



Site Layout Design Heuristics

Αθανάσιος Ανδρούτσος



Page Layout (1)

- Λέγοντας **Page Layout** εννοούμε δύο πράγματα
 - Τη **συνολική δομή της σελίδας**, όπως την ζωγραφίζαμε σε ένα χαρτί. Για παράδειγμα αν θα περιλαμβάνει ένα λογότυπο, ένα μενού, σε ποιο σημείο της σελίδας θα βρίσκονται, πόσο μεγάλα θα είναι, αν θα είναι στοιχισμένα, κλπ..
 - Το **styling των HTML στοιχείων** (το πραγματικό περιεχόμενο) που βρίσκεται μέσα (*nested*) στην πρωταρχική δομή της σελίδας

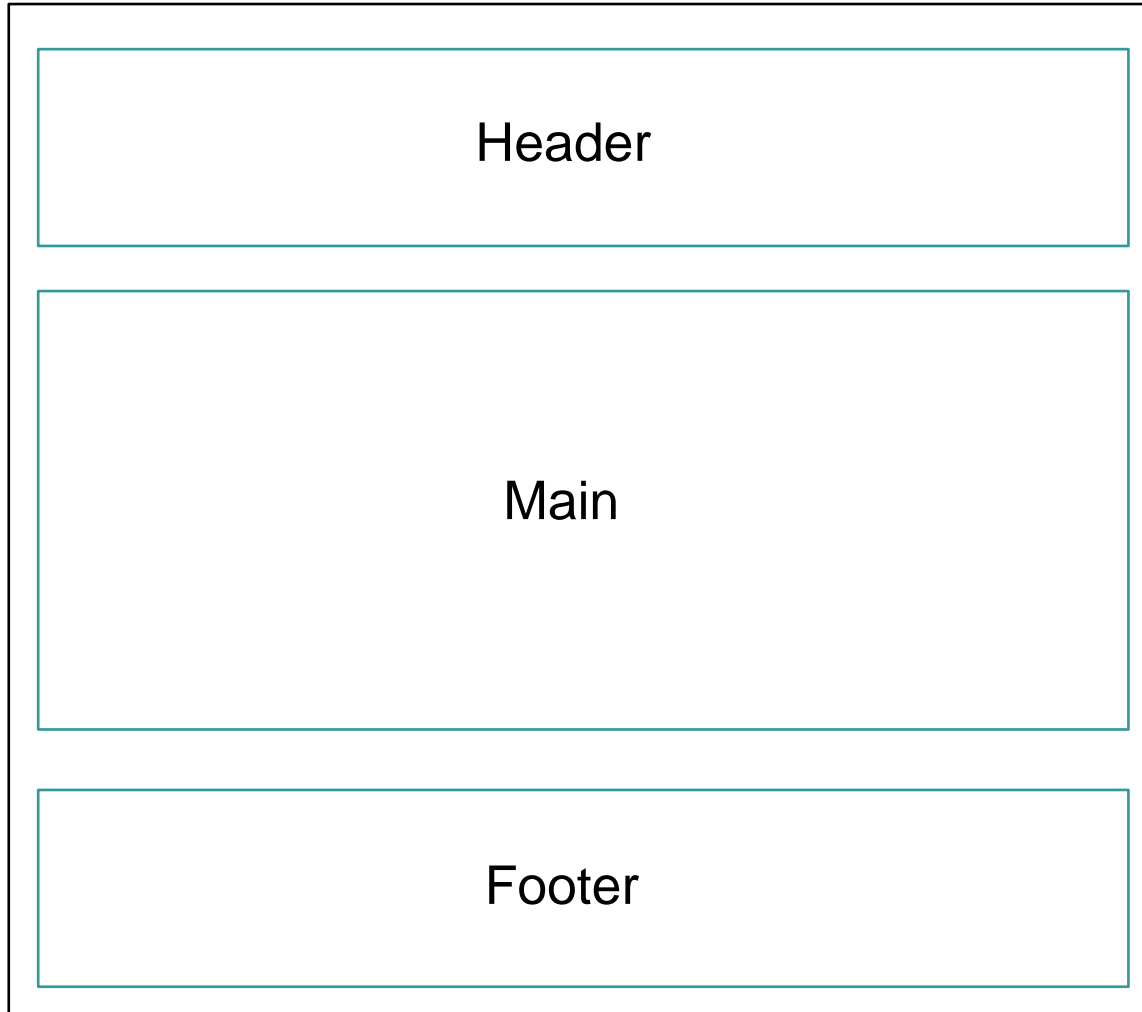


Page Layout (2)

- Η βασική ιδέα του Layout είναι το nesting (στοιχεία μέσα σε boxes)
- Σε πιο περίπλοκα Layouts μπορεί να έχουμε περισσότερα επίπεδα nesting (στοιχεία μέσα σε boxes, μέσα άλλα boxes, κλπ.)
- Επομένως η βασική ιδέα για την δημιουργία layouts είναι πως όλα είναι ένα κουτί (box / container) μέσα σε ένα άλλο κουτί. μέσα σε ένα άλλο κουτί, κλπ.



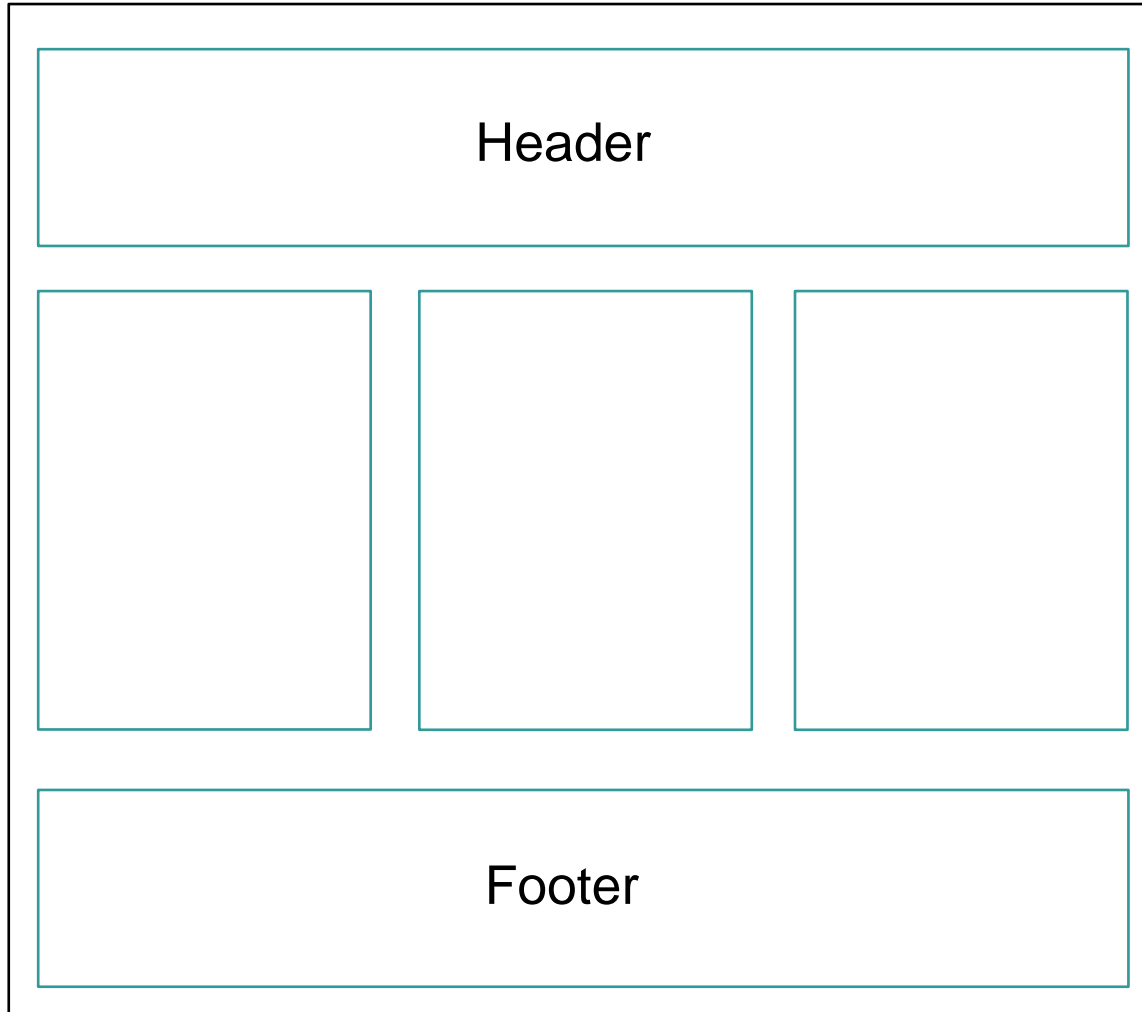
Τυπικά Layout (1)



- Ένα τυπικό Layout μπορεί να περιλαμβάνει ένα **Header** (που μπορεί με τη σειρά του να περιλαμβάνει ένα λογότυπο / ένα μενού), ένα Main section που μπορεί να περιλαμβάνει άλλα υπό-sections με περιεχόμενο και άλλα στοιχεία HTML και ένα Footer που μπορεί να περιλαμβάνει επίσης άλλα στοιχεία, όπως πληροφορίες copyright, social, κλπ.



