

# Web Development with HTML/CSS/JavaScript jQuery/AJAX & Bootstrap

Ανάπτυξη Web Εφαρμογών

## Εισαγωγή στην HTML

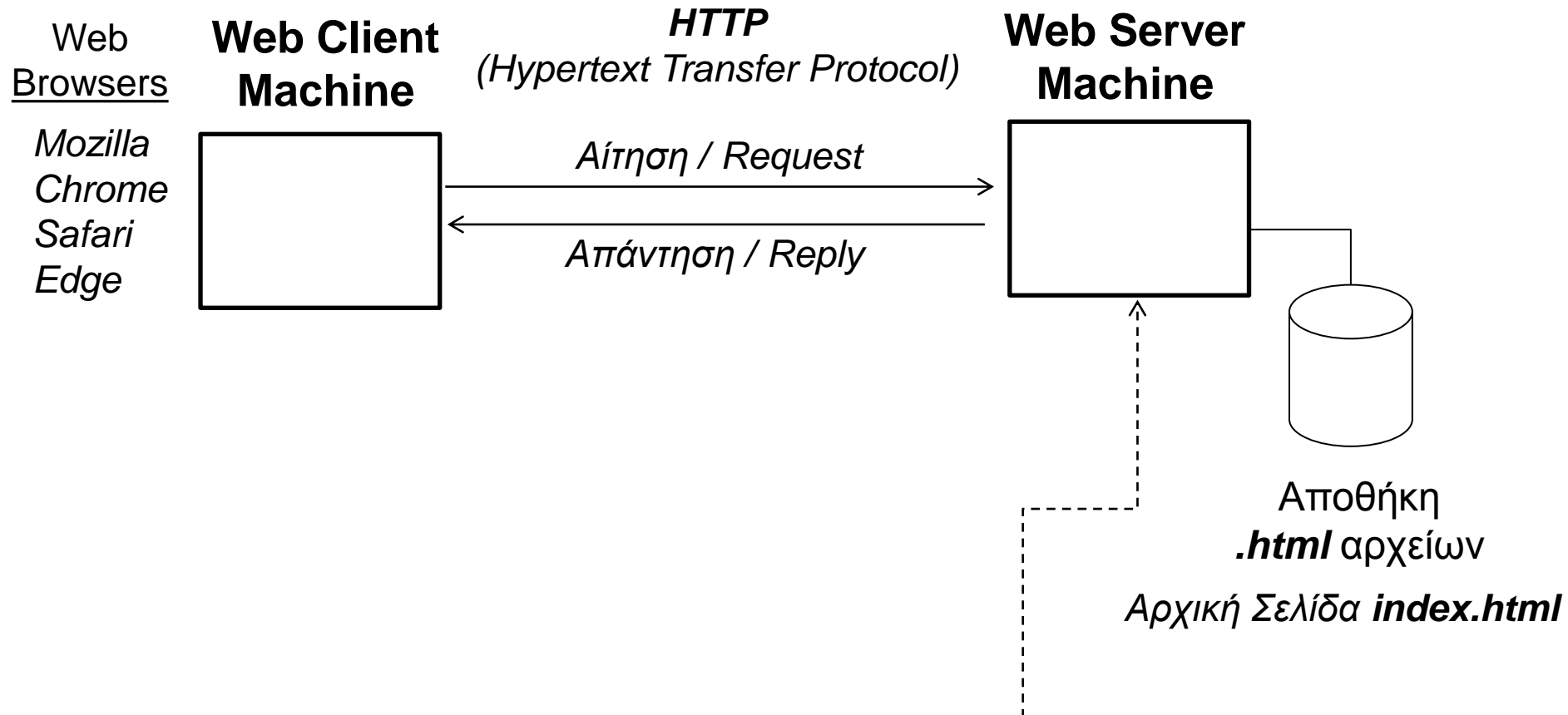
## Αθανάσιος Ανδρούτσος

# World Wide Web

- Ο Sir Timothy John Berners-Lee το 1980, όταν εργαζόταν στο CERN, πρότεινε ένα νέο τρόπο διαμοιρασμού εγγράφων
- Το 1989, ο Tim Berners-Lee (TimBL) εφηύρε το World Wide Web (WWW) υλοποιώντας την πρώτη επιτυχημένη επικοινωνία client-server μέσω του πρωτοκόλλου HTTP

# Αρχιτεκτονική World Wide Web

Ανάπτυξη Web Εφαρμογών



- IP Διεύθυνση: 195.251.255.156
  - Ονοματοδοσία (URL): <http://www.aueb.gr/>
- ↻ **DNS** (Domain Name Service)

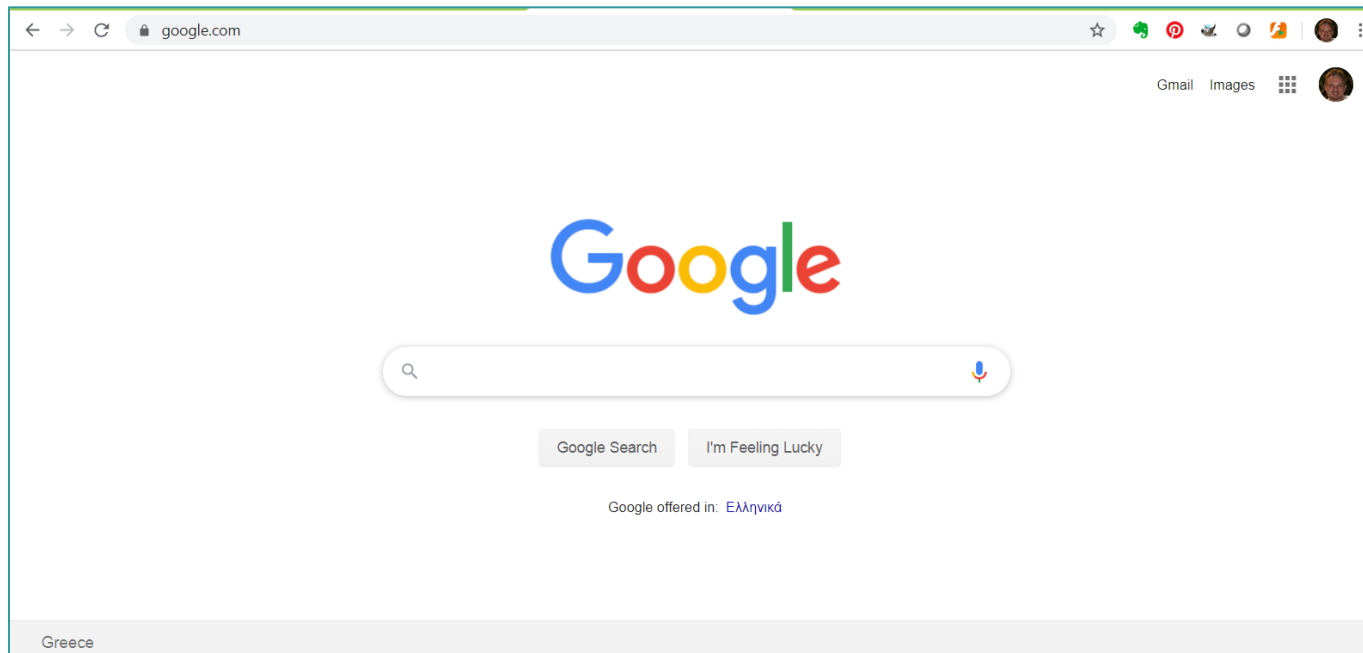
- Ο Tim Berners-Lee είναι ο Διευθυντής του W3C (World Wide Web Consortium) ενός οργανισμού προτυποποίησης του WWW που ιδρύθηκε το 1994 που εκδίδει οδηγίες, εργαλεία, πρότυπα, κλπ. για το Web

(<https://www.w3.org/People/Berners-Lee/>)

- 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

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



- Δύο κατηγορίες
  - **Στατικές** σελίδες, είναι οι σελίδες που δεν αλληλοεπιδρούν με εφαρμογές και επομένως 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. Συνήθως οι δυναμικές σελίδες επικοινωνούν με Βάσεις Δεδομένων

- 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 (annotation ή markup) με τη χρήση tags του περιεχομένου (content) της 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



# Web Browsers

- Πρόκειται για εφαρμογές που χρησιμοποιούνται για να διαβάζουν και να εμφανίζουν HTML σελίδες
- Πρόκειται δηλαδή για HTML Readers
- Οι browsers διαβάζουν τα HTML tags και μπορούν και 'καταλαβαίνουν' τι πρέπει να εμφανίσουν
- Υπάρχουν διάφοροι Web Browsers, όπως ο Chrome, Microsoft Edge, Mozilla Firefox, Apple Safari, Internet Explorer, MS Edge, Opera, κλπ.

