και άλλο δευτερεύον πληροφοριακό υλικό (π.χ. όροι χρήσης).

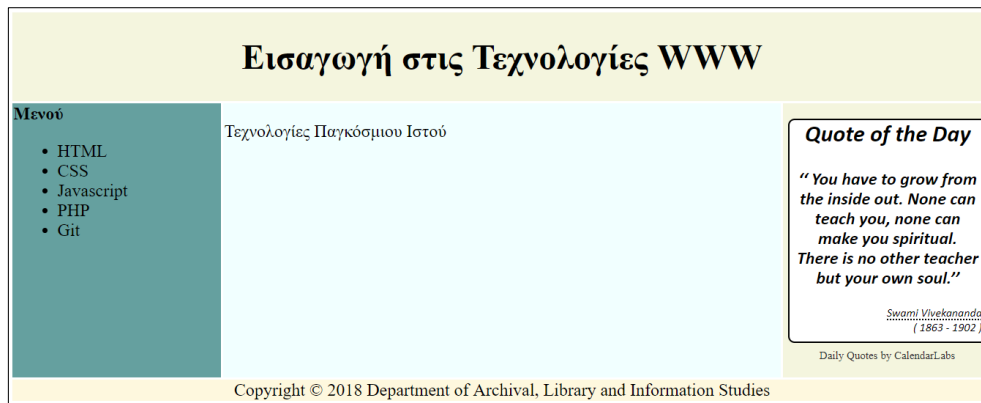


Εικόνα 2.9. Τυπική δομή ιστοσελίδας

Η διαδικασία διαμόρφωσης της δομής μιας ιστοσελίδας αποτελεί μια από τις σημαντικότερες προϋποθέσεις για τη δημιουργία ενός λειτουργικού και ευπαρουσίαστου ιστότοπου. Όσο εξελίσσονταν η HTML και η υποστήριξη των φυλλομετρητών χρησιμοποιήθηκαν διαφορετικοί τρόποι για να επιτευχθεί ο σχεδιασμός της δομής των ιστοσελίδων.

### 2.8.2 Δομή με Πίνακες

Αποτέλεσε έναν από τους δημοφιλέστερους τρόπους οργάνωσης της δομής μιας ιστοσελίδας με χρήση της ετικέτας `<table>...</table>`, ο οποίος εξασφάλιζε ενιαία αντιμετώπιση από τους φυλλομετρητές, αλλά **δημιουργεί σύγχυση στις συσκευές ανάγνωσης** για το τι αποτελεί πραγματικό περιεχόμενο και τι δομή της σελίδας.



Εικόνα 2.10. Δομή με χρήση πίνακα

Παράδειγμα 2.52. Παράδειγμα δομής με χρήση πίνακα

```

1. <!DOCTYPE html>
2. <html lang="el">
3. <head>
4.     <title>Δομή Σελίδας με Πίνακα</title>
5.     <meta charset="UTF-8">
6. </head>
7. <body>
8.     <table style="width:100%; border: 0;">
9.         <!-- Τίτλος -->
10.        <tr>
11.            <td style="background-color: beige; text-align: center" colspan="2">
12.                <h1> Εισαγωγή στις Τεχνολογίες WWW</h1>
13.            </td>
14.        </tr>
15.        <!-- Μενου & Περιεχόμενο -->
16.        <tr>
17.            <!-- Μενου -->
18.            <td style="background-color: cadetblue; vertical-align: top"
width="25%">
19.                <b> Μενού </b>
20.                <ul>
21.                    <li> HTML</li>
22.                    <li> CSS</li>
23.                    <li> Javascript </li>
24.                    <li> PHP </li>
25.                    <li> Git </li>
26.                </ul>
27.            </td>
28.            <!-- Κύριο περιεχόμενο -->
29.            <td style="background-color: azure;" width="75%">
30.                <p>Τεχνολογίες Παγκόσμιου Ιστού </p>
31.                <br>
32.                <br>
33.                ...
34.                <br>
35.                <br>
36.            </td>
37.        </tr>
38.        <!-- Υποσέλιδο -->
39.        <tr>
40.            <td style="background-color: cornsilk; text-align: center" colspan="2">

```

```

41. Copyright © 2018 Department of Archival, Library and Information Studies
42. </td>
43. </tr>
44. </table>
45. </body>
46. </html>

```

## Άσκηση 16

```
<!-- Πρόγνωση καιρού -->
```

Η εμφάνιση της πρόγνωσης καιρού θα πραγματοποιηθεί μέσω κώδικα που θα προκύψει από την παραμετροποίηση του widget στη διεύθυνση <https://www.okairos.gr/widget/>

```
<!-- Φράση της ημέρας -->
```

Η εμφάνιση της φράσης της ημέρας θα πραγματοποιηθεί μέσω κώδικα που προκύπτει από την παραμετροποίηση του widget στη διεύθυνση <https://www.calendarlabs.com/calendars/web-content/free-quotes-widget.php>

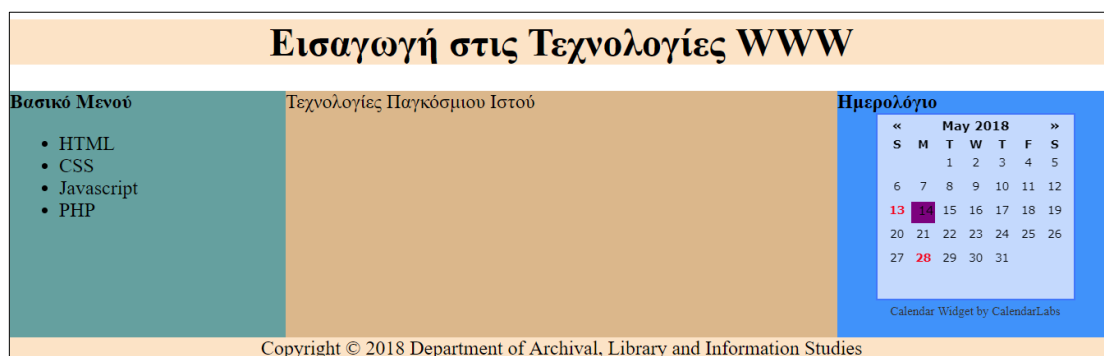
### 2.8.3 Δομή με <div>...</div>

Το πιο δημοφιλές στοιχείο ομαδοποίησης πληροφορίας για τον ορισμό δομής στις ιστοσελίδες είναι το <div>...</div>. Αποτελεί ένα στοιχείο (block) η χρήση του οποίου δεν προσφέρει **καμία σημασιολογική αξία**, αλλά χρησιμοποιείται κυρίως για να ομαδοποιήσει περιεχόμενο. Σε συνδυασμό με τα χαρακτηριστικά “id” και “class” μπορεί να αποδώσει στα περιεχόμενα που οργανώνει τη σημασία που πρέπει, κυρίως για λόγους μορφοποίησης μέσω CSS, αλλά χωρίς αυτό να γίνεται κατανοητό από τις μηχανές (agents) που την επεξεργάζονται.

W3C Specification HTML5.1 – παραδείγματα και σενάρια χρήσης:

<https://www.w3.org/TR/html51/grouping-content.html#the-div-element>

Το παρακάτω παράδειγμα εμφανίζει μια δομή με 3 στήλες όπως στην περίπτωση της προηγούμενης ενότητας αντικαθιστώντας τα στοιχεία <tr> με <div>.



Εικόνα 2.11. Δομή με χρήση DIV

Παράδειγμα 2.53. Παράδειγμα δομής με χρήση πίνακα

```
1. <!DOCTYPE html>
2. <html lang="el">
3. <head>
4.     <title>Δομή Σελίδας με χρήση DIV</title>
5.     <meta charset="UTF-8">
6. </head>
7. <body>
8.
9. <!-- Τίτλος -->
10. <div id="header" style="width: 100%">
11.     <div style="background-color: bisque; width: 100%; text-align: center">
12.         <h1> Εισαγωγή στις Τεχνολογίες WWW</h1>
13.     </div>
14. </div>
15.
16. <!-- Βασικό Μενου -->
17. <div id="nav" style="background-
18.     color: cadetblue; height:200px; width: 25%; float: left">
19.     <div>
20.         <b> Βασικό Μενού </b>
21.         <ul>
22.             <li> HTML</li>
23.             <li> CSS</li>
24.             <li> Javascript </li>
25.             <li> PHP </li>
26.         </ul>
27.     </div>
28. </div>
29. <!-- Κύριο περιεχόμενο -->
30. <div id="main" style="background-
31.     color: burlywood; height:200px; width: 50%; float: left">
32.     <div>
33.         Τεχνολογίες Παγκόσμιου Ιστού
34.     </div>
35. </div>
36. <!-- Δευτερεύον Μενου -->
37. <div id="aside" style="background-
38.     color: dodgerblue; height:200px; width: 25%; float: right">
39.     <div>
40.         <b>Ημερολόγιο</b>
41.         <!--Do not change below this line----->
42.         <div align="center"><iframe align="center" src="http://widget.calendarlab
43.             s.com/v1/calendar.php?cid=1001&uid=7346730493&c=8&l=en&cbg=C3D9FF&cfg=000000&hfg=
44.             000000&hfg1=000000&ct=1&cb=1&cbc=2275FF&cf=verdana&cp=&sw=0&hp=t&ib=0&ibc=&i=" wi
45.             dth="170" height="155" marginwidth=0 marginheight=0 frameborder=0 scrolling=no al
46.             lowtransparency='true'>Loading...</iframe><div align="center" style="width:140px;
47.             font-
48.             size:10px;color:#333;">Calendar Widget by <a href="https://www.calendarlabs.com/"
49.             target="_blank" rel="nofollow" style="font-size:10px;text-
50.             decoration:none;color:#333;">CalendarLabs</a></div></div>
51.         <!--Do not change above this line----->
52.     </div>
53. </div>
```

```

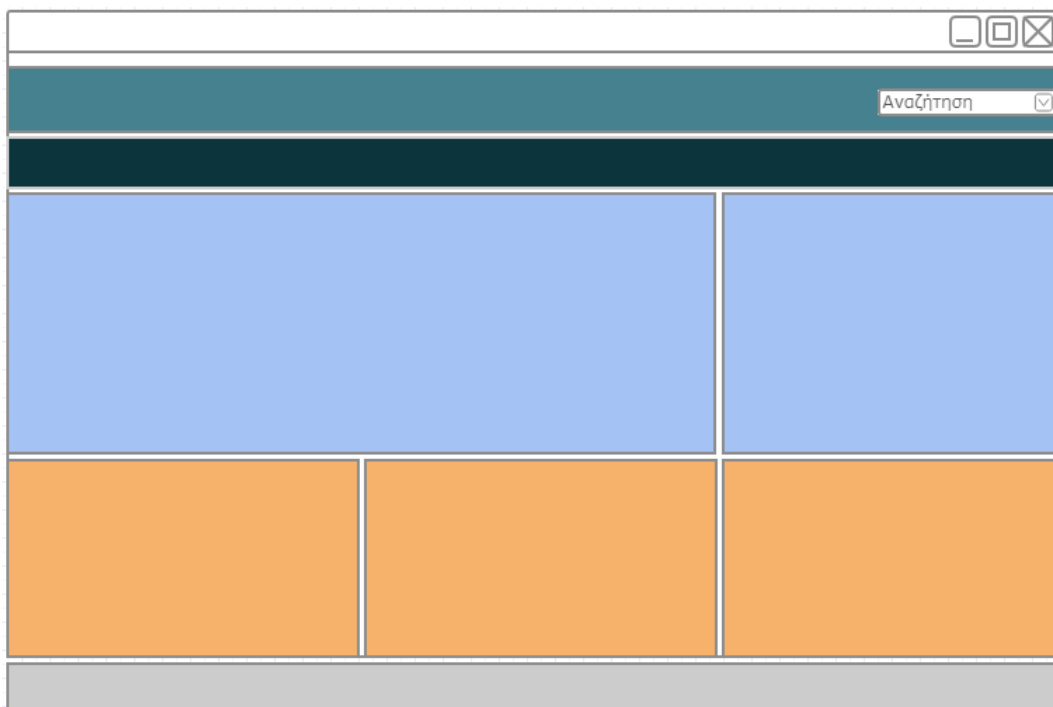
45. <!-- Υποσέλιδο -->
46. <div id="footer" style="background-color: bisque;clear: both; text-
    align: center">
47.     <p>Copyright © 2018 Department of Archival, Library and Information Stu
        dies </p>
48. </div>
49. </body>
50. </html>

```

## Άσκηση 17

### 2.8.4 Δομή με Σημασιολογικά Στοιχεία της HTML5

Ένα από τα πλεονεκτήματα της HTML5 είναι η δυνατότητα να ορίζουμε τη δομή των εγγράφων κατά τέτοιο τρόπο ώστε **αυτόματα να προβάλλεται η «σημασία του περιεχομένου»** (semantics). Για παράδειγμα, οι μηχανές αναζήτησης και άλλες βοηθητικές συσκευές αντιμετωπίζουν με άγνοια τη δομή της ιστοσελίδας (Εικόνα 2.9) με αποτέλεσμα να «αντιλαμβάνονται» αυτό που εμφανίζεται παρακάτω:

