



# Coding Factory

## Εισαγωγή στην HTML

**Αθανάσιος Ανδρούτσος**



# World Wide Web

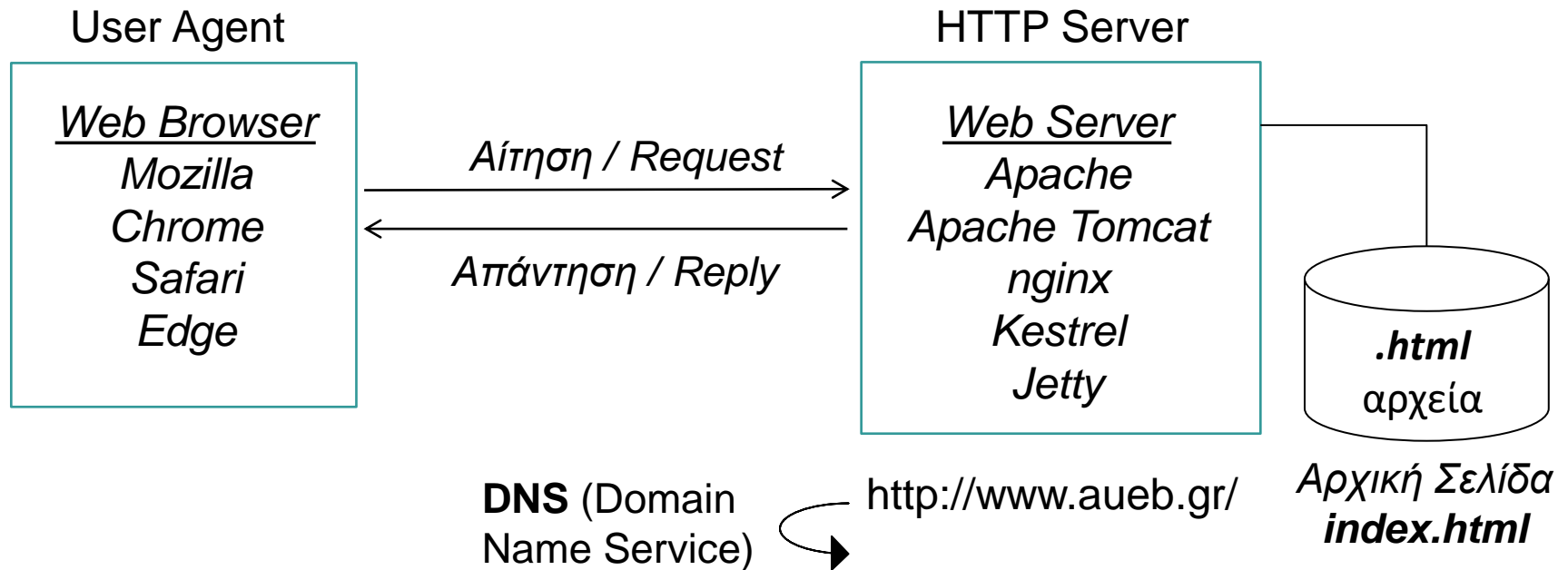
Προγραμματισμός στο Web

- Ο Sir Timothy John Berners-Lee (TimBL) το 1989, όταν εργαζόταν στο CERN, πρότεινε ένα νέο τρόπο διαμοιρασμού εγγράφων βασισμένο στο World Wide Web (WWW)
- Υλοποίησε την πρώτη επιτυχημένη επικοινωνία client-server μέσω του πρωτοκόλλου HTTP (HyperText Transfer Protocol)



# World Wide Web (WWW) (1)

Προγραμματισμός στο Web

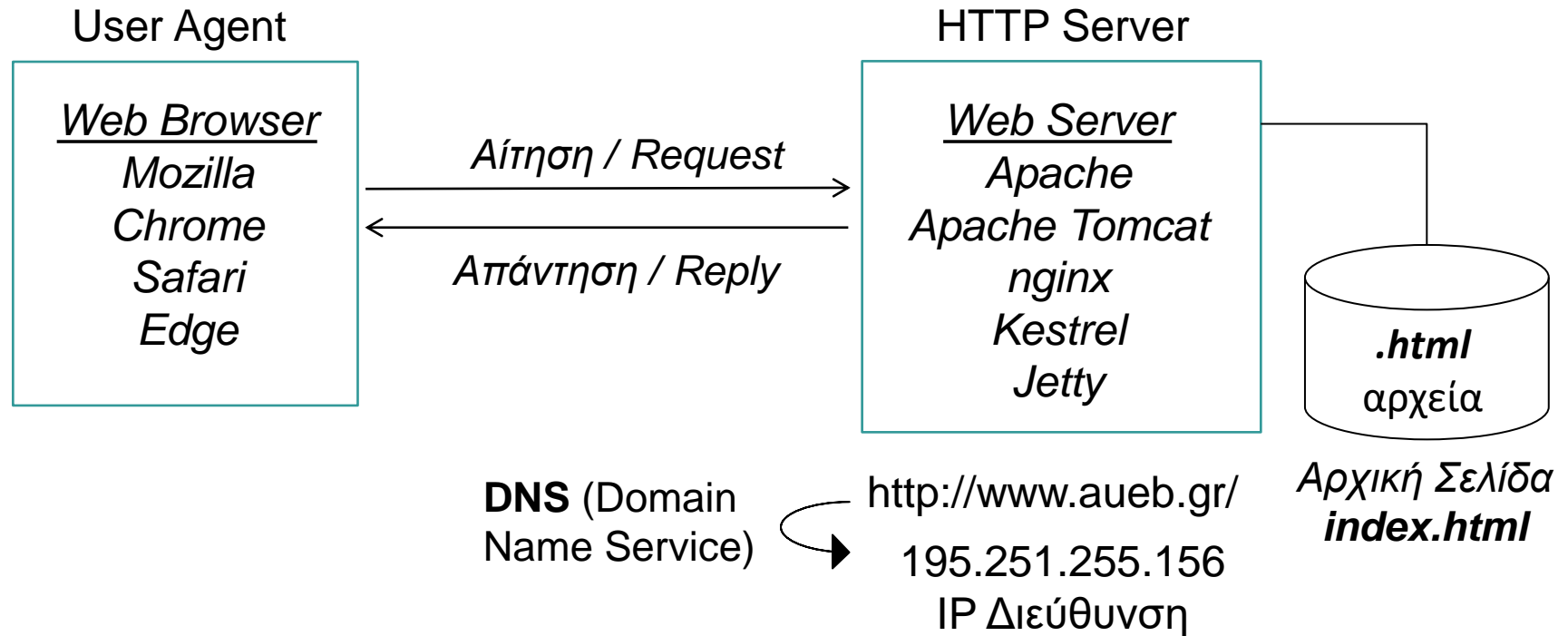


- Στο βασικό μοντέλο επικοινωνίας, ο client ή αλλιώς user agent (π.χ. ένας web browser) επικοινωνεί μέσω ενός URL (Uniform Resource Location, π.χ. [www.aueb.gr](http://www.aueb.gr)) και αιτείται μίας υπηρεσίας από ένα Εξυπηρετητή που εδώ είναι ένας Web Server (που είναι βασικά ένας HTTP Server) και ο Web Server επιστρέφει ένα αρχείο με κατάληξη .html που ονομάζεται web σελίδα



# World Wide Web (WWW) (2)

Προγραμματισμός στο Web



- Επειδή οι άνθρωποι καταλαβαίνουν λογικές διευθύνσεις (DNS διευθύνσεις) ενώ οι υπολογιστές ταυτοποιούνται μέσω IP διευθύνσεων, η υπηρεσία DNS (Domain Name Service) κάνει τη μετατροπή από DNS σε IP διεύθυνση



- Ο Tim Berners-Lee είναι ο Διευθυντής του **W3C (World Wide Web Consortium)** ενός οργανισμού προτυποποίησης του WWW που ιδρύθηκε το 1994 που εκδίδει οδηγίες, εργαλεία, πρότυπα, κλπ. για το Web
- <https://www.w3.org/People/Berners-Lee/>



# Ψηφιακή Εποχή

Προγραμματισμός στο Web

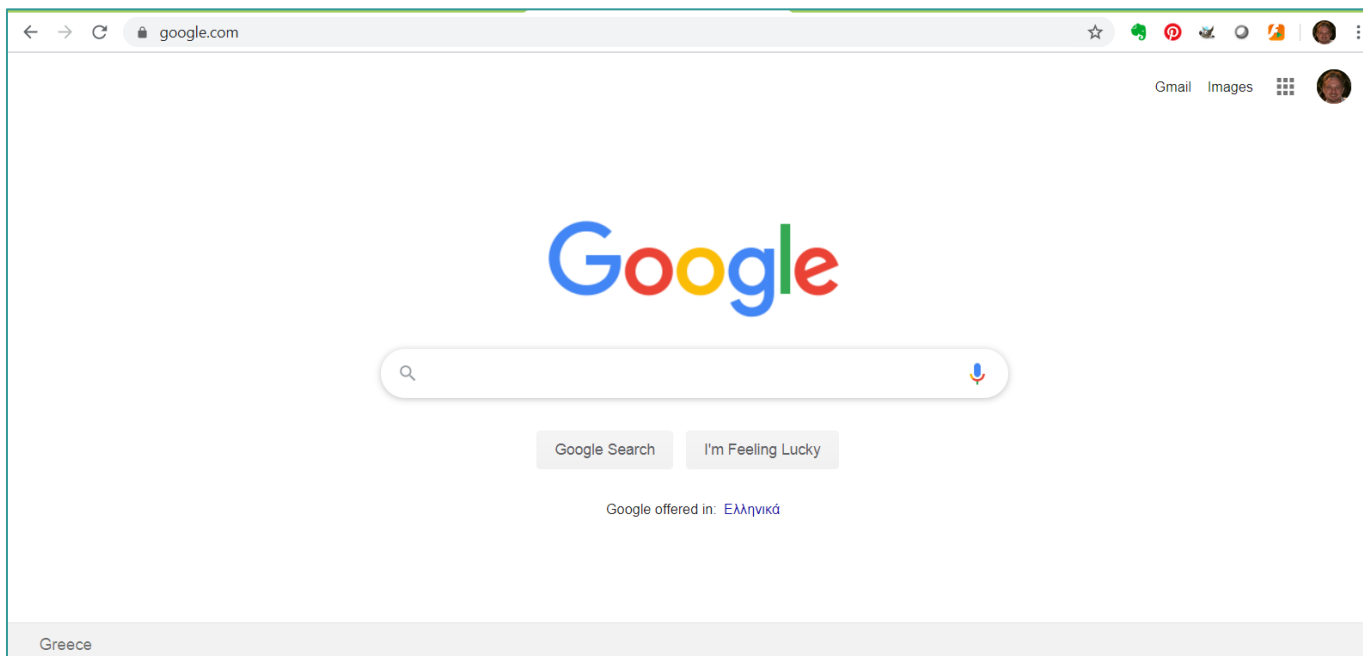
- Το **World Wide Web** αποτελεί το μηχανισμό για την απομακρυσμένη επικοινωνία εκατομμυρίων ανθρώπων, την πώληση προϊόντων, την εξ αποστάσεων παροχή υπηρεσιών
- Αποτελεί τον θεμέλιο λίθο της ψηφιακής οικονομίας (Digital Economy) και της εποχής της Πληροφορίας (Information Age)



# Web Σελίδες

Προγραμματισμός στο Web

- Οι Web Σελίδες είναι αρχεία κειμένου δομημένα και μορφοποιημένα με ειδικές σημάνσεις της γλώσσας HTML





- Δύο κατηγορίες
  - **Στατικές** σελίδες, είναι οι σελίδες που το περιεχόμενό τους δεν μεταβάλλεται, αλλά παραμένει σταθερό
  - **Δυναμικές** σελίδες, είναι οι σελίδες που το περιεχόμενό τους μπορεί να αλλάζει ανάλογα με το τι ζητάει ο χρήστης, π.χ. καλάθι αγορών. Οι δυναμικές σελίδες επικοινωνούν με πηγές δεδομένων (π.χ. Βάσεις Δεδομένων).





# Τεχνολογίες Web

Προγραμματισμός στο Web

- **HTML** (HyperText Markup Language). Προσδίδει σημασιολογία και δομή στο περιεχόμενο (κείμενο, εικόνες, ήχο, βίντεο, σύνδεσμοι, κλπ.)
- **CSS** (Cascading Style Sheets). Styling του περιεχομένου, π.χ. χρώματα, μέγεθος γραμματοσειράς, κλπ.
- **JavaScript**. Γλώσσα προγραμματισμού για αλληλεπίδραση με τον χρήστη και επικοινωνία με πηγές δεδομένων στο δίκτυο



# HTML

- Η γλώσσα **HTML (HyperText Markup Language)** χρησιμοποιείται για τον ορισμό της δομής Web σελίδων
- Η ετυμολογία της λέξης HTML περιλαμβάνει τις λέξεις:
  - **HyperText** που αναφέρεται στο βασικό χαρακτηριστικό των Web Σελίδων που είναι οι **υπέρ-σύνδεσμοι** (hyperlinks ή απλά links) προς άλλες σελίδες
  - **Markup** που αναφέρεται στην σήμανση (markup) με τη χρήση **tags** του περιεχομένου μιας Web σελίδας ώστε το περιεχόμενο να αποκτήσει νόημα (semantics). Για παράδειγμα η χρήση του `<p>` element `<p> Hello World </p>` σημαίνει ότι το Hello World είναι παράγραφος



# Έγγραφα HTML

Προγραμματισμός στο Web

- Μία Web σελίδα είναι ένα αρχείο HTML που περιέχει ένα σύνολο από στοιχεία του User Interface (UI elements)
- Τα HTML elements μπορούν να υλοποιήσουν διάφορα UI elements όπως επικεφαλίδες, παραγράφους, υπέρ-συνδέσμους, εικόνες, λίστες, πίνακες, κλπ. καθώς επίσης και να δομήσουν ένα HTML έγγραφο σε λογικά μέρη (header, menu, content, footer, κλπ.)



# Web Browsers

Προγραμματισμός στο Web

- Πρόκειται για εφαρμογές που χρησιμοποιούνται για να διαβάζουν και να εμφανίζουν HTML σελίδες
- Οι browsers διαβάζουν τα HTML tags και μπορούν και 'καταλαβαίνουν' τη σημασία τους, και επομένως εμφανίζουν (render / paint) το αντίστοιχο UI Element
- Υπάρχουν διάφοροι Web Browsers, όπως ο Google Chrome, Microsoft Edge, Mozilla Firefox, Apple Safari, Opera, κλπ.