Τυπικό Layout (1)



- Ένα άλλο layout μπορεί να περιλαμβάνει περισσότερα από ένα columns, π.χ. τρία columns



Οριζόντια Στοίχιση

Προγραμματισμός στο Web

- Ένα πρόβλημα που ανακύπτει κατά την υλοποίηση Layouts είναι η οριζόντια στοίχιση των ‘κουτιών’
- Μιας και όπως γνωρίζουμε το normal flow των div είναι κάθετο -καθότι τα div είναι block elements- η βασική στοίχιση των div είναι σε μία διάσταση, την κάθετη, δηλαδή το ένα div κάτω από το άλλο



Floats

- Κάτι τέτοιο μας περιορίζει σε μία διάσταση και δημιουργεί το πρόβλημα του πως να δημιουργήσουμε Layouts και στις δύο διαστάσεις, πως να στοιχίσουμε δηλαδή τα 'κουτιά' και οριζόντια
- Τα Floats είναι ιστορικά η πρώτη λύση που δόθηκε στο πρόβλημα αυτό και παρότι τα Floats έχουν αντικατασταθεί πλέον από το Flexbox, υπάρχει μεγάλο codebase μιας και για πάνω από μία δεκαετία υπήρξαν ο βασικός τρόπος σχεδιασμού Layouts



Layout - HTML

Προγραμματισμός στο Web

<> floats.html X

testbed > <> floats.html > ...

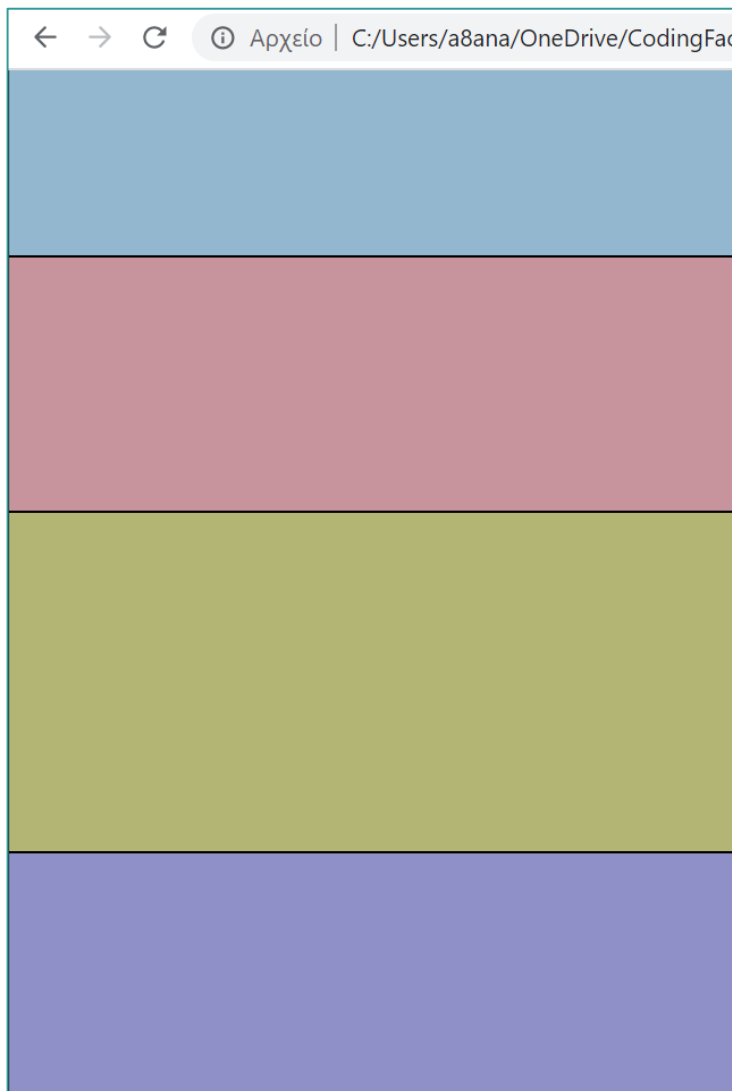
```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1">
7      <title>Floats</title>
8      <link rel="stylesheet" href="./css/floats.css">
9  </head>
10 <body>
11     <div class="container">
12         <div class="header"></div>
13         <div class="aside"></div>
14         <div class="main"></div>
15         <div class="footer"></div>
16     </div>
17 </body>
18 </html>
```

- Έστω μία σελίδα που αποτελείται από ένα header, ένα aside που είναι συνήθως ένα section (π.χ. sidebar) που σχετίζεται έμμεσα με το main section και ένα footer
- Όλα αυτά είναι container div μέσα στα οποία μπορούμε να βάλουμε περιεχόμενο



Layout – CSS (1)

```
1 * {  
2   box-sizing: border-box;  
3   margin: 0;  
4   padding: 0;  
5 }  
6  
7 .header {  
8   height: 120px;  
9   border: 1px solid black;  
10  background-color: rgb(143, 182, 206);  
11 }  
12  
13 .aside {  
14   height: 150px;  
15   border: 1px solid black;  
16   background-color: rgb(203, 148, 157);  
17 }  
18  
19 .main {  
20   height: 200px;  
21   border: 1px solid black;  
22   background-color: rgb(178, 182, 118);  
23 }  
24  
25 .footer {  
26   height: 150px;  
27   border: 1px solid black;  
28   background-color: rgb(143, 143, 199);  
29 }
```



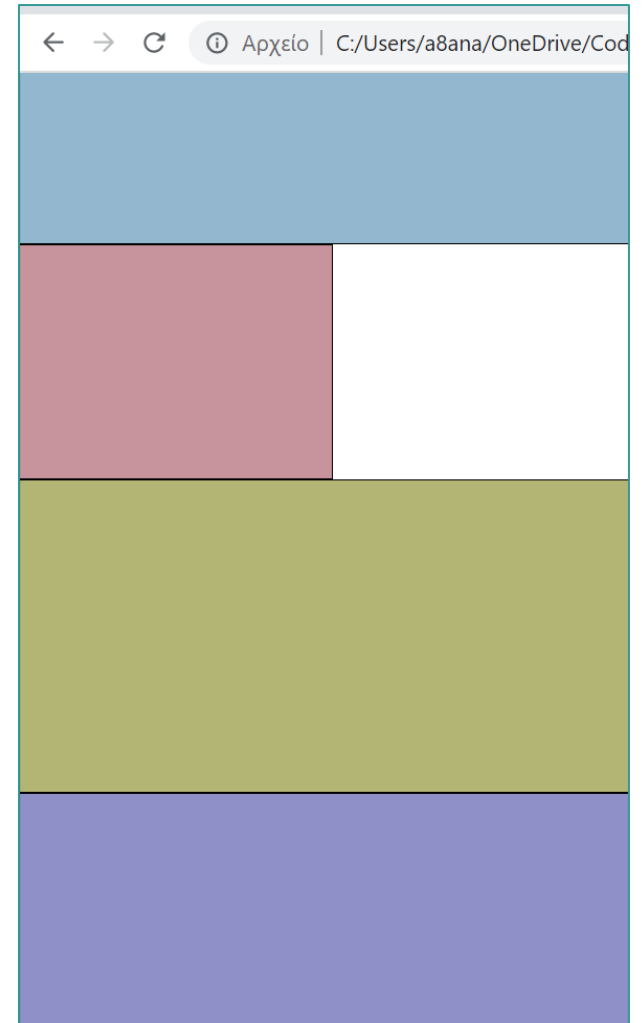
- Όπως βλέπουμε τα div στοιχίζονται το ένα κάτω από το άλλο



Layout – CSS (2)

```
.aside {  
  height: 150px;  
  width: 200px;  
  border: 1px solid black;  
  background-color: rgb(203, 148, 157);  
}
```

- Προσθέτοντας στο aside, πλάτος (width) δεν αλλάζει κάτι στο normal flow. Το main section συνεχίζει και είναι από κάτω