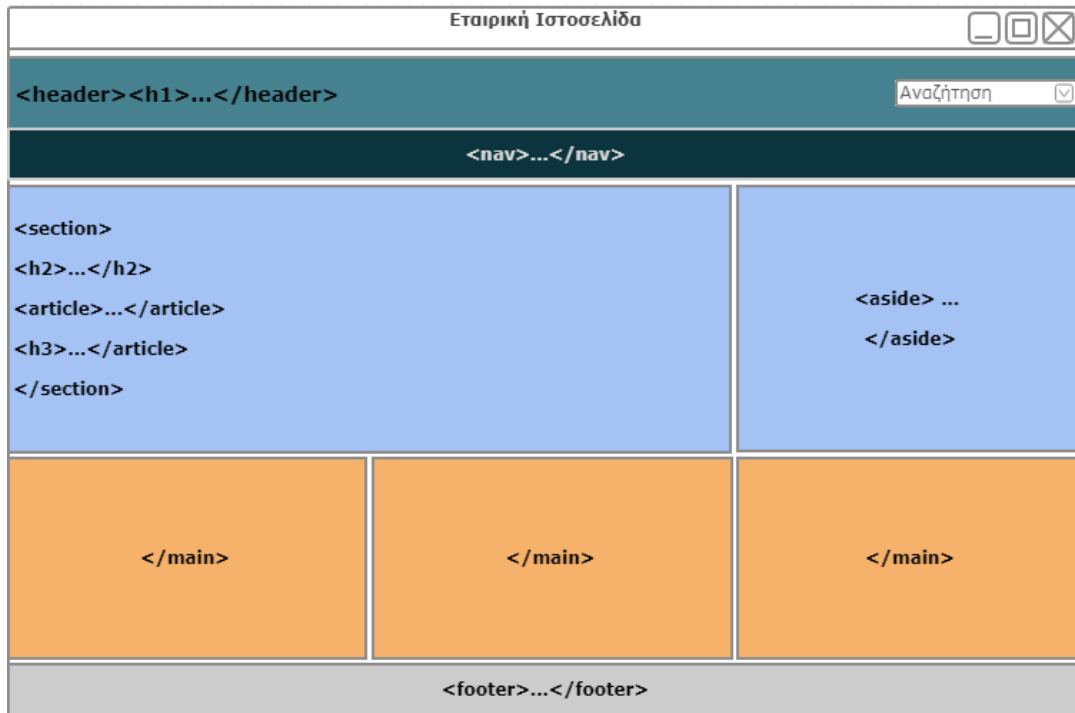


Εικόνα 2.12. Δομή ιστοσελίδας...χωρίς σημασιολογική αξία

Έτσι, τα σημασιολογικά στοιχεία της HTML5 διευκολύνουν τη δημιουργία δομής η οποία ξεκάθαρα αποκαλύπτει τη σημασία του περιεχομένου τόσο στον επισκέπτη όσο και σε και βοηθητικές συσκευές. Ένα ακόμα πλεονέκτημα αυτών των στοιχείων, σε σχέση με τα στοιχεία που αναφέρθηκαν σε προηγούμενες ενότητες, είναι η αξιοποίησή τους σε τεχνικές βελτιστοποίησης

για την καλύτερη κατάταξη των σελίδων στις μηχανές αναζήτησης. Τα στοιχεία αυτά, όπως απεικονίζονται στην επόμενη εικόνα είναι :

- Στοιχεία σημασιολογικής **ομαδοποίησης** (sectioning)
  - <article>, <section>, <aside>, <nav>, <h1> ... <h6>
- Στοιχεία σημασιολογικής **μορφοποίησης** (formatting)
  - <header>, <main>, <footer>



Εικόνα 2.13. Σημασιολογικά στοιχεία δομής σελίδων (HTML5)

Ένας εναλλακτικός σημασιολογικός τρόπος δόμησης σελίδων παραπλήσιος με αυτόν της εικόνας απεικονίζεται στο επόμενο παράδειγμα.

Παράδειγμα 2.54. Παράδειγμα σημασιολογικής δομής HTML5 (<https://gist.github.com/thomd/9220049>)

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4.   <meta charset="utf-8" />
5.   <title>Title</title>
6.   <link href="stylesheets/main.css" rel="stylesheet" />
7. </head>
8. <body>
9.   <header>
10.    <hgroup>
11.      <h1>Header</h1>
12.      <h2>Subheader</h2>
13.    </hgroup>
14.  </header>
15.  <nav>
16.    <ul>
17.      <li><a href="#">Menu Option 1</a></li>
18.      <li><a href="#">Menu Option 2</a></li>
```



```

19.         </ul>
20.     </nav>
21.     <section>
22.         <article>
23.             <header>
24.                 <h1>Article #1</h1>
25.             </header>
26.             <section>
27.                 This is the first article.
28.             </section>
29.         </article>
30.         <article>
31.             <header>
32.                 <h1>Article #2</h1>
33.             </header>
34.             <section>
35.                 This is the second article.
36.             </section>
37.         </article>
38.     </section>
39.     <aside>
40.         <section>
41.             <h1>Links</h1>
42.             <ul>
43.                 <li><a href="#">Link 1</a></li>
44.                 <li><a href="#">Link 2</a></li>
45.             </ul>
46.         </section>
47.         <figure>
48.             
51.             <figcaption>Foobar</figcaption>
52.         </figure>
53.     </aside>
54.     <footer>Footer</footer>
55. </body>
56. </html>

```

Στις επόμενες παραγράφους περιγράφουμε τα σημασιολογικά στοιχεία δομής (semantic/sectioning elements) της HTML5, τη χρήση τους και τη διαδικασία με την οποία αυτά συνδυάζονται για να έχουμε ένα καλό περίγραμμα σελίδας.

#### 2.8.4.1 Περίγραμμα Σελίδας

Η διαδικασία δημιουργίας δομής μιας σελίδας θα πρέπει να έχει ως στόχο ένα αποτελεσματικό **περίγραμμα εγγράφου** (document outline). Η HTML χρησιμοποιεί τα σημασιολογικά στοιχεία δομής (sectioning elements) για να δημιουργήσει το περίγραμμα, που πρακτικά λειτουργεί ως πίνακας περιεχομένων. Το περίγραμμα αξιοποιείται από συσκευές για να αναζητηθεί, να εντοπισθεί το πραγματικό περιεχόμενο μιας σελίδας - οι ενότητες που υπάρχουν σε αυτό καθώς και ο τρόπος που σχετίζονται μεταξύ τους. Έτσι πρέπει να γίνει κατανοητό πως η δημιουργία του περιγράμματος αποτελεί βασικό παράγοντα επιτυχίας μιας ιστοσελίδας.

Στο επόμενο παράδειγμα απεικονίζεται μια σελίδα ενός ιστολογίου (blog) με θέμα τα ενδιαφέροντα και hobbies του συγγραφέα.

Παράδειγμα 2.55. Περίγραμμα Σελίδας - blog

```
1. <!DOCTYPE html>
2. <html lang="el">
3. <head>
4. <meta charset="UTF-8">
5. <title>Ελέγχοντας το Περίγραμμα της Σελίδας</title>
6. </head>
7. <body>
8. <p>Το Blog μου </p>
9. <p>Μενού Πλοήγησης </p>
10. <p>Τα ενδιαφέροντά μου </p>
11. <p>Αγαπημένα Βιβλία </p>
12. <p>Φαντασίας </p>
13. <ul>
14. <li>The Girl with the Dragon Tattoo </li>
15. <li>The Martian</li>
16. <li>The Andromeda Strain </li>
17. </ul>
18. <p>Μυθιστορήματα</p>
19. <ul>
20. <li>The Count of Monte Cristo</li>
21. <li>The Little Prince </li>
22. <li>1984 </li>
23. </ul>
24. <p>Αγαπημένες Ταινίες και Σειρές </p>
25. <p>Ταινίες </p>
26. <ul>
27. <li>The Wall</li>
28. <li>Apocalypse Now</li>
29. <li>Fight Club</li>
30. </ul>
31. <p>Σειρές </p>
32. <ul>
33. <li>Dexter</li>
34. <li>Breaking Bad</li>
35. <li>Elementary</li>
36. </ul>
37. <p>Αγαπημένη Μουσική </p>
38. <p>Jazz </p>
39. <ul>
40. <li>Chuck Mangione</li>
41. <li>Miles Davis</li>
42. <li>John Coltrane</li>
43. </ul>
44. <p>Rock </p>
45. <ul>
46. <li>AC/DC</li>
47. <li>Deep Purple</li>
48. <li>Queen</li>
49. <li>Tool</li>
50. </ul>
51. <p>Αγαπημένο όλων των Εποχών</p>
```

```

52. <p>Ένα από τα αγαπημένα μου albums είναι το Moontan των Golden Earring που κυκλοφ
    όρησε το 1973. Το τραγούδι που με συναρπάζει ειδικά την ώρα που οδηγώ είναι το "R
    adar Love" το οποίο με κάνει να σκέφτομαι πως "Καμία απόσταση δεν είναι αρκετά με
    γάλη για να με κρατήσει μακριά από αυτούς που αγαπώ!" </p>
53. <p> copyright &copy; 2018. </p>
54. </body>
55. </html>

```

Εμφανίζοντας τη σελίδα στο φυλλομετρητή εύκολα μπορούμε να διακρίνουμε ενότητες, υποενότητες, επικεφαλίδες και γενικά περιεχόμενο που μπορεί να ομαδοποιηθεί. Στα στοιχεία που έχουν χρησιμοποιηθεί κυριαρχεί αυτό της παραγράφου <p>, και πρακτικά όλο το έγγραφο αντιμετωπίζεται ως μια παράγραφος. Για να ελέγξουμε το περίγραμμα της σελίδας μπορούμε:

- Να χρησιμοποιήσουμε το πρόσθετο "Document Outline" του Brackets
- Να εισάγουμε τον κώδικά μας στην σελίδα <http://gsnedders.html5.org/outliner>

Σε κάθε περίπτωση, διαπιστώνουμε πως το έγγραφο δεν έχει δομή! Το **πρώτο στάδιο** δημιουργίας δομής απαιτεί τη *χρήση επικεφαλίδων* (<h1> ... <h6>). Πριν την HTML5, η χρήση των headings αποτελούσε τον μοναδικό τρόπο για να ορίσει κάποιος δομή στο έγγραφο. Με τον εμπλουτισμό των σημασιολογικών sectioning στοιχείων της HTML5 γίνεται η δομή της σελίδας ακόμα από πιο κατανοητή.

#### 2.8.4.2 Περιοχή Πλοήγησης <nav>...</nav>

Αποτελεί το στοιχείο που θα φιλοξενήσει οποιοδήποτε περιεχόμενο πλοήγησης όπως μενού ή συνδέσμους προς άλλες σελίδες. Πιο συγκεκριμένα, προορίζεται για εμφάνιση **κύριων συνδέσμων πλοήγησης** που δεν αφορούν πληροφορία που προβάλλεται στο κύριο μέρος του περιεχομένου. Για τους συνδέσμους που ενδεχομένως σχετίζονται με το περιεχόμενο (article) θα μπορούν είτε να το ακολουθούν στο κάτω μέρος του (footer), να προβάλλονται ως συμπληρωματικό περιεχόμενο σε αντίστοιχο στοιχείο (aside), ή και να περιέχονται σε εμφωλευμένο στοιχείο <nav> μέσα στο περιεχόμενο.

Παράδειγμα 2.56. Παράδειγμα χρήσης στοιχείου <nav>