# HTML Versions (Εκδόσεις)

Προγραμματισμός στο Web

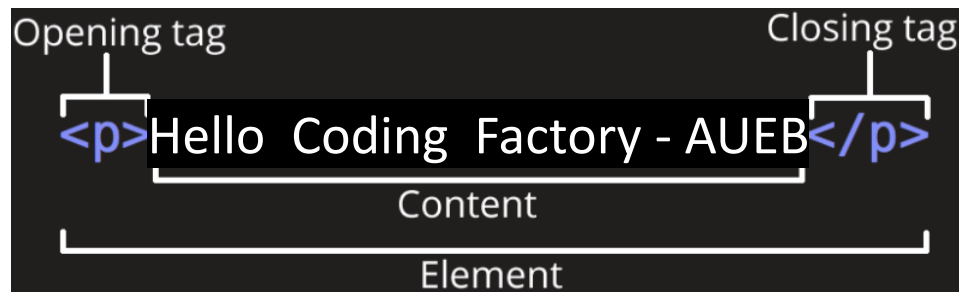
Έκδοση	Έτος
HTML	1991
HTML 2.0	1995
HTML 3.2	1997
HTML 4.01	1999
XHTML	2000
HTML5	2014

- Η **HTML5** είναι σήμερα το βασικό πρότυπο συγγραφής Web σελίδων, ένα living standard!
- Η XHTML (eXtensible HTML) είναι μία πιο αυστηρή συντακτικά έκδοση της HTML



# HTML Elements (1)

- Το συντακτικό της γλώσσας HTML αποτελείται από στοιχεία HTML (**HTML elements**). Τα HTML Elements αποτελούνται από Tags και μπορεί να περιλαμβάνουν και Attributes
- Τα Tags είναι ζεύγη ετικετών (tags): Opening/Closing tag. Π.χ. **<p>Hello Coding Factory - AUEB</p>**



- Υπάρχουν και **empty** ή **void elements** δηλαδή tags χωρίς closing tag. Π.χ. **<br>** που σημαίνει *break* και χρησιμοποιούμε για αλλαγή γραμμής



# HTML Elements (2)

- Opening Tag. Είναι το όνομα του element μέσα σε angle brackets `<>`, π.χ. `<p>`
- Closing Tag. Όπως το opening tag αλλά περικλείει και ένα forward slash / πριν το όνομα του element, π.χ. `</p>`
- Content. Το περιεχόμενο του element, π.χ. κείμενο
- HTML Element. Opening tag, closing tag και περιεχόμενο



# HTML Elements (3)

- Τα HTML elements μπορεί να έχουν και **ιδιότητες (attributes)** που είναι έξτρα ιδιότητες του στοιχείου που το συγκεκριμενοποιούν περαιτέρω το HTML Element

Diagram illustrating an HTML element with an attribute:

```
graph TD
    Attribute[Attribute] --- Element["<p class='coding-factory-text'>Hello Coding Factory - AUEB</p>"]
```

The diagram shows the HTML element `<p class="coding-factory-text">Hello Coding Factory - AUEB</p>`. The attribute `class="coding-factory-text"` is highlighted, and a line connects it to the label "Attribute" above it.

- Για παράδειγμα, η ιδιότητα `class` προσδίδει στο element ένα αναγνωριστικό ώστε να μπορεί να εντοπισθεί και μορφοποιηθεί με styling
- Οι ιδιότητες ακολουθούνται από το `=` και την τιμή της ιδιότητας μέσα σε διπλά εισαγωγικά (quotation marks)





# Nested Elements

- Μπορούμε να έχουμε elements μέσα σε άλλα elements (nesting), π.χ.
- `<p>Hello <strong>coding factory</strong></p>`
- Μπορούμε να κάνουμε wrap (περιτυλίξουμε) τις λέξεις coding factory μέσα σε ένα `<strong>` element ώστε να δώσουμε strong importance στο κείμενο (όχι μόνο bold, αλλά και σημασιολογία, για παράδειγμα οι screen readers που απευθύνονται σε άτομα με προβλήματα όρασης θα 'διαβάσουν' το strong με έμφαση). Το απλό `<b>` που εμφανίζει bold έχει μόνο styling ερμηνεία όχι σημασιολογία (όπως αναφέραμε η σημασιολογία μπορεί να γίνει κατανοητή από το Google ή screen readers)



# Empty Elements

- Μερικά HTML elements μπορεί να μην περιέχουν περιεχόμενο, επομένως ούτε closing tag. π.χ.

```

```

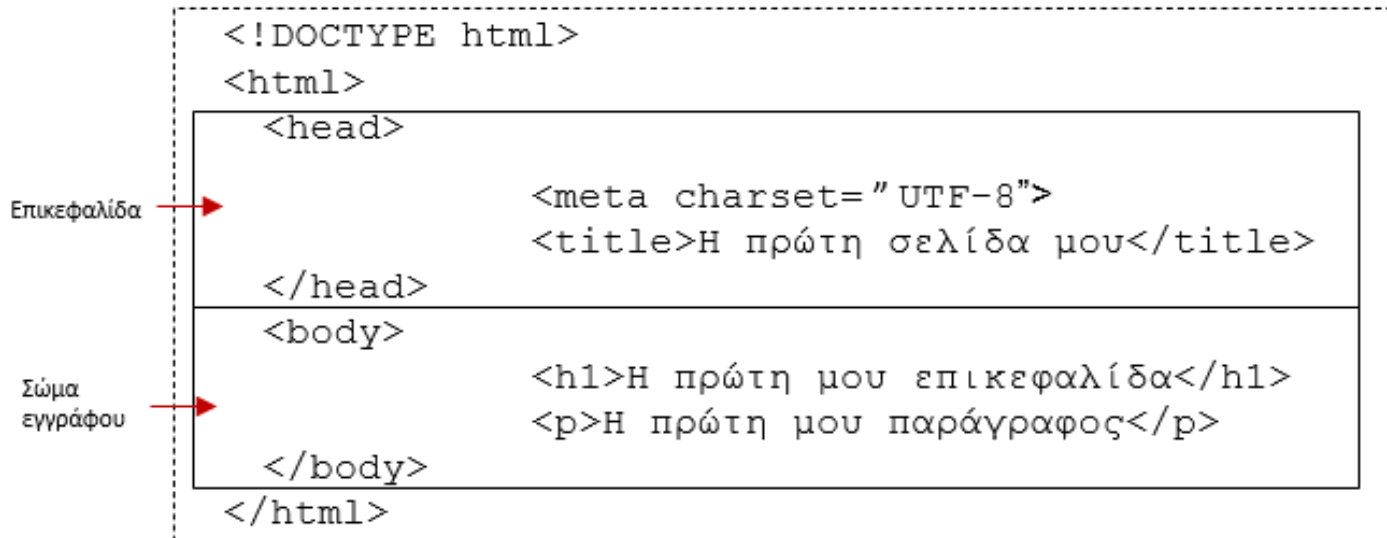
Για παράδειγμα, το `img` element που αναπαριστά εικόνες δεν έχει εσωτερικό περιεχόμενο. Σκοπός του είναι μόνο να κάνει render (εμφανίσει στον browser) ένα image, στο σημείο που βρίσκεται το `<img>`



# Δομή Εγγράφου HTML (1)

Προγραμματισμός στο Web

- Κάθε έγγραφο HTML5 ξεκινά με μία δήλωση **<!DOCTYPE html>** και στη συνέχεια αποτελείται από δύο μέρη: Επικεφαλίδα (head) και Σώμα (body) μέσα στο <html> (root element (entire page))

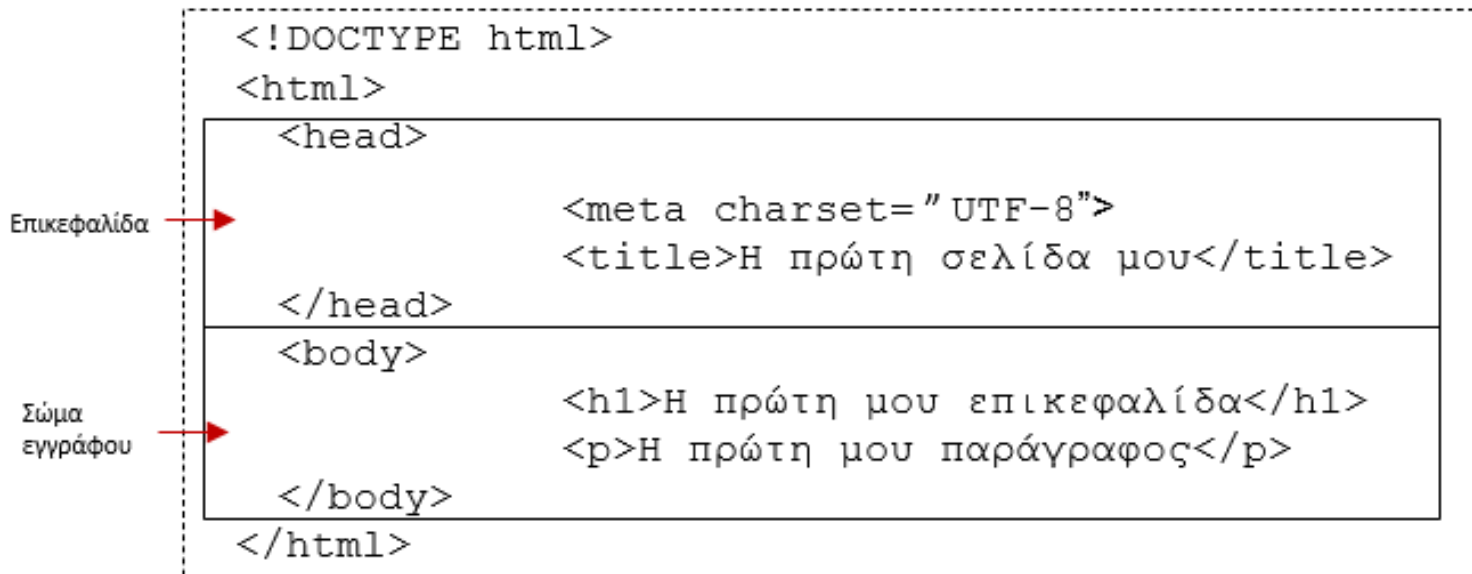




# Δομή Εγγράφου HTML (2)

Προγραμματισμός στο Web

- Το `<head>` περιέχει μεταδεδομένα και άλλες πληροφορίες που δεν εμφανίζονται στη σελίδα, ενώ το `<body>` περιέχει περιεχόμενο (content) που εμφανίζεται στη σελίδα.

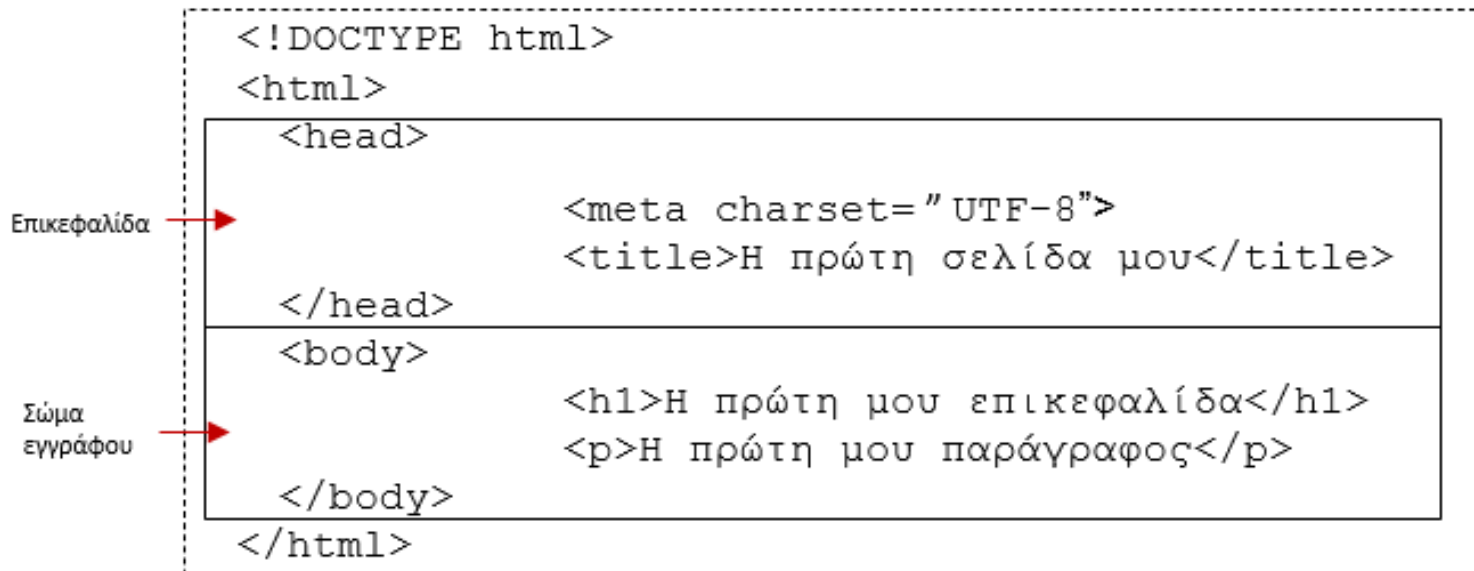




# Δομή Εγγράφου HTML (3)

Προγραμματισμός στο Web

- Το charset ορίζει ως κωδικοποίηση χαρακτήρων της σελίδας UTF-8 (το UTF-8 κωδικοποιεί και περιλαμβάνει χαρακτήρες και σύμβολα από όλες τις γλώσσες του κόσμου) και επομένως μπορούμε να έχουμε ως περιεχόμενο μέσα στο έγγραφό μας οποιονδήποτε χαρακτήρα οποιασδήποτε γλώσσας, π.χ. ελληνικά.

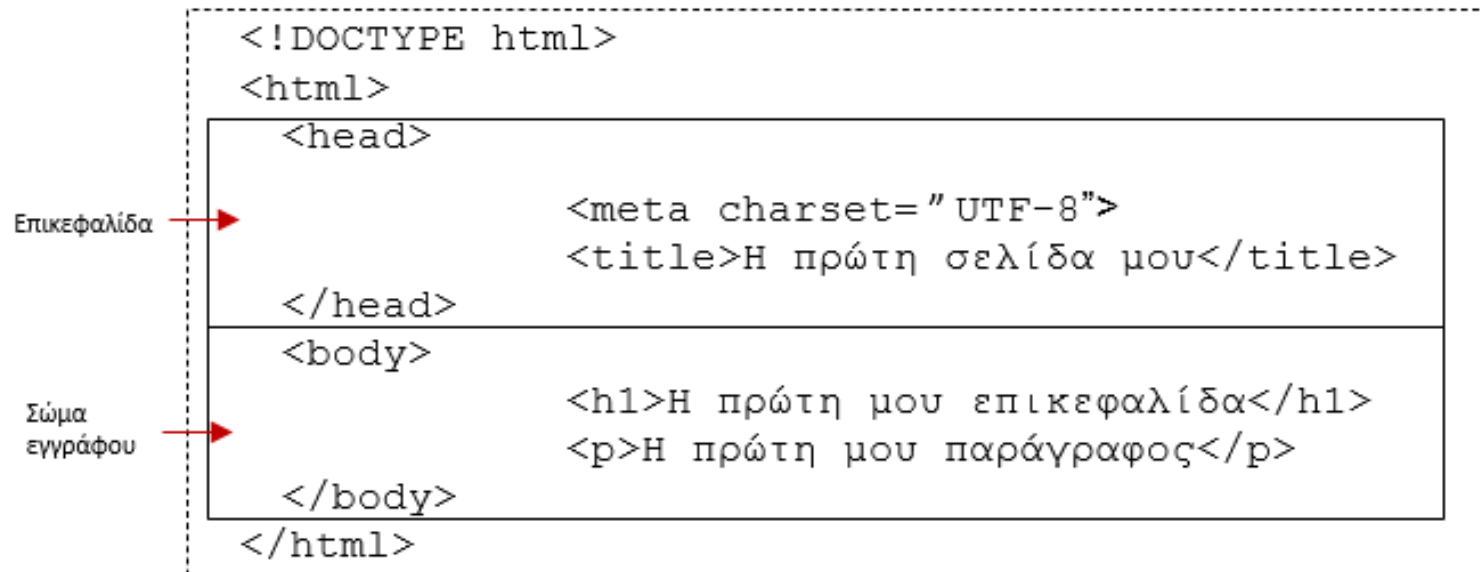




# Δομή Εγγράφου HTML (4)

Προγραμματισμός στο Web

- Ο τίτλος `<title>` δεν είναι περιεχόμενο της σελίδας. Εμφανίζεται στο tab του browser.
- Τα `<h1>` και `<p>` μέσα στο `<body>` είναι περιεχόμενο: Επικεφαλίδα και παράγραφος