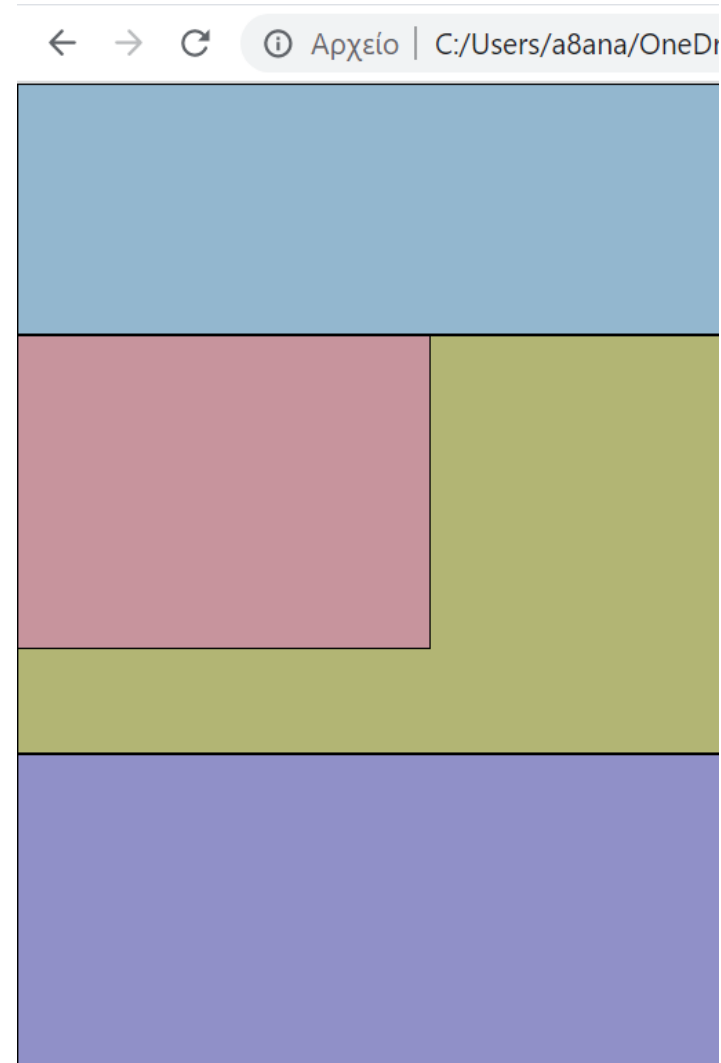




Float (1)

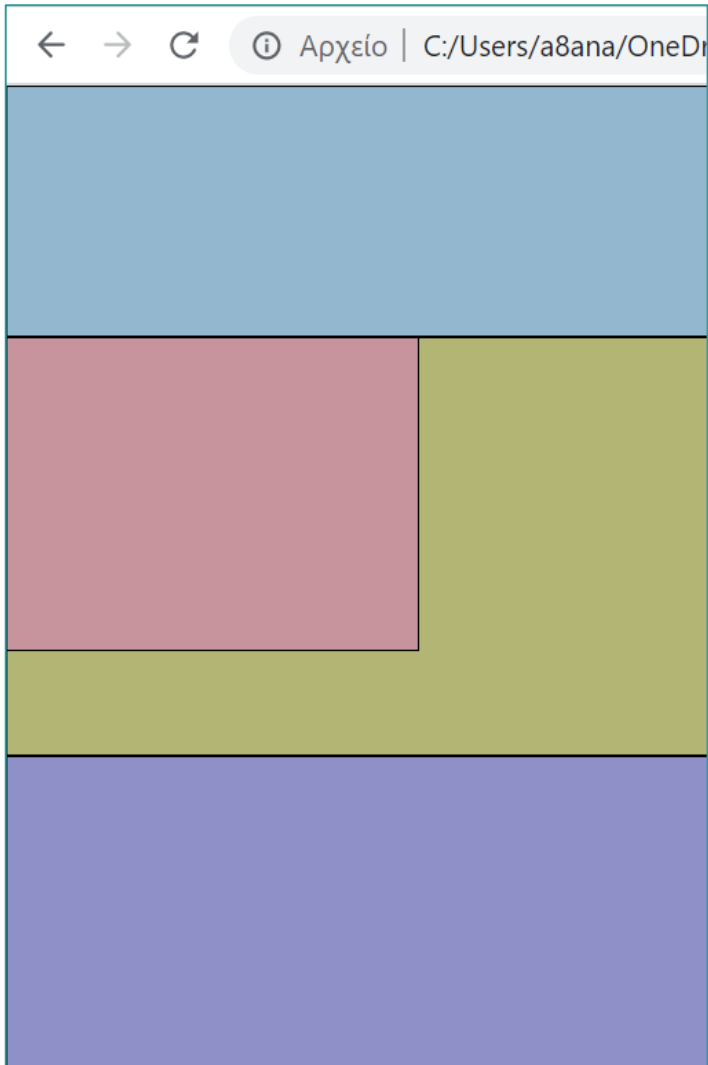
```
13 .aside {  
14     float: left;  
15     height: 150px;  
16     width: 200px;  
17     border: 1px solid black;  
18     background-color: #c99999;  
19 }
```

- Δίνοντας `float: left` στο `aside`, ουσιαστικά **βγάζουμε το `aside` από το `normal flow`** και το τοποθετούμε αριστερά **σε σχέση με το `parent element` (τον `container`)** ενώ τα άλλα στοιχεία συνεχίζουν να υπάρχουν στο `normal flow`
- Επομένως ένα `block element` όπως ο `container` ορίζει ένα `Block Formatting Context (BFC)` μέσα στο οποίο έχουν `effect` τα `floats`, που περιέχει. Δεν επηρεάζονται τα στοιχεία έξω από το `BFC`





Float (2)



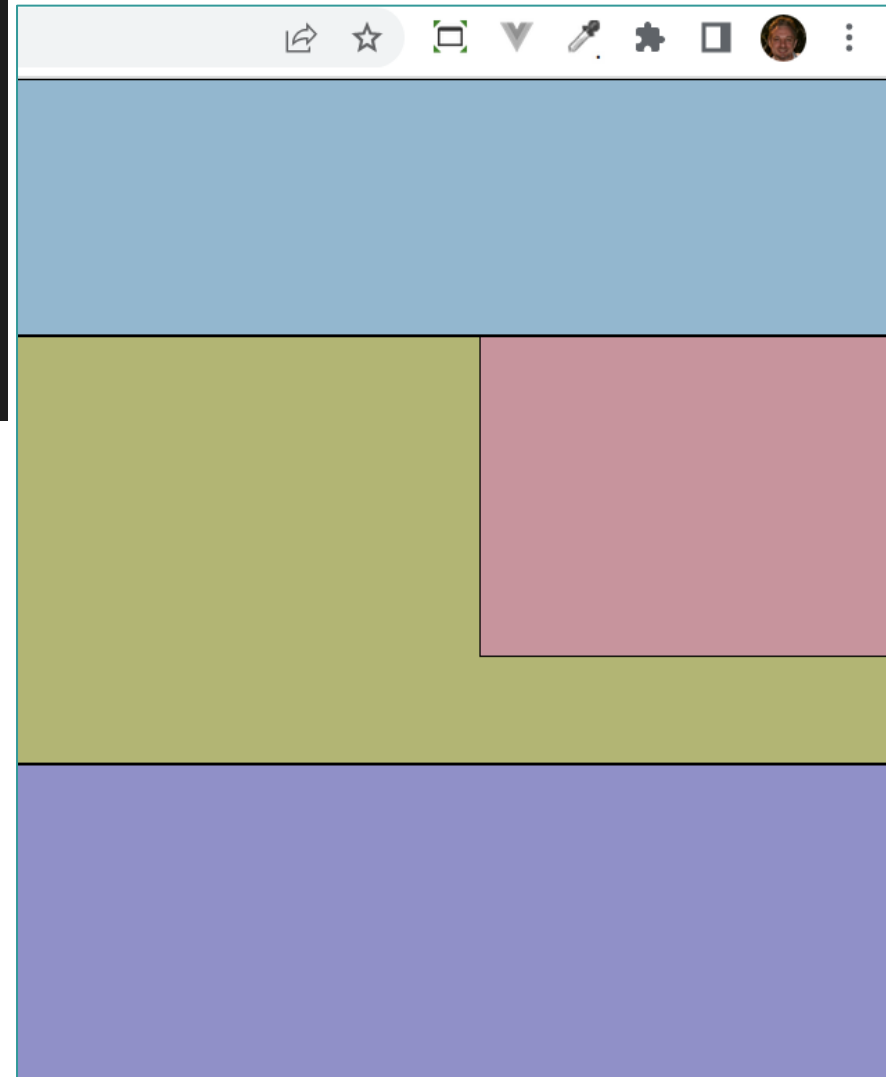
- Βγάζοντας το aside από το normal flow, και ενώ τα άλλα στοιχεία παραμένουν στο normal flow, το aside εμφανίζεται πάνω από το main, στο foreground



Float (3)

```
13 .aside {  
14     float: right;  
15     height: 150px;  
16     width: 200px;  
17     border: 1px solid black;  
18     background-color: rgb(203, 148, 157);  
19 }
```

- Το ίδιο μπορούμε να κάνουμε και με `float: right`
- Ενώ με `float: none` βγαίνουμε από το float





Στοιίχιση

- Επομένως μπορούμε πλέον να στοιχίσουμε τα boxes (div, outer box) αριστερά, δεξιά με float, και στο κέντρο με auto-margins
- Ενώ στο inner box (του div) μπορούμε να στοιχίσουμε κείμενο και inline boxes με text-align στον parent
- Τα δε εσωτερικά block elements πάλι μπορούν να στοιχηθούν όπως τα εξωτερικά boxes με float και auto-margins