```

1. <nav>
2.   <a href="/html/">HTML</a> |
3.   <a href="/css/">CSS</a> |
4.   <a href="/js/">JavaScript</a> |
5.   <a href="/jquery/">jQuery</a>
6. </nav>

```

W3C Specification HTML5.1 – παραδείγματα και σενάρια χρήσης:  
<https://www.w3.org/TR/html51/sections.html#the-nav-element>

#### 2.8.4.3 Ενότητα <section>...</section>

Σύμφωνα με την W3C τεκμηρίωση το στοιχείο <section> εκφράζει μια **θεματική ομαδοποίηση περιεχομένου** και τυπικά αναγνωρίζεται από μια επικεφαλίδα (heading), ενώ μπορεί να περιέχει τη δική του κεφαλίδα (<header>) και υποσέλιδο (<footer>). Τυπικά μια σελίδα περιέχει ενότητες όπως:

- μια εισαγωγική ενότητα,
- μια ενότητα με πραγματικό περιεχόμενο και
- μια ενότητα με στοιχεία επικοινωνίας.

W3C Specification HTML5.1 – παραδείγματα και σενάρια χρήσης:  
<https://www.w3.org/TR/html51/sections.html#the-section-element>

#### 2.8.4.4 Άρθρο <article>...</article>

Ένα άρθρο πρέπει να περιλαμβάνει **περιεχόμενο** το οποίο μπορεί να είναι **ανεξάρτητο** (independent) και **αυτοδύναμο** (self-contained). Το στοιχείο αυτό θα πρέπει να περικλείει περιεχόμενο που μπορεί να διαβαστεί ανεξάρτητα από τις γειτονικές του ενότητες ή και ακόμα ανεξάρτητα από το υπόλοιπο περιεχόμενο του ιστότοπου. Η χρήση του <article> είναι αναγκαία όταν θέλουμε στα περιεχόμενα μιας σελίδας να εμφανιστεί το περιεχόμενο που περικλείει βάσει του <h1...6> στοιχείου που το συνοδεύει. Παραδείγματα αποτελούν:

- Αναρτήσεις σε ένα forum
- Αναρτήσεις σε blog
- Άρθρα εφημερίδων

Για τα στοιχεία <section> και <article> υπάρχει μια ασάφεια για το ποιο θεωρείται γονικό, καθώς το πρώτο προσδιορίζει μια ενότητα, ενώ το δεύτερο αυτόνομο περιεχόμενο μέσα σε ένα έγγραφο. Είναι φανερό πως βάσει ορισμού δεν είναι απόλυτος ο τρόπος συνδυασμού τους. Έτσι, είναι σύνηθες να δούμε σελίδες με <section> που περιέχει <article>, όπως και <article> να περιέχει <section>. Αλλά δεν είναι απίθανο να συναντήσουμε σελίδες που περιέχουν <section> με περιεχόμενο <section>, όπως και <article> περιεχόμενο <article>. Σε επόμενη ενότητα (2.8.4.9) αναλύουμε καλές πρακτικές για τη χρήση τους.

W3C Specification HTML5.1 – παραδείγματα και σενάρια χρήσης:  
<https://www.w3.org/TR/html51/sections.html#the-article-element>

#### 2.8.4.5 Συμπληρωματικό περιεχόμενο <aside>...</aside>

Το στοιχείο <aside> υποδέχεται περιεχόμενο που λειτουργεί **συμπληρωματικά με το βασικό περιεχόμενο**, και του οποίου η αφαίρεση δε θα αλλάξει το νόημα ή τη συνοχή του περιεχομένου. Η θέση που οι περισσότεροι επιλέγουν για το στοιχείο aside είναι δεξιά του περιεχομένου. Η πληροφορία που συνηθίζεται να φιλοξενεί μπορεί να είναι σύνδεσμοι σχετικοί (προτεινόμενο περιεχόμενο) με το τρέχον άρθρο, διαφημίσεις σε μορφή κειμένου ή και εικόνες.

W3C Specification HTML5.1 – παραδείγματα και σενάρια χρήσης:  
<https://www.w3.org/TR/html51/sections.html#the-aside-element>

#### 2.8.4.6 Κύριο περιεχόμενο μιας σελίδας <main>...</main>

Το στοιχείο <main> δε συμμετέχει στην δημιουργία του περιγράμματος, αλλά χρησιμοποιείται για να οριστεί το βασικό περιεχόμενο μιας σελίδας, σε αντίθεση με υλικό που επαναλαμβάνεται σε όλες τις σελίδες όπως συνδέσμους πλοήγησης, λογότυπα σελίδας, φόρμες αναζήτησης.

W3C Specification HTML5.1 – παραδείγματα και σενάρια χρήσης:

<https://www.w3.org/TR/html51/grouping-content.html#the-main-element>

#### 2.8.4.7 Κεφαλίδα <header>...</header>

Το στοιχείο <header> μπορεί να χρησιμοποιηθεί για να δηλώσει την κεφαλίδα της σελίδας ή ακόμα και την επικεφαλίδα ενός τμήματός (ενότητας) της. Χρησιμοποιείται για να ομαδοποιηθεί **εισαγωγικό περιεχόμενο** που μπορεί να επαναλαμβάνεται σε κάθε σελίδα όπως για παράδειγμα ένα λογότυπο και τον τίτλο/επωνυμία της σελίδας. Μπορεί επίσης να φιλοξενήσει τον πίνακα περιεχομένων, μια φόρμα αναζήτησης ή κάποιο άλλο γενικό περιεχόμενο που μπορεί να επαναλαμβάνεται.

W3C Specification HTML5.1 – παραδείγματα και σενάρια χρήσης:

<https://www.w3.org/TR/html51/sections.html#the-header-element>

#### 2.8.4.8 Υποσέλιδο <footer>...</footer>

Το στοιχείο <footer> περιλαμβάνει το περιεχόμενο που συνήθως εμφανίζεται στο κάτω μέρος της σελίδας (υποσέλιδο) παρέχοντας πληροφορίες όπως στοιχεία επικοινωνίας, διευθύνσεις κατασκευαστών, συνδέσμους για τα μέσα κοινωνικής δικτύωσης και γενικά ό, τι αφορά την ιστοσελίδα συνολικά. Μπορεί όμως να χρησιμοποιηθεί σε μια ενότητα ως το στοιχείο που ολοκληρώνει το περιεχόμενο.

W3C Specification HTML5.1 – παραδείγματα και σενάρια χρήσης:

<https://www.w3.org/TR/html51/sections.html#the-footer-element>

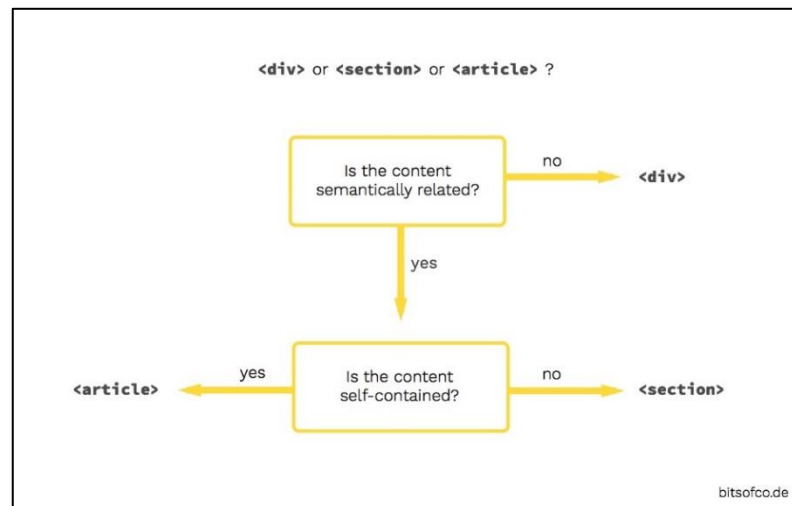
#### 2.8.4.9 Καλή Πρακτική: <section>, <article> ή <div>;

Η HTML5 έχει ήδη αποδείξει την αξία της για το πώς μπορεί κάποιος να δομήσει μια ιστοσελίδα ώστε να έχει νόημα τόσο για τους χρήστες όσο και για τις μηχανές. Από τις προηγούμενες ενότητες, όμως, μπορεί να δημιουργηθεί μια σύγχυση για το πότε θεωρείται καταλληλότερη η χρήση του <article> και πότε του <section> και αν υπάρχει ένας κανόνας για όλες τις περιπτώσεις για τον τρόπο που αυτά μπορούν να συνδυάζονται. Επίσης, υπάρχει και το ερώτημα που αφορά τη συνέχιση ή μη χρήσης του στοιχείου <div>. Συγκρίνουμε τα στοιχεία ομαδοποίησης βάσει της καταλληλότητάς τους για το περιεχόμενο που θα υποδεχτούν:

- Το στοιχείο <div> αποτελεί το γενικότερο στοιχείο ομαδοποίησης και σκοπός του είναι να ομαδοποιεί περιεχόμενο που δεν σχετίζεται σημασιολογικά. Πρακτικά είναι άχρηστο σε μηχανές αναζήτησης και βοηθητικές συσκευές, αλλά πάρα πολύ χρήσιμο στους σχεδιαστές για τη μορφοποίηση των σελίδων με CSS. Η χρήση του θα πρέπει να περιορίζεται μόνο σε θέματα σχεδίασης και μόνο στην περίπτωση που η χρήση των σημασιολογικών στοιχείων δεν είναι η κατάλληλη.

- Το στοιχείο `<section>` προσφέρει στα περιεχόμενά της μια πολύ συγκεκριμένη (θεματική) ομαδοποίηση σε σχέση με το `<div>`. Έτσι θα πρέπει πάντα να συνοδεύεται με κάποιο στοιχείο heading (`<h1>...<h6>`), το οποίο συνήθως αποτελεί το πρώτο στοιχείο του `<section>`.
- Το στοιχείο `<article>` προσφέρει σημασιολογική ομαδοποίηση όπως και το `<section>` με τη διαφορά ότι τα περιεχόμενά του θα πρέπει να είναι αυτοδύναμα.

Βάσει των παραπάνω προκύπτει το διάγραμμα που φαίνεται στην Εικόνα 2.14, ως ένας **οδηγός επιλογής**.



Εικόνα 2.14. Επιλογή σημασιολογικών στοιχείων δομής <https://bitsofco.de/sectioning-content-in-html5/>

Τέλος, πέρα από την επιλογή του καταλληλότερου στοιχείου, σημαντικό είναι να γνωρίζουμε πώς τα στοιχεία αυτά μπορούν να συνδυαστούν:

- **<article> μέσα σε <article>**: Στην περίπτωση που υπάρχει αυτοδύναμο περιεχόμενο το οποίο σχετίζεται με το περιεχόμενο που το “φιλοξενεί”.

Παράδειγμα 2.57. Παράδειγμα `<article>` σε `<article>`

```

1. <article>
2.   <h1>Article Title</h1>
3.   <p class="author">John Smith</p>
4.   <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
5.
6.   <article>
7.     <h2>Another Article</h2>
8.     <p class="author">Jane Doe</p>
9.     <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
10.  </article>
11.
12.  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
13. </article>
  
```

- **<article> μέσα σε <section>**: Στην περίπτωση που πολλαπλά άρθρα (αυτοδύναμα) ομαδοποιούνται θεματικά σε μια ενότητα `<section>` (container). Για παράδειγμα, οι πρόσφατες ανακοινώσεις “posts” μπορούν να ομαδοποιηθούν σε ένα ξεχωριστό section.

Παράδειγμα 2.58. Παράδειγμα <article> σε <section>

```
1. <section>
2.   <h1>Latest Blog Posts</h1>
3.
4.   <article>
5.     <h2>Blog Post Title</h2>
6.     <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. </p>
7.   </article>
8.
9.   <article>
10.    <h2>Blog Post Title</h2>
11.    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. </p>
12.  </article>
13. </section>
```

- <section> μέσα σε <article>: Κάθε αυτοδύναμο περιεχόμενο μπορεί να αποτελείται από πολλαπλές ενότητες (θεματικές περιοχές). Για παράδειγμα αν περιγράψουμε την τρέχουσα ενότητα θα προκύψει ο κώδικας HTML5 που ακολουθεί.

Παράδειγμα 2.59. Παράδειγμα <section> σε <article>