# Περιεχόμενο Web Σελίδων

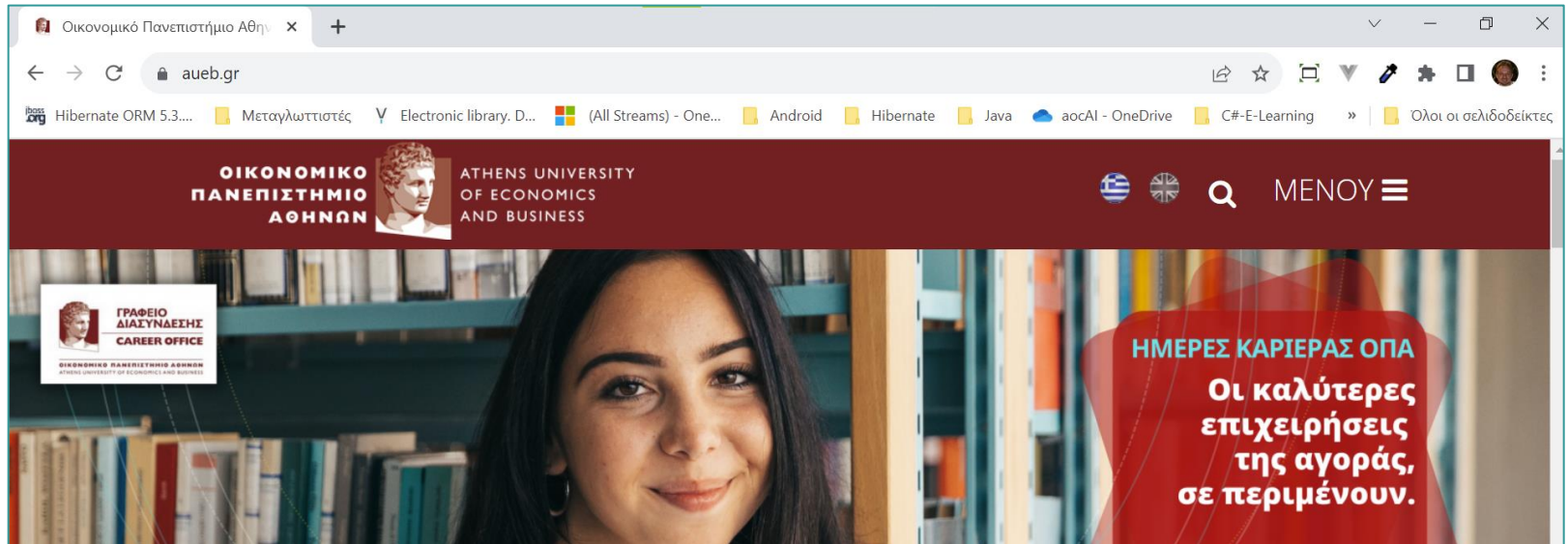
Προγραμματισμός στο Web

- Γενικά, το περιεχόμενο μιας Web Σελίδας, μπορεί να είναι πολυμεσικό: κείμενο, εικόνες, βίντεο, ήχος και το πιο σημαντικό υπερσύνδεσμοι
- Οι υπερσύνδεσμοι είναι είτε κείμενο ή εικόνα που είναι clickable. Μπορεί ο χρήστης να κάνει click και να οδηγηθεί σε μία άλλη σελίδα



# Στοιχεία Σελίδας

## Προγραμματισμός στο Web



- Στην παραπάνω σελίδα κάποια στοιχεία που διακρίνουμε είναι το **title** (Οικονομικό Πανεπιστήμιο Αθηνών) στο πάνω αριστερά μέρος του tab ενώ βλέπουμε και άλλα UI Elements όπως **Logo**, **Εικονίδια γλώσσας**, **Αναζήτησης**, **Μενού** (τα οποία είναι οργανωμένα σε ένα οριζόντιο container που ονομάζεται Header)
- Επίσης κάτω από τον Header υπάρχει ένας άλλος λογικός container που περιέχει μία εικόνα background, μία εικόνα 'ΓΡΑΦΕΙΟ ΔΙΑΣΥΝΔΕΣΗΣ' και μία ακόμα εικόνα 'ΗΜΕΡΕΣ ΚΑΡΙΕΡΑΣ ΟΠΑ'





# Προαπαιτούμενα Εργαλεία

Προγραμματισμός στο Web

- **Download και Εγκατάσταση εργαλείων**
  - IDE. **Visual Studio Code** (VS Code), Atom, Sublime, κλπ.
  - Web browser. **Chrome**, Firefox, Safari, κλπ.
  - Graphics editor. **GIMP**, **Figma**, Photoshop, Adobe XD
  - Version Control System. **Git/GitHub**
  - Βιβλιοθήκες. **Bootstrap**, **jQuery**, **AJAX**, **Vue**
  - Web Server. **Apache Tomcat**



# Συμβάσεις ονοματοδοσίας (1)

Προγραμματισμός στο Web

- Ως **σύμβαση** για την ονοματοδοσία φακέλων και αρχείων θα χρησιμοποιούμε μικρά γράμματα και ως word separator θα χρησιμοποιούμε την παύλα (hyphen ή dash) που η Google θεωρεί ως διαχωριστικό λέξεων αντί της κάτω παύλας
- Επομένως θα χρησιμοποιούμε για φακέλους και αρχεία Kebab-case: **Φάκελοι και αρχεία με μικρά γράμματα και dash ως διαχωριστικό**



# Συμβάσεις ονοματοδοσίας (2)

Προγραμματισμός στο Web

- Ο λόγος είναι πως πολλά συστήματα web servers (π.χ. Linux) είναι case sensitive και διαχωρίζουν κεφαλαία και πεζά (αν ένα αρχείο το ονομάσουμε Image.png και μετά το καλέσουμε ως image.png, μπορεί να μη δουλεύει εκτός εάν ο Web Server έχει γίνει config ως case-insensitive) ενώ το κενό το αντικαθιστούν με %20 καταλήγοντας σε links που δεν δουλεύουν (broken)



# Visual Studio Code (1)

Προγραμματισμός στο Web

- Για τη συγγραφή HTML κώδικα θα χρησιμοποιήσουμε το Visual Studio Code (VS Code) της Microsoft που είναι δωρεάν προς χρήση
- <https://code.visualstudio.com/>
- Δείτε στις επόμενες διαφάνειες τις οδηγίες εγκατάστασης



# Visual Studio Code (2)

Προγραμματισμός στο Web

- <https://code.visualstudio.com/>

The screenshot shows the Visual Studio Code website with the headline "Code editing. Redefined." and a download table. Below the table is a mockup of the Visual Studio Code IDE interface, showing the Extensions Marketplace, a code editor with JavaScript code, and a terminal window.

**Download for Windows**  
Stable Build

		Stable	Insiders
<b>macOS</b>	Universal	↓	↓
<b>Windows x64</b>	User Installer	↓	↓
<b>Linux x64</b>	.deb .rpm	↓ ↓	↓ ↓

**EXTENSIONS: MARKETPLACE**

- Python** 2019.6.24221 54.9M ★ 4.5  
Linting, Debugging (multi-threaded, ...  
Microsoft **Install**
- GitLens — Git sup...** 9.8.5 23.1M ★ 5  
Supercharge the Git capabilities built...  
Eric Amodio **Install**
- C/C++** 0.24.0 23M ★ 3.5  
C/C++ IntelliSense, debugging, and ...  
Microsoft **Install**
- ESLint** 1.9.0 21.9M ★ 4.5  
Integrates ESLint JavaScript into VS ...  
Dirk Baumer **Install**
- Debugger for Ch...** 4.11.6 20.6M ★ 4  
Debug your JavaScript code in the C...  
Microsoft **Install**
- Language Supp...** 0.47.0 18.7M ★ 4.5  
Java Linting, Intellisense, formatting, ...  
Red Hat **Install**
- vscode-icons** 8.8.0 17.2M ★ 5  
Icons for Visual Studio Code  
VSCode Icons Team **Install**
- Vetur** 0.21.1 17M ★ 4.5

**JS serviceWorker.js**

```
src > JS serviceWorker.js > register > window.addEventListener('load') callback
39
40
41
42 // Add some additional logging to localhost, p
43 // service worker/PWA documentation.
44
45
46 product
47 productSub
48 removeSiteSpecificTrackingException
49 removeWebWideTrackingException
50 requestMediaKeySystemAccess
51 sendBeacon
52 serviceWorker (property) Navigator.serviceWorke...
53 storage
54 storeSiteSpecificTrackingException
55 storeWebWideTrackingException
56
57 }
58 userAgent
59 vendor
60
61
62 function registerValidSW(swUrl, config) {
63   navigator.serviceWorker
64     .register(swUrl)
65     .then(registration => {
```

**TERMINAL** 1: node



# Download

code.visualstudio.com/Download

Visual Studio Code Docs Updates Blog API Extensions FAQ Learn Search Docs Download

Free and built on open source. Integrated Git, debugging and extensions.

**Windows**  
Windows 10, 11

User Installer x64 x86 Arm64  
System Installer x64 x86 Arm64  
.zip x64 x86 Arm64  
CLI x64 x86 Arm64

**.deb**  
Debian, Ubuntu

**.rpm**  
Red Hat, Fedora, SUSE

.deb x64 Arm32 Arm64  
.rpm x64 Arm32 Arm64  
.tar.gz x64 Arm32 Arm64  
Snap Snap Store  
CLI x64 Arm32 Arm64

**Mac**  
macOS 10.15+

.zip Intel chip Apple silicon Universal  
CLI Intel chip Apple silicon

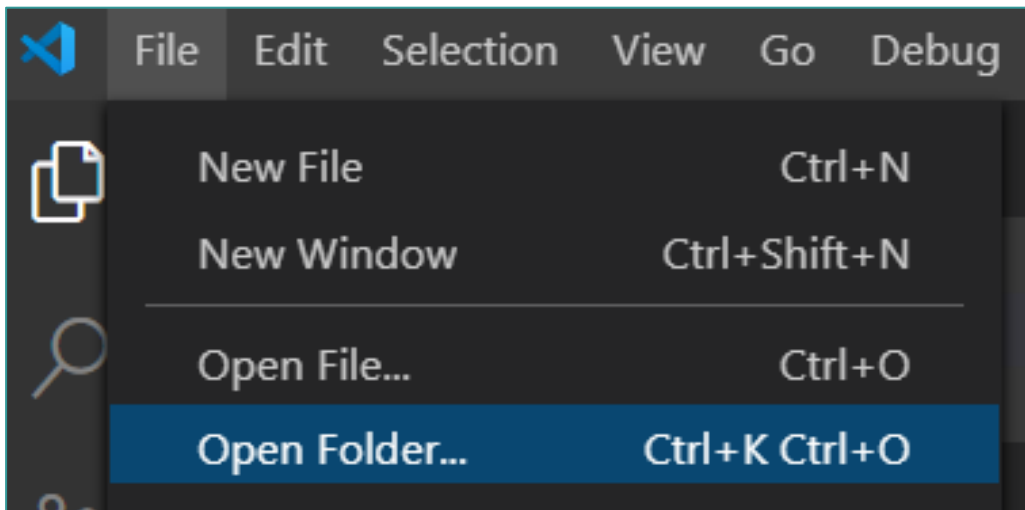
- <https://code.visualstudio.com/download>
- Εναλλακτικά από το κουμπί του Download (πάνω δεξιά) και εγκαθιστούμε την έκδοση που αντιστοιχεί στο Λειτουργικό Σύστημα που έχουμε



# Δημιουργία Αρχικού Φακέλου

Προγραμματισμός στο Web

- Δημιουργούμε ένα φάκελο στα Windows / Linux / Mac, μέσα στον οποίο θα αποθηκεύσουμε όλα τα αρχεία του project μας και στη συνέχεια από το μενού File του VSC επιλέγουμε Open Folder

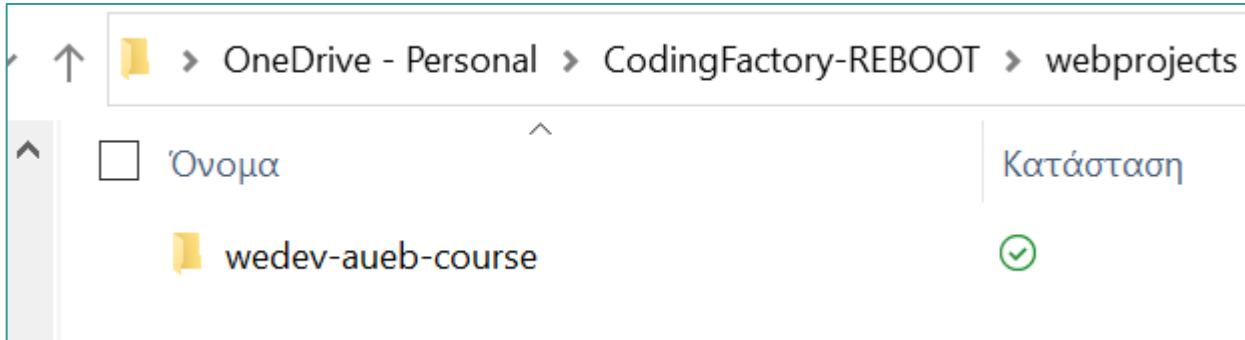


- Ή μπορούμε να επιλέξουμε File/Open Folder και να δημιουργήσουμε τον φάκελο και ταυτόχρονα να τον επιλέξουμε
- Μπορούμε επίσης -αν θέλουμε να έχουμε περισσότερους root folders- να αποθηκεύσουμε ως workspace



# Δημιουργία και Επιλογή Φακέλου (1)

Προγραμματισμός στο Web



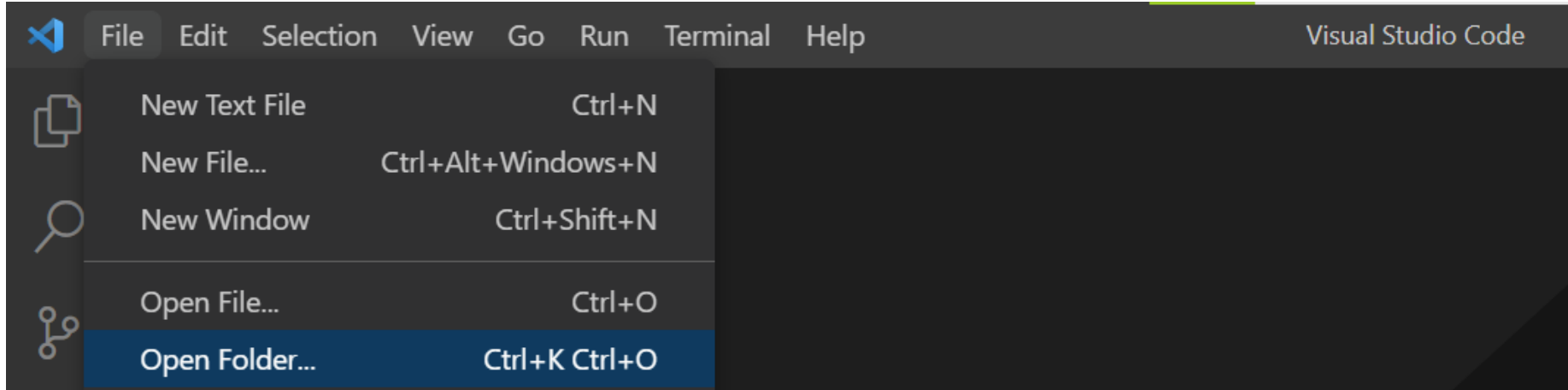
- Πρώτα δημιουργούμε νέο φάκελο (Δημιουργία / Νέος Φάκελος) έστω με όνομα webdev-aueb-course