# HTML Versions (Εκδόσεις)

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

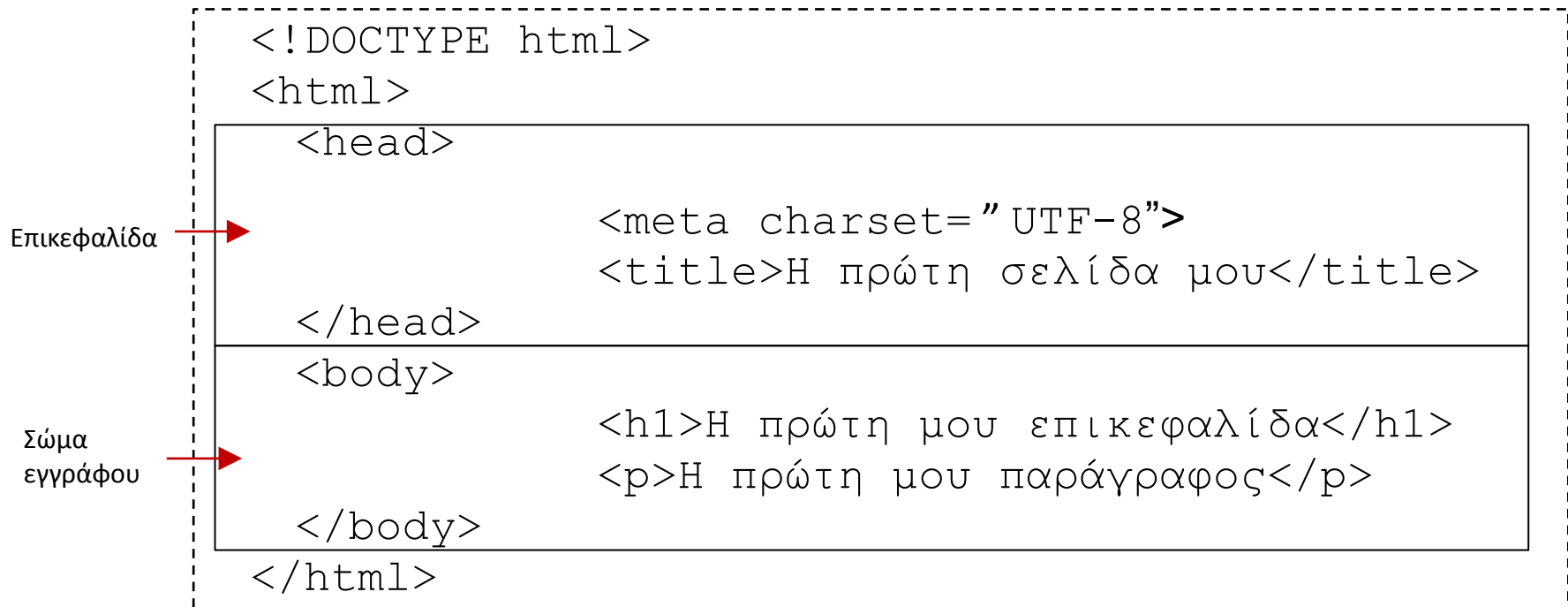
Οι εκδόσεις που κυρίως χρησιμοποιούνται είναι η XHTML (eXtensible HTML) και **HTML5**. Η XHTML έχει πιο αυστηρό συντακτικό.

# HTML Elements

- Το συντακτικό της γλώσσας HTML αποτελείται από στοιχεία HTML (**HTML elements**)
- Τα HTML elements είναι συνήθως ζεύγη ετικετών (tags): Opening/Closing tag. Π.χ. **<p>Hello World</p>**
- Ωστόσο υπάρχουν και **empty ή void elements** δηλαδή tags χωρίς closing tag. Π.χ. **<br>** που σημαίνει *break* και χρησιμοποιούμε για αλλαγή γραμμής

# Δομή Εγγράφου HTML

- Κάθε έγγραφο HTML αποτελείται από δύο μέρη: Επικεφαλίδα (head) και Σώμα (body)

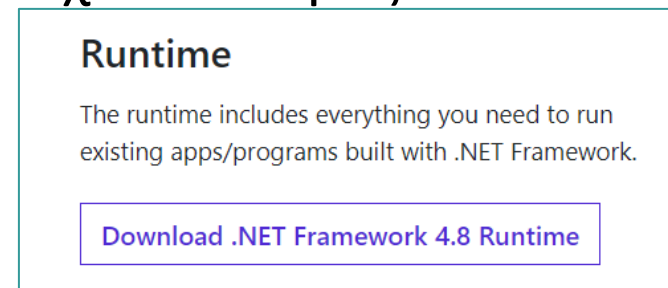


# Visual Studio Code (1)

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

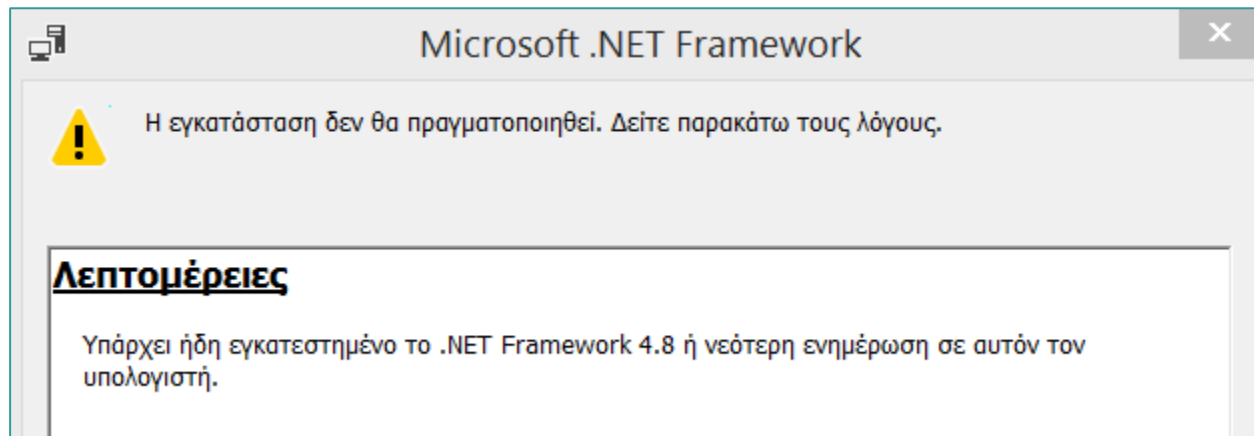
# 1<sup>ο</sup> βήμα: .NET Framework Runtime (1)

- Πρώτα εγκαθιστούμε το **.NET Framework**, ένα περιβάλλον εκτέλεσης προγραμμάτων της Microsoft, μιας και το VSC το χρειάζεται για να μπορεί να εκτελείται
  - Στα **Windows 7, 8.1** και **Windows 10** εγκαταστήστε το **.NET Framework 4.8**
  - Στα **Windows 8**, μπορεί να εγκατασταθεί το **.NET Framework 4.6.1**
  - Στα **Windows Vista**, μπορεί να εγκατασταθεί το **.NET Framework 4.6**
- Τα παραπάνω πακέτα βρίσκονται εδώ:  
<https://dotnet.microsoft.com/download/dotnet-framework/>
- Αφού επιλέξετε την έκδοση, στη συνέχεια κατεβάζετε το **.NET Framework Runtime**



# 1<sup>ο</sup> βήμα: .NET Framework Runtime (2)

- Το .NET Framework είναι ήδη εγκατεστημένο στα Windows 10 αν έχουν κάνει το 10 May 2019 Update
- Σε κάθε περίπτωση αν πάτε να εγκαταστήσετε και είναι ήδη εγκατεστημένο θα εμφανιστεί αντίστοιχο μήνυμα



- Για περιβάλλον Linux, macOS μπορείτε να εγκαταστήσετε το .NET Core 3.1
- <https://dotnet.microsoft.com/download>





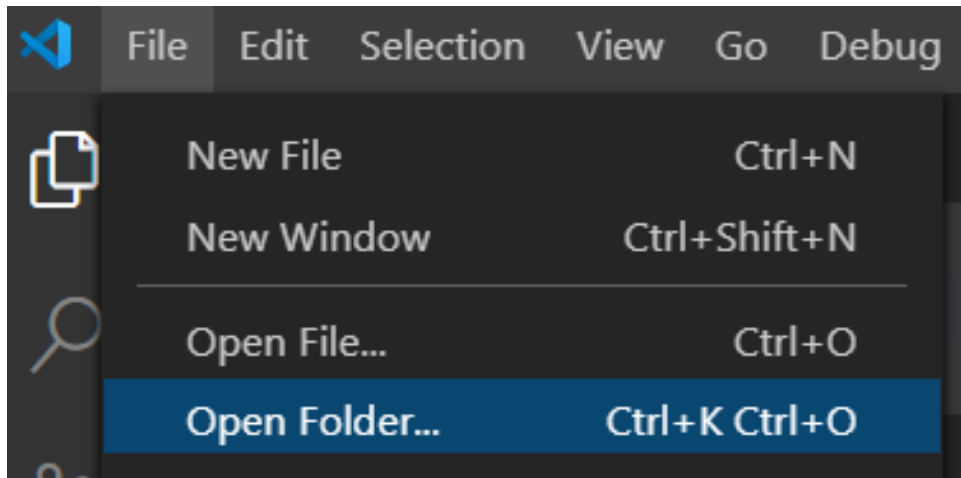
# 2ο Βήμα: Visual Studio Code

- Στη συνέχεια εγκαθιστάτε το Visual Studio Code, αφού πρώτα το κατεβάσετε από τον σύνδεσμο:

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

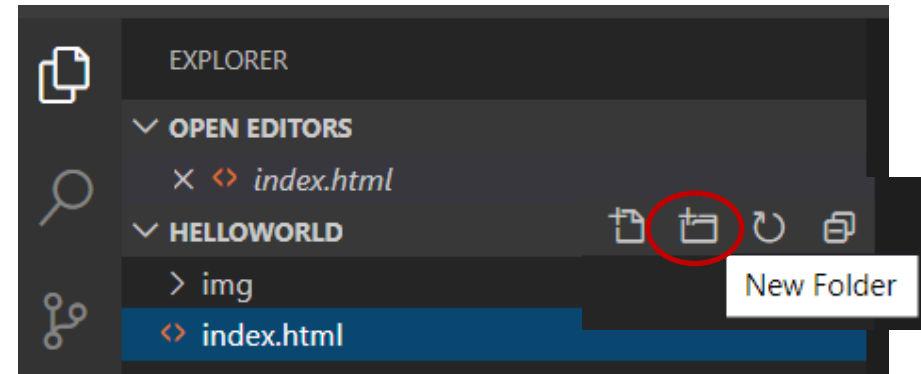
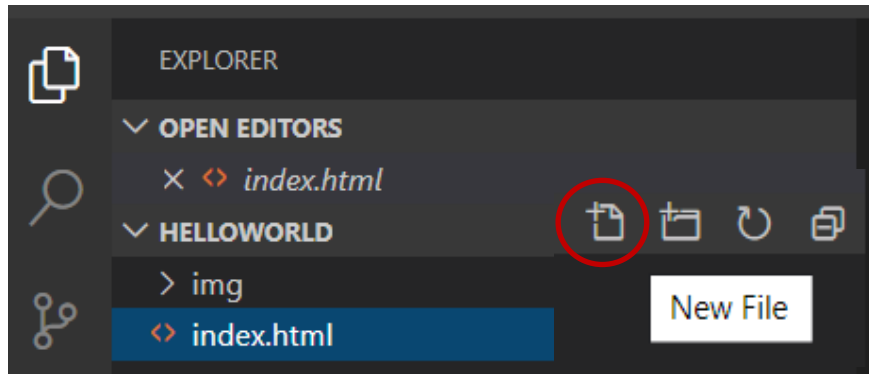
# Δημιουργία Αρχικού Φακέλου