```
1. <article>
2.   <h1>Δημιουργώντας δομή στη HTML5 - div, section ή article;</h1>
3.
4.   <section>
5.     <h2>Τα στοιχεία γενικά</h2>
6.     <section>
7.       <h3>div</h3>
8.       <p>Το div αποτελεί το πιο γενικό στοιχείο ομαδοποίησης χωρίς σημασιο
       λογική αξία.</p>
9.     </section>
10.    <section>
11.      <h3>section</h3>
12.      <p>Το section προσφέρει σημασιολογική (θεματική) ομαδοποίηση στο περ
       ιεχόμενό του.</p>
13.    </section>
14.    <section>
15.      <h3>article</h3>
16.      <p>Το article είναι το πιο ειδικό στοιχείο για περιεχόμενο που θεωρε
       ίται αυτοδύναμο.</p>
17.    </section>
18.  </section>
19.
20.  <section>
21.    <h2>Οδηγός Επιλογής: div, section ή article;</h2>
22.    <!-- ... -->
23.  </section>
24.  <section>
25.    <h2>Σενάρια: Συνδυάζοντας τα στοιχεία</h2>
26.    <!-- ... -->
27.  </section>
28.
29. </article>
```

## Άσκηση 18





### 3. Παράρτημα: Συστήματα Ελέγχου Εκδόσεων Κώδικα (VCS)

#### 3.1 Εισαγωγή

Μια άλλη ανάγκη που προκύπτει τόσο από τη συνεργασία μας με άλλους, όσο και από την αυξανόμενη πολυπλοκότητα των εργασιών που θα αναλαμβάνουμε, σχετίζεται με την παρακολούθηση αλλαγών μέσα από Συστήματα Ελέγχου Εκδόσεων Κώδικα (“Version Control Systems” - VCS). Αν και η τυπική χρήση ενός VCS αφορά την παρακολούθηση κώδικα, τα συστήματα αυτά μπορούν να χρησιμοποιηθούν πρακτικά με οποιασδήποτε μορφής «ηλεκτρονικά τεκμήρια» τα οποία θέλουμε να προστατεύσουμε αλλά και να παρακολουθούμε την εξέλιξή τους.

Τα Συστήματα Ελέγχου Εκδόσεων (Version Control Systems) αποτελούν εργαλεία που βοηθούν ομάδες να διαχειριστούν τις αλλαγές που πραγματοποιούνται σε αρχεία μέσα στο χρόνο, με δυνατότητα **ελέγχου, σύγκρισης** με παλαιότερες εκδόσεις και την **επαναφορά** αν αυτό απαιτηθεί. Ταυτόχρονα, επιτρέπουν στις ομάδες να εργάζονται **συνεργατικά αλλά και αυτόνομα** χωρίς να διαταράσσεται η ροή εργασίας.

Τα μέλη των ομάδων που υλοποιούν ένα project μπορούν να λειτουργούν ασύγχρονα με βάση πάντα τον ίδιο κώδικα, όπου κάποιος μπορεί να διορθώνει προβλήματα που έχουν εντοπιστεί (bugs), ενώ κάποιος άλλος να ξεκινά μια νέα έκδοση, η οποία θα δίνει νέα χαρακτηριστικά στην εφαρμογή. Σε κάθε περίπτωση, τα συστήματα αυτά προσφέρουν τη δυνατότητα καταγραφής σχολίων και παρατηρήσεων για κάθε αλλαγή που έχει καταγραφεί εμπλουτίζοντας την **τεκμηρίωση** του έργου.

Στην περίπτωση που οι ομάδες είναι μεγάλες, τα συστήματα αυτά βοηθούν να διευκρινιστούν θέματα αλλαγών (**ποιος-πότε-τι**) καταγράφοντας τις ενέργειες κάθε συμμετέχοντα. Αυτό βοηθά στην αποφυγή προβλημάτων όπου οι αλλαγές που πραγματοποίησε ένα μέλος αφορούν κώδικα που ταυτόχρονα έχει τροποποιηθεί (ή τροποποιείται) από κάποιο άλλο μέλος της ομάδας. Έτσι προσφέρει εργαλεία με τα οποία εντοπίζονται και λύνονται αυτού του είδους οι “**συγκρούσεις**” (conflict resolution).

Τα συστήματα ελέγχου εκδόσεων κώδικα αποτελούν μονόδρομο για την **κλιμάκωση** (scale up) και **επιτυχή ολοκλήρωση** ενός έργου είτε αυτό υλοποιείται από μεγάλες απομακρυσμένες ομάδες, είτε αφορά το έργο ενός προγραμματιστή. Τα πλεονεκτήματα που προσφέρουν συνοψίζονται ως εξής:

1. Διαθέτουμε το **πλήρες ιστορικό των αλλαγών ενός αρχείου** – καταγράφοντας όλα τα μέλη που το επεξεργάστηκαν. Οι αλλαγές δεν αφορούν μόνο τα περιεχόμενα των αρχείων, αλλά και τη διαδικασία δημιουργίας και διαγραφής τους. Το ιστορικό περιλαμβάνει τον συγγραφέα, την ημερομηνία αλλαγής και σημειώσεις ή παρατηρήσεις που αφορούν την αλλαγή. Έτσι προσφέρεται η δυνατότητα να προσπελάσουμε προηγούμενες εκδόσεις του αρχείου για να συσχετίσουμε τις αλλαγές με κάποιο πρόβλημα που έχει αναφερθεί.
2. Η δυνατότητα **διακλάδωσης και ενσωμάτωσης** (Branching and Merging) επιτρέπει στα μέλη μιας ομάδας να λειτουργούν αυτοδύναμα πάνω σε ξεχωριστά κομμάτια κώδικα του ίδιου συστήματος, χωρίς ο ένας να επηρεάζει τη δουλειά του άλλου. Δημιουργούμε ένα branch όταν θέλουμε να δοκιμάσουμε κάποια νέα χαρακτηριστικά, χωρίς να επηρεάζεται

η βασική ροή της κύριας έκδοσης. Αν όλα πάνε καλά ολοκληρώνοντας τους ελέγχους, μπορούν αυτές οι αλλαγές να ενσωματωθούν στο βασικό προϊόν ολοκληρώνοντας τον κύκλο ζωής του branch.

3. Η δυνατότητα **ιχνηλασιμότητας** (traceability) επιτρέπει τον πλήρη έλεγχο των αλλαγών που έχουν καταγραφεί και σε συνδυασμό με συστήματα διαχείρισης έργων (project management) ή καταγραφής αιτημάτων (bug tracking) μπορεί να υποσημειωθεί κάθε αλλαγή με την πληροφορία που την προκάλεσε. Με αυτό τον τρόπο το ιστορικό αλλαγών εμπλουτίζεται με πληροφορία που μπορεί να χρησιμοποιηθεί τόσο για σκοπούς εκπαίδευσης, τεκμηρίωσης ακόμα και εγκληματολογικής έρευνας (forensic analysis).