# Δημιουργία και Επιλογή Φακέλου (2)

Προγραμματισμός στο Web

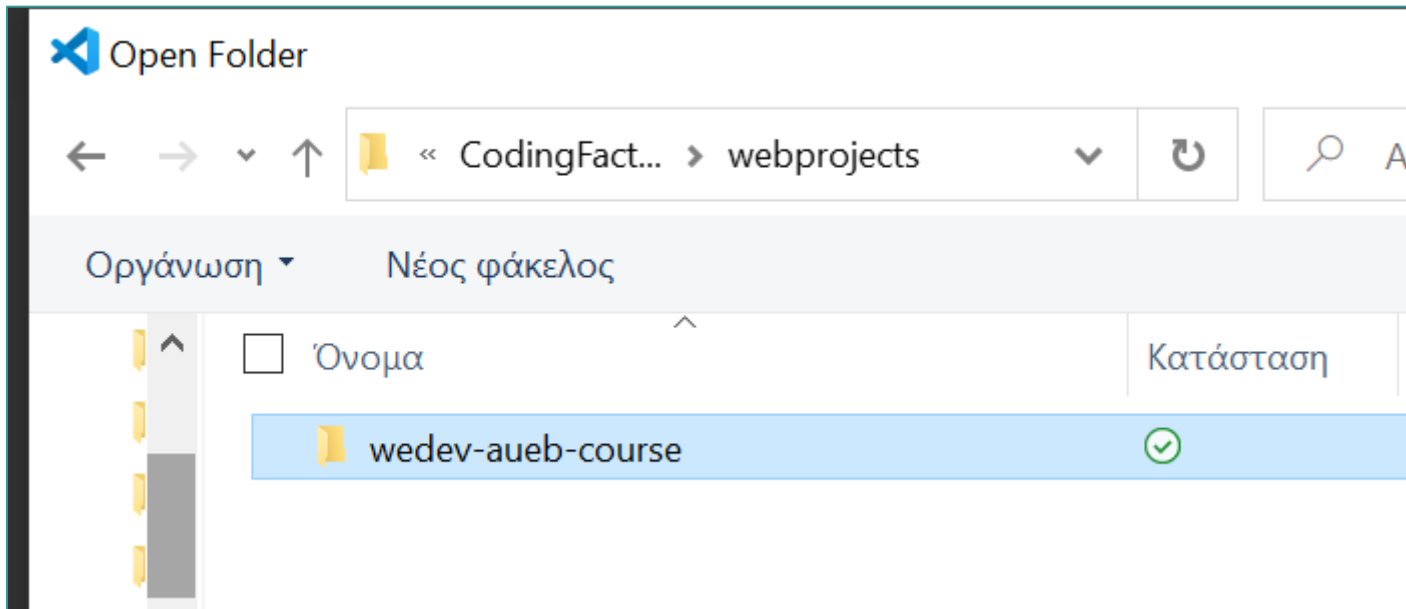


- Στη συνέχεια ανοίγουμε το VS Code και μετά μέσα από το VSC επιλέγουμε File/Open Folder και επιλέγουμε τον φάκελο που δημιουργήσαμε και πατάμε το κουμπί *Επιλογή Φακέλου*



# Δημιουργία και Επιλογή Φακέλου (3)

Προγραμματισμός στο Web

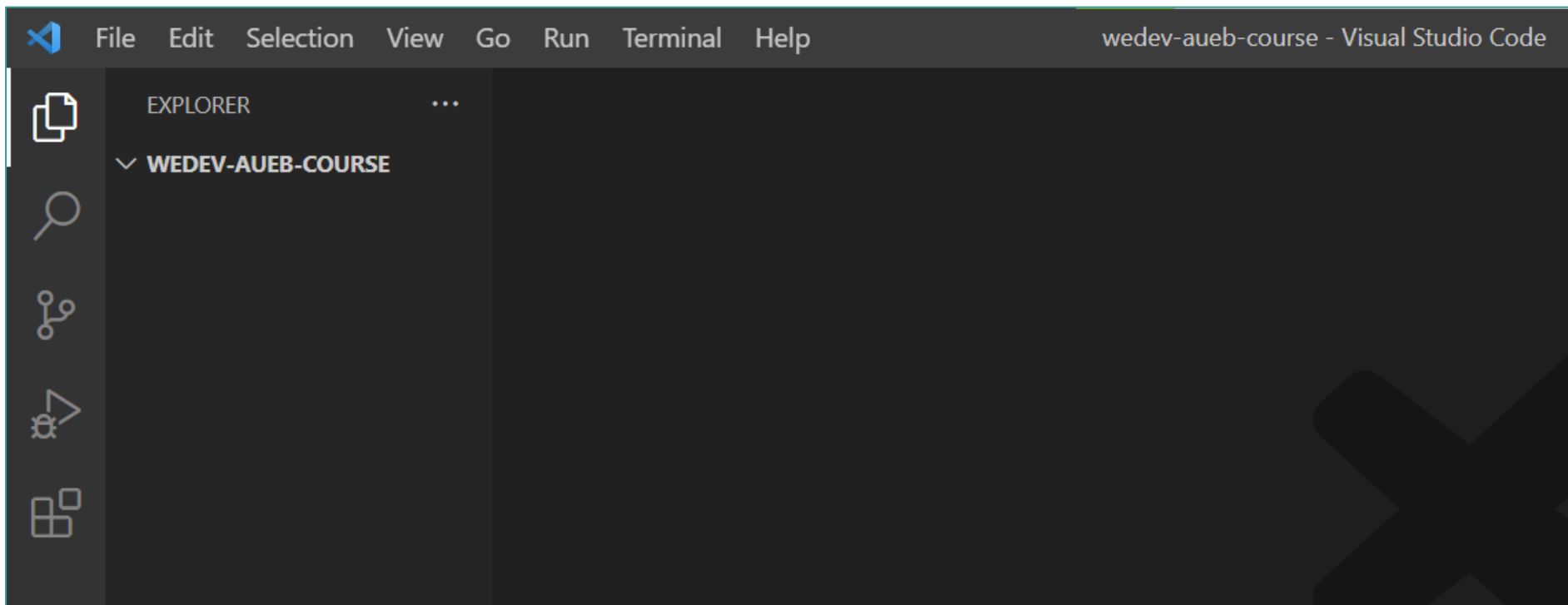


- Επιλέγουμε τον φάκελο που είχαμε δημιουργήσει. Αυτός ο φάκελος θα είναι ο φάκελος του Project μας



# VSC Project

Προγραμματισμός στο Web

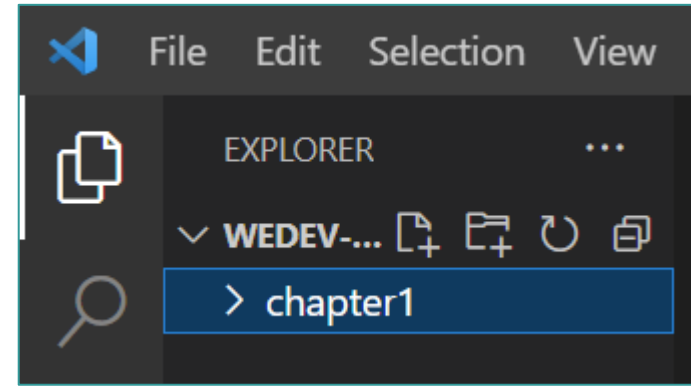
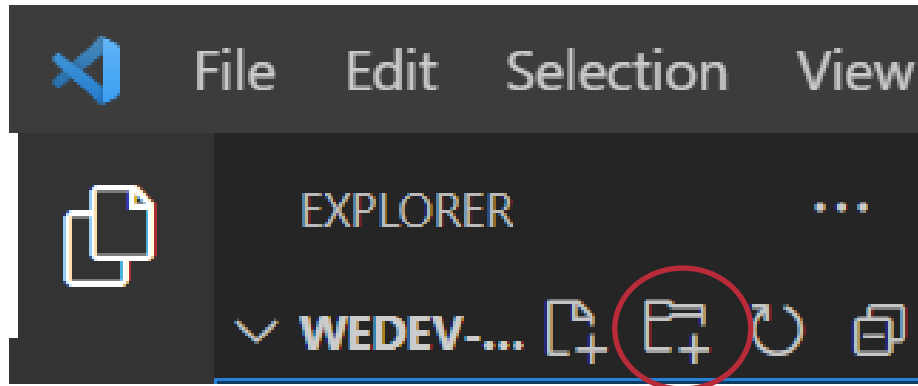


- Μέσα σε αυτό το χώρο (φάκελο) του Project μας θα δημιουργήσουμε όλες τις σελίδες μας



# Δημιουργία φακέλου

Προγραμματισμός στο Web

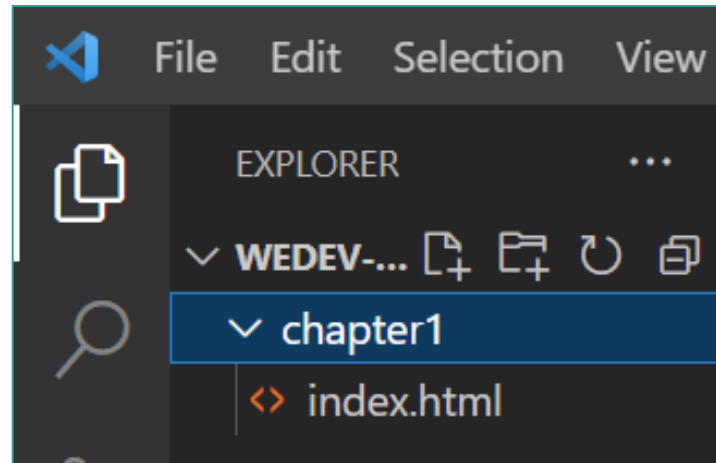


- Δημιουργούμε ένα νέο φάκελο (υποφάκελο του βασικού φακέλου) με όνομα chapter1



# Δημιουργία index.html

Προγραμματισμός στο Web



- Δημιουργούμε νέο αρχείο με όνομα index.html



# Βασική δομή HTML εγγράφου

Προγραμματισμός στο Web

```
index.html X
chapter1 > <> index.html > html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8 </head>
9 <body>
10
11 </body>
12 </html>
```

- Γράφουμε **!** και αμέσως μετά πατάμε το πλήκτρο **tab** ή **enter** και δημιουργείται αυτόματα η δομή του html εγγράφου (το **!** είναι shortcut του *Emmet*, ενός plugin που υπάρχει ήδη εγκατεστημένο στο VS Code)



# Δομή HTML εγγράφου (1)

Προγραμματισμός στο Web

```
<> index.html X
<> index.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Document</title>
8  </head>
9  <body>
10
11 </body>
12 </html>
```

- Κάθε έγγραφο που συμμορφώνεται με την έκδοση HTML5 ξεκινά έχοντας στην 1η γραμμή την οδηγία **<!DOCTYPE html>**
- Στη γραμμή 2 ξεκινά το **<html>** tag που κλείνει στη γραμμή 12 με **</html>**. Μέσα σε opening tags όπως αναφέραμε μπορούμε να έχουμε και attributes (ιδιότητες) που προσδιορίζουν περαιτέρω τη σημασία του tag.



# Δομή HTML εγγράφου (2)

Προγραμματισμός στο Web

```
<> index.html X
<> index.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Document</title>
8  </head>
9  <body>
10
11 </body>
12 </html>
```

- Το **lang** attribute προσδιορίζει ότι η βασική γλώσσα του εγγράφου είναι τα Αγγλικά (*lang* = "en" για Ελληνικά, βλ. [https://en.wikipedia.org/wiki/List\\_of\\_ISO\\_639-1\\_codes](https://en.wikipedia.org/wiki/List_of_ISO_639-1_codes)) και χρησιμοποιείται από την Google για language specific results καθώς και από screen readers





# Δομή HTML εγγράφου (3)

```
<> index.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Document</title>
7  </head>
8  <body>
9
10 </body>
11 </html>
```

- Μέσα στο `<html>` element βρίσκονται τα `<head>` και `<body>` elements
- Μέσα στο `<head>` μπορούμε να δίνουμε metadata (ξεκινά με το *meta* tag), που δεν εμφανίζονται στο browser window. Για παράδειγμα το `charset="UTF-8"` όπως αναφέραμε είναι η κωδικοποίηση των χαρακτήρων του εγγράφου. Το *viewport* θα το δούμε στο πλαίσιο του responsive design. Το `<title>` tag είναι ο τίτλος του εγγράφου όπως φαίνεται στο browser window.



# <head> και <body>

- Το **<head>** περιέχει μεταδεδομένα που προσδιορίζουν τη μορφή των δεδομένων και δεν εμφανίζονται στον browser
- Το **<body>** περιλαμβάνει περιεχόμενο – content που εμφανίζεται στο viewport του browser



# Hello World (1)

```
chapter1 > <> hello.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Hello</title>
8  </head>
9  <body>
10
11      <p>Hello World!</p>
12
13  </body>
14  </html>
```

- Στο <head> το tag **<meta>** υποδηλώνει μεταδεδομένα (metadata) όπου στο παράδειγμα δηλώνουμε το *charset* δηλαδή την κωδικοποίηση του εγγράφου σε utf-8. Το charset το χρησιμοποιεί ο browser για να μεταφράσει σωστά τη σελίδα
- Θα πρέπει πάντα η κωδικοποίηση να είναι UTF-8 ενώ θα πρέπει και να έχει αποθηκευτεί το έγγραφο ως UTF-8
- Επίσης, με το tag **<title>** ορίζουμε τον τίτλο της σελίδας, όπως θα φαίνεται στην 1<sup>η</sup> γραμμή του παραθύρου του browser



# Hello World (2)

```
chapter1 > <> hello.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Hello</title>
8  </head>
9  <body>
10
11      <p>Hello World!</p>
12
13 </body>
14 </html>
```

- Αν δεν οριστεί charset, το default charset που χρησιμοποιούν οι browsers για να διαβάσουν είναι UTF-8. Επίσης, αν δεν οριστεί lang, η default lang είναι "en"



# To <p> element

```
<> index.html X
<> index.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Hello World from Coding Factory</title>
8  </head>
9  <body>
10
11      <p>Hello World! Hello Coding Factory!</p>
12
13  </body>
14  </html>
```

- Μέσα στο body εισάγουμε το <p> element. Το <p> σημαίνει paragraph και το περιεχόμενο που δίνουμε μέσα στα <p> </p> εμφανίζεται (γίνεται render) στο browser window ως κείμενο



# Default Styling για το <p>

```
<> index.html X
<> index.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Hello World from Coding Factory</title>
8  </head>
9  <body>
10
11      <p>Hello World! Hello Coding Factory!</p>
12
13  </body>
14  </html>
```

- Αν δεν ορίσουμε εμείς συγκεκριμένη μορφοποίηση (styling), η μορφοποίηση προσδιορίζεται από τον browser.
- Στον chrome οι default τιμές είναι : μέγεθος 16px, χρώμα μαύρο, γραμματοσειρά *Times New Roman*, αφήνοντας μία κενή γραμμή πριν και μία μετά για το <p>