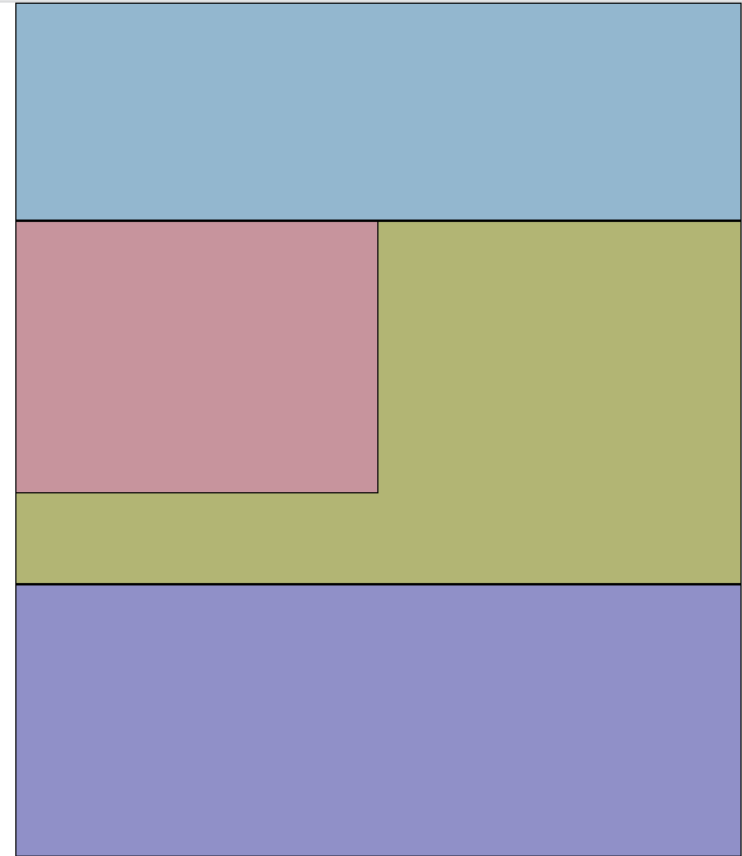


Parent Element

```
7  .container {  
8      width: 400px;  
9      margin-left: auto;  
10     margin-right: auto;  
11 }
```

- Όπως αναφέραμε και βλέπουμε και εδώ τα floats στοιχίζονται σε σχέση με το parent element

rive/CodingFactory-REBOOT/webprojects/coding-factory/testbed/floats.html





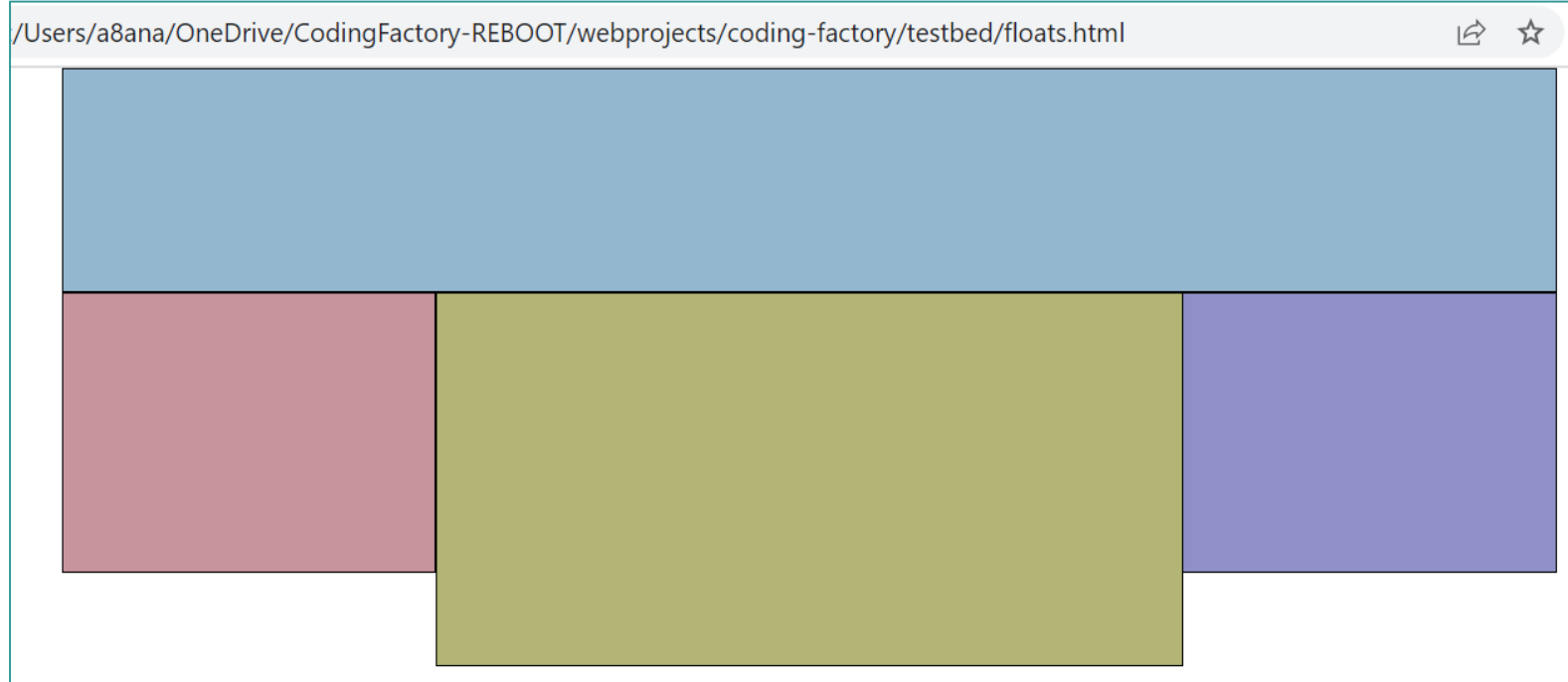
Float floats over main



- Παρατηρούμε επίσης πως το float element βρίσκεται πάνω από το main section
- Κάτι τέτοιο θα ήταν επιθυμητό για magazine-style layout
- Για το συγκεκριμένο Layout ωστόσο θα θέλαμε το main να είναι δίπλα-δεξιά από το aside



Float, main float over footer



```
19 .aside {  
20     float: left;  
21     height: 150px;  
22     width: 200px;  
23     border: 1px solid black;  
24     background-color: rgb(203, 148, 157);  
25 }
```

- Κάνοντας το main float: left, βγαίνει και το main από το normal flow, πάει αριστερά και μένει μόνο το header και το footer στο normal flow



Συνδυασμοί left-right

Προγραμματισμός στο Web

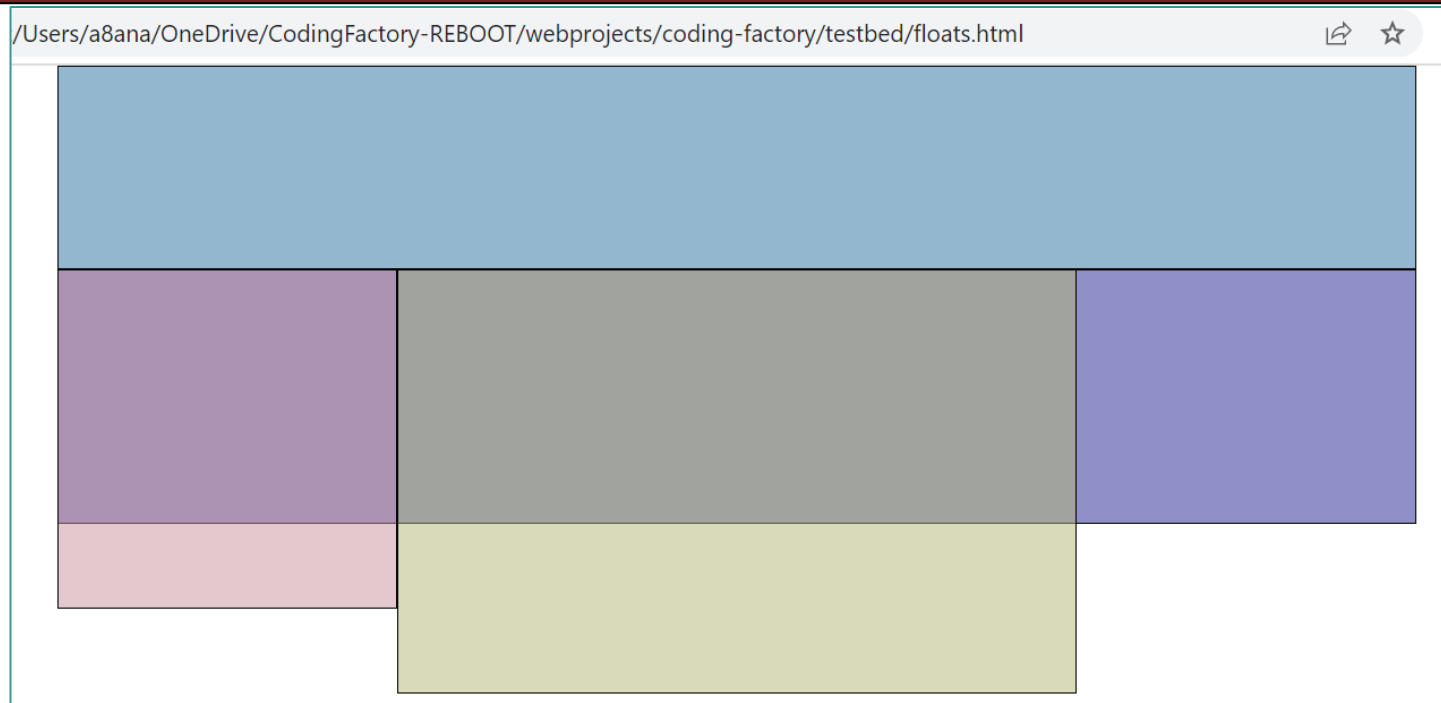


```
19 .aside {
20     float: right;
21     height: 150px;
22     width: 200px;
23     border: 1px solid black;
24     background-color: rgb(203, 148, 157);
25 }
26
27 .main {
28     float: left;
29     height: 200px;
30     width: 400px;
31     border: 1px solid black;
32     background-color: rgb(178, 182, 118);
33 }
```