### 3.2 Γενικά για το Git

Το Git είναι με διαφορά το πιο διαδεδομένο σύγχρονο VCS. Δημιουργήθηκε το 2005 από τον δημιουργό του Linux, Linus Torvalds, και αποτελεί ένα κατανεμημένο VCS. Σε σχέση με άλλα συστήματα που χρησιμοποιούν μια πιο κεντρικοποιημένη αρχιτεκτονική (πχ. SVN, CVS), επιτρέπει στους δημιουργούς να διατηρούν **τοπικά μια πλήρη έκδοση του κώδικα**, ενώ ταυτόχρονα υποστηρίζει λειτουργίες όπως commit, blame, diff, merge με μεγάλη απόδοση. Ταυτόχρονα υποστηρίζει branching, merging και τα πανίσχυρα pull requests. Η ισχύς των pull requests είναι αυτή που ενισχύει την ικανότητα συνεργασίας, επίβλεψης και καθιστά το Git το standard για ανάπτυξη λογισμικού.

Οι πιο διαδεδομένες υπηρεσίες βασισμένες στο Git είναι το GitHub, το Bitbucket και το Gitlab. Ενώ διαφοροποιούνται στο επιχειρηματικό μοντέλο και στη φιλοσοφία διάθεσης των υπηρεσιών, προσφέρουν ένα κοινό σετ χαρακτηριστικών:

- Pull requests
- Code review
- Inline editing
- Issue tracking
- Markdown support
- Advanced permission management
- Hosted static web pages
- Fork / Clone Repositories
- Snippets

### 3.3 Μοντέλο (Συν)εργασίας

Το μοντέλο (συν)εργασίας περιγράφει τους κανόνες και τεχνικές μέσω των οποίων κάποιος μπορεί να ξεκινήσει να αξιοποιεί τα πλεονεκτήματα του Git είτε αυτό γίνεται σε συνεργατικό επίπεδο, είτε σε ατομικό. Ειδικά στη δεύτερη περίπτωση, δεν απαιτείται η χρήση ενός λογαριασμού σε υπηρεσία cloud (όπως το GitHub ή το Bitbucket) αλλά το λογισμικό που ενεργοποιεί τη χρήση ενός αποθετηρίου σε τοπικό επίπεδο (H/Y). Βέβαια, για λόγους ασφάλειας και προστασίας της εργασίας μας (π.χ. αστοχίες υλικού) θεωρείται καλή πρακτική η παράλληλη χρήση μιας cloud υπηρεσίας. Τα προτεινόμενα βήματα για την οργάνωση του περιβάλλοντος εργασίας είναι τα εξής:

1. Δημιουργία Λογαριασμού σε μια Git Υπηρεσία (π.χ. GitHub)
2. Εγκατάσταση Λογισμικού Διαχείρισης Αποθετηρίων (π.χ. GitHub Desktop)
3. Εγκατάσταση και παραμετροποίησης Επεξεργαστή Κώδικα (π.χ. Brackets)
4. Δημιουργία Project – Τοπικού Αποθετηρίου
5. Διαχείριση Περιεχομένων Αποθετηρίου

Τα παραπάνω βήματα μπορούν να εκτελεστούν γραμμικά ή ανάλογα με το σημείο εκκίνησης να μετατοπιστούν ή και να εξαιρεθούν. Σε κάθε περίπτωση, οι επόμενες ενότητες περιγράφουν τη διαδικασία υλοποίησης των παραπάνω βημάτων. Αρχικά, περιγράφονται δύο τυπικά σενάρια χρήσης Git.

### 3.3.1 Σενάριο 1: Τοπικά χωρίς τη χρήση λογαριασμού Cloud

Για την εκκίνηση ενός project, και την παρακολούθησή του με τεχνολογίες Git δεν απαιτείται καν η πρόσβαση στο διαδίκτυο. Το μόνο που χρειάζεται είναι ο επεξεργαστής κώδικα της προτίμησής μας και τα εργαλεία Git. Με αυτό τον τρόπο εργαζόμαστε τοπικά ακολουθώντας τα παρακάτω βήματα:

1. Στον κεντρικό κατάλογο του project προκαλούμε τη δημιουργία τοπικού αποθετηρίου μέσω της εντολής “git init” ή επιλέγοντας την αντίστοιχη εντολή μέσα από το γραφικό περιβάλλον του επεξεργαστή κώδικα.
2. Ολοκληρώνουμε την επεξεργασία των αρχείων και στην περίπτωση που αυτές αποτελούν μια αυτοτελή αλλαγή (π.χ. διόρθωση ενός προβλήματος) ομαδοποιούμε τα αρχεία που θα αποτελέσουν την επόμενη καταχώρηση του αποθετηρίου μέσω της εντολής “git add”.
3. Τέλος, κατοχυρώνουμε την ολοκλήρωση αυτής της επεξεργασίας μέσω της εντολής “git commit -m [message]” μέσω της οποίας δημιουργείται η επόμενη καταχώρηση στο αποθετήριο με μια σύντομη περιγραφή της εργασίας που εκτελέστηκε (π.χ. αποκατάσταση του προβλήματος XY)

### 3.3.2 Σενάριο 2: Συνεργασία σε ομάδα με χρήση GitHub.com

Ως μέλος ομάδας, θα έχουμε πρόσβαση σε ένα online αποθετήριο στο GitHub, όπου μέσω εκχωρημένων δικαιωμάτων, θα μπορούμε να λαμβάνουμε τις αλλαγές και να στέλνουμε τη δουλειά που έχουμε ολοκληρώσει, ώστε να γίνει διαθέσιμη σε όλους. Αυτό δε σημαίνει ότι θα πρέπει συνεχώς να εργαζόμαστε με πρόσβαση στο διαδίκτυο, οπότε ο τρόπος εργασίας κατά ένα μέρος πλησιάζει σε αυτό που περιγράψαμε προηγουμένως.

Σε μια τυπική μέρα, για να ξεκινήσουμε την ενασχόλησή μας με το project θα πρέπει:

1. Πριν ξεκινήσουμε την επεξεργασία αρχείων, θα συγχρονίσουμε (pull) το τοπικό μας αποθετήριο με όσες αλλαγές έχουν προκύψει από την τελευταία φορά που συνδεθήκαμε σε αυτό. Δεν πρέπει να ξεχνάμε πως η συνεργασία σε ομάδες μπορεί – και συνήθως συμβαίνει – να πραγματοποιείται σε διαφορετικά time zones!
2. Να μπορούμε να αντιμετωπίσουμε το ενδεχόμενο σύγκρουσης (conflict) στα αρχεία που έχουμε αφήσει υπό επεξεργασία στο τοπικό με αυτά που υπάρχουν στο online αποθετήριο. Σε αυτή την περίπτωση θα πρέπει να εξετάσουμε τις αλλαγές μέσω

διαθέσιμων εργαλείων (diff tools) και δομών του Git όπως η Stash, για να αποφασίσουμε ποιες αλλαγές θα πρέπει να παραμείνουν και τον τρόπο διευθέτησής τους.

3. Το μέρος της επεξεργασίας των τοπικών μας αρχείων δεν έχει διαφορά από αυτό που περιγράψαμε στην προηγούμενη ενότητα. Στην περίπτωση εργασίας χωρίς σύνδεση στο διαδίκτυο, χρησιμοποιούμε αποκλειστικά το τοπικό αποθετήριο για να καταγράψουμε τη δουλειά μας.
4. Στο τέλος της ημέρας, ή ακόμα καλύτερα όταν ολοκληρώνεται μια κρίσιμη αλλαγή στον κώδικα, μπορούμε να την συγχρονίσουμε (push) με το GitHub.com, ώστε να την έχουν διαθέσιμη οι συνεργάτες μας.