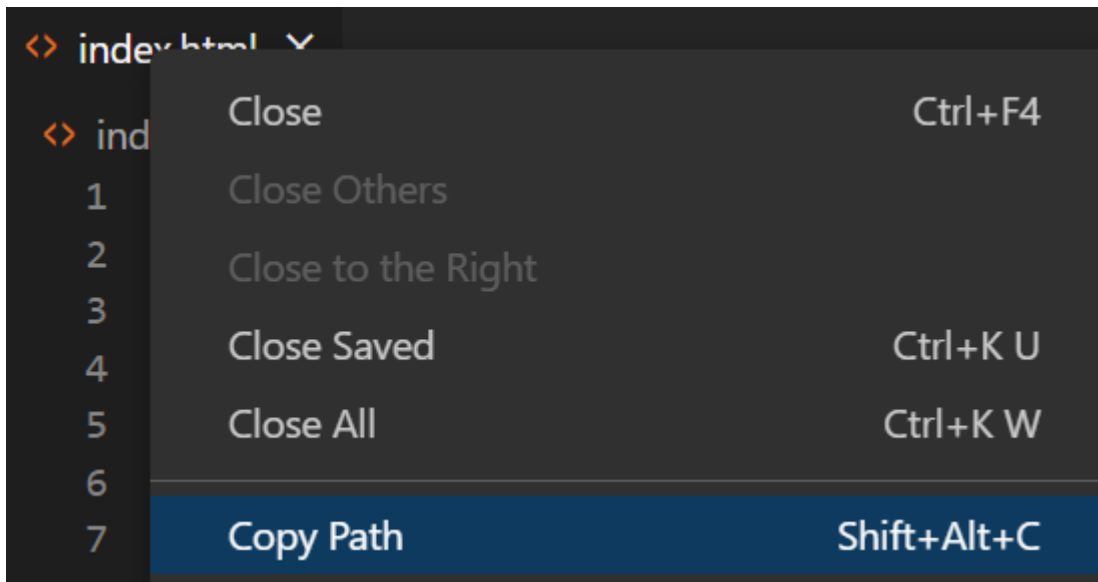
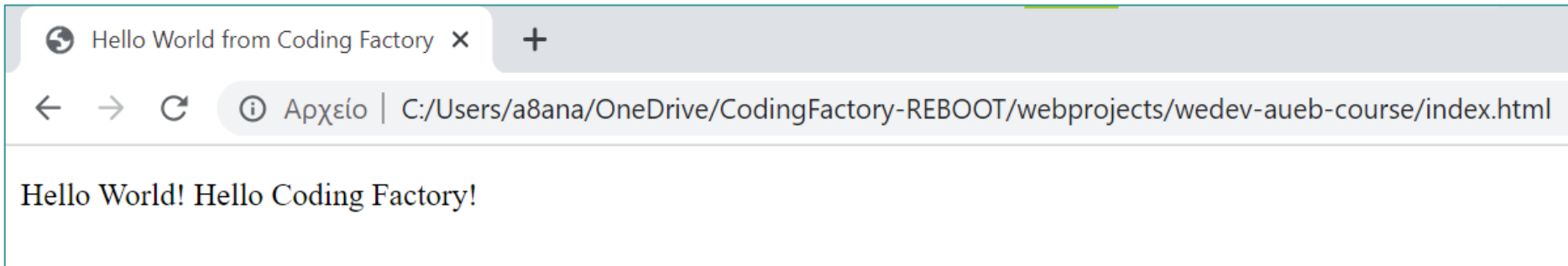
# Default Styling

- Γενικά, οι browsers όπως ο Chrome δίνουν ένα default styling σε όλα τα στοιχεία HTML, Για παράδειγμα το κείμενο εμφανίζεται με γραμματοσειρά Times New Roman και μέγεθος γραμματοσειράς 16px
- Διαφορετικά όλα θα εμφανίζονταν σαν μία ακολουθία χαρακτήρων και στοιχείων και δεν θα ήταν αναγνώσιμα
- Περαιτέρω όπως θα δούμε μπορούμε κι εμείς να δώσουμε styling μιας και το default styling είναι βαρετό και δεν οδηγεί σε μεγιστοποίηση της εμπειρίας του χρήστη



# Άνοιγμα html με Copy Path

Προγραμματισμός στο Web



- Δεξί κλικ στο html αρχείο και επιλέγουμε **Copy Path**. Κάνουμε paste στον browser που επιθυμούμε





# Live Server

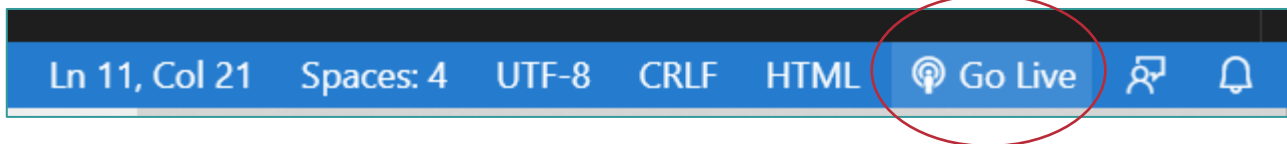
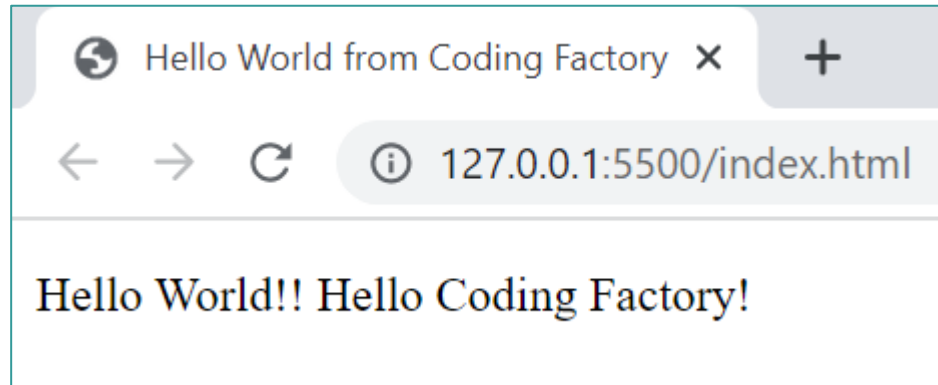
Προγραμματισμός στο Web

The screenshot shows the Visual Studio Code interface. The 'EXTENSIONS: MARKETPLACE' view is active, displaying a search for 'Live Server'. The first result, 'Live Server' by Ritwick Dey, is selected and highlighted with a red circle. The extension details panel on the right shows the 'Live Server' extension (v5.7.9) by Ritwick Dey, with 26,457,834 downloads and a 5-star rating (393 reviews). The description states: 'Launch a development local Server with live reload feature for static & dynamic pages'. The 'Disable' and 'Uninstall' buttons are visible, along with a gear icon for settings. A note at the bottom indicates 'This extension is enabled globally.' The left sidebar shows the 'Extensions' icon (a square with four smaller squares) circled in red.

- Ο Live Server είναι ένα extension, που μας δίνει έναν Web Server ώστε να τρέχουμε τις σελίδες μας σε περιβάλλον Web Server. Επίσης κάθε φορά που κάνουμε αλλαγές στη σελίδα μας και save, η σελίδα στον Web Server ανανεώνεται αυτόματα
- Πάμε στα Extensions, κάνουμε αναζήτηση για Live Server, επιλέγουμε το πρώτο αποτέλεσμα και κάνουμε Install



# Live Server



- Κάτω δεξιά στο status bar του VSC επιλέγουμε Go Live και ανοίγει ο default browser. Η γραμμή διεύθυνσης είναι τώρα 127.0.0.1:5500/index.html, όπου το 127.0.0.1 αντιστοιχεί στον τοπικό web server (localhost), το 5500 είναι το port, ένα συγκεκριμένο σημείο στη μνήμη όπου ο Web Server δέχεται αιτήσεις από browsers και index.html που είναι το όνομα της σελίδας



# HTML Tags

```
<> index.html X
<> index.html > html > body > p
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>My first html page</title>
6      </head>
7      <body>
8          <p>
9              Hello World!
10         </p>
11     </body>
12 </html>
```

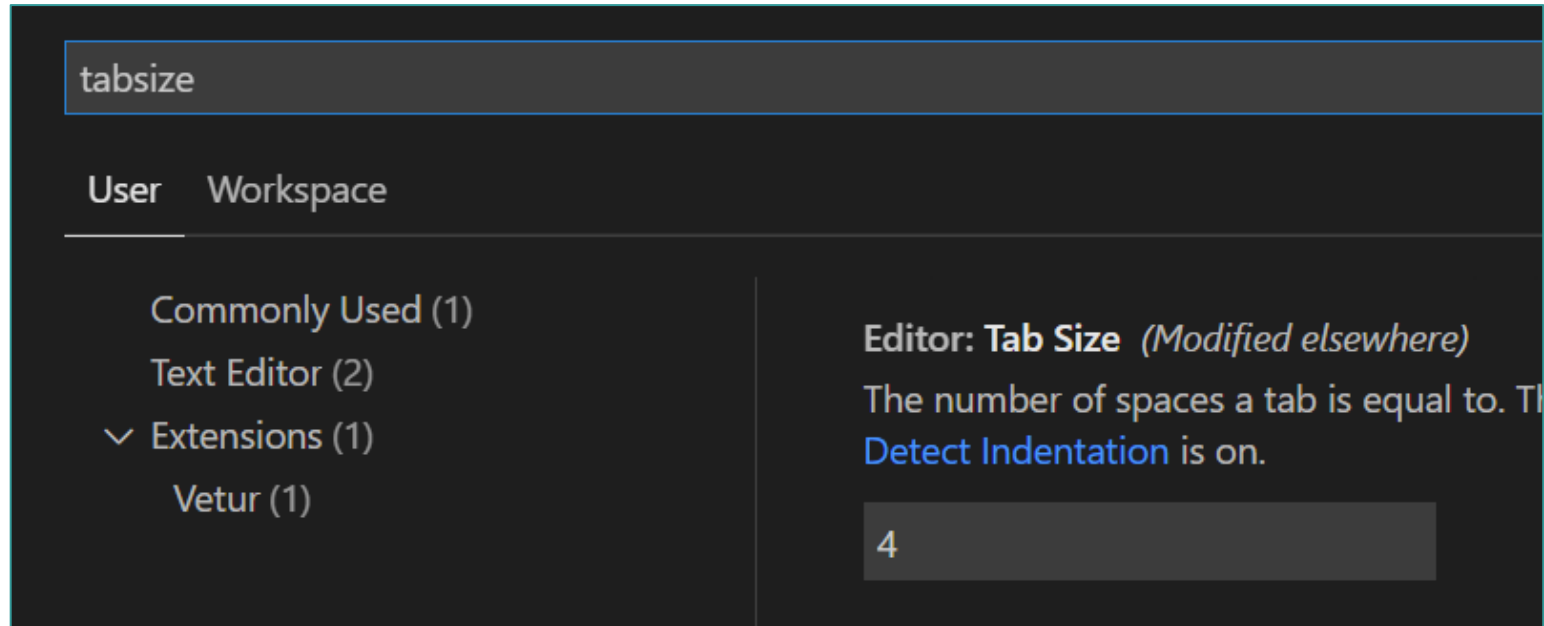
- Στην HTML5, τα HTML tags δεν είναι case sensitive, δηλαδή δεν έχει σημασία αν τα γράφουμε με μικρά ή κεφαλαία γράμματα αλλά **συνίσταται ως καλή πρακτική να γράφουμε με πεζά**
- Εξαίρεση είναι μόνο η λέξη DOCTYPE που το convention είναι με κεφαλαία

Επίσης, παρατηρήστε τη στοίχιση. Στοιχίζουμε ένα tab δεξιά (εσοχή) τα εμφωλιασμένα HTML elements ώστε το έγγραφο να είναι καλύτερα αναγνώσιμο (readability)



# Στοίχιση κειμένου

Προγραμματισμός στο Web



- Σύμφωνα με τα guidelines της Google και του mozilla.org καλό είναι να χρησιμοποιούμε 2 spaces για εσοχές (indentation). Γενικά 2-4 spaces είναι αποδεκτό indentation
- Στο File/Preferences/Settings, αναζητούμε για *tabsize* και μπορούμε να ορίσουμε το **tabsize** δηλαδή πόσα κενά θα αφήνει αυτόματα ο VS Code σε κάθε HTML κώδικα



# Βασικά UI Elements

Προγραμματισμός στο Web

- Τα βασικά UI Elements που περιέχει μία σελίδα είναι
  - Κείμενο: `<h1> .. <h6>, <p>`
  - Εικόνες: `<img>`
  - Βίντεο: `<video>`
  - Ήχος: `<audio>`
  - Υπερσύνδεσμοι: `<a>`



# HTML elements κειμένου

Προγραμματισμός στο Web

- Επικεφαλίδες.
  - `<h1>` έως `<h6>` μας επιτρέπουν να ορίσουμε ότι κάποιο κείμενο είναι μεγάλη επικεφαλίδα `<h1>` ή μικρότερου μεγέθους επικεφαλίδα `<h2>` έως `<h6>`
  - `<h1> Η επικεφαλίδα μου</h1>`
- Παράγραφος
  - `<p>`
  - `<p>Η παράγραφός μου</p>`



# Παράδειγμα – h1 & p elements

```
<> index.html X
<> index.html > ...

3   <head>
4   |   <meta charset="UTF-8" />
5   |   <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6   |   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7   |   <title>Hello World from Coding Factory</title>
8   |   </head>
9   |   <body>
10  |   <h1>Hello Coding Factory!</h1>
11  |
12  |   <p>
13  |   |   Lorem ipsum dolor sit amet consectetur adipisicing elit. Deserunt dolor
14  |   |   quaerat vel exercitationem amet possimus sint aut. Deserunt error porro
15  |   |   rerum similique sequi excepturi. Dolorem nostrum reprehenderit et sit
16  |   |   sapiente!
17  |   </p>
18  |   </body>
19  </html>
```

- Στον κώδικα υπάρχει ένα `<h1>` element που μορφοποιεί το κείμενο Hello Coding Factory ως Επικεφαλίδα-1, δηλαδή με μέγεθος γραμματοσειράς διπλάσιο από το προεπιλεγμένο του browser, bold και μία κενή γραμμή πάνω και κάτω



# Αποτέλεσμα

Προγραμματισμός στο Web



- Το Lorem ipsum είναι placeholder για κείμενο, για να βλέπουμε πως φαίνεται το layout της σελίδας μας, αντί για το κείμενο αυτό καθαυτό. Το χρησιμοποιούμε κατά τη φάση του σχεδιασμού.
- Εισάγεται με lorem και tab





# Σχόλια (Comments)

Προγραμματισμός στο Web

```
<!--index.html file-->
<!DOCTYPE html>
```

- Σχόλια  
βάζουμε με  
`<!--` Εδώ  
σχόλια `-->`

Τα σχόλια χρησιμοποιούνται για  
εσωτερική τεκμηρίωση



# Κενά Διαστήματα / Αλλαγές γραμμής, Κενή, Οριζόντια γραμμή

Προγραμματισμός στο Web

- Τα κενά και οι αλλαγές γραμμής στο περιεχόμενο αγνοούνται (θεωρούνται ως **ένα** κενό)
- Για ρητή αλλαγή γραμμής χρησιμοποιούμε το **<br>** (break)
- **<hr>** για οριζόντια γραμμή (horizontal rule)
- Για διατήρηση της δομής του περιεχομένου με τα κενά και τις αλλαγές γραμμής χρησιμοποιούμε το tag:

**<pre> </pre>** (preformatted)



# Σύνδεσμοι - Links

- Τα Links δηλαδή οι υπερσύνδεσμοι ορίζονται με το element **<a>** που σημαίνει anchor
- Το **<a>** όπως και άλλα tags στην HTML μπορούν να περιέχουν και **ιδιότητες (attributes)**, που προσδιορίζουν περαιτέρω τη λειτουργία των tags
- Έτσι το **<a>** έχει την ιδιότητα **href** (hypertext reference) που προσδιορίζει την URL διεύθυνση. Για παράδειγμα:|

`<a href="https://www.aueb.gr/">Οικ. Παν. Αθηνών</a>`

- Η URL διεύθυνση πρέπει να περιέχει το `https://` (ή `http://`, αλλά πλέον οι browsers προειδοποιούν για σελίδες που δεν είναι https, δηλαδή secure http, δηλαδή ασφαλείς όσο αφορά την κρυπτογραφημένη μεταφορά δεδομένων)
- Το κείμενο του συνδέσμου Οικ. Παν. Αθηνών προσδιορίζει το ορατό μέρος



# Παράδειγμα – anchor element

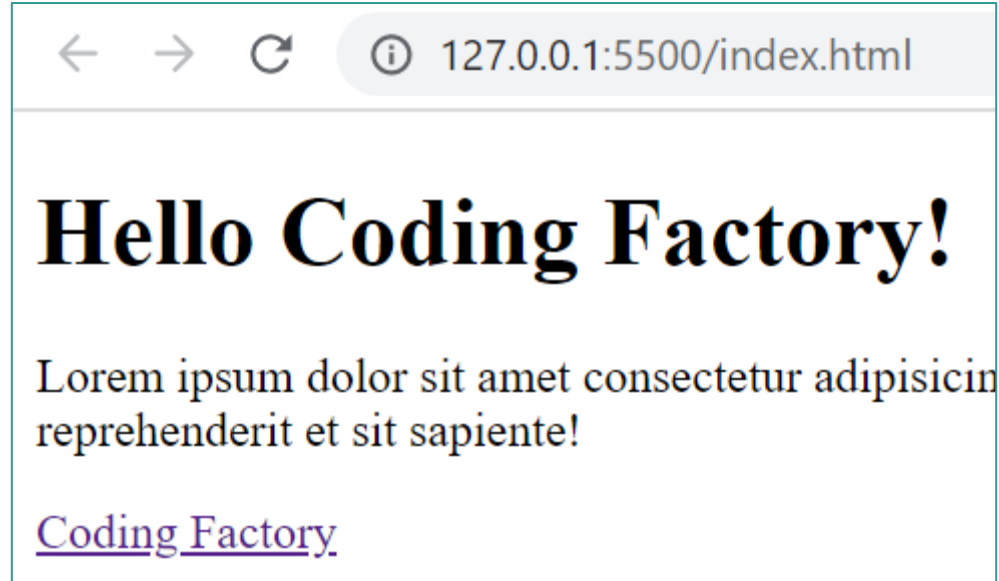
```
<> index.html X
<> index.html > ...
3   <head>
4       <meta charset="UTF-8" />
5       <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6       <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7       <title>Hello World from Coding Factory</title>
8   </head>
9   <body>
10      <h1>Hello Coding Factory!</h1>
11
12      <p>
13          Lorem ipsum dolor sit amet consectetur adipisicing elit. Deserunt dolor
14          quaerat vel exercitationem amet possimus sint aut. Deserunt error porro
15          rerum similique sequi excepturi. Dolorem nostrum reprehenderit et sit
16          sapiente!
17      </p>
18
19      <a href="https://codingfactory.aueb.gr/" target="_blank">Coding Factory</a>
20  </body>
21 </html>
```

- Η ιδιότητα ***target="\_blank"*** ανοίγει τον σύνδεσμο σε νέο παράθυρο



# Αποτέλεσμα

Προγραμματισμός στο Web





# Εικόνες

- Εικόνες εισάγουμε με το **<img>** στο σημείο που θέλουμε να εμφανίζεται (το **<img>** είναι empty tag και δεν έχει closing tag)
- Το **<img>** έχει τις ιδιότητες **src** όπου ορίζουμε τη θέση του αρχείου εικόνας στο δίσκο ή στο web, **alt** που ορίζει ένα εναλλακτικό κείμενο που εμφανίζεται αν δεν μπορεί να εμφανιστεί η εικόνα ή αν το HTML αρχείο διαβάζεται από ειδικά προγράμματα (screen readers) καθώς και τις ιδιότητες **width** και **height** που ορίζουν το πλάτος και το ύψος αντίστοιχα της εικόνας. Π.χ.
- ****



# Void elements < />

- Μπορούμε να χρησιμοποιήσουμε και το notation της XHTML που είναι πιο αυστηρή σε θέματα σύνταξης της γλώσσας θα έπρεπε να κλείσουμε με /> το <img> tag, δηλαδή:
- ``
- Και γενικά για κάθε empty tag μπορεί να χρησιμοποιείται αυτό το notation, αλλά στην HTML5 δεν είναι υποχρεωτικό



# Εικόνες – Ιδιότητα `source` (1)

- Ιδιότητα ***src*** (`source`). Αν δεν προσδιορίζεται φάκελος τότε υπονοείται ο ίδιος φάκελος που βρίσκεται το `index.html`, π.χ. ``
- Αλλιώς πρέπει να προσδιορίζεται ο φάκελος. Π.χ. αν το `opa.png` βρίσκεται στον υποφάκελο `img`, τότε γράφουμε:
- ``
- Το παραπάνω είναι ένας *relative* τρόπος προσδιορισμού της θέσης στο δίσκο ενός αρχείου, όπου η τελεία συμβολίζει τον τρέχον φάκελο. Δεν δίνουμε δηλαδή το *absolute path*, π.χ. `C:\Users\asana\OneDrive\CodingFactory-REBOOT\webprojects\weden-aueb-course\chapter1\img\opa.png`. Εφόσον το `html` αρχείο είναι στο φάκελο `chapter1`, τότε μπορούμε να προσδιορίσουμε τη θέση του `opa.png` σε σχέση με τη τρέχουσα (του `html file`) θέση αρχείου, άρα αν με τελεία αναφερόμαστε στον τρέχον φάκελο, τότε η θέση του αρχείου είναι `./img/opa.png`





# Εικόνες – Ιδιότητα `source` (2)

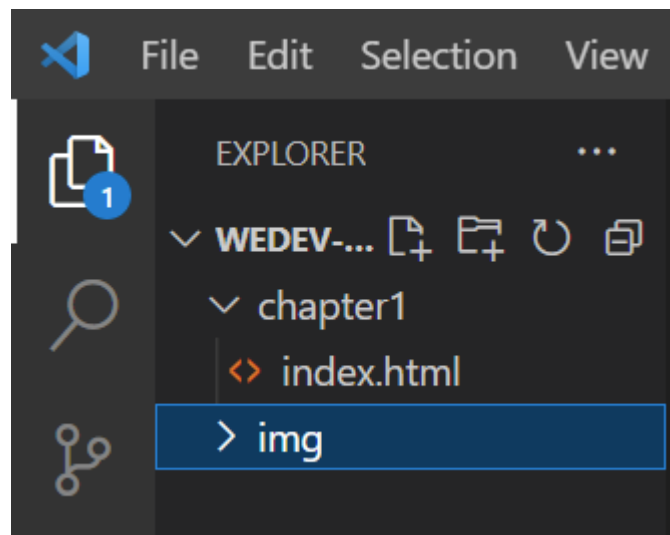
Προγραμματισμός στο Web

- Άλλοι ***relative*** τρόποι προσδιορισμού της θέσης ενός αρχείου είναι
  - Με `../img/head.png`, η διπλή τελεία συμβολίζει τον parent folder
  - *Root-relative links*, π.χ. `/resources/images/large.png` όπου το `/` στην αρχή συμβολίζει τον root folder (τον υψηλότερο δηλαδή folder σε μία ιεραρχία φακέλων) ενός συστήματος αρχείων



# Φάκελος img (1)

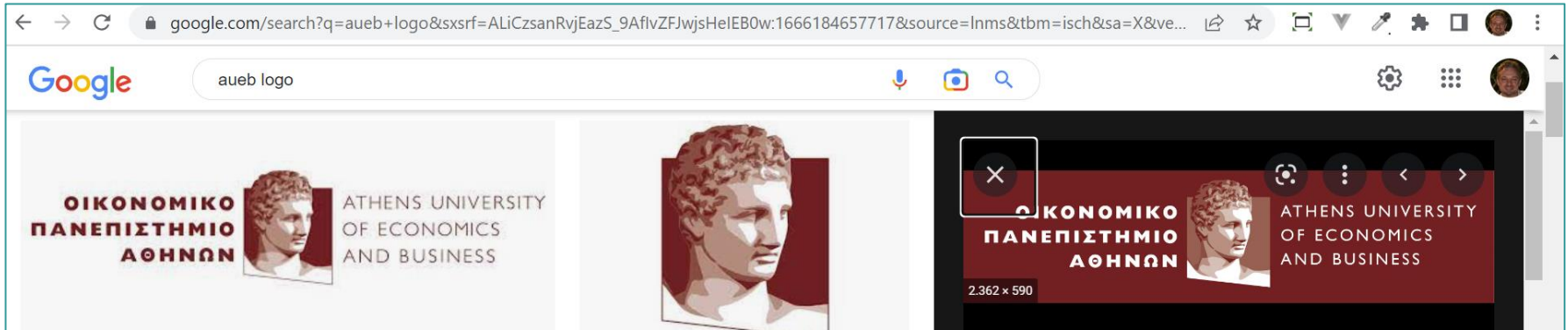
- Θα δημιουργήσουμε ένα υποφάκελο του αρχικού μας φακέλου (του φακέλου δηλαδή webdev-aueb-course) με όνομα img και σε αυτό τον φάκελο θα εισάγουμε όλες τις εικόνες μας





# Αναζήτηση εικόνας

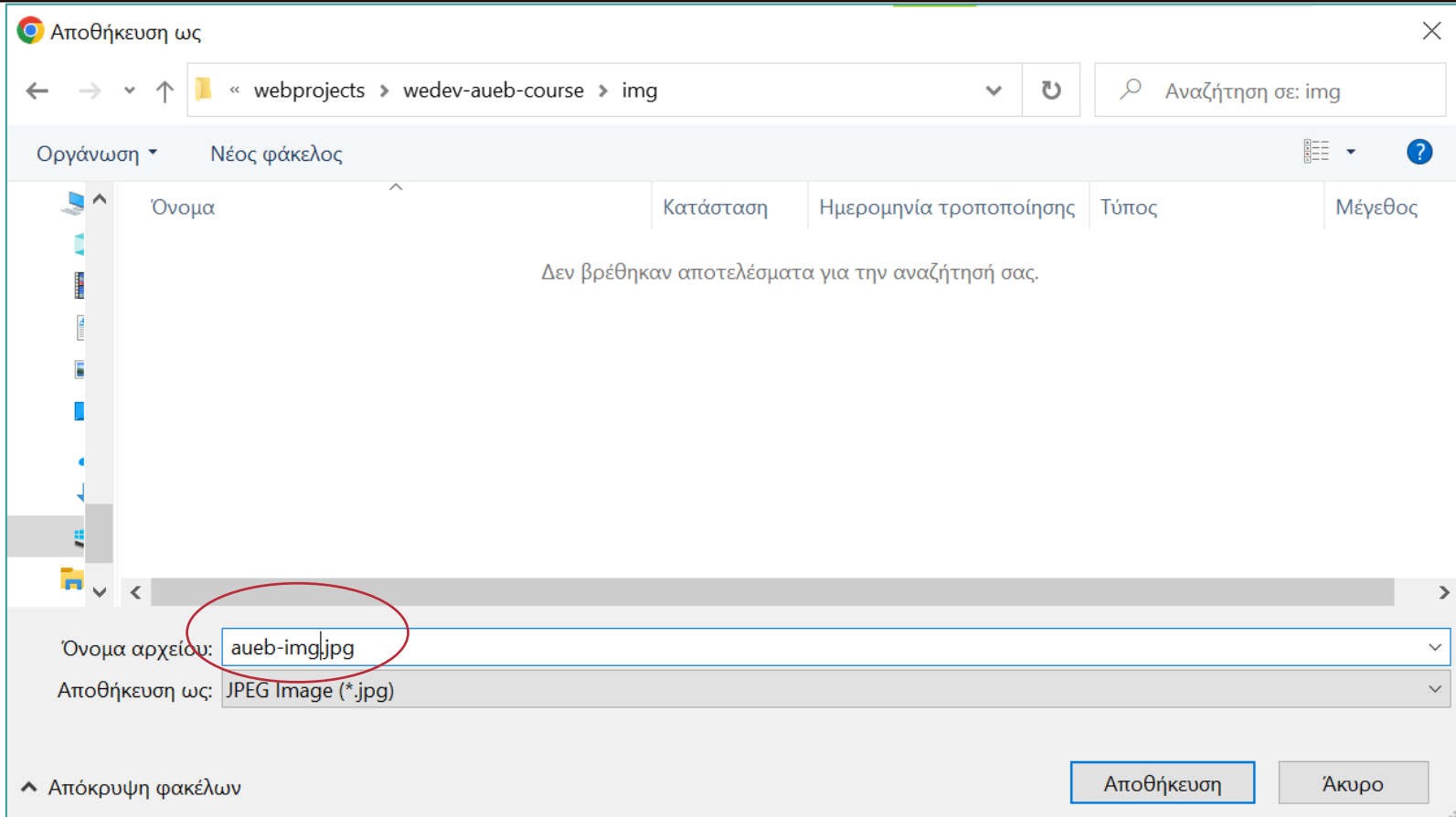
Προγραμματισμός στο Web



- Η εικόνα έχει ανάλυση 2362 x 590 pixels. Είναι αρκετά μεγάλη επομένως και θα φαίνεται καλά και σε μεγάλες οθόνες με υψηλή ανάλυση
- Αν η εικόνα είχε μικρή ανάλυση, π.χ. 400 x 100, τότε σε διαστάσεις μεγαλύτερες από 400 px θα γινόταν pixelated, θα χάλαγε η ποιότητά της