- Μπορούμε να κάνουμε διάφορους συνδυασμούς left και right μεταξύ aside και main (left/left, right/right, left/right, right/left)



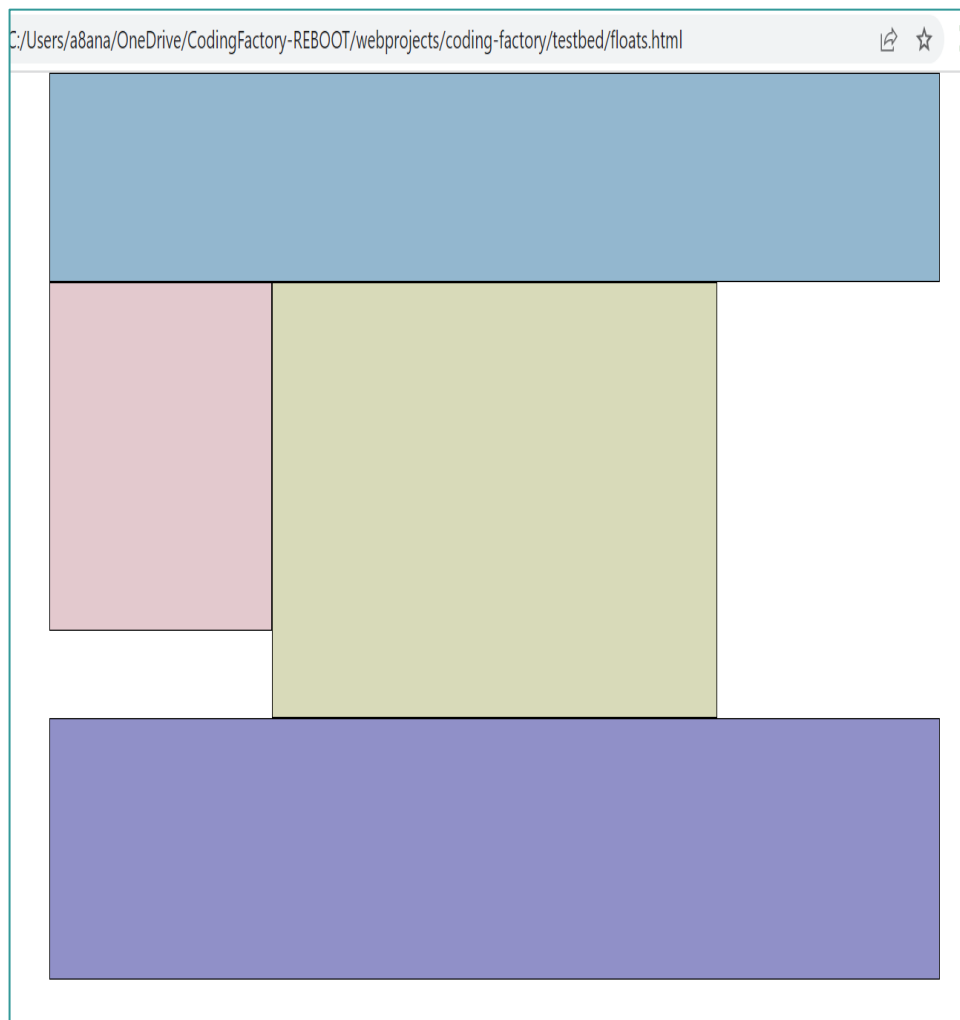
Layout



- Βάζοντας `opacity` παρατηρούμε καλύτερα ότι τα `aside` και `main` είναι πάνω στο `footer`



Clear Float



```
.footer {  
  clear: both;  
  height: 150px;  
  border: 1px solid black;  
  background-color: rgb(143, 143, 199);  
}
```

- Κάνοντας `Clear: both` (ή `clear: left`, ή `clear: right`) προστάζουμε το block να μετακινηθεί κάτω από κάθε προηγούμενο floating element
- Δημιουργεί δηλαδή ένα νέο block formatting context



Full-bleed layout (1)

```
<body>
  <div class="header"></div>

  <div class="container">
    <div class="aside"></div>
    <div class="main"></div>
  </div>

  <div class="footer"></div>
</body>
</html>
```

- Σε ένα άλλο τύπο Layout μπορεί να έχουμε επίσης τα float elements μέσα σε ένα container element



Full-bleed layout (2)

```
13  .container {
14      width: 800px;
15      margin-left: auto;
16      margin-right: auto;
17      border: 2px solid red;
18  }
19
20  .aside {
21      float: left;
22      height: 200px;
23      width: 200px;
24      border: 1px solid black;
25      background-color: rgb(203, 148, 157, 0.5);
26  }
27
28  .main {
29      float: left;
30      height: 250px;
31      width: 400px;
32      border: 1px solid black;
33      background-color: rgb(178, 182, 118, 0.5);
34  }
```

- Βάζουμε κόκκινο background στο container ώστε να δούμε το πλαίσιο στο οποίο εκτείνεται



Full-bleed layout (3)



- Παρατηρούμε ότι ο container έχει μηδενικό ύψος μιας και τα δύο στοιχεία (aside και main) έχουν φύγει από το normal flow και δεν υπάρχει κανένα άλλο στοιχείο στο normal flow για να δώσει ύψος στον container



Overflow: hidden (1)

- Το `clear: both` δουλεύει όταν το `element` είναι μέσα στον `container (child)` όπως στο προηγούμενο παράδειγμα το `Footer` ήταν μέσα στο `Container`
- Για να δημιουργήσουμε ένα `block formatting context` στο `parent element` κάνουμε **`overflow: hidden`**
- Το `overflow: hidden` είναι ένα `hack` για να ορίσουμε ρητά ένα `BFC` στο `container div` και να του δώσουμε την ιδιότητα να περιλαμβάνει `floats`, οπότε στην δική μας περίπτωση το `div` θα έχει ύψος



Overflow: hidden (2)



```
13 .container {  
14     overflow: hidden;  
15     width: 800px;  
16     margin-left: auto;  
17     margin-right: auto;  
18     border: 2px solid red;  
19 }
```

- Οπότε με overflow: hidden δημιουργούμε ένα νέο Blocking Formatting Context μέσα στο οποίο μπορούμε να έχουμε floats



Full-bleed Layout (1)

Προγραμματισμός στο Web

- Για να δώσουμε full-bleed layout (full width) στο main και aside δεν μπορούμε να το κάνουμε με τον container γιατί έχει περιορισμένο width, έτσι ώστε να μπορεί να στοιχήσει στο κέντρο
- Θα εισάγουμε ένα wrapper container με full width και θα του δώσουμε background-color



Full-bleed Layout (2)

Προγραμματισμός στο Web



```
.container-fluid {  
  background-color: rgba(96, 103, 103, 0.543);  
}
```



Ίδιου πλάτους columns (1)

Προγραμματισμός στο Web

```
<div class="footer">
  <div class="footer-item"></div>
  <div class="footer-item"></div>
  <div class="footer-item"></div>
</div>
</body>
</html>
```

```
48 .footer-item {
49   float: left;
50   height: 100px;
51   width: 30%;
52   background-color: #232323;
53   border: 1px solid black;
54   margin: 10px 1.5%;
55 }
```

- Δίνουμε το πλάτος σε ποσοστό % ώστε να προσαρμόζονται οι στήλες σε διαφορετικά μεγέθη οθονών



Ίδιου πλάτους columns (2)