- 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



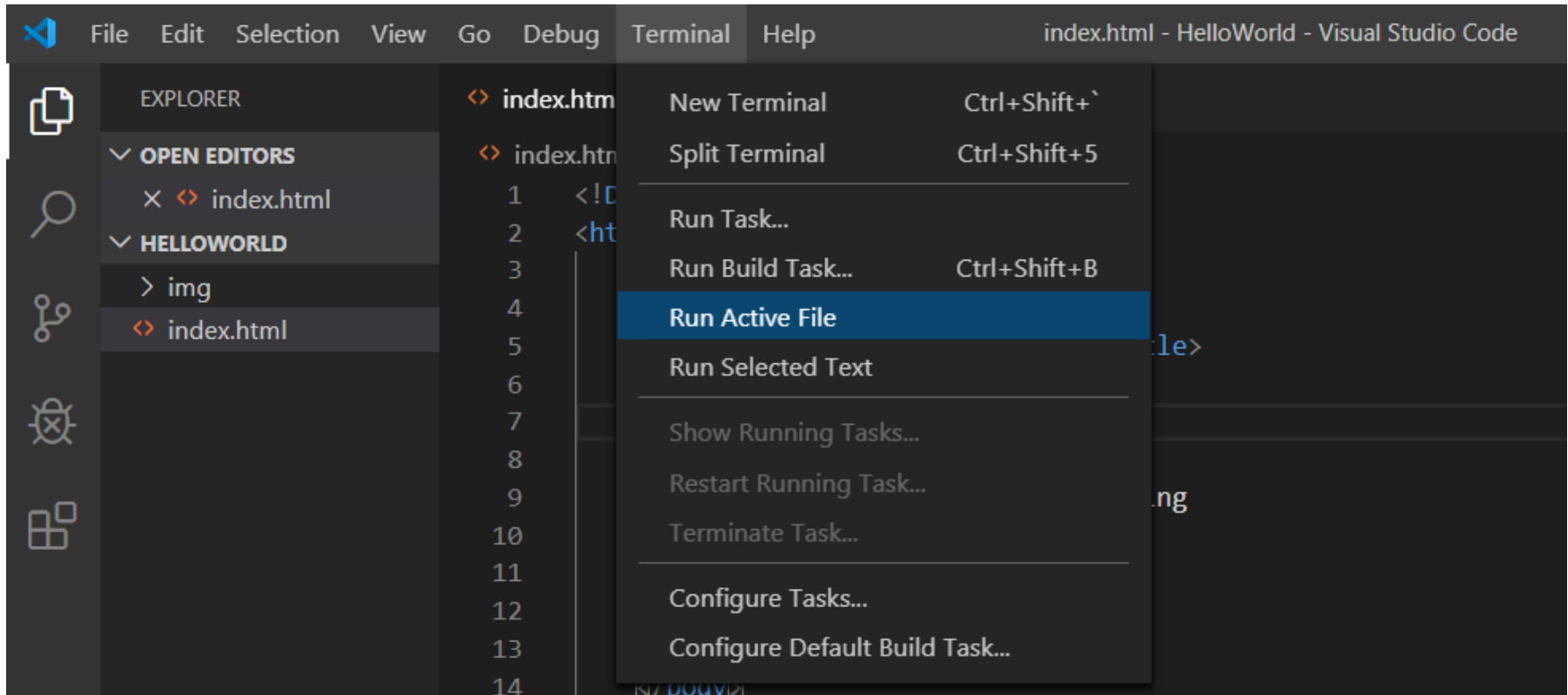
- Ή μπορούμε να επιλέξουμε File/Open Folder και να δημιουργήσουμε το φάκελο και ταυτόχρονα να τον επιλέξουμε
- Μπορούμε επίσης -αν θέλουμε να έχουμε περισσότερους root folders- να αποθηκεύσουμε ως workspace

# Δημιουργία αρχείων / υποφακέλων



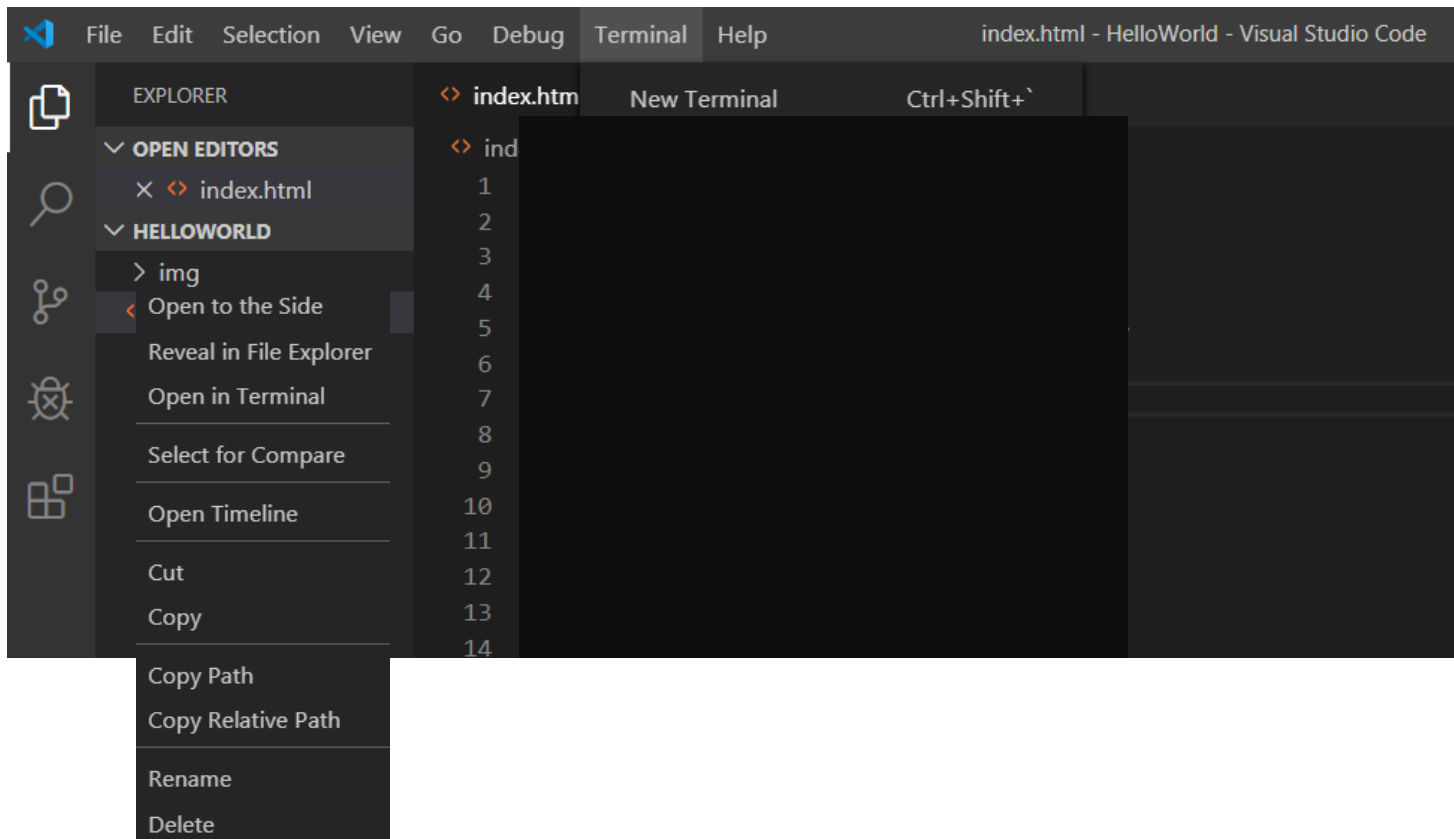
- Μπορούμε να δημιουργήσουμε νέα αρχεία και υποφακέλους. Εδώ έχουμε προσθέσει το αρχείο **index.html** και τον υποφάκελο **img**

# Άνοιγμα html σε default browser



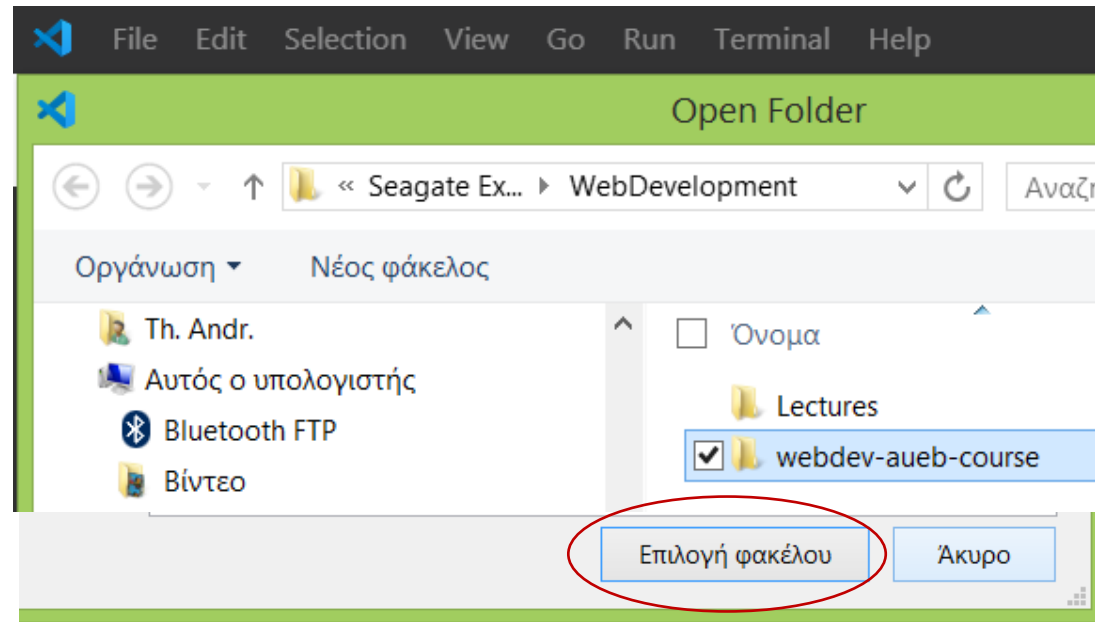
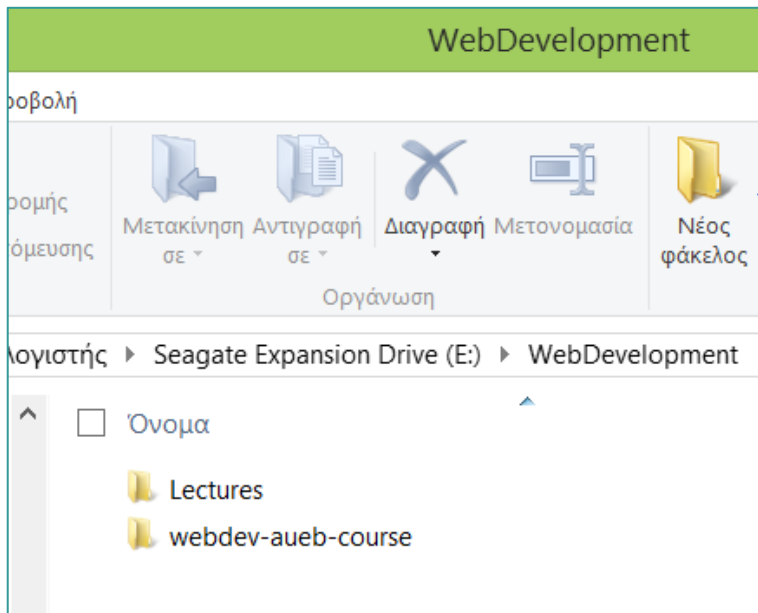
- Εκτελούμε το HTML αρχείο με Terminal / Run Active File και ανοίγει η σελίδα στον default browser