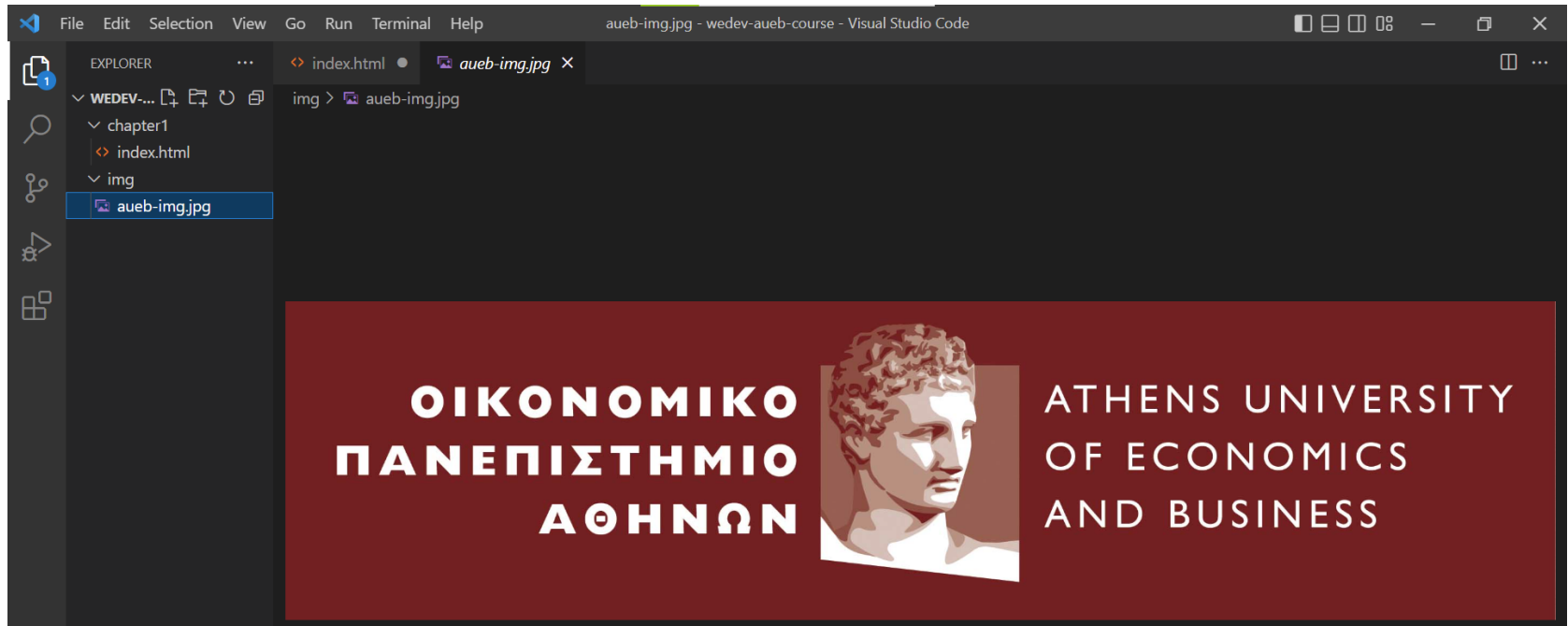
# Φάκελος img (2)



- Αποθηκεύουμε ως aueb-img.jpg ή άλλο πρόσφορο όνομα αρχείου



# Εικόνα



- Συνοπτικά, δημιουργούμε **φάκελο img** μέσα στον αρχικό φάκελο του project μας. Αναζητούμε στο google για AUEB Logo εικόνες και αποθηκεύουμε μέσα στον φάκελο που δημιουργήσαμε
- Άλλος τρόπος είναι να εισάγουμε στο src κατευθείαν το URL



# Παράδειγμα – img tag

Προγραμματισμός στο Web

```
11 <body>
12   
13   <h1>Hello Coding Factory!</h1>
14
15   <p>
16     Lorem ipsum dolor sit, amet consectetur adipisicing elit.
17     Labore blanditiis nostrum, illum temporibus doloribus impedit
18     quibusdam harum et animi sequi! Repudiandae accusamus,
19     iste ipsa assumenda earum iure explicabo laudantium molestiae.
20   </p>
21
22   <a href="https://codingfactory.aueb.gr/" target="_blank">Coding Factory</a>
23 </body>
24
25 </html>
```

- Στο **width** εισάγουμε **800** (υπονοείται pixels). Το auto στο height θα ρυθμίσει αυτόματα το ύψος της εικόνας ώστε το aspect ratio (αναλογία πλάτους-ύψους) να μείνει σταθερό.



# Αποτέλεσμα

Προγραμματισμός στο Web



## Hello Coding Factory!

Lorem ipsum dolor sit, amet consectetur adipisicing elit. Labore blanditiis nostrum, illum temporibus doloribus impedit quibus assumenda earum iure explicabo laudantium molestiae.

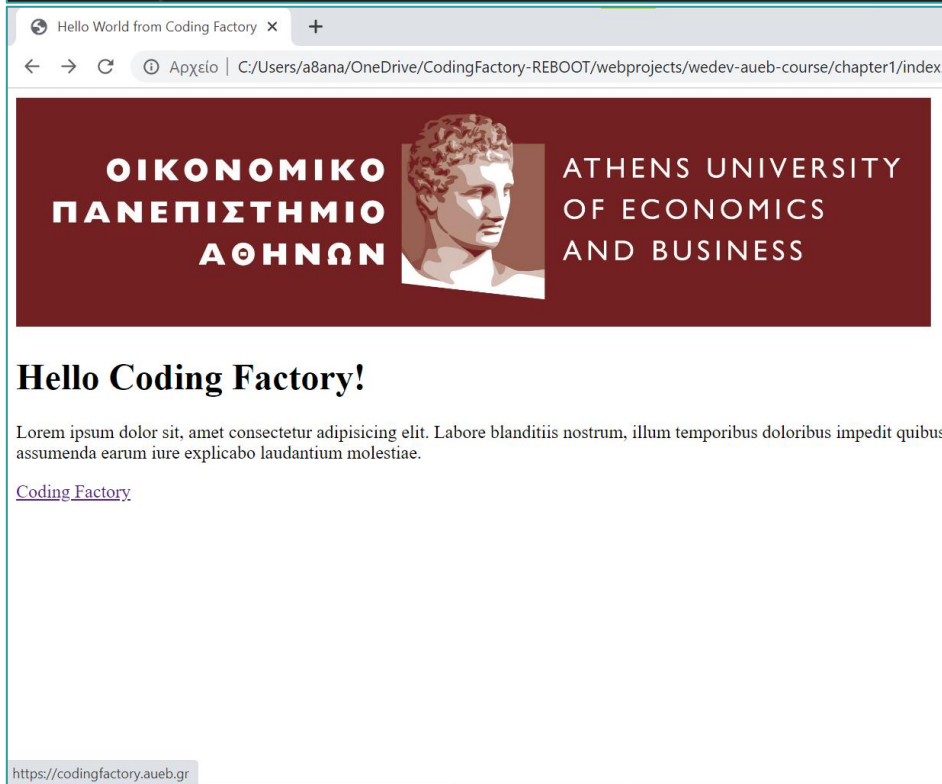
[Coding Factory](#)



# Image και Link

Προγραμματισμός στο Web

```
11 <body>
12   <a href="https://codingfactory.aueb.gr/" target="_blank">
13     
14   </a>
15
```



- Αντί για ορατό κείμενο στο `<a>` εισαγάγαμε το `img`
- Όταν βάλουμε το ποντίκι πάνω στο `image`, κάτω αριστερά στον browser (στο status bar) εμφανίζεται ο σύνδεσμος





# Πνευματικά Δικαιώματα

Προγραμματισμός στο Web

- Προσοχή, οι περισσότερες εικόνες στο Web είναι copyrighted. Απαγορεύεται να χρησιμοποιούμε, ιδιαίτερα για εμπορικούς σκοπούς εικόνες που δεν είναι δικές μας ή που δεν έχουμε άδεια να χρησιμοποιήσουμε, ή που δεν είναι στο public domain, δηλαδή είναι ελεύθερες προς χρήση
- Για να μπορούμε να δούμε με τι δικαιώματα συνοδεύονται διάφορα ψηφιακά προϊόντα έχουμε τα creative commons, <https://creativecommons.org/> που είναι ένας standard τρόπος ενημέρωσης για τα πνευματικά δικαιώματα που συνοδεύουν ένα ψηφιακό αγαθό



# Πίνακες

- Για τη δόμηση tabular data η HTML μας παρέχει τα **<table>** element.
- **<table>**
  - **<tr>** γραμμή (table row)
  - **<th>** επικεφαλίδες στηλών ή γραμμών (table header)
  - **<td>** δεδομένα-στήλη (table data)



# Δομή table

- Ένας πίνακας περιέχει header, body και footer. Τα HTML στοιχεία είναι `<thead>`, `<tbody>` και `<tfoot>`. Δεν είναι απαραίτητο να υπάρχουν αλλά οι browsers χρησιμοποιούν αυτά τα tags για να κάνουμε scrolling στο table body καθώς και για να εκτυπώνονται τα header και footer σε κάθε σελίδα, ενώ επίσης μπορούμε να εφαρμόσουμε styling ξεχωριστά σε κάθε section

			← Header
			} ← Body
			← Footer



# Απλός Πίνακα


```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible"
6   <meta name="viewport" content="width=device-width, initial-scale=1">
7   <title>Διδάσκοντες</title>
8 </head>
9 <body>
10
11   <table>
12     <tr>
13       <th>Μάθημα</th>
14       <th>Ώρες Διδασκαλίας</th>
15     </tr>
16     <tr>
17       <td>Java</td>
18       <td>100</td>
19     </tr>
20     <tr>
21       <td>.NET</td>
22       <td>40</td>
23     </tr>
24   </table>
```

- Στην απλή του μορφή ένας πίνακας αποτελείται από γραμμές `<tr>` και στήλες `<th>` ή `<td>`
- Τα `<th>` γίνονται render από τον browser ως bold με στοίχιση στο κέντρο, ενώ τα `<td>` δεν είναι bold και είναι στοιχισμένα αριστερά



# Αποτέλεσμα

Προγραμματισμός στο Web

		Αρχείο   C
Μάθημα		Ώρες Διδασκαλίας
Java	100	
.NET	40	



# Πιο σύνθετος Πίνακας

## Προγραμματισμός στο Web

```
26 <table>
27   <caption>Διδάσκοντες</caption>
28   <thead>
29     <tr><th>Όνομα</th><th>Επώνυμο</th><th>Μάθημα</th></tr>
30   </thead>
31   <tbody>
32     <tr>
33       <td>Μάκης</td>
34       <td>Καπέτης</td>
35       <td>Βάσεις Δεδομένων</td>
36     </tr>
37     <tr>
38       <td>Αθ.</td>
39       <td>Ανδρούτσος</td>
40       <td>Java</td>
41     </tr>
42     <tr>
43       <td>Άννα</td>
44       <td>Γιαννούτσου</td>
45       <td>Drupal</td>
46     </tr>
47   </tbody>
48   <tfoot>
49     <tr><th colspan="3">Coding Factory AUEB</th></tr>
50   </tfoot>
51 </table>
```

- Το <caption> δίνει ένα τίτλο στον πίνακα ακριβώς πάνω από το <thead>
- Τα <thead> , <tbody> και <tfoot> είναι απλά containers, δηλαδή ομαδοποιούν τα <tr> ώστε να δώσουμε μία κοινή μορφοποίηση
- Η ιδιότητα colspan συγχωνεύει στήλες (και η αντίστοιχη rowspan συγχωνεύει γραμμές)



# Αποτέλεσμα

← → ↻ ⓘ Αρχείο   C:/Users/a8ana/		
<b>Μάθημα Ώρες Διδασκαλίας</b>		
Java	100	
.NET	40	
Διδάσκοντες		
<b>Όνομα</b>	<b>Επώνυμο</b>	<b>Μάθημα</b>
Μάκης Καπέτης	Βάσεις Δεδομένων	
Αθ. Ανδρούτσος	Java	
Άννα Γιαννούτσου	Drupal	
<b>Coding Factory AUEB</b>		

- Παρατηρούμε ότι η τελευταία γραμμή καλύπτει (span) τρεις στήλες



# Colspan - Rowspan

```
55     <table>
56         <tr><th colspan="2">Όνοματεπώνυμο</th><th>Ηλικία</th></tr>
57         <tr><td>Παπαδόπουλος</td><td> Νίκος</td><td>32</td></tr>
58         <tr><td>Παπαδόπουλος</td><td>Κώστας</td><td>20</td></tr>
59         <tr><td colspan="2" rowspan="2">Min/Max</td><td>20</td></tr>
60         <tr><td>32</td></tr>
61     </table>
62
63
64 </body>
65 </html>
```

- Με colspan όπως αναφέραμε συγχωνεύουμε στήλες και με rowspan συγχωνεύουμε γραμμές





# Αποτέλεσμα

Προγραμματισμός στο Web

Όνοματεπώνυμο	Ηλικία
Παπαδόπουλος Νίκος	32
Παπαδόπουλος Κώστας	20
Min/Max	20
	32



# Λίστες - Lists

- Οι λίστες είναι σημαντικά HTML στοιχεία και χρησιμοποιούνται πολύ συχνά στην HTML.
- Παρέχονται τρεις τύποι
  - `<ul>` unordered lists
  - `<ol>` ordered lists
  - `<dl>` description lists



- Unordered lists: `<ul><li></li></ul>`
  - Τα `<li>` είναι list items
- ordered lists: `<ol><li></li></ol>`
- Description lists: `<dl><dt></dt><dd></dd></dl>`
  - Τα `<dt>`, `<dd>` είναι description term και description data αντίστοιχα

```
<ul>
  <li>Ανατολική Αττική</li>
  <li>Δυτική Αττική </li>
</ul>
```

```
<ol>
  <li>Ανατολική Αττική</li>
  <li>Δυτική Αττική </li>
</ol>
```



# Nested Lists

```
<body>
<h2>Λίστα μέσα σε λίστα</h2>
<ul>
  <li>Στερεά Ελλάδα</li>
  <li>Αττική
    <ul>
      <li>Ανατολική Αττική</li>
      <li>δυτική Αττική </li>
    </ul>
  </li>
  <li>Κρήτη</li>
</ul>
</body>
```



# Παράδειγμα - Λίστες

Προγραμματισμός στο Web

- Οι `<ul>` λίστες είναι οι πιο σημαντικές γιατί με αυτές φτιάχνουμε – όπως θα δούμε – menu

```
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Lists</title>
8 </head>
9 <body>
10
11   <h2>Υποχρεωτικά Μαθήματα</h2>
12   <ul>
13     <li>HTML/CSS</li>
14     <li>JavaScript</li>
15     <li>Bootstrap</li>
16   </ul>
17
18   <h2>Μαθήματα Επιλογής</h2>
19   <ol>
20     <li>HTML/CSS</li>
21     <li>JavaScript</li>
22     <li>Bootstrap</li>
23   </ol>
24
25   <dl>
26     <dt>HTTP</dt>
27     <dd>HyperText transfer Protocol</dd>
28     <dt>HTML/CSS</dt>
29     <dd>HyperText Markup Language/Cascading Style Sheets</dd>
30   </dl>
31
32 </body>
```



# Emmet (1)

- Το Emmet είναι ένα extension για text editors που είναι ενσωματωμένο στο VS Code
- Κάνει expand abbreviations, δηλαδή μπορούμε να δίνουμε συντομεύσεις
- Για παράδειγμα το ! και tab που δίνουμε είναι abbreviation (συντομογραφία) του `<!DOCTYPE>` και γίνεται expand η βασική δομή του project. Αυτό μας δίνεται από το Emmet



# Emmet (2)

- Χάρη στο Emmet μπορούμε και δίνουμε και άλλες συντομεύσεις
- Για παράδειγμα το `ul>li` κάνει expand σε `<ul><li></li></ul>` (Το σύμβολο `>` σημαίνει άμεσος απόγονος - child)
- Επίσης, `ul>li*3` και κάνει expand σε `<ul><li></li><li></li><li></li></ul>` (Το `*n` σημαίνει n φορές - Multiplication)



# HTML Forms (1/3)

- Οι φόρμες χρησιμοποιούνται για **εισαγωγή δεδομένων των χρηστών**
- `<form> </form>`
- Οι φόρμες αποτελούνται από αντικείμενα ελέγχου (controls) ή αλλιώς widgets
  - **Label** (ετικέτα) – Εκτύπωση κειμένου
  - **Text box** (Εισαγωγή κειμένου) – π.χ. `<input type="text" name="lastname">`
  - **Text Area** (Εκτύπωση κειμένου σε γραμμές και στήλες). π.χ. `<textarea name="details" rows="5" cols="50">Δώστε το επώνυμό σας</textarea>`
  - **Combo box** (drop-down list) – Μία επιλογή από Λίστα π.χ.  
`<select name="fruits">`  
    `<option value="Apples">Apples</option>`  
    `<option value="Oranges">Oranges</option>`  
`</select>`