### 3.4 Λογισμικά για τη Διαχείριση Αποθετηρίων

Υπάρχει πληθώρα λογισμικών με τα οποία μπορούμε να διαχειριστούμε τα αποθετήρια κώδικα προσφέροντας διαφορετικές δυνατότητες. Αρχικά, πριν αποφασίσουμε τα εργαλεία επιλογής μας μπορούμε να αξιοποιήσουμε το GitHub.com για να:

1. Δημιουργούμε αρχεία, καταλόγους και δομή στο project μας.
2. Να ανεβάζουμε αρχεία που έχουμε στον Η/Υ.
3. Να τροποποιούμε απευθείας τα περιεχόμενα των αρχείων που βρίσκονται ήδη στο αποθετήριο, χωρίς να απαιτείται να τα κατεβάσουμε και να τα ξανανεβάσουμε.
4. Να διαγράφουμε αρχεία ή καταλόγους που δε χρειαζόμαστε πλέον.
5. Να παρακολουθούμε το ιστορικό εξέλιξης του αποθετηρίου και των αρχείων που περιέχει.

Πέρα από αυτό, υπάρχουν πολλοί τρόποι να συνδυαστούν τα διαθέσιμα εργαλεία και να προσαρμοστούν στον τρόπο που προτιμά ο δημιουργός:

- **Μέσω κονσόλας (console):** Υπάρχει δυνατότητα διαχείρισης του αποθετηρίου αποκλειστικά μέσω κονσόλας χρησιμοποιώντας εντολές GIT.
- **Αυτόνομη Διαχείριση με Εφαρμογή (client):** Μέσω γραφικού περιβάλλοντος το οποίο επιτρέπει τη διαχείριση πολλαπλών αποθετηρίων.
- **Συγγραφή & Διαχείριση “2-σε-1” μέσω πρόσθετων (plugins):** Συγγραφή, εποπτεία και διαχείριση του αποθετηρίου μέσα από την ίδια εφαρμογή. Όλοι οι σύγχρονοι επεξεργαστές κώδικα (Atom, Brackets, Sublime Text, Visual Studio Code) υποστηρίζουν πρόσθετα διαχείρισης αποθετηρίων.

Στις επόμενες ενότητες, περιγράφονται οι διαδικασίες δημιουργίας λογαριασμού στο GitHub.com, η εγκατάσταση προγράμματος (client) και η παραμετροποίηση ενός editor με υποστήριξη σε Git.

#### 3.4.1 Δημιουργία Λογαριασμού στο GitHub

Το GitHub είναι πρώτο σε επιλογή Git cloud, αν και δωρεάν προσφέρει μόνο δημόσια αποθετήρια - αντίθετα το BitBucket υποστηρίζει δωρεάν ιδιωτικά αποθετήρια για μικρές ομάδες. Η διαδικασία εγγραφής στην υπηρεσία GitHub (<https://github.com/>) ολοκληρώνεται μέσα σε 3 βήματα και αρχικά απαιτεί:

- την επιλογή ενός μοναδικού username,

- την ύπαρξη ενός email (για την επικύρωση της εγγραφής) και
- την επιλογή ενός συνθηματικού (password) με μήκος 7 χαρακτήρων το οποίο θα περιλαμβάνει τουλάχιστον 1 πεζό (αγγλικό) γράμμα και έναν αριθμό.

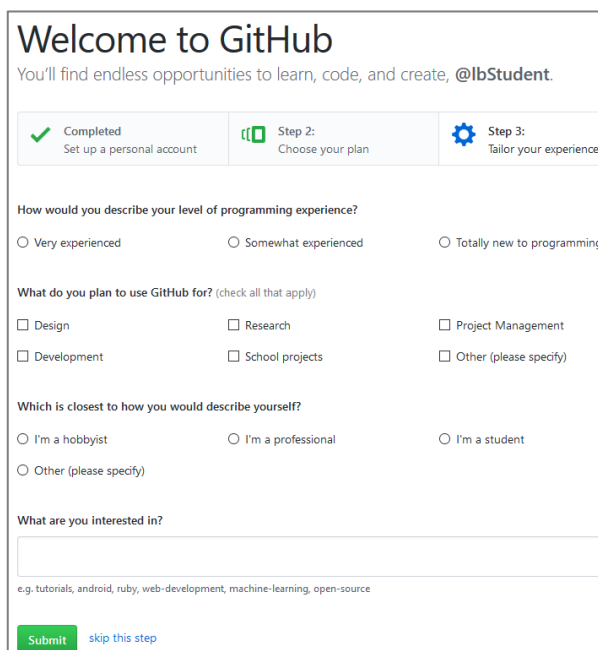
Εικόνα 3.1. Δημιουργία Λογαριασμού GitHub

Στη συνέχεια (βήμα 2), απαιτεί την επιλογή του personal plan και σε αυτή την περίπτωση θα επιλέξουμε το δωρεάν πλάνο με απεριόριστα δημόσια αποθετήρια (public repositories).

Εικόνα 3.2. Επιλογή δωρεάν απεριόριστων δημόσιων αποθετηρίων (Βήμα 2)

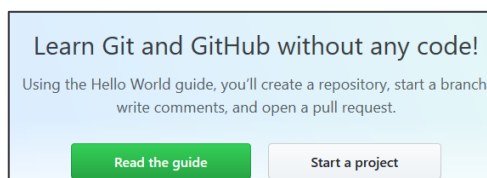
Τέλος, στο βήμα 3 αν και προαιρετικό (μπορούμε να το παραλείψουμε πατώντας “Skip this step”), θα μπορούσε να φανεί χρήσιμο για κάποιον που μόλις ξεκινάει να δηλώσει τα

ενδιαφέροντά του ώστε να λάβει μια περισσότερο προσωποποιημένη πληροφόρηση ή εκπαιδευτικό υλικό.



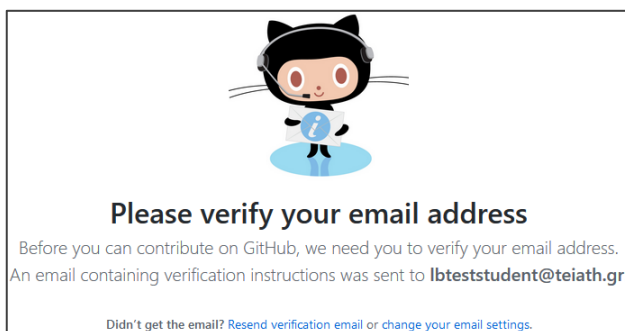
Εικόνα 3.3. Ολοκλήρωση Εγγραφής στο GitHub (Βήμα 3)

Ολοκληρώνοντας την εγγραφή στο GitHub βλέπουμε την παρακάτω εικόνα, από την οποία μπορούμε να ανοίξουμε έναν πολύ χρήσιμο οδηγό για το Git (10 λεπτών), τον οποίο προτείνουμε να διαβάσετε. Αλλιώς, μπορούμε να ξεκινήσουμε τη διαχείριση/δημιουργία ενός νέου αποθετηρίου (Project).



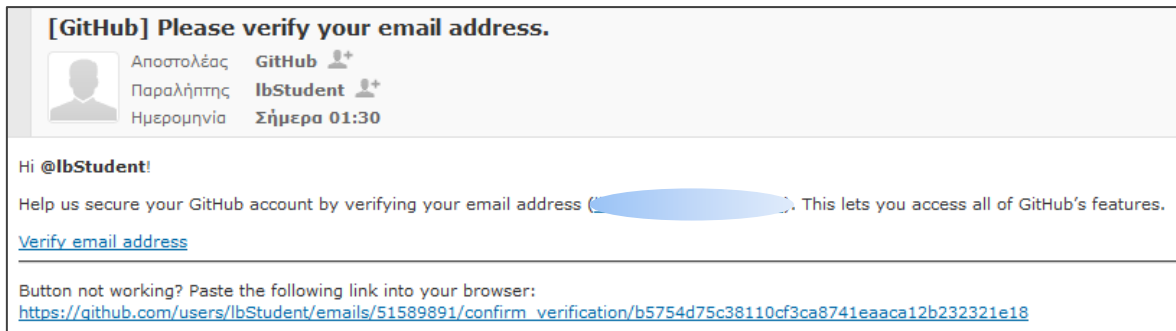
Εικόνα 3.4. Ολοκλήρωση εγγραφής – βασικός οδηγός / δημιουργία project

Στην περίπτωση που απαιτηθεί η επικύρωση του λογαριασμού που δημιουργήσαμε, μέσα από ένα ηλεκτρονικό μήνυμα που μας έχει αποσταλεί στην διεύθυνση που χρησιμοποιήσαμε κατά τη δημιουργία του λογαριασμού.



Εικόνα 3.5.Επικύρωση του λογαριασμού αλληλογραφίας

Σε αυτή την περίπτωση αρκεί να επισκεφθούμε τον σύνδεσμο “Verify email address” που υπάρχει στο μήνυμα.



Εικόνα 3.6. E-mail επικύρωσης λογαριασμού GitHub

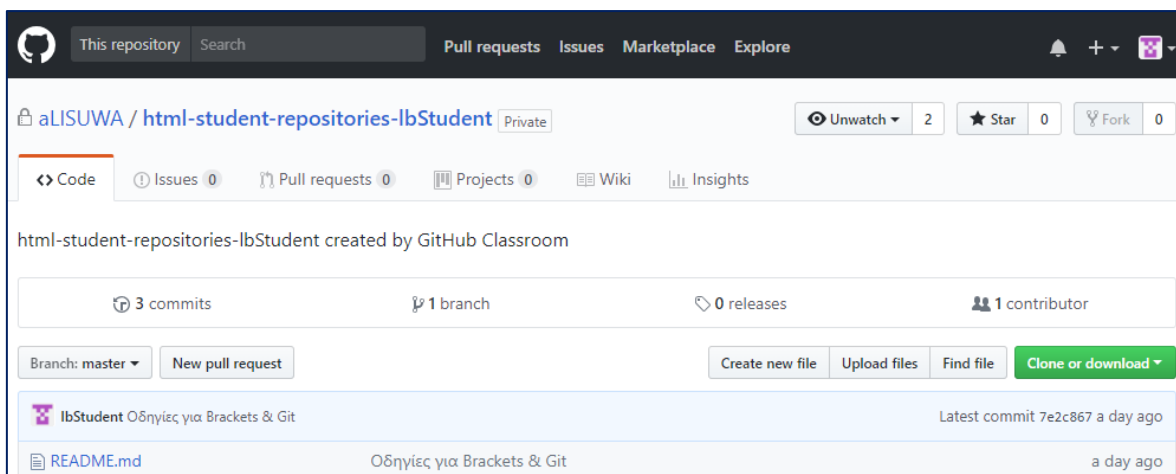
### 3.4.2 Δημιουργία Project – Τοπικό Αποθετήριο

Η δημιουργία ενός καταλόγου project πάντα αποτελεί καλή πρακτική, ώστε να αποφύγουμε τα αρχεία της δουλειάς μας να μπερδεύονται με άλλα έγγραφα του Η/Υ. Επιλέγουμε έναν κατάλογο που γνωρίζουμε εξ αρχής τη θέση του όπως C:\MyProjects\LiSProject ή στην επιφάνεια εργασίας των Windows το οποίο μεταφράζεται σε C:\Users\XXXX\Desktop\LiSProject.

Σε αυτή τη φάση ο κατάλογος θα είναι κενός ή μπορεί να περιέχει μέρος κώδικα που έχουμε λάβει από τους συνεργάτες μας (π.χ. σε μορφή .zip). Το πρώτο βήμα είναι η αρχικοποίηση του αποθετηρίου (init), το οποίο μπορεί να πραγματοποιηθεί με 2 τρόπους:

- Με χρήση των εργαλείων Git
- Με χρήση της εφαρμογής (client) π.χ. GitHub Desktop
- Με χρήση επεξεργαστή κώδικα που αξιοποιεί πρόσθετο Git

Θεωρούμε, όπως στις περισσότερες περιπτώσεις, πως το αποθετήριο που θα δημιουργήσουμε προέρχεται από ένα online αποθετήριο στο οποίο έχουμε πρόσβαση - **όπως αυτό για τις ανάγκες του μαθήματος.**



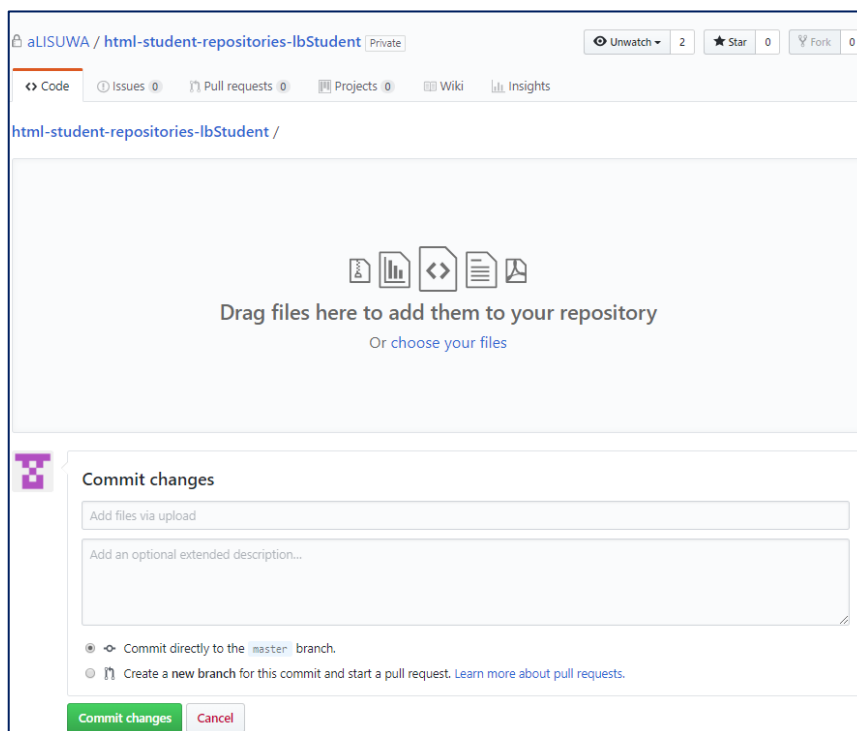