# Άνοιγμα html με Copy Path



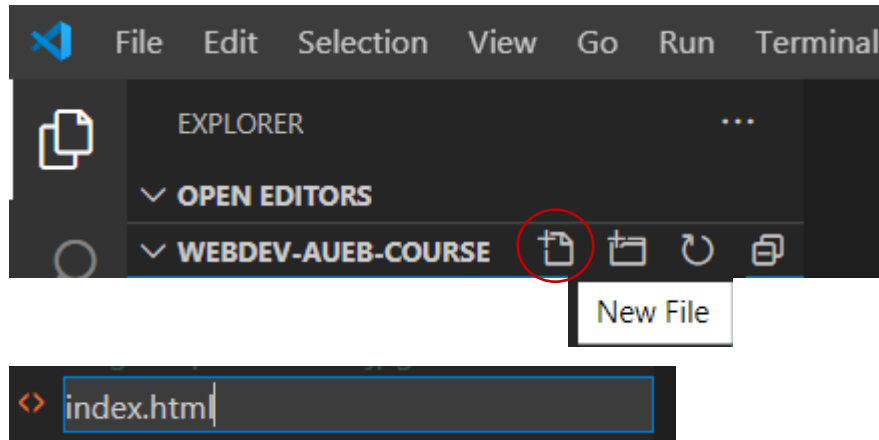
- Δεξί κλικ στο html αρχείο και επιλέγουμε Copy Path. Κάνουμε paste στον browser που επιθυμούμε

# Δημιουργία και Επιλογή Φακέλου



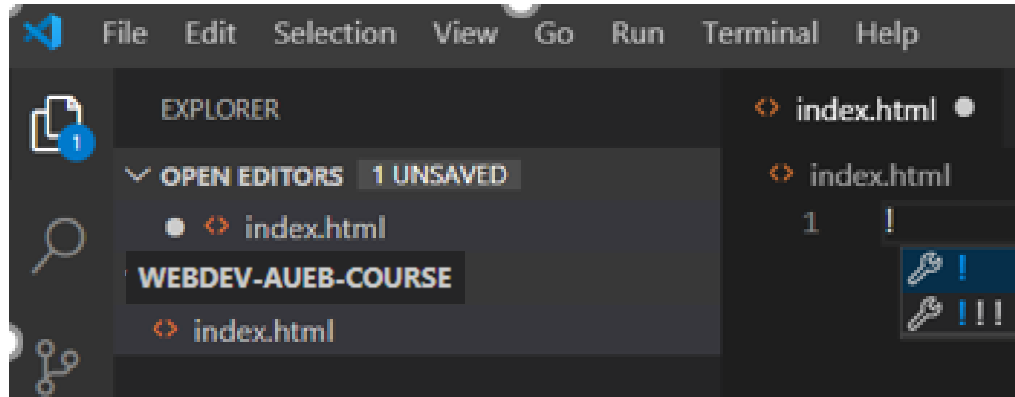
- Πρώτα δημιουργούμε (Δημιουργία / Νέος Φάκελος) και μετά μέσα από το VSC επιλέγουμε File/Open Folder και επιλέγουμε τον φάκελο που δημιουργήσαμε και πατάμε το κουμπί *Επιλογή Φακέλου*

# Δημιουργία index.html

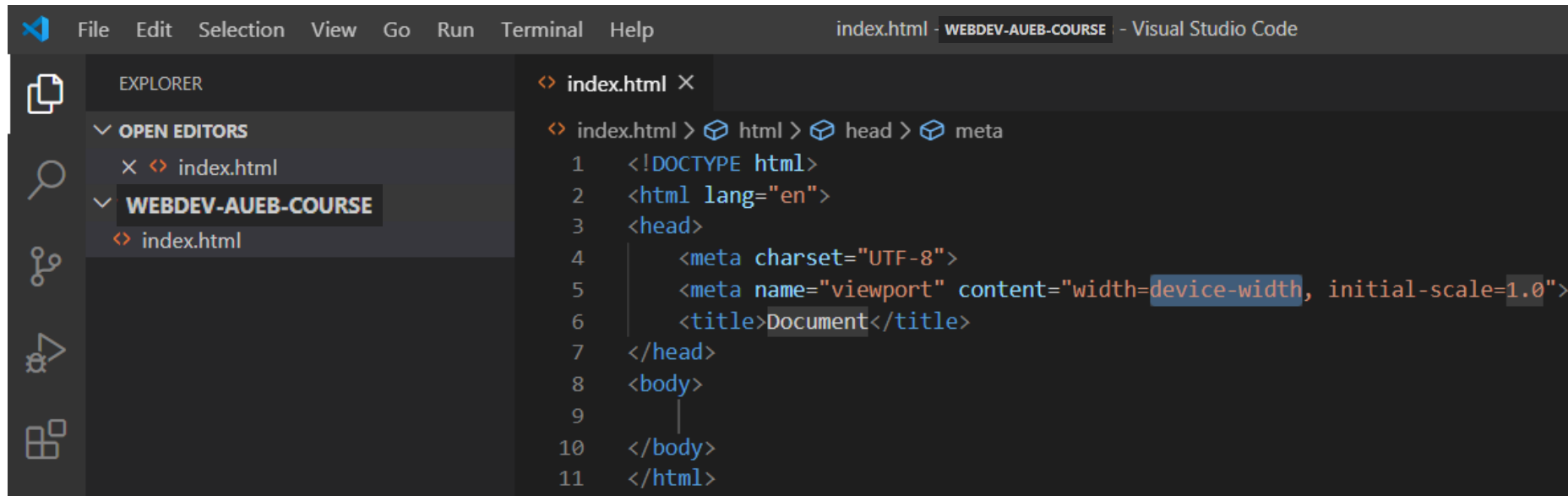


- Δημιουργούμε νέο αρχείο με όνομα index.html

# Βασική δομή HTML εγγράφου



- Γράφουμε ! και αμέσως μετά πατάμε το πλήκτρο **tab** ή **enter** και δημιουργείται αυτόματα η δομή του html εγγράφου





# Βασική δομή HTML εγγράφου

```
<> 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>
```

- 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 11 με **</html>**. Μέσα σε opening tags μπορούμε να έχουμε και attributes (ιδιότητες) που προσδιορίζουν περαιτέρω τη σημασία του tag. 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))
- 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"* 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. Μέσα στα elements που περιέχουν κείμενο, όπως το **<title>** δίνουμε το κείμενο ως περιεχόμενο (Document στο παραπάνω έγγραφο)

# <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 στον browser

# <head>

```
<> 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>
```

- Στο <head> το tag **<meta>** υποδηλώνει μεταδεδομένα (metadata) όπου στο παράδειγμα δηλώνουμε το *charset* δηλαδή την κωδικοποίηση του εγγράφου σε utf-8.
- 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

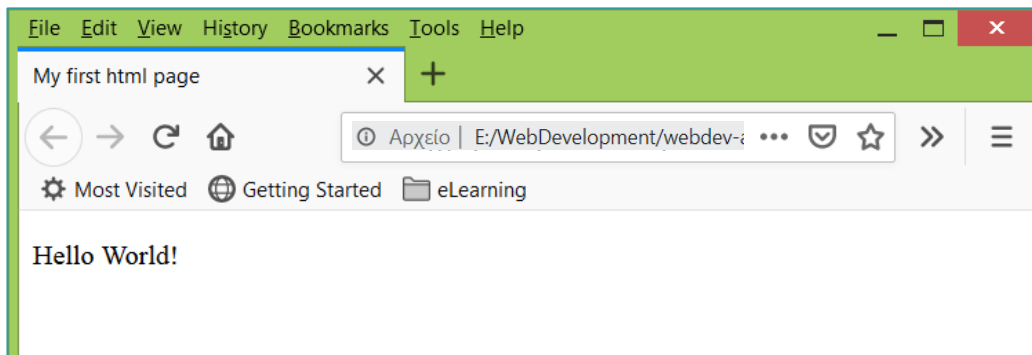
Αν δεν οριστεί charset, το default charset που χρησιμοποιούν οι browsers για να διαβάσουν είναι UTF-8. Επίσης, αν δεν οριστεί lang, η default lang είναι “en”

# Πρώτο πρόγραμμα – Hello World

## Το `<p>` element

```
<> 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>
```

- Μέσα στο body εισάγουμε το `<p>` element. Το `<p>` σημαίνει paragraph και το περιεχόμενο που δίνουμε μέσα στα `<p>` `</p>` εμφανίζεται στο browser window ως κείμενο.
- Αν δεν ορίσουμε εμείς συγκεκριμένη μορφοποίηση, 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