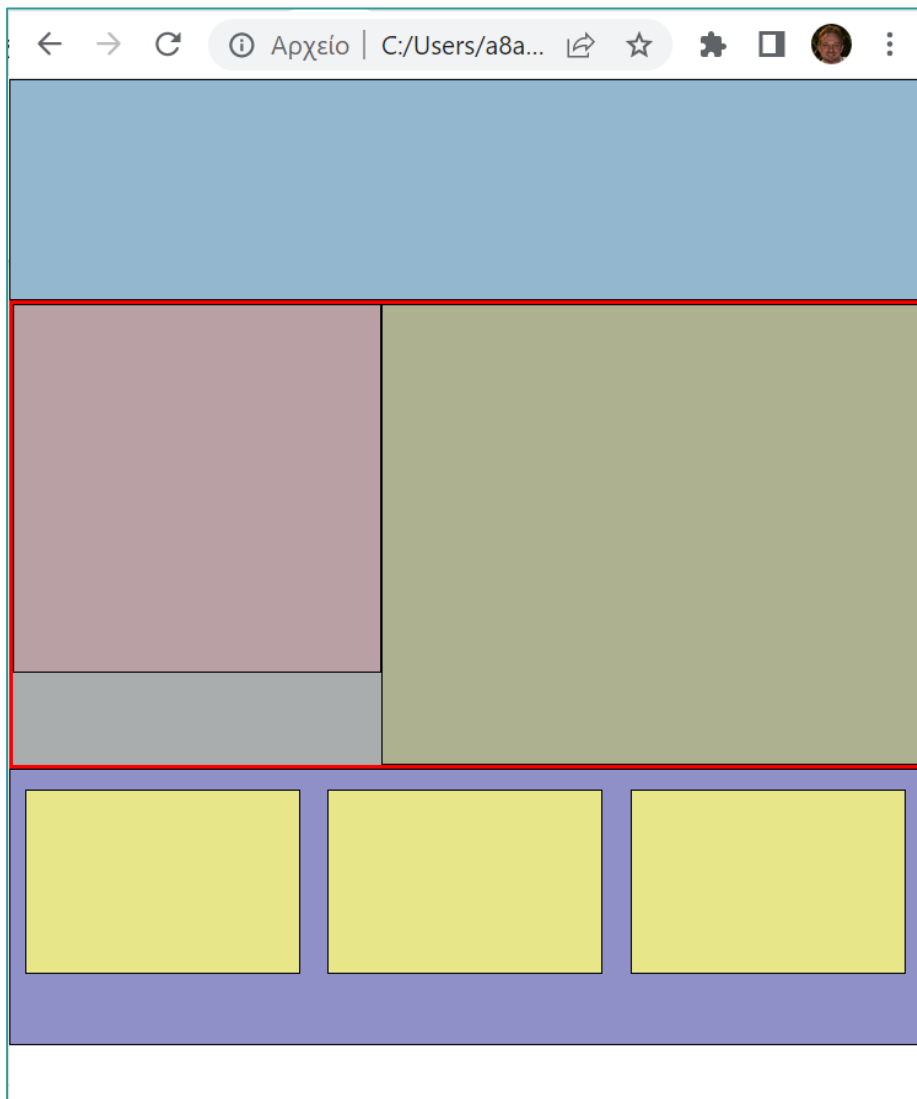
Προγραμματισμός στο Web





Resize και Responsive Design

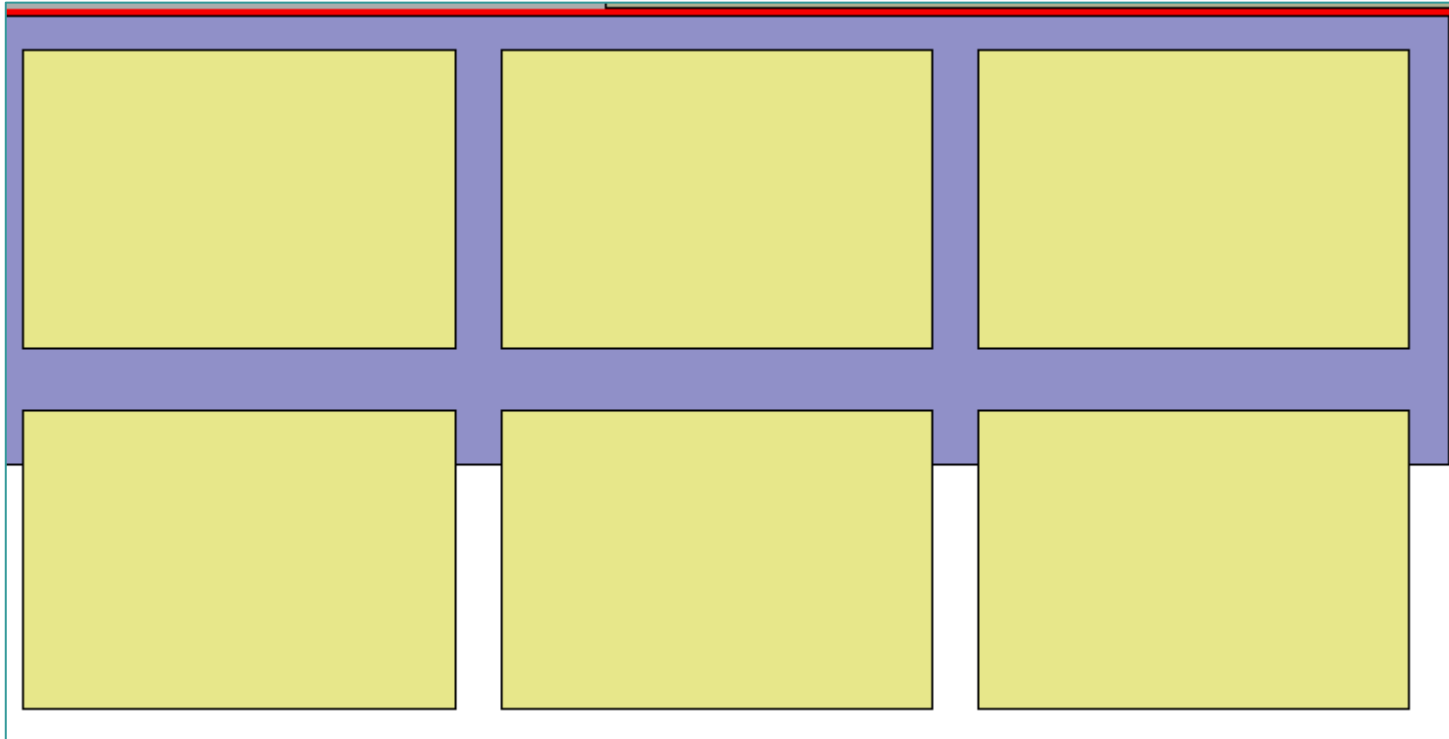
Προγραμματισμός στο Web



- Η βασική ιδέα στο fluid και responsive design είναι το width στα layout elements να είναι δοσμένο σε ποσοστό



Grid (1)



- Όταν δεν υπάρχει άλλος οριζόντιος χώρος τα floats πάνε από κάτω



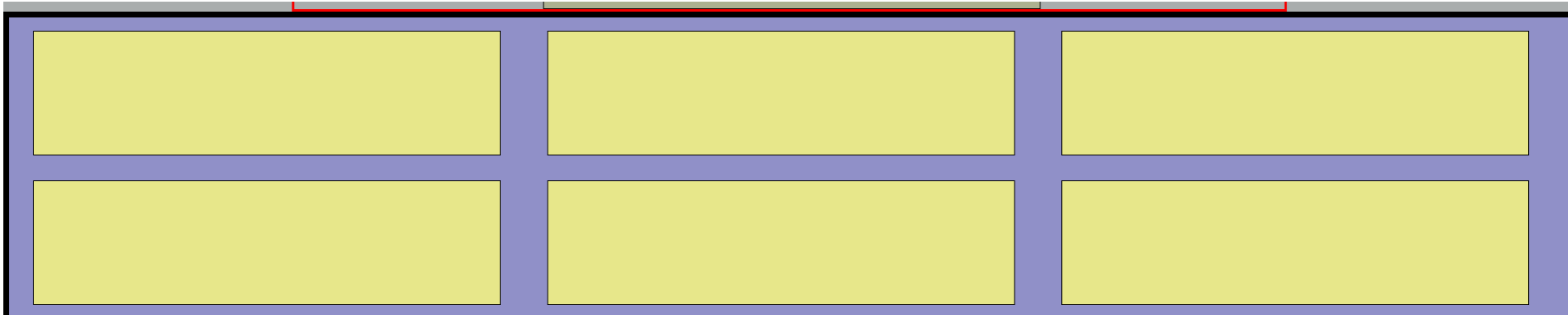
Grid (2)

```
40
41 .footer {
42     overflow: hidden;
43     min-height: 150px;
44     border: 5px solid black;
45     background-color: rgb(143, 143, 199);
46 }
47
48 .footer-item {
49     float: left;
50     height: 100px;
51     width: 30%;
52     background-color: rgb(232, 232, 141);
53     border: 1px solid black;
54     margin: 10px 1.5%;
55 }
56
```

- Πρέπει να κάνουμε τον **footer** **overflow: hidden** ώστε να δημιουργηθεί block context
- Επίσης πρέπει να δώσουμε κατάλληλο ύψος ή min-height αντί για height
- Το min-height στην περίπτωση που τα περιεχόμενα κάνουν overflow δεν γίνονται crop (δεν κόβονται)