Εικόνα 3.7. Το προσωπικό αποθετήριο στο GitHub

Από το πράσινο κουμπί “Clone or Download” στην (Εικόνα 3.7) μπορούμε να κατεβάσουμε τον κώδικα του αποθετηρίου στον προσωπικό μας Η/Υ, και συγκεκριμένα στον κατάλογο του project. Η διαδικασία αυτή μπορεί να πραγματοποιηθεί με τη χρήση του GitHub Desktop (3.4.3).

Έχοντας αποθηκεύσει τα αρχεία του project σε τοπικό επίπεδο μπορούμε να αρχίσουμε την επεξεργασία τους με τον επεξεργαστή κώδικα της επιλογής μας. Κάθε φορά που ολοκληρώνουμε



μια άσκηση την αποθηκεύουμε και την ανεβάζουμε στο αποθετήριό μας χρησιμοποιώντας την επιλογή “Upload files” δίνοντας μια σύντομη περιγραφή για το “Commit”.

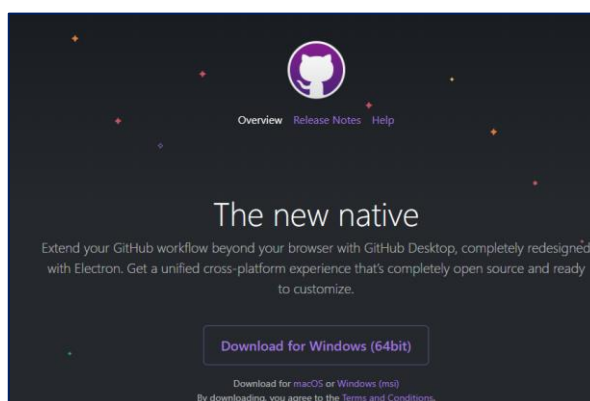


Εικόνα 3.8. Ανέβασμα αρχείων στο GitHub και Δημιουργία Commit

Η παραπάνω διαδικασία είναι αρκετά απλή, αλλά απαιτεί την εναλλαγή προγραμμάτων για την επεξεργασία κώδικα και την αποθήκευση αλλαγών (commits) στο αποθετήριο.

### 3.4.3 GitHub Desktop

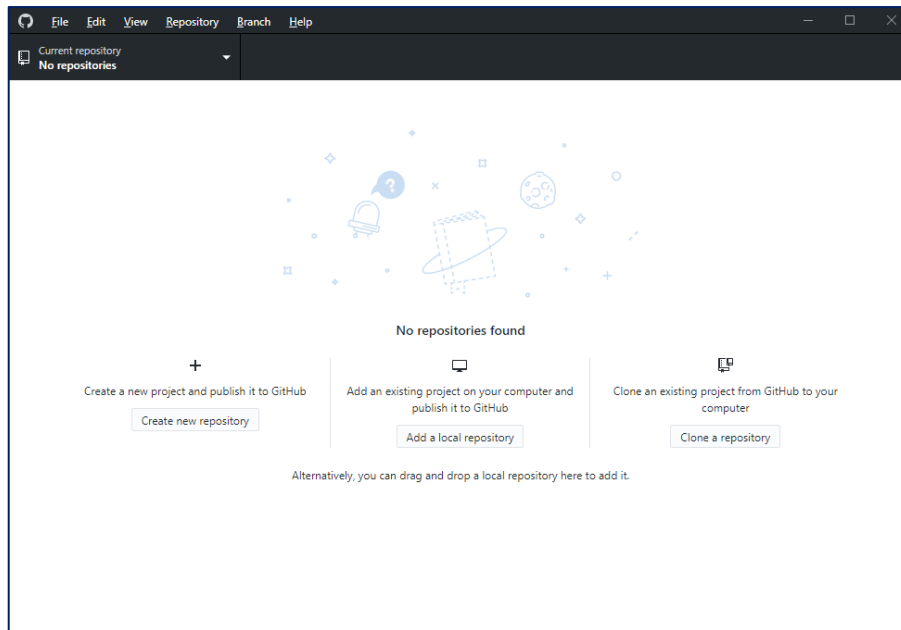
Το GitHub Desktop αποτελεί μια εφαρμογή (ανοιχτού κώδικα) με την οποία μπορούμε να επεκτείνουμε τον τρόπο με τον οποίο διαχειριζόμαστε τα αποθετήριά μας. Είναι διαθέσιμο για Windows και MacOS και μπορούμε να το κατεβάσουμε από τη διεύθυνση <https://desktop.github.com/>.



Εικόνα 3.9. Εγκατάσταση GitHub Desktop

Η εγκατάσταση ολοκληρώνεται σε ελάχιστο χρόνο χωρίς να απαιτεί ρυθμίσεις εμφανίζοντας το κεντρικό παράθυρο (Εικόνα 3.10).



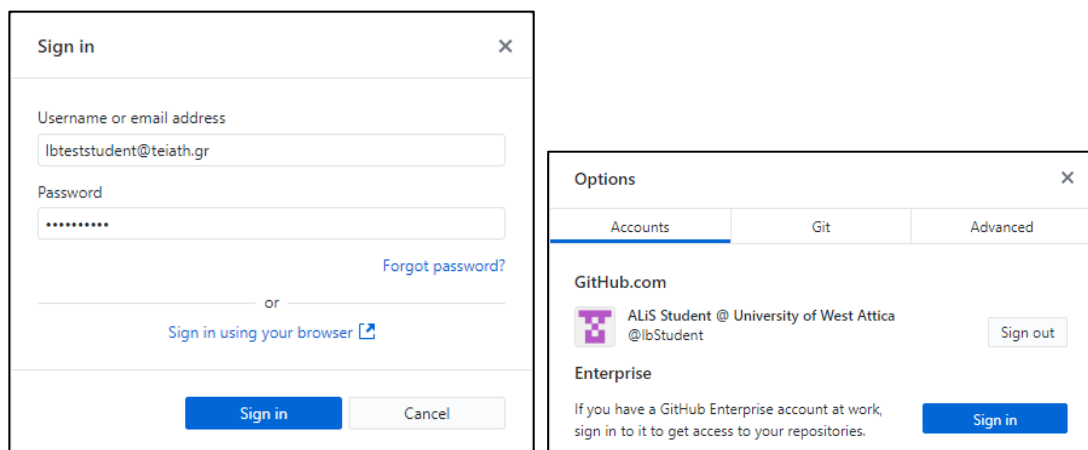


Εικόνα 3.10. Κεντρικό παράθυρο GitHub

Αρχικά, βλέπουμε πως δεν υπάρχει κάποιο «εγκατεστημένο» αποθετήριο (“No repositories”), ενώ προτείνονται 3 εναλλακτικές:

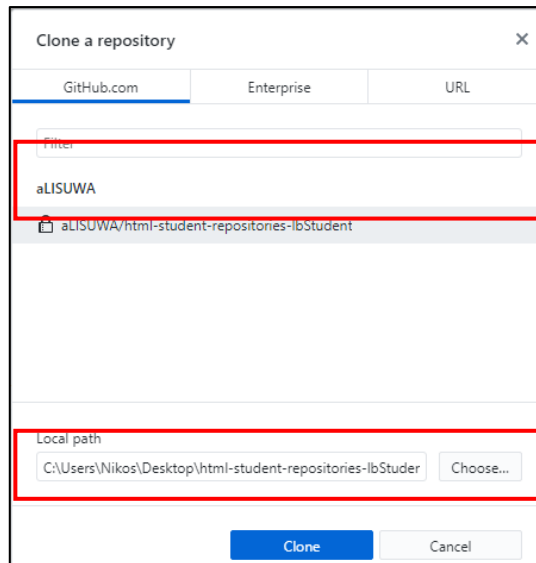
- Η **δημιουργία** ενός νέου project και δημοσίευσή του στο GitHub (Create new repository).
- Η **δημοσίευση ενός τοπικού** αποθετηρίου στο GitHub (Add a local repository)
- Η **κλωνοποίηση** ενός αποθετηρίου που είναι ήδη αναρτημένο στο GitHub.com

Για την πραγματοποίηση των ασκήσεων του εργαστηρίου, θα επιλέξουμε την κλωνοποίηση καθώς το αποθετήριο έχει ήδη δημιουργηθεί. Στη συνέχεια, χρησιμοποιώντας τους προσωπικούς κωδικούς για το GitHub.com θα επιλέξουμε “Sign in” (Εικόνα 3.11).



Εικόνα 3.11. Σύνδεση στο αποθετήριο

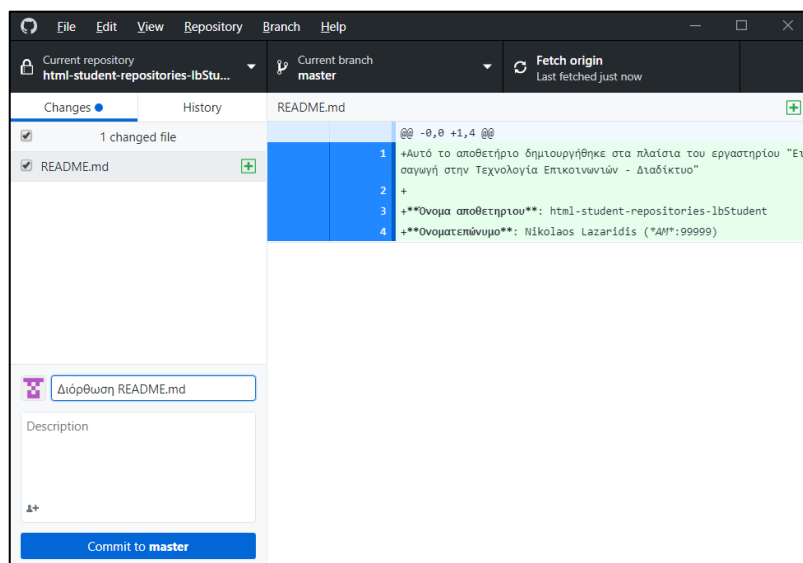
Στη συνέχεια, μπορούμε να επιλέξουμε το **αποθετήριο** που θέλουμε να διαχειριστούμε μέσω της εφαρμογής και τον **κατάλογο** αποθήκευσης των αρχείων στο δίσκο.



Εικόνα 3.12. Κλωνοποίηση Αποθετηρίου στον τοπικό δίσκο του Η/Υ

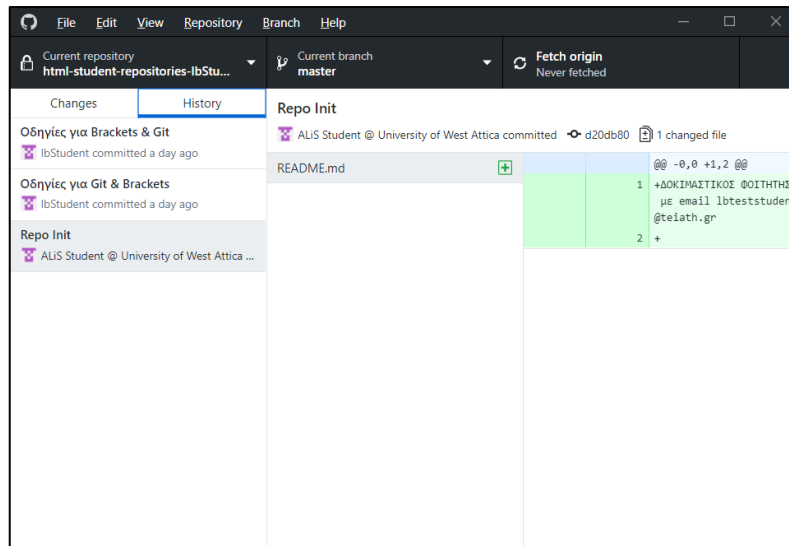
Στην παρούσα φάση, το αποθετήριο μας έχει αντιγραφεί στον Η/Υ και μέσω της εφαρμογής μπορούμε να διαχειριστούμε:

- Τις **αλλαγές** (Changes) που έχουν καταγραφεί στα τοπικά αρχεία από το τελευταίο commit στο αποθετήριο (Ctrl+1) π.χ. την επεξεργασία του αρχείου README.md



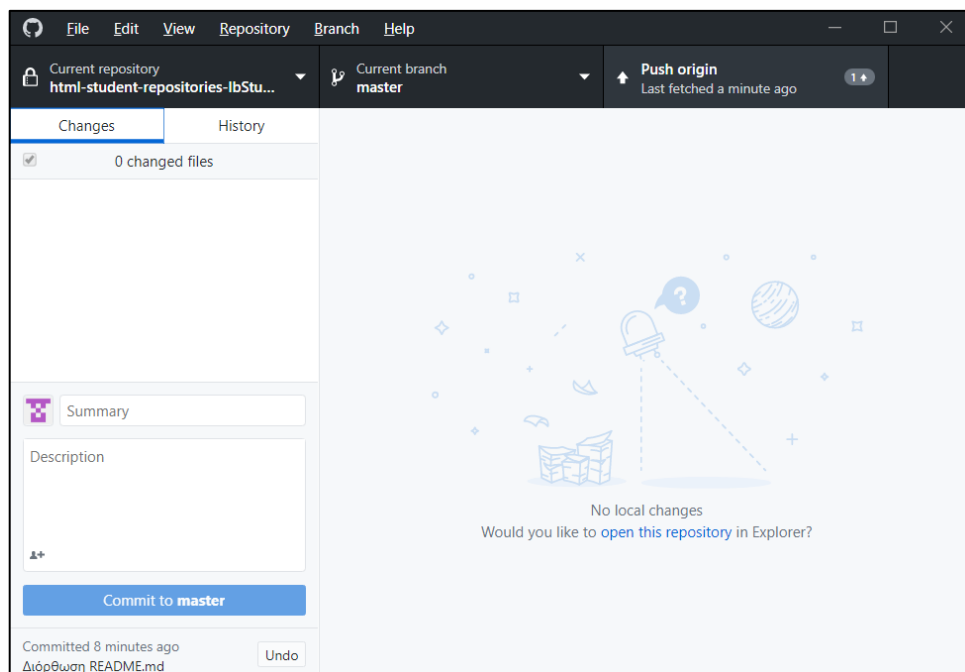
Εικόνα 3.13. Έλεγχος Αλλαγών - Δημιουργία Commit

- Το **ιστορικό αλλαγών** (History) του αποθετηρίου (Ctrl+2)



Εικόνα 3.14. Ιστορικό αλλαγών

- Εντολές κατακύρωσης των αλλαγών (**commits**) που έχουν πραγματοποιηθεί στα αρχεία σε τοπικό επίπεδο για το κεντρικό branch (**master**).
- Τον συγχρονισμό της τοπικής έκδοσης του αποθετηρίου με αυτό που βρίσκεται στο GitHub.com μέσω της εντολής **Push** (Ctrl+P)
- Την **ανάκτηση** όλων των αλλαγών που έχουν καταχωρηθεί στο online αποθετήριο (στο GitHub.com) με σκοπό την ενημέρωση της τοπικής έκδοσης, μέσω της εντολής **Pull** (Ctrl+Shift+P).



Εικόνα 3.15. Συγχρονισμός Αποθετηρίων με Push & Pull

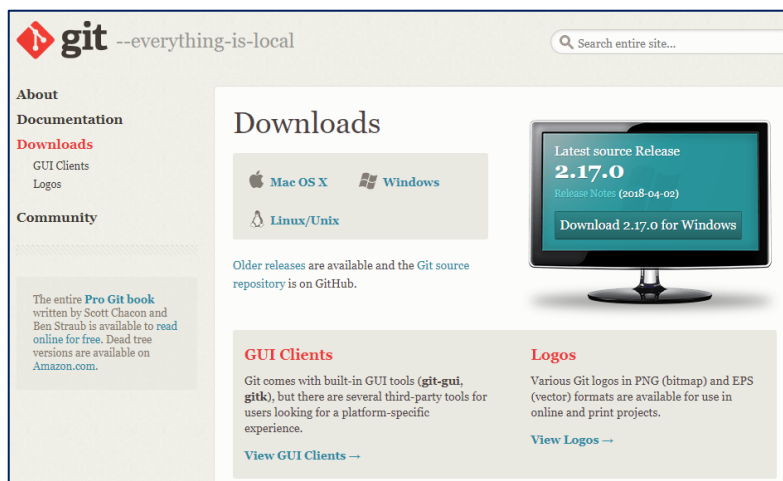
- Την **διαγραφή** του αποθετηρίου, η οποία αφαιρεί το αποθετήριο από την εφαρμογή GitHub Desktop αλλά δεν διαγράφει τα αρχεία από τον δίσκο του Η/Υ.

Επιπλέον, επιτρέπει τη δημιουργία και τη διαχείριση branches, επιτρέποντας τη συγχώνευσή τους με το τρέχον, καθώς και τη δημιουργία pull requests τα οποία είναι μεν πολύ χρήσιμα, αλλά ξεφεύγουν από τους σκοπούς αυτού του κειμένου.

### 3.4.4 Git Tools

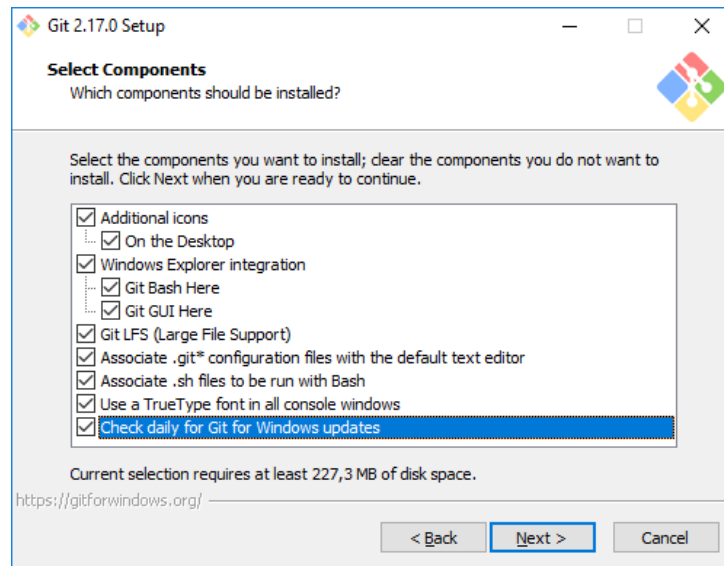
Για να μπορούμε να διαχειριστούμε το αποθετήριο τόσο μέσω εντολών σε κονσόλα, όσο και μέσα από πρόσθετα που υπάρχουν στους επεξεργαστές κώδικα όπως το Brackets, απαιτείται η εγκατάσταση των Git Tools. Αυτά είναι διαθέσιμα στην διεύθυνση <https://git-scm.com/downloads> για λειτουργικά MacOS, Windows και Linux, ενώ μπορούμε να κατεβάσουμε δωρεάν το βιβλίο “ProGit” (<https://git-scm.com/book/gr/v2>) και σε ελληνική μετάφραση. Συμπληρωματικά, εμφανίζεται μια μεγάλη λίστα εφαρμογών (<https://git-scm.com/downloads/guis>) διαχείρισης αποθετηρίων, που προσφέρουν ένα πιο φιλικό περιβάλλον χρήσης, με προτεινόμενα τα παρακάτω:

- GitHub Desktop (<https://desktop.github.com/>)
- Source Tree (<https://www.sourcetreeapp.com/>)



Εικόνα 3.16.

Τα βήματα της εγκατάστασης είναι αρκετά αλλά η εγκατάσταση ολοκληρώνεται γρήγορα με το πάτημα του Next> επιλέγοντας τις προκαθορισμένες επιλογές (defaults).