# 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>
```

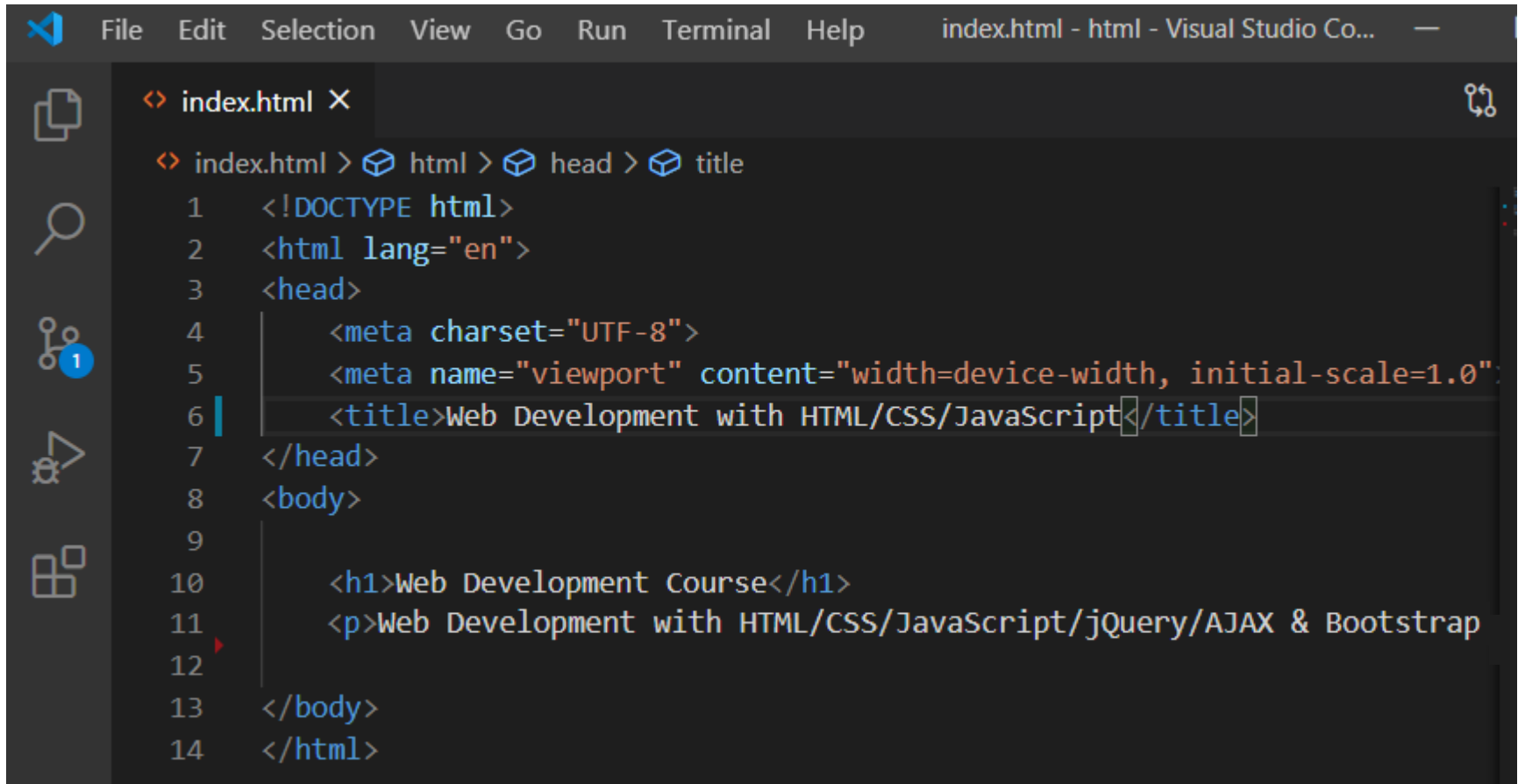
- Στην HTML5, Τα HTML tags δεν είναι case sensitive, δηλαδή δεν έχει σημασία αν τα γράφουμε με μικρά ή κεφαλαία γράμματα αλλά **συνίσταται ως καλή πρακτική να γράφουμε με πεζά**
- Στην XHTML πρέπει τα tags να τα γράφουμε με πεζά

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

# Βασικά HTML elements κειμένου

- Επικεφαλίδες
  - `<h1>` έως `<h6>`
  - `<h1>` Η επικεφαλίδα μου`</h1>`
- Παράγραφος
  - `<p>`
  - `<p>`Η παράγραφός μου`</p>`

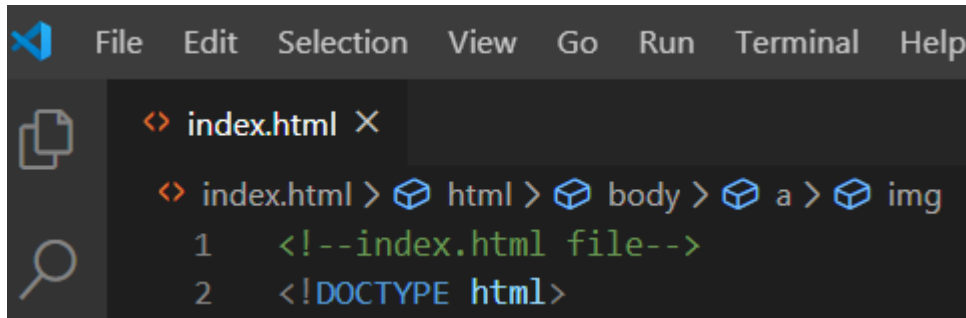
# Παράδειγμα – h1 & p elements



```
<> index.html X
<> index.html > html > head > title
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>Web Development with HTML/CSS/JavaScript</title>
7  </head>
8  <body>
9
10     <h1>Web Development Course</h1>
11     <p>Web Development with HTML/CSS/JavaScript/jQuery/AJAX & Bootstrap
12
13 </body>
14 </html>
```

- Στον κώδικα έχει προστεθεί ένα `<h1>` element που μορφοποιεί το κείμενο Web Development Course ως Επικεφαλίδα-1, δηλαδή με μέγεθος γραμματοσειράς διπλάσιο από το προεπιλεγμένο του browser, bold και μία κενή γραμμή πάνω και κάτω

# Σχόλια (Comments)



A screenshot of a code editor window. The title bar shows 'index.html' with a close button. The menu bar includes 'File', 'Edit', 'Selection', 'View', 'Go', 'Run', 'Terminal', and 'Help'. The breadcrumb navigation shows 'index.html > html > body > a > img'. The code editor displays two lines of HTML code: line 1 is `<!--index.html file-->` and line 2 is `<!DOCTYPE html>`.

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

Τα σχόλια χρησιμοποιούνται για  
εσωτερική τεκμηρίωση ή εξήγηση του  
κώδικα σε δύσκολα σημεία



# Κενά Διαστήματα / Αλλαγές γραμμής

## Κενή, Οριζόντια γραμμή

Ανάπτυξη Web Εφαρμογών

- 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>**

# 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** που προσδιορίζει την URL διεύθυνση. Για παράδειγμα:
  - `<a href="http://www.aueb.gr/">Οικ. Παν. Αθηνών</a>`

# Παράδειγμα – anchor element

```
<h1>Web Development Course</h1>
<p>Web Development with HTML/CSS/JavaScript/jQuery/AJAX & Bootstrap
  is the world's best course!</p>
<br>
<a href="https://elearning.aueb.gr/" target="_blank">Visit E-Learning AUEB</a>
```

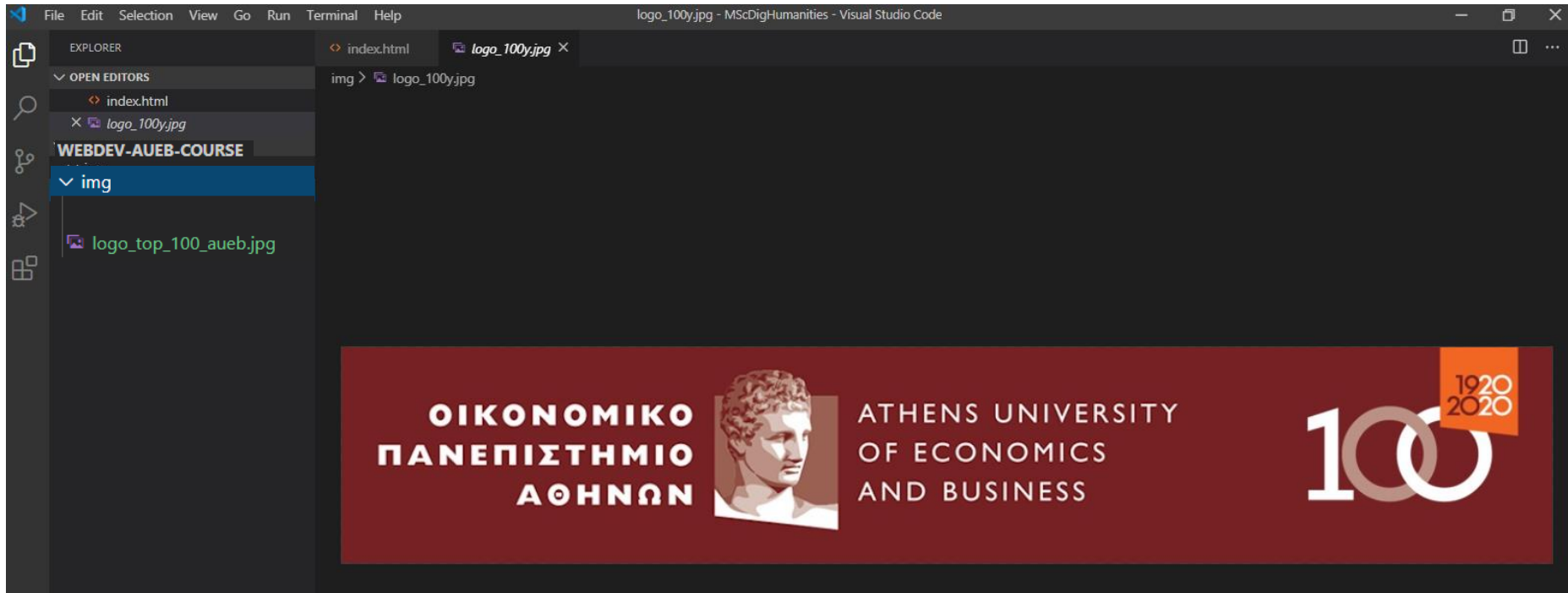
- `<a href="https://elearning.aueb.gr/" target="_blank">E-Learning AUEB</a>`
  - Ιδιότητα **href** (hypertext reference) -- ορίζει την URL διεύθυνση ή τοπικό σύνδεσμο (π.χ. `href="oranews.html"`)
  - Ιδιότητα **target="\_blank"** -- ανοίγει τον σύνδεσμο σε νέο παράθυρο
  - **Κείμενο συνδέσμου** -- προσδιορίζει το ορατό μέρος

- Εικόνες ορίζουμε με το **<img>** που είναι empty tag δηλαδή δεν έχει closing tag.
- Το **<img>** έχει τις ιδιότητες **src** που ορίζουμε τη θέση στο δίσκο της εικόνας, **alt** που ορίζει το 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.
  - ****
- Στην XHTML που είναι πιο αυστηρή σε θέματα σύνταξης της γλώσσας θα έπρεπε να κλείσουμε με **/>** το **<img>** tag και γενικά κάθε empty tag δηλ.
  - ****

# Εικόνες – Ιδιότητα source

- Ιδιότητα **src** (source). Αν δεν προσδιορίζεται φάκελος τότε υπονοείται ο ίδιος φάκελος που βρίσκεται το index.html, π.χ. `<img src= "opa.png" alt="OPA" />`
- Αλλιώς πρέπει να προσδιορίζεται ο φάκελος. Π.χ. αν το opa.png βρίσκεται στον υποφάκελο img, τότε γράφουμε:
- ``
- Το παραπάνω είναι ένα *relative link* όπου η τελεία συμβολίζει τον τρέχον φάκελο.
- 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 του web server

# Εικόνα



- 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