Grid (3)



- Για να φτιάξουμε μία photo gallery μπορούμε απλά να εισάγουμε σε κάθε item ένα image και να χρησιμοποιήσουμε το main section αντί του footer



Floats στο content (1)

```
<div class="main">
  
  <p>Lorem ipsum dolor, sit amet consectetur
    adipisci inventore deleniti velit vel
    Similique.
  </p>
  <p>Lorem ipsum dolor sit amet consectetur
    eligendi officia exercitationem corru
    asperiores saepe?
  </p>
</div>
```

- Έστω ότι έχουμε ένα `img` και δύο `p` μέσα στο `main`



Floats στο content (2)

```
33 .main {
34     float: left;
35     height: 300px;
36     width: 400px;
37     border: 1px solid black;
38     background-color: rgb(178, 182, 118, 0.5);
39     padding: 20px;
40 }
41
42 .main > img {
43     float: left;
44     width: 100px;
45     height: auto;
46     margin-right: 20px;
47     margin-bottom: 20px;
48 }
49
50 .main > p {
51     margin-bottom: 10px;
52 }
53
```

- Δίνουμε float την εικόνα
- Έχουμε έτσι float μέσα σε float



Floats στο content (3)

Προγραμματισμός στο Web



Lorem ipsum dolor, sit amet
consectetur adipisicing elit. Quas ad
pariatur itaque error perferendis
aliquam quae sapiente dolore ipsum
ducimus adipisci inventore deleniti
velit vel, dolor atque perspiciatis!

Similique.

Lorem ipsum dolor sit amet consectetur adipisicing elit.
Omnis, fugit. Ducimus aliquam reprehenderit at
quidem non? Magnam laudantium officiis odio eligendi
officia exercitationem corrupti dolorum. Dolor totam ea
asperiores saepe?

- Η εικόνα έχει φύγει από το normal flow και έχει τοποθετηθεί αριστερά
- Τα υπόλοιπα στοιχεία ακολουθούν το normal flow



Τύποι σελίδων

- Υπάρχουν διάφοροι τύποι σελίδων
 - **Home Page (Αρχική Σελίδα).** Όλα τα web sites έχουν μία Αρχική Σελίδα που είναι η σημαντικότερη σελίδα ενός site γιατί είναι η 1^η σελίδα που βλέπει ο χρήστης καθώς και η σελίδα που πηγαίνουμε (Landing Page) όταν πατάμε κλικ σε μία διαφήμιση του site μας
 - **Σελίδες πληροφοριών / προϊόντων.** Πρόκειται για σελίδες που στόχο έχουν είτε να πληροφορήσουν τους χρήστες για πτυχές του site μας ή να προωθήσουν και να πουλήσουν κάποιο προϊόν ή υπηρεσία
 - **Σελίδες εφαρμογών** που εισερχόμαστε με κωδικούς χρήστη (credentials – username / password) και που υλοποιούν την επιχειρηματική λογική της εφαρμογής



Αρχική Σελίδα – Home Page

Προγραμματισμός στο Web

- Η Αρχική σελίδα είναι η σημαντικότερη σελίδα ενός Site γιατί ενσωματώνει όλα τα χαρακτηριστικά που μπορεί να **προτρέψουν ή να αποτρέψουν τους χρήστες** να συνεχίσουν την περιήγηση τους στο Site μας και να ολοκληρώσουν ή όχι συναλλαγές
- Να δούμε στη συνέχεια τα βασικά οπτικά χαρακτηριστικά σχεδιασμού, δηλαδή το User Interface (UI) μιας Αρχικής Σελίδας



Βασικά στοιχεία UI - Header

Προγραμματισμός στο Web

- Λογότυπο και Μενού - Header
 - Μπορεί να είναι σε ξεχωριστές ‘γραμμές’ αλλά συνήθως είναι σε ένα λογικό ‘κουτί’ που ονομάζεται **Header**
 - Το Μενού είναι από τα σημαντικότερα χαρακτηριστικά μιας σελίδας γιατί υλοποιεί το *navigational model* δηλαδή τη δομή συνολικά της σελίδας μας και τον τρόπο που μπορεί ο χρήστης να περιηγηθεί σε αυτή



Slider / Images

Προγραμματισμός στο Web

- Μετά το Header μπορούμε να χρησιμοποιήσουμε μία **μεγάλη εικόνα** ή **ένα Slider** (περισσότερες εικόνες που περνάνε μία-μία) ώστε να προσελκύσουμε οπτικά τους χρήστες, να κρατήσουμε αμείωτο το ενδιαφέρον τους και να τους προτρέψουμε να αλληλεπιδράσουν μαζί μας, να αυξήσουμε δηλαδή το engagement



Main Section

- Μετά το Image υπάρχει συνήθως ένα Main Section που μπορεί να λάβει διάφορες οπτικές μορφές, συνήθως με μία/δύο ή τρεις στήλες, καθώς και περιλαμβάνει μία ή περισσότερες λογικές 'γραμμές' στη σελίδα μας
- Σκοπός του main section είναι η παροχή πληροφοριών προς τον χρήστη



Footer

- Τέλος, υπάρχει ένα Footer στο κάτω μέρος της σελίδας με, πιθανώς, πληροφορίες επικοινωνίας, πνευματικά δικαιώματα ή/και συνδέσμους προς social media
- Σκοπός του Footer είναι να συνεχίσουμε την αλληλεπίδραση/επικοινωνία με τον χρήστη και αφού φύγει από τη σελίδα μας ζητώντας του πιθανά το e-mail του ή παρέχοντας τα δικά μας στοιχεία επικοινωνίας (e-mail, τηλέφωνο, διεύθυνση, κλπ.)



Απλότητα

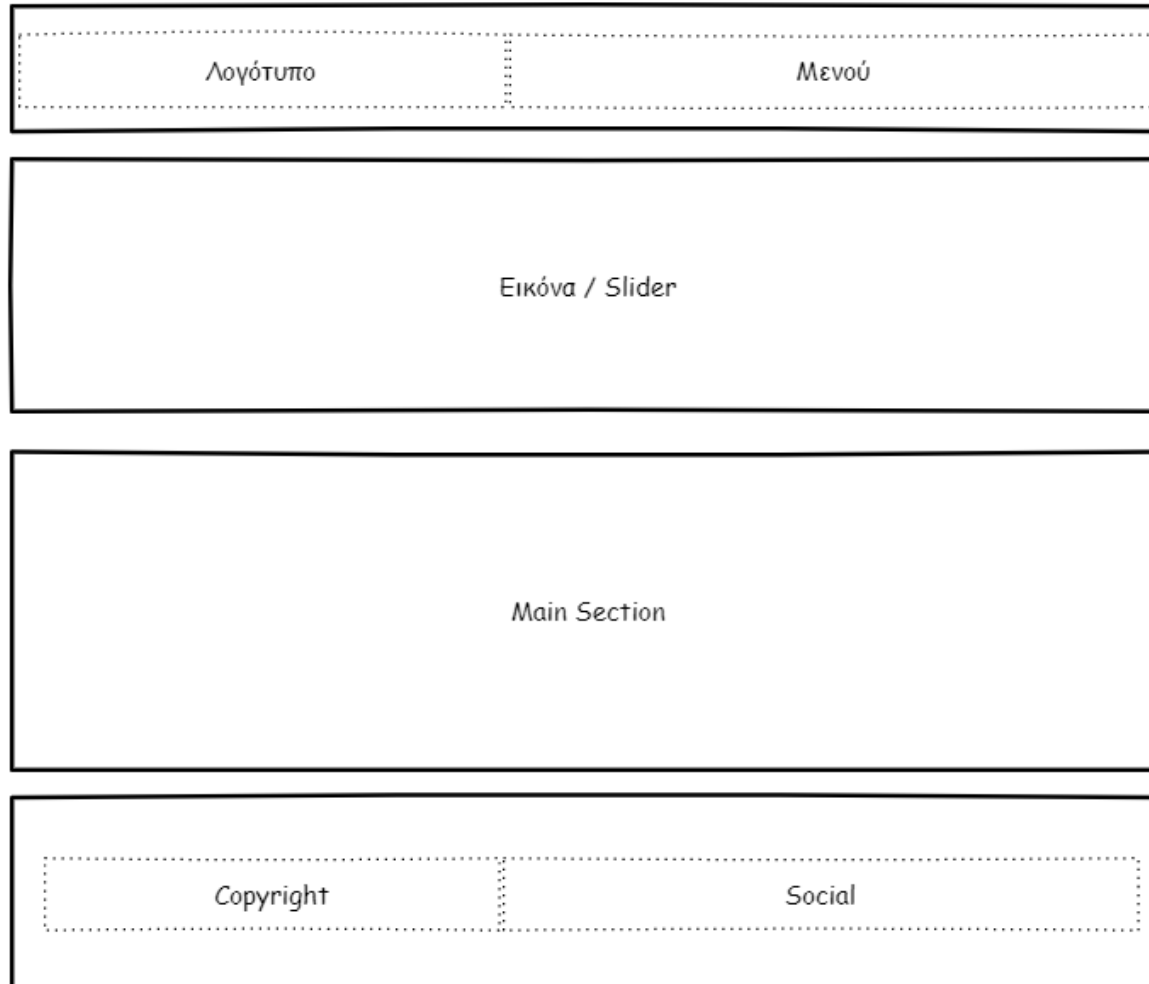
Προγραμματισμός στο Web

- Η απλότητα στο σχεδιασμό μιας σελίδας είναι βασική σχεδιαστική αρχή
- Οι χρήστες θέλουν να χρησιμοποιούν σελίδες που είναι εύκολα κατανοητές και ευκολομνημόνευτες δίχως περιττή πληροφορία ή μαξιμαλιστικό UI