# HTML Forms (2/3)

- **checkbox** – Επιλογή σε square-box μία ή περισσότερες τιμές  
`<input type="checkbox" name="fruits" value="Apples" />Apples<br>`  
`<input type="checkbox" name="fruits" value="Oranges" checked /> Oranges<br>`
- **radio button** – Μία μόνο επιλογή  
`<input type="radio" name="mstatus" value="nmar" checked>non-married<br>`  
`<input type="radio" name="mstatus" value="mar">Married`
- **Submit button** – Για αποστολή δεδομένων φόρμας  
`<input type="submit" value="Πατήστε Click">`



# HTML Forms (3/3)

- Άλλα input types:
  - Date
  - Time
  - Number
  - Range



# Form (1)

## Προγραμματισμός στο Web

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8 </head>
9 <body>
10
11   <form>
12     <div>
13       <label for="username">Username</label>
14       <input type="text" name="username" id="username">
15     </div>
16     <div>
17       <label for="email">E-mail</label>
18       <input type="email" name="email" id="email">
19     </div>
20     <div>
21       <label for="message">Message</label>
22       <textarea name="message" id="message" cols="30" rows="10"></textarea>
23     </div>
24     <div>
25       <input type="submit" value="Submit">
26     </div>
27   </form>
28 </body>
29 </html>
```

- Τα `<div>` είναι containers και χρησιμοποιούνται για να δώσουν δομή στη φόρμα ώστε τα στοιχεία της φόρμας (ζεύγη label/input) να είναι το ένα κάτω από το άλλο
- Για usability και accessibility έχουμε χρησιμοποιήσει την ιδιότητα `for` σε κάθε label με τιμή το `id` του αντίστοιχου input, ώστε όταν κάνει κλικ ο χρήστης στο Label να πατιέται το αντίστοιχο input



# Usability - Accessibility

Προγραμματισμός στο Web

- Το `for` του `label` σε συνδυασμό με το `id` του `input element` και άρα ο συσχετισμός αυτών των δύο `controls` επιτρέπει να κάνουμε κλικ στο `label` και να ενεργοποιούμε το αντίστοιχο `input control` γεγονός σημαντικό ιδιαίτερα σε συσκευές με μικρές οθόνες και `controls` όπως `check boxes`, `radio buttons`, που είναι μικρά
- Επίσης, σημαντικό για το `usability` είναι η ιδιότητα `type` στα `input`, όπως για παράδειγμα το `email` που ενεργοποιεί τον έλεγχο στις περιπτώσεις που ο χρήστης δεν δώσει έγκυρη διεύθυνση `e-mail`
- Επίσης, τα `labels` είναι σημαντικά για `accessibility` σε άτομα με χαμηλή όραση γιατί οι `screen readers` διαβάζουν το `label` και το αναφέρουν (το κάνουν 'speak')



# Input submit vs button

- Τα `<input>` δεν έχουν closing tag. Επομένως δεν έχουν ορατό μέρος
- Το `<input type="submit" value="Submit">` για να εμφανίζει κάποιο κείμενο μέσα στο button έχει την τιμή της ιδιότητας value
- Επίσης, το default action του submit είναι η αποστολή των data και το refresh της σελίδας



# Αποτέλεσμα

Προγραμματισμός στο Web

A screenshot of a web browser window. The address bar shows 'Αρχείο | C:/Users/a8ana/OneDr'. The form contains three input fields: 'Username' (a single-line text box), 'E-mail' (a single-line text box), and 'Message' (a large multi-line text area). Below the 'Message' field is a 'Submit' button.

- Έχει εφαρμοστεί μόνο το default styling του browser και για αυτό δεν είναι στοιχισμένα τα στοιχεία



# Σχεδιασμός Φόρμας

Προγραμματισμός στο Web

Firstname/Lastname

Lastname

Firstname

☒ Male ☐ FEMALE

☐ Married

BSc ▼

Submit

- Μία φόρμα συνήθως αποτελείται όπως είδαμε από ζεύγη: Label / Control. Στο παράδειγμα έχουμε για το Επώνυμο και το Όνομα, μέσα σε fieldset container με ένα legend (Firstname/Lastname).
- Τα Labels χρησιμοποιούνται ως απλό κείμενο και για λόγους accessibility, ενώ τα Textboxes για την εισαγωγή στοιχείων από τον χρήστη
- Άλλα controls για εισαγωγή στοιχείων από τον χρήστη είναι τα **Radio buttons** (που επιτρέπουν μόνο μία επιλογή, π.χ. Άνδρας-Γυναίκα), τα **Combo boxes** που είναι dropdown lists, όπως π.χ. το Επίπεδο Σπουδών (BSc, MSc, PhD) και το **checkbox (Married)** που είναι true/false, όπως το Married, αλλά αν έχουμε πολλά checkboxes μπορούμε να επιλέξουμε όσα θέλουμε, εν αντιθέσει με τα radio buttons, που επιλέγουμε μόνο ένα



# Υλοποίηση φόρμας

```
11 <form>
12   <fieldset style="width:450px;">
13     <legend>Firstname/Lastname</legend>
14     <div>
15       <label for="lastname">Lastname</label>
16       <input type="text" name="lastname" id="lastname" size="50">
17     </div>
18     <div>
19       <label for="firstname">Firstname</label>
20       <input type="text" name="firstname" id="firstname" size="50">
21     </div>
22   </fieldset>
```

- Το style στο fieldset ορίζει το styling του στοιχείου και θέτει ως πλάτος (width) τα 450 pixels. Η σημειολογία είναι key:value





# Form

## Προγραμματισμός στο Web

```
24 <div>
25   <input type="radio" name="gender" id="male" value="Male" checked>
26   <label for="gender">Male</label>
27   <input type="radio" name="gender" id="female" value="Female">
28   <label for="gender">FEMALE</label>
29 </div>
30 <div>
31   <input type="checkbox" name="married" id="married" value="married">
32   <label for="married">Married</label>
33 </div>
34 <div>
35   <label for="level"></label>
36   <select name="level" id="level">
37     <option value="1">BSc</option>
38     <option value="2">MSc</option>
39     <option value="3">PhD</option>
40   </select>
41 </div>
42 <div>
43   <input type="submit" value="Submit">
44 </div>
45 </form>
46
47 </body>
48 </html>
```

- Το Radio button επιτρέπει μία μόνο επιλογή και για αυτό το name είναι ίδιο (gender). Λειτουργεί ως group
- Το checkbox επιτρέπει πολλές επιλογές
- Το value αναφέρεται στην τιμή όταν είναι checked (είτε το radio button ή το checkbox) ενώ μπορούμε να ορίσουμε το default checked με το checked



# Αποτέλεσμα

← → ↻ ⓘ Αρχείο | C:/Users/a8ana/OneDrive/CodingFactory-REBOOT

Firstname/Lastname

Lastname

Firstname

☒ Male ☐ FEMALE

☐ Married

BSc ▼

Submit

- Με λίγο styling όπως θα δούμε στο μάθημα του CSS θα έχουμε καλύτερο οπτικό αποτέλεσμα



# Placeholders

```
<div>
  <label for="submission-date">Submission Date</label>
  <input type="text" placeholder="DD-MM-YYYY">
</div>
```

- Έχουμε προσθέσει ένα input πεδίο για ημερομηνία υποβολής
- Η ιδιότητα placeholder μας εμφανίζει μέσα στο textbox ένα hint για τη μορφή της ημερομηνίας

A screenshot of a web browser window. The address bar shows the file path: C:/Users/a8ana/OneDrive/CodingFactory-REBOOT/. The form contains the following elements:

- A label "Firstname/Lastname" above two stacked text input fields, one for "Lastname" and one for "Firstname".
- Radio buttons for "Male" (selected) and "FEMALE".
- A checkbox for "Married".
- A dropdown menu with "BSc" selected.
- A "Submission Date" label above a text input field with the placeholder "DD-MM-YYYY".
- A "Submit" button.



# Σχεδιασμός (1)

- Πολλές φορές είναι χρήσιμο να **σχεδιάζουμε**, πριν υλοποιήσουμε, ιδιαίτερα όσο αφορά γραφικές διεπαφές, ώστε:
  1. Να έχουμε feedback από το μέλη της ομάδας (Alpha Testing)
  2. Να έχουμε feedback ενδιαφερομένων (stakeholders)
  3. Να έχουμε το feedback και τελικά τη συναίνεση των χρηστών
  4. Να παρέχουμε μία οπτική αναπαράσταση προς την ομάδα των developers που πρόκειται να υλοποιήσει



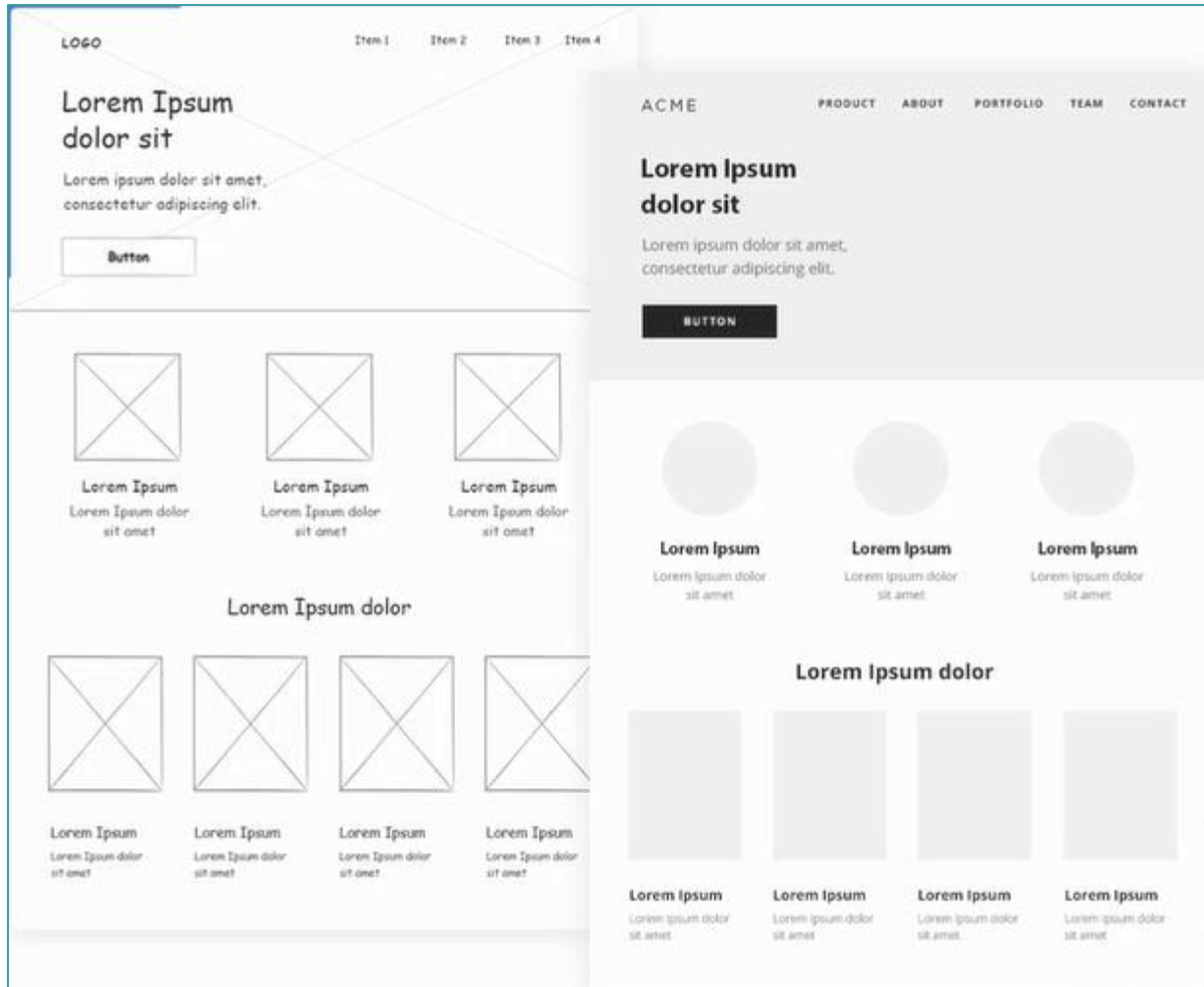
# Σχεδιασμός (2)

- Τα αρχικά σχέδια αυτά ονομάζονται *wireframes*. Τρεις τύποι:
  - **Low Fidelity**. Στο χαρτί, αναπαριστούν βασικά τη δομή της σελίδας
  - **Mid Fidelity**. Με εργαλεία, έμφαση στη δομή, π.χ. εικόνες ως placeholders, τετράγωνα με X
  - **High Fidelity**. Πιο κοντά στο πραγματικό, π.χ. εικόνες με το σχήμα που θέλουμε πραγματικά, π.χ. στρογγυλές, χωρίς X (δείτε επόμενη διαφάνεια)
- Τα mock-ups είναι high fidelity wireframes, που περιέχουν και χρώματα και φτιάχνονται σε αντίστοιχα σχεδιαστικά προγράμματα. Τα mock-ups είναι σελίδες έτοιμες που πάνε προς υλοποίηση.



# Mid Fidelity vs High Fidelity

Προγραμματισμός στο Web



- Το ίδιο wireframe αριστερά mid-fidelity και δεξιά high fidelity
- Έχουν φύγει τα X και έχει πάρει μία μορφή πιο κοντά στο πραγματικό. Δεν έχει χρώματα



# Wireframing, Prototype, Mockup Design Tools

Προγραμματισμός στο Web

- **Figma** Θα το χρησιμοποιήσουμε στο UX/UI Design
- Adobe XD
- Sketch
- Pencil
- Azure
- Mockflow
- Balsamiq
- Miro
- Wireframe CC




# Άσκηση 1

← → ↻ ⓘ Αρχείο | E:/WebDevelopment/atom-exercises/html-chapter1/tableform.html

## Στοιχεία Εκπαιδευόμενου

Επώνυμο:

Όνομα:

Ημερομηνία Γέννησης:  

Έτος Σπουδών:

Login (e-mail):

Password:

- Υλοποιήστε την παραπάνω φόρμα χωρίς μορφοποίηση και στοίχιση, παρά μόνο με το default styling του browser
- Αριθμούς στο Έτος Σπουδών μπορούμε να ορίσουμε με `input type="number"` και ημερομηνίες με `input type="date"`





# Άσκηση 2

## Φόρμα Πληρωμής

Τα υποχρεωτικά πεδία ακολουθούνται από \*.

### Προσωπικά Στοιχεία

Title  
☐ Mr.  
☐ Mrs.  
☐ Dr.

Name: \*

E-mail: \*

Password: \*

### Πληροφορίες Πληρωμής

Card type:

Card number: \*

Expiration date: \*

Προχωρήστε στην πληρωμή

- Υλοποιήστε την φόρμα αριστερά χωρίς styling
- Δώστε έμφαση στη δομή (με div)
- Το <abbr> δίνει το abbreviation (συντομογραφία). Για παράδειγμα όταν βάζουμε το ποντίκι πάνω στο \* θα εμφανίζεται η τιμή της ιδιότητας title (=Required field)

```
<p><abbr title="Required field">*</abbr></p>
```