```

<h1>Web Development Course</h1>
<p>Web Development with HTML/CSS/JavaScript/jQuery/AJAX & Bootstrap
    is the world's best course!</p>
<br>
<a href="https://elearning.aueb.gr/" target="_blank">Visit E-Learning AUEB</a>
```

- Αν στο **width** εισάγουμε **100%** τότε το πλάτος της εικόνας θα ισούται με το πλάτος του viewport (δηλ. του browser window). Μπορούμε να εισάγουμε και συγκεκριμένη τιμή σε px. Αν στο **height** εισάγουμε **auto**, τότε το ύψος της εικόνας θα προσδιορίζεται αυτόματα με βάση το πλάτος της (διατηρώντας το aspect ratio σταθερό)
- Αυτό είναι ένα πρώιμο παράδειγμα fluid & responsive design. Αντί δηλαδή για απόλυτες τιμές pixels χρησιμοποιούμε ποσοστά με αποτέλεσμα να λειτουργεί σε όλα τα μεγέθη οθονών

# Image και Link

```
<a href="https://aueb.gr/"></a></a>
<h1>Web Development Course</h1>
<p>Web Development with HTML/CSS/JavaScript/jQuery/AJAX & Bootstrap
is the world's best course!</p>
<br>
<a href="https://elearning.aueb.gr/" target="_blank">Visit E-Learning AUEB</a>
```

- Αντί για κείμενο στο πρώτο <a> εισαγάγαμε το img



# Αποτέλεσμα



- `<table>`
  - `<tr>` γραμμή (table row)
  - `<th>` επικεφαλίδες στηλών (table header)
  - `<td>` δεδομένα-στήλη (table data)

# Παράδειγμα Πίνακα

```
index.html X
index.html > html > body > table > tr
18
19 <h1>Διδάσκοντες</h1>
20 <table>
21   <tr>
22     <th>Ονοματεπώνυμο</th>
23     <th>Διδασκόμενο Μάθημα</th>
24   </tr>
25   <tr>
26     <td>Ανδρούτσος Αθανάσιος</td>
27     <td>Web Development</td>
28   </tr>
29   <tr>
30     <td>Γιαννούτσου Άννα</td>
31     <td>Drupal</td>
32   </tr>
33   <tr>
34     <td>Καπέτης Μάκης</td>
35     <td>Βάσεις Δεδομένων</td>
36   </tr>
37 </table>
38
39 </body>
40 </html>
```



- Σε επόμενο στάδιο θα δούμε πως προσθέτουμε περιγράμματα στον πίνακα και στα κελιά του πίνακα

- Unordered lists: `<ul><li></li></ul>`
  - Τα `<li>` είναι list items
  - `<ul style="list-style-type:disc">` (circle, square, none)
- ordered lists: `<ol><li></li></ol>`
  - `<ol type="1">` (1, A, a, I, i)
- Description lists: `<dl><dt></dt><dd></dd></dl>`
  - Τα `<dt>`, `<dd>` είναι description term και description data αντίστοιχα

# Πολλά επίπεδα - Παράδειγμα

```
<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

```
index.html X
index.html > html > body > ol > li
34      <td>Καπέτης Μάκης</td>
35      <td>Βάσεις Δεδομένων</td>
36    </tr>
37  </table>
38
39  <h2>Υποχρεωτικά Μαθήματα</h2>
40  <ul>
41    <li>HTML/CSS</li>
42    <li>JavaScript</li>
43    <li>Bootstrap</li>
44    <li>Angular</li>
45  </ul>
46
47  <h2>Μαθήματα Επιλογής</h2>
48  <ol>
49    <li>Βάσεις Δεδομένων</li>
50    <li>Drupal</li>
51    <li>Προγραμματισμός με Python</li>
52    <li>Προγραμματισμός με Java</li>
53  </ol>
54
55  <h2>Λεξικό Όρων</h2>
56  <dl>
57    <dt>HTML/CSS</dt>
58    <dd>Hypertext Markup Language / Cascading Style
59    <dt>HTTP</dt>
60    <dd>Hypertext Transfer Protocol</dd>
61  </dl>
62
63  </body>
```

## Διδάσκοντες

Ονοματεπώνυμο	Διδασκόμενο Μάθημα
Ανδρούτσος Αθανάσιος	Web Development
Γιαννούτσου Άννα	Drupal
Καπέτης Μάκης	Βάσεις Δεδομένων

## Υποχρεωτικά Μαθήματα

- HTML/CSS
- JavaScript
- Bootstrap
- Angular

## Μαθήματα Επιλογής

1. Βάσεις Δεδομένων
2. Drupal
3. Προγραμματισμός με Python
4. Προγραμματισμός με Java

## Λεξικό Όρων

HTML/CSS

Hypertext Markup Language / Cascading Style Sheets

HTTP

Hypertext Transfer Protocol

# HTML Forms (1/3)

- Οι φόρμες χρησιμοποιούνται για **εισαγωγή δεδομένων των χρηστών**
- `<form> </form>`
- **Οι φόρμες αποτελούνται από αντικείμενα ελέγχου (controls)**
  - *Label (ετικέτα)* – Εκτύπωση κειμένου
  - *Text Area (Εκτύπωση κειμένου σε γραμμές και στήλες).* π.χ.  
`<textarea name="details" rows="5" cols="50">Δώστε το επώνυμό σας</textarea>`
  - *Text box (Εισαγωγή κειμένου)* – π.χ.  
`<input type="text" name="lastname" />`
  - *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* – Για αποστολή δεδομένων φόρμας  
`<input type="submit" value="Πατήστε Click" />`
- *Button* - `<input type="button" onclick="alert('AUEB!') " />`



# HTML Forms (3/3)

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

# Σχεδιασμός (1)

- Πολλές φορές είναι χρήσιμο να **σχεδιάζουμε**, πριν υλοποιήσουμε, ιδιαίτερα όσο αφορά γραφικές διεπαφές, ώστε:
  1. Να έχουμε το feedback και τελικά τη συναίνεση των χρηστών (έτσι ώστε να μην προχωρήσουμε στην υλοποίηση και μετά μας πουν οι χρήστες ότι περίμεναν να δουν κάτι άλλο, οπότε τότε το κόστος για εμάς θα είναι μεγάλο)
  2. Να έχουμε κι εμείς μία οπτική αναπαράσταση του πως θα φαίνεται αυτό που πρόκειται να υλοποιήσουμε, ώστε στη συνέχεια να το υλοποιήσουμε ευκολότερα και ταχύτερα

# Σχεδιασμός (2)

- Τα σχέδια αυτά ονομάζονται wireframes ή mockups
- Τα wireframes αναπαριστούν βασικά τη δομή της σελίδας
- Τα mock-ups μπορεί να περιέχουν πιο πλούσια οπτική αναπαράσταση, πιο κοντά στο τελικό αποτέλεσμα