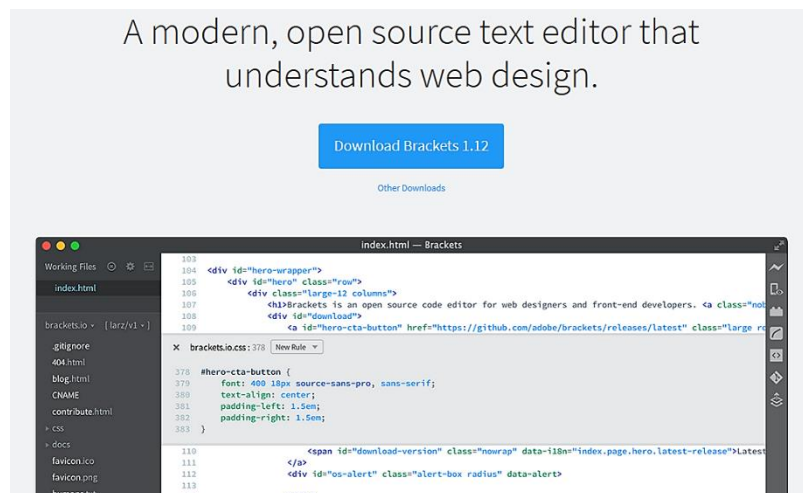


Εικόνα 3.17. Παραμετροποίηση Εγκατάστασης Git

Η αξιοποίηση των εργαλείων Git μέσω εντολών κονσόλας περιγράφονται σε επόμενη ενότητα (3.4.5). Επίσης, ένα εξαιρετικό περιβάλλον εκμάθησης με ιδιαίτερα χιουμοριστικό στυλ βρίσκεται στην διεύθυνση <https://try.github.io/levels/1/challenges/1>

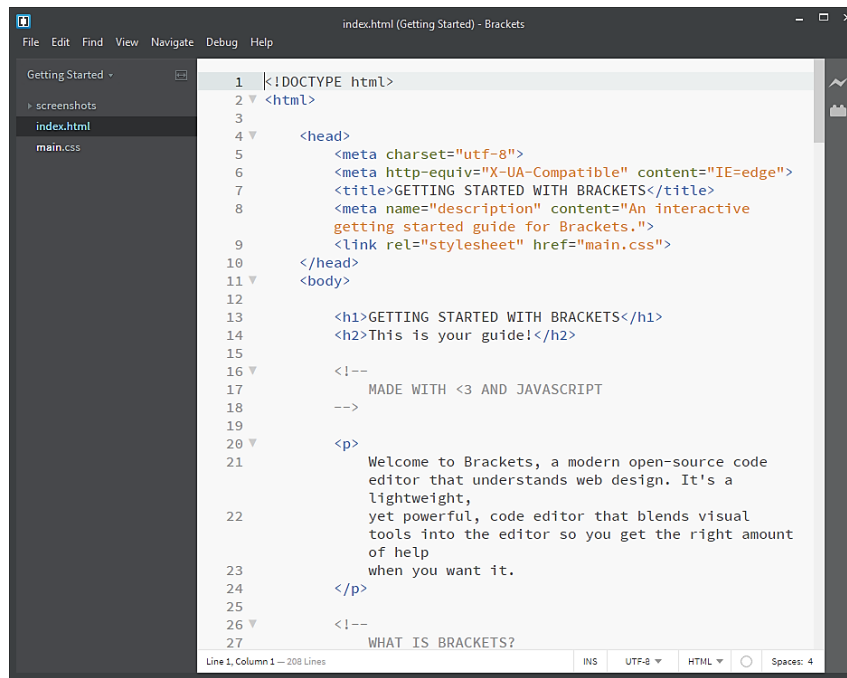
### 3.4.5 Brackets + Git plugin

Υπάρχει πληθώρα επεξεργαστών κώδικα στο διαδίκτυο οι οποίοι είναι εξαιρετικοί, με μηδενικό κόστος και προσφέρουν αυτοματισμούς που διευκολύνουν τη συγγραφή κώδικα. Ένας από αυτούς είναι ο Brackets τον οποίο μπορούμε να κατεβάσουμε από τη διεύθυνση <http://brackets.io/>. Η εγκατάσταση του είναι απλή και ολοκληρώνεται σε λίγα βήματα.



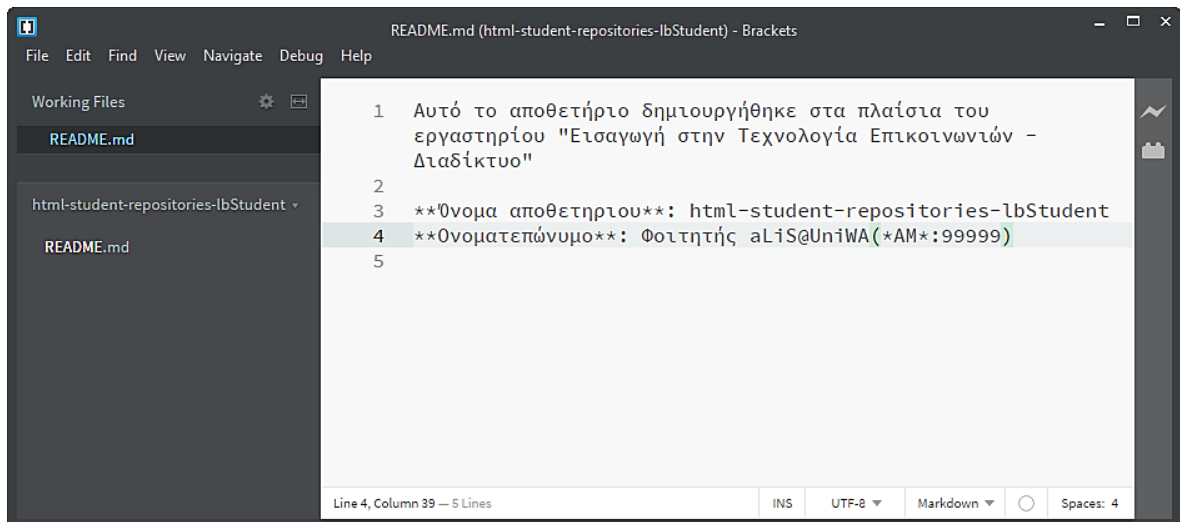
Εικόνα 3.18. Brackets

Ολοκληρώνοντας την εγκατάσταση εμφανίζεται το κεντρικό παράθυρο της εφαρμογής όπως παρακάτω:



Εικόνα 3.19. Κεντρικό παράθυρο Brackets

Ένα από τα πλεονεκτήματα χρήσης ενός επεξεργαστή κώδικα όπως ο Brackets είναι η **δυνατότητα επεξεργασίας ενός ολόκληρου καταλόγου** (project) σε αντίθεση με το σημειωματάριο στο οποίο μπορούμε να επεξεργαζόμαστε ένα αρχείο κάθε φορά. Ανοίγοντας τον φάκελο του project File→Open Folder... (Ctrl+Alt+O) μπορούμε να έχουμε άμεσα προσβάσιμα όλα τα αρχεία του καταλόγου. Στην περιοχή “Working Files” εμφανίζονται τα πιο πρόσφατα αρχεία, ενώ ακριβώς από κάτω εμφανίζεται όλη η δομή αρχείων του project .



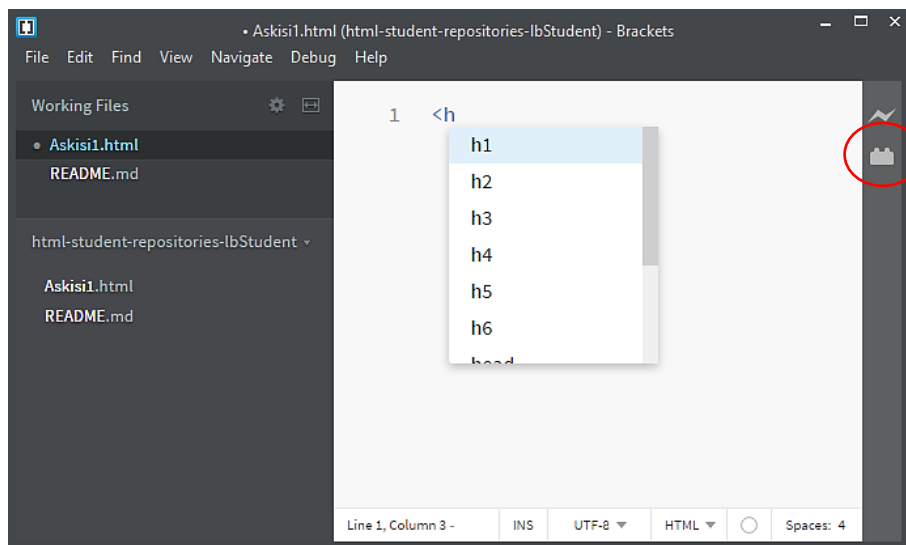
Εικόνα 3.20. Επεξεργασία φακέλου – project στο Brackets

Με δεξί κλικ στις περιοχές αυτές μπορούμε να ενεργοποιήσουμε μενού επεξεργασίας φακέλων ή αρχείων όπως παρακάτω, όπου μπορούμε να δημιουργήσουμε αρχεία/φακέλους και να τα διαχειριστούμε (μετονομασία/αποθήκευση/αναζήτηση-αντικατάσταση περιεχομένου/διαγραφή/κλείσιμο) όπως παρακάτω:



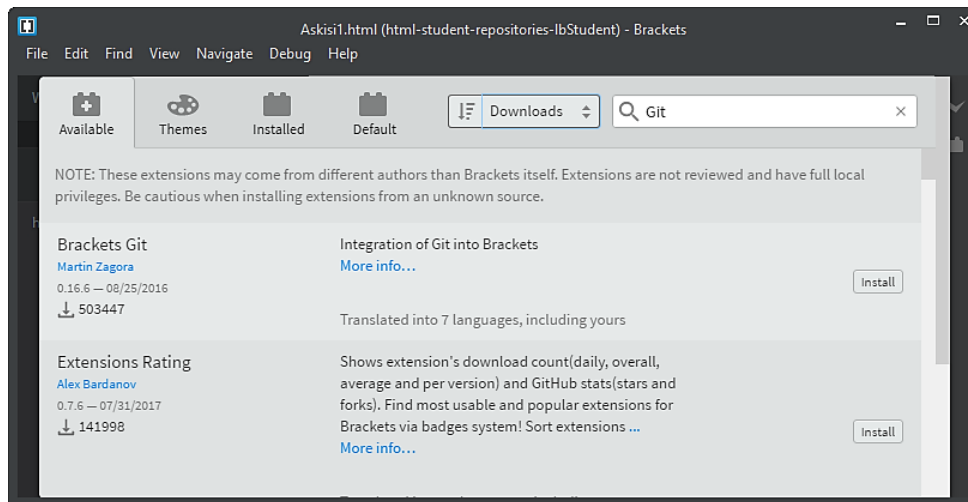
Εικόνα 3.21. Μενού επεξεργασίας αρχείων/φακέλων στο Brackets

Ένα ακόμα πλεονέκτημα που προσφέρει το Brackets είναι η **συντακτική υποβοήθηση** στη συγγραφή των αρχείων, ανάλογα με τον τύπο αρχείων που επεξεργαζόμαστε. Για να ενεργοποιηθεί αυτή η λειτουργία θα **πρέπει να αποθηκεύσουμε το αρχείο από την αρχή επιλέγοντας τον σωστό τύπο αρχείου**. Για παράδειγμα, στην επεξεργασία των αρχείων HTML θα πρέπει η επέκταση του αρχείου που επεξεργαζόμαστε να είναι `.html`. Έτσι, ενεργοποιείται η εισαγωγή στοιχείων HTML από ελεγχόμενη λίστα, η χρωματική απεικόνισή τους καθώς και το αυτόματο κλείσιμο των tags (Μενού Edit → Autoclose Braces).



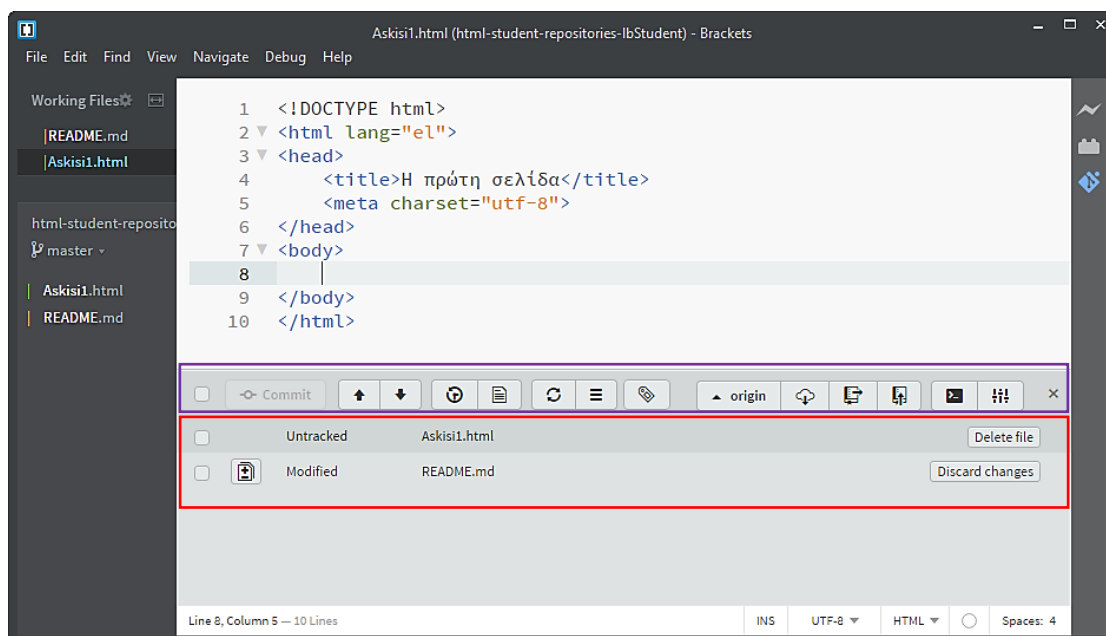
Εικόνα 3.22. Αυτοματισμοί συγγραφής HTML στο Brackets

Η διαχείριση αποθετηρίων Git μέσα το Brackets μπορεί να γίνει με τη βοήθεια πρόσθετων προγραμμάτων (plugins), τα οποία ενεργοποιούνται από τον **Διαχειριστή Πρόσθετων** (Extension Manager)



Εικόνα 3.23. Διαχείριση πρόσθετων στο Brackets

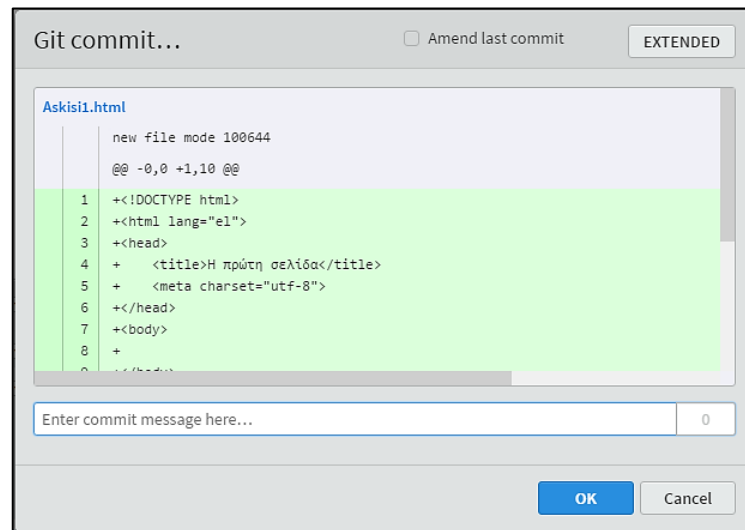
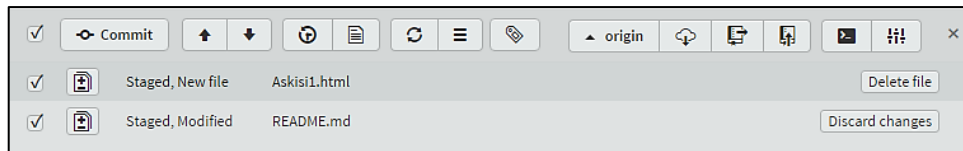
Επιλέγοντας το επιθυμητό πρόσθετο (Brackets Git) και ενεργοποιώντας το Install, ολοκληρώνεται η εγκατάστασή του και το πρόσθετο εμφανίζεται ακριβώς κάτω από τον “Extensions Manager”. Για την ενεργοποίηση του Git plugin αρκεί να το επιλέξουμε από τη λίστα, όπου θα εμφανιστεί η περιοχή διαχείρισής του στο κάτω μέρος της οθόνης.



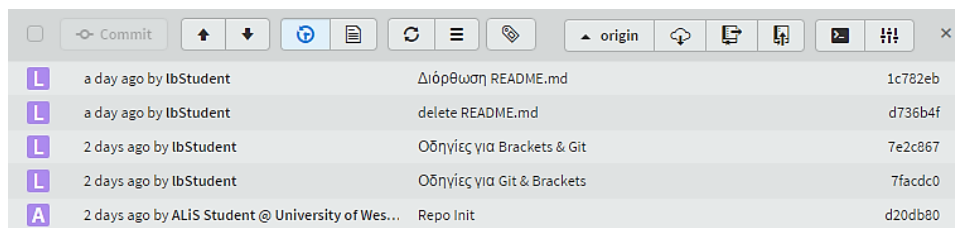
Εικόνα 3.24. Διαχείριση Αποθετηρίου μέσα από το Brackets

- Στο κάτω μέρος της περιοχής Git (κόκκινο πλαίσιο) εμφανίζονται τα αρχεία που έχουν προστεθεί στο αποθετήριο αλλά δεν παρακολουθούνται (Untracked), καθώς και αυτά των οποίων το περιεχόμενο έχει αλλάξει. Επιλέγοντας τα αρχεία έχουμε την επιλογή διαγραφής, αναίρεσης αλλαγών (Discard changes) ή υποβολής στο αποθετήριο (commit).

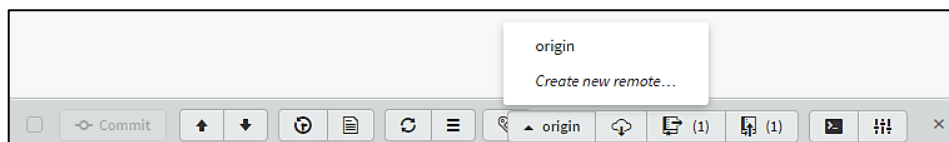






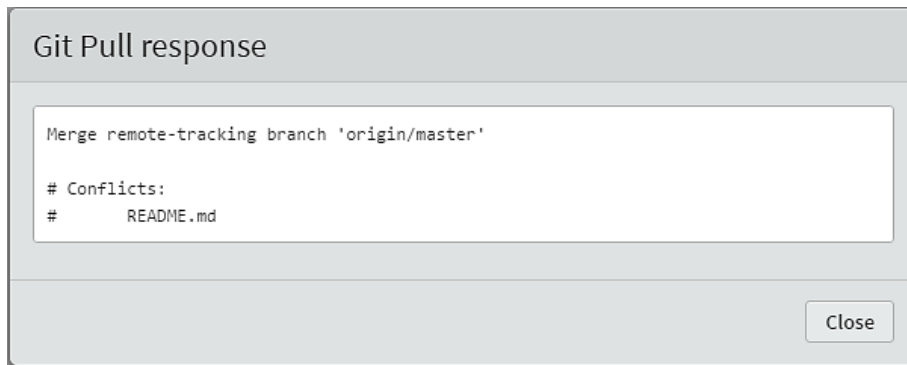
- **Επισκόπηση ιστορικού του αποθετηρίου**  και **επιλεγμένων αρχείων** 



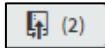
- **Δημιουργία σύνδεσης με απομακρυσμένο αποθετήριο (remote/origin)** 



- **Λήψη όλων των ενημερώσεων (Fetch all remotes)** και ενημέρωση των δεικτών (counters)  - ως αποτέλεσμα έχει την ενημέρωση των επόμενων στοιχείων που αφορούν τις αλλαγές που πρέπει να συγχρονιστούν (τόσο στο τοπικό όσο και στο απομακρυσμένο).
- **Λήψη των αλλαγών του απομακρυσμένου αποθετηρίου (Git pull)** και ενημέρωση της τοπικής έκδοσης  (1) - το πλήθος αλλαγών φαίνεται στην παρένθεση. Στην περίπτωση που στο απομακρυσμένο αποθετήριο υπάρχουν αλλαγές που έρχονται σε σύγκρουση με τις εκδόσεις των αρχείων που βρίσκονται αποθηκευμένες τοπικά, θα πρέπει διαχειριστούμε το τελικό περιεχόμενο του αρχείου επιλέγοντας ποια μέρη από τα δύο αρχεία θα παραμείνουν.



- **Αποστολή των αλλαγών** του τοπικού αποθετηρίου στο απομακρυσμένο (remote/origin)



- το πλήθος αλλαγών φαίνεται στην παρένθεση.