# Σχεδιασμός Φόρμας (1)

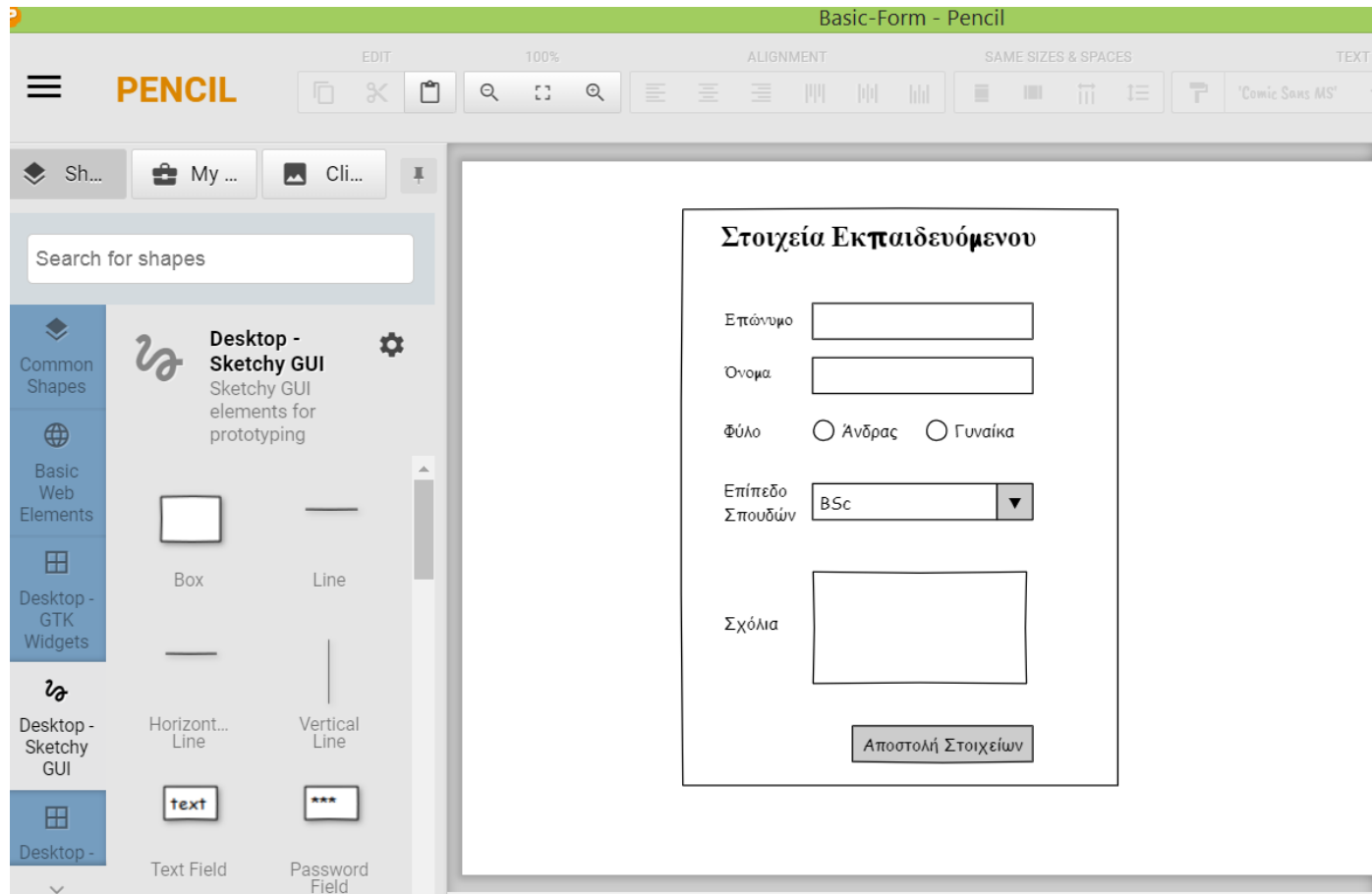
- Η διαδικασία του σχεδιασμού ισχύει σε όλες τις διαδικασίες, κατά κύριο λόγο όμως όπως αναφέραμε στη γραφικές διεπαφές
- Θα δούμε ένα πρώτο παράδειγμα με το σχεδιασμό μιας φόρμα εισαγωγικής στοιχείων Εκπαιδευομένων, αλλά η ίδια διαδικασία μπορεί να ακολουθηθεί για ολόκληρες web σελίδες και web sites

# Σχεδιασμός Φόρμας (2)

- Θα μπορούσαμε να σχεδιάσουμε σε μια κόλλα χαρτί, αλλά ωστόσο υπάρχουν χρήσιμα εργαλεία για να δημιουργήσουμε wireframes
- Στη συνέχεια θα χρησιμοποιήσουμε ένα απλό δωρεάν εργαλείο, το Pencil, που μπορείτε να κατεβάσετε από τον ακόλουθο σύνδεσμο:

<https://pencil.evolus.vn/>

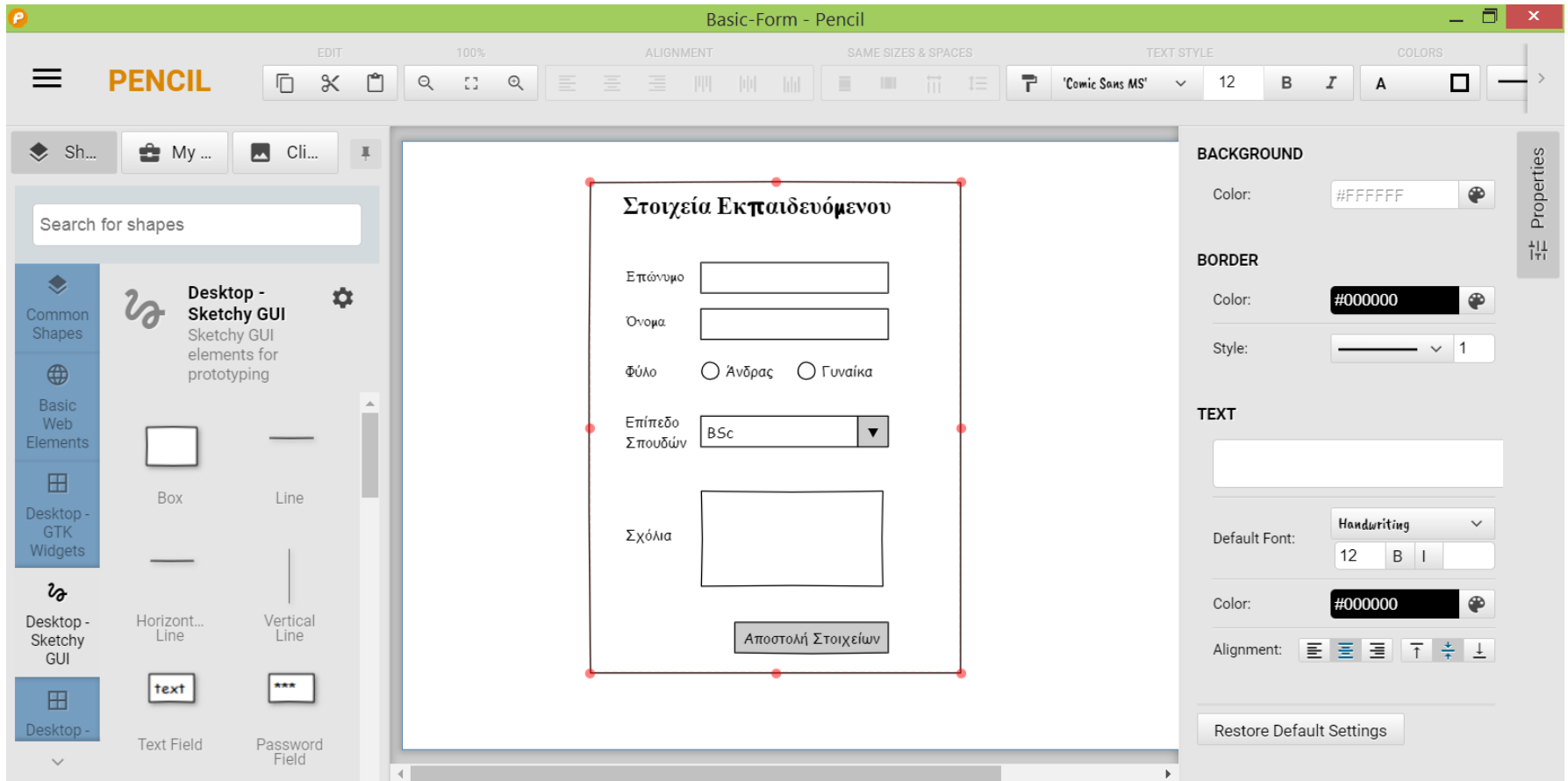
# Pencil (1)



- Με drag & drop μεταφέρουμε αντικείμενα από τις εργαλειοθήκες αριστερά, στον καμβά
- Για φόρμες χρησιμοποιούμε κυρίως την Desktop – Sketchy GUI και την Basic Web Elements

Σχεδιάζουμε την παραπάνω φόρμα την οποία θα υλοποιήσουμε στη συνέχεια

# Pencil (2)



- Κάθε αντικείμενο μπορούμε να το επιλέξουμε και στη συνέχεια στο δεξί μέρος να επεξεργαστούμε, αν θέλουμε, τις ιδιότητές του και κυρίως το κείμενο που θα εμφανίζεται.

# Επεξήγηση αντικειμένων

## Στοιχεία Εκπαιδευόμενου

Επώνυμο

Όνομα

Φύλο ☐ Άνδρας ☐ Γυναίκα

Επίπεδο Σπουδών  ▼

Σχόλια

- Πριν περάσουμε στην υλοποίηση να εξηγήσουμε ότι μια φόρμα αποτελείται συνήθως από ζεύγη: Label / Input
- Στο παράδειγμα έχουμε για το Επώνυμο και το Όνομα, ένα **Label** και ένα **Textbox**, αντίστοιχα.
- Τα Labels χρησιμοποιούνται ως απλό κείμενο, 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 βασικά dropdown lists, όπως π.χ. το Επίπεδο Σπουδών και το **Textarea** που είναι ένα μεγάλο textbox
- Βασικά controls είναι επίσης τα **buttons**, στα οποία μπορούμε να κάνουμε κλικ και να εκτελεστεί κώδικας και να γίνει κάποια ενέργεια



# Παράδειγμα απλής φόρμας

```
10 <h1>Στοιχεία Εκπαιδευομένου</h1>
11
12 <form>
13   <label for="lastname">Επώνυμο</label>
14   <input type="text" id="lastname"><br>
15   <label for="firstname">Όνομα</label>
16   <input type="text" id="firstname"><br>
17   <label for="">Φύλο</label>
18   <input type="radio" id="male" name="gender" checked><label for="male">Ανδρας</label>
19   <input type="radio" id="female" name="gender"><label for="female">Γυναίκα</label><br>
20   <label for="levelOfStudies">Επίπεδο Σπουδών</label>
21   <select id="levelOfStudies">
22     <option value="BSc">BSc</option>
23     <option value="MSc">MSc</option>
24     <option value="PhD">PhD</option>
25   </select><br>
26   <label for="comments">Σχόλια</label>
27   <textarea id="comments" cols="30" rows="10"></textarea><br>
28   <button type="submit">Αποστολή Στοιχείων</button>
29 </form>
30
31 <button type="button" onclick="console.log('Hello World!')">Check Submission</button>
32
33 </body>
34 </html>
```

- Το for στο label σχετίζεται με το id στο control
- Έτσι αυτά τα δύο: *label* και *control* είναι συνδεδεμένα και ομαδοποιημένα
- Αν πατήσουμε κλικ πάνω στο 'Ανδρας για παράδειγμα ενεργοποιείται το radio button

# Παράδειγμα φόρμας με πίνακα

```
9 <h1>Στοιχεία Εκπαιδευομένου</h1>
10 <form>
11   <table>
12     <tr>
13       <td><label for="lastname">Επώνυμο</label></td>
14       <td><input type="text" id="lastname"></td>
15     </tr>
16     <tr>
17       <td><label for="firstname">Όνομα</label></td>
18       <td><input type="text" id="firstname"></td>
19     </tr>
20     <tr>
21       <td><label for="">Φύλο</label></td>
22       <td><input type="radio" id="male" name="gender" checked=""><label for="male">Ανδρας</label>
23         <input type="radio" id="female" name="gender"><label for="female">Γυναίκα</label></td>
24     </tr>
25     <tr>
26       <td><label for="levelOfStudies">Επίπεδο Σπουδών</label></td>
27       <td><select id="levelOfStudies">
28         <option value="BSc">BSc</option>
29         <option value="MSc">MSc</option>
30         <option value="PhD">PhD</option>
31       </select></td>
32     </tr>
33     <tr>
34       <td><label for="comments">Σχόλια</label></td>
35       <td><textarea id="comments" cols="30" rows="10"></textarea></td>
36     </tr>
37     <tr>
38       <td></td>
39       <td><button type="submit">Αποστολή Στοιχείων</button></td>
40     </tr>
41   </table>
42 </form>
43 <br>
44 <hr>
45 <button type="button" onclick="console.log('Hello World!')">Check Submission</button>
```

- Χρήση πίνακα για στοίχιση των στοιχείων της φόρμας
- Το πρόβλημα με τους πίνακες είναι πως δεν είναι responsive αλλά εδώ το πλάτος είναι μικρό και φαίνεται και σε μικρές οθόνες χωρίς πρόβλημα

# Παράδειγμα φόρμας (2)

← → ↻ ⓘ Αρχείο | E:/WebDevelopment/w... ☆

## Στοιχεία Εκπαιδευομένου

Επώνυμο

Όνομα

Φύλο ☒ Άνδρας ☐ Γυναίκα

Επίπεδο Σπουδών

Σχόλια

Αποστολή Στοιχείων

Check Submission

- Η φόρμα εμφανίζεται όπως την έχουμε σχεδιάσει

- Θέλουμε να **εμφανίζονται μηνύματα λάθους**
  - Αν εισάγουμε δεδομένα σε μορφή μη-έγκυρη, ή
  - Αν δεν εισάγουμε ενώ είναι υποχρεωτικό να εισάγουμε

# Client-Side Validation (1)

- Client-side validation, εν αντιθέσει με την server-side validation
- Για παράδειγμα ο browser ελέγχει ένα e-mail ότι είναι έγκυρο αλλά ο server μπορεί να το απορρίψει γιατί υπάρχει ήδη στη Βάση Δεδομένων

# Client-Side Validation (2)

```
12 <form>
13   <table>
14     <tr>
15       <td><label for="lastname">Επώνυμο</label></td>
16       <td><input type="text" id="lastname" maxlength="50" required pattern="[a-zA-ZΑ-Ω]+"></td>
17     </tr>
18     <tr>
19       <td><label for="firstname">Όνομα</label></td>
20       <td><input type="text" id="firstname" maxlength="50" required pattern="[a-zA-ZΑ-Ω]+"></td>
21     </tr>
22     <tr>
23       <td><label for="birthdate">Ημερομηνία Γέννησης</label></td>
24       <td><input type="date" id="birthdate" maxlength="10" max="2002-01-01"></td>
25     </tr>
```

- **required** πρέπει υποχρεωτικά να δοθεί – αν δεν δοθεί βγάζει μήνυμα λάθους
- **pattern** – μπορούν να δοθούν συγκεκριμένοι συνδυασμοί χαρακτήρων. Στο παράδειγμα στο Επώνυμο μπορούν να δοθούν μόνο αγγλικά πεζά ή κεφαλαία ή ελληνικά κεφαλαία μία ή περισσότερες φορές
- **minlength / maxlength** – περιορίζει το μήκος του εισαγόμενου αλφαριθμητικού

# Client-Side Validation (3)


```
26      <tr>
27      |      <td><label for="yearOfStudies">Έτος Σπουδών</label></td>
28      |      <td><input type="number" id="yearOfStudies" min="1" max="6"></td>
29      |      </tr>
30      |      <tr>
31      |      |      <td><label for="login">Login(e-mail)</label></td>
32      |      |      <td><input type="email" id="login" required></td>
33      |      |      </tr>
34      |      |      <tr>
35      |      |      |      <td><label for="password">Password</label></td>
36      |      |      |      <td><input type="password" id="password" required minlength="8"></td>
37      |      |      |      </tr>
38      |      |      |      <tr>
39      |      |      |      |      <td><button type="submit">Αποστολή</button></td>
40      |      |      |      |      <td></td>
41      |      |      |      |      </tr>
42      |      |      |      </table>
43      |      </form>
```

- **type = "e-mail"** – ελέγχει αν το αλφαριθμητικό είναι έγκυρη διεύθυνση e-mail (για παράδειγμα να περιέχει @)
- **type = "password"** – εμφανίζει το αλφαριθμητικό με κουκίδες
- **min / max** – περιορίζει το εύρος τιμών της εισαγόμενης τιμής

# Client-Side Validation (4)

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

## Στοιχεία Εκπαιδευόμενου

Επώνυμο:	<input type="text" value="ΑΝΔΡΟΥΤΣΟΣ"/>
Όνομα:	<input type="text" value="Γεώργιος"/>
Ημερομηνία Γέννησης:	<input type="text" value="27/08/2002"/> 
Έτος Σπουδών:	<input type="text" value="2"/>
Login (e-mail):	<input type="text" value="a8anassis@gmail.com"/>
Password:	<input type="password" value="....."/>
<input type="button" value="Αποστολή"/>	


- Αν δώσουμε έγκυρα στοιχεία δεν εμφανίζεται κανένα μήνυμα



# Μηνύματα Λάθους (1)

## Στοιχεία Εκπαιδευόμενου


Επώνυμο:	<input type="text" value="Ανδρούτσος"/>
Όνομα:	<input type="text" value="Γ"/>
Ημερομηνία Γέννησης:	<input type="text" value="2"/>
Έτος Σπουδών:	<input type="text" value="2"/>
Login (e-mail):	<input type="text" value="a8anassisgmail.com"/>
Password:	<input type="password" value="....."/>
<input type="button" value="Αποστολή"/>	

 Αντιστοιχίστε τη ζητούμενη μορφή.

- Αν δεν δώσουμε Αγγλικά ή κεφαλαία ελληνικά, εμφανίζει μήνυμα λάθους

# Μηνύματα Λάθους (2)

## Στοιχεία Εκπαιδευόμενου

Επώνυμο:	<input type="text" value="ΑΝΔΡΟΥΤΣΟΣ"/>
Όνομα:	<input type="text" value="Γεώργιος"/>
Ημερομηνία Γέννησης:	<input type="text" value="27/08/2002"/> 
Έτος Σπουδών:	<input type="text" value="2"/>
Login (e-mail):	<input type="text" value="a8anassisgmail.com"/>



Συμπεριλάβετε το σύμβολο "@" στη διεύθυνση ηλεκτρονικού ταχυδρομείου. Από τη διεύθυνση "a8anassisgmail.com" λείπει το σύμβολο "@".

- Αν δεν δώσουμε το @ στην e-mail διεύθυνση εμφανίζεται μήνυμα λάθους

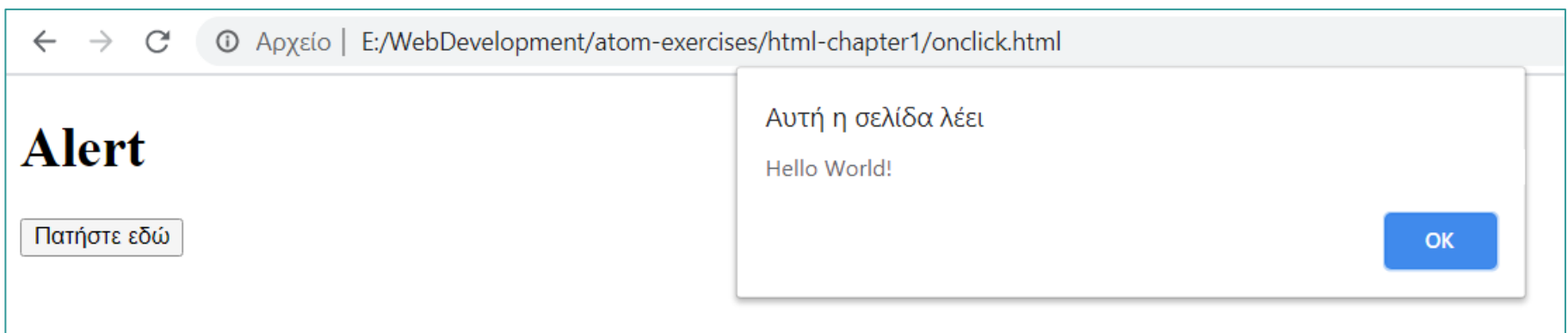
# Buttons

- Buttons τύπου είτε *button* ή *submit*
- Το `type="button"` που είναι και το default μπορεί να έχει την ιδιότητα `onclick` που να εκτελείται μία ενέργεια όταν πατηθεί
- Το `type="submit"` όταν πατηθεί αποστέλλει όλα τα πεδία της φόρμας στον server (αν κάνουμε validation πριν αποστείλει, ελέγχει την εγκυρότητα)

# onclick

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title></title>
6   </head>
7   <body>
8
9     <h1>Alert</h1>
10    <button type="button" onclick="alert('Hello World!')">Πατήστε εδώ</button>
11
12  </body>
13 </html>
```

- Όταν ο χρήστης πατήσει το button, τότε δημιουργείται ένα event, που ονομάζεται onclick και τότε έχουμε ορίσει με μία εντολή απλή JavaScript να εμφανίζεται ένα pop-up alert window.



# Βίντεο Κεφαλαίου

1. Intro & VSC
2. Basic HTML Elements
3. Εικόνες / BR / HL
4. Λίστες / Πίνακες
5. Design & Pencil
6. Φόρμες / Events
7. Φόρμες και Πίνακες
8. Form Validation