Wireframe

Προγραμματισμός στο Web

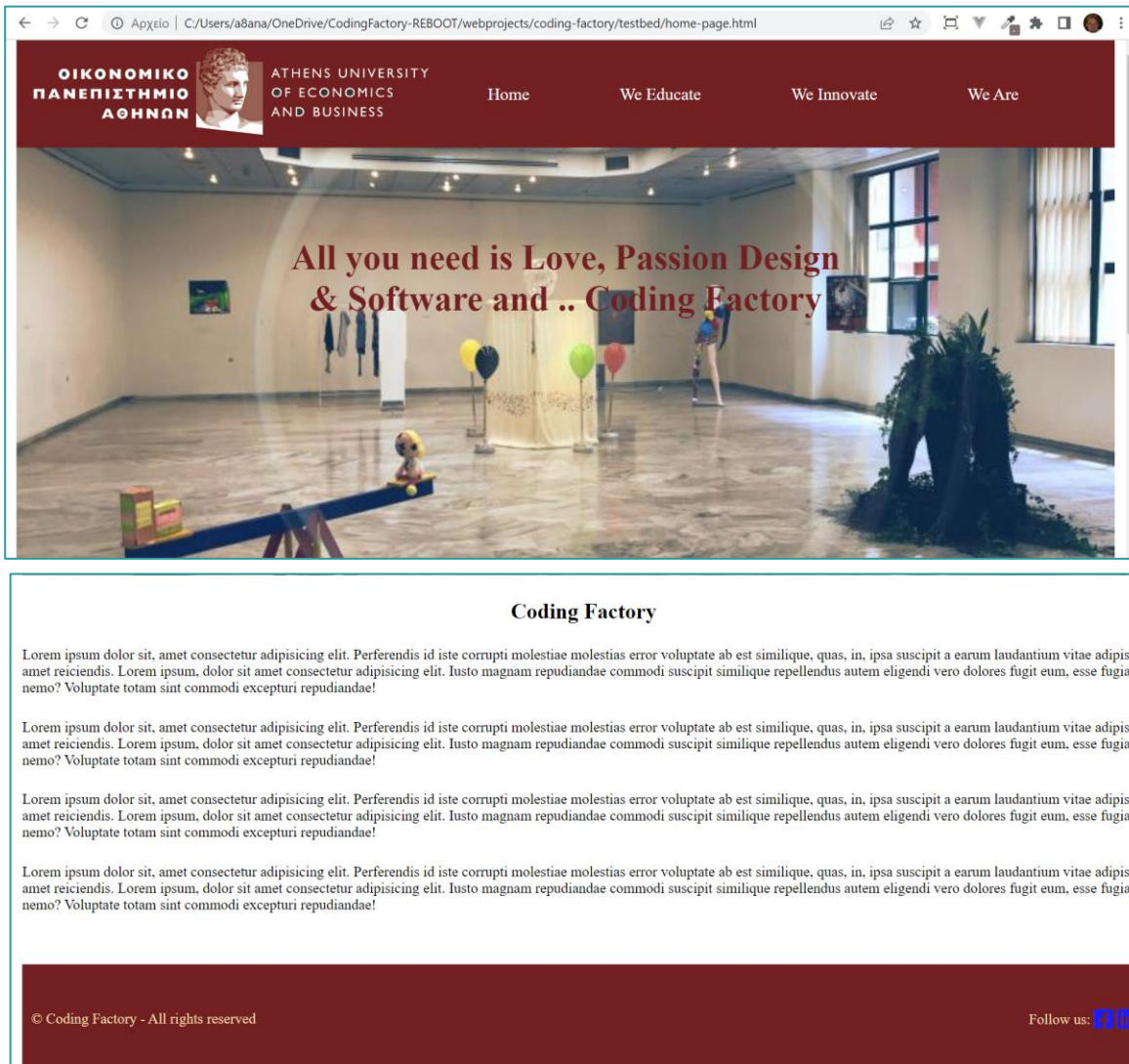


- Το Header περιλαμβάνει το Λογότυπο και το Μενού
- Στη συνέχεια μία εικόνα συνήθως ως background για να εμφανίζουμε και κάποιο μήνυμα από πάνω ή ένας Slider για να δημιουργούμε και κάποιο story
- Ένα Main Section που μπορεί να πάρει πάρα πολλές μορφές
- Και ένας Footer που περιλαμβάνει ένα Copyright καθώς και Social (Facebook, Twitter)



Mock-up

Προγραμματισμός στο Web



Το mock-up είναι ο τελικός σχεδιασμός που δίνεται από τους UX/UI Designers στους Developers



Home Page – HTML (1)

Προγραμματισμός στο Web

```
<> home-page.html x
testbed > <> home-page.html > html > body > div.container > div.home-img-gap-bottom
5  <meta http-equiv="X-UA-Compatible" content="IE=edge">
6  <meta name="viewport" content="width=device-width, initial-scale=1.0">
7  <title>Home Page</title>
8  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.2.1/css/all.min.css" integ
9  <link rel="stylesheet" href="./css/home-page.css">
10 <ad>
11 />
12 <div class="container">
13   <div class="header">
14     <div class="logo">
15       
16     </div>
17     <div class="nav">
18       <ul>
19         <li><a href="">Home</a></li>
20         <li><a href="">We Educate</a></li>
21         <li><a href="">We Innovate</a></li>
22         <li><a href="">We Are</a></li>
23       </ul>
24     </div>
25   </div>
26 </div>
27 <div class="home-img gap-bottom">
28   <div class="home-title">All you need is Love, Passion Design & Software and .. Coding Factory</div>
29   <div class="home-subtitle"></div>
30 </div>
```

- Παρατηρούμε μέσα στον header έχουμε logo και nav. Παρατηρούμε ότι το <nav> περιέχει ένα μενού, που το υλοποιούμε ως και τα μενού items ως
- Επίσης υπάρχει ένα <div> για <home-img>



Home Page – HTML (2)

Προγραμματισμός στο Web

```
31 <div class="main">
32   <h1 class="main-title gap-bottom">Coding Factory</h2>
33   <section class="gap-bottom">
34     Lorem ipsum dolor sit, amet consectetur adipisicing elit. Perferendis
35     suscipit a earum laudantium vitae adipisci amet
36     reiciendis.
37     Lorem ipsum, dolor sit amet consectetur adipisicing elit.
38     Iusto magnam repudiandae commodi suscipit similique repellendus autem
39     Voluptate totam sint commodi excepturi repudiandae!
40   </section>
41   <section class="gap-bottom">
42     Lorem ipsum dolor sit, amet consectetur adipisicing elit. Perferendis
43     suscipit a earum laudantium vitae adipisci amet
44     reiciendis.
45     Lorem ipsum, dolor sit amet consectetur adipisicing elit.
46     Iusto magnam repudiandae commodi suscipit similique repellendus autem
47     Voluptate totam sint commodi excepturi repudiandae!
48   </section>
49   <section class="gap-bottom">
50     Lorem ipsum dolor sit, amet consectetur adipisicing elit. Perferendis
51     suscipit a earum laudantium vitae adipisci amet
52     reiciendis.
53     Lorem ipsum, dolor sit amet consectetur adipisicing elit.
54     Iusto magnam repudiandae commodi suscipit similique repellendus autem
55     Voluptate totam sint commodi excepturi repudiandae!
56   </section>
57   <section class="gap-bottom">
58     Lorem ipsum dolor sit, amet consectetur adipisicing elit. Perferendis
59     suscipit a earum laudantium vitae adipisci amet
60     reiciendis.
61     Lorem ipsum, dolor sit amet consectetur adipisicing elit.
62     Iusto magnam repudiandae commodi suscipit similique repellendus autem
63     Voluptate totam sint commodi excepturi repudiandae!
64   </section>
65 </div>
```

- Το <div> με class-name = main περιέχει <section>
- Το <section> όπως και τα <header>, <nav>, <main>, <footer> κλπ είναι στην πραγματικότητα div αλλά έχουν και accessibility semantics για τους screen readers
- Επομένως, τα προτιμάμε και θα τα δούμε και στην ενότητα του Accessibility



Home Page – HTML (3)

Προγραμματισμός στο Web

```
66     <div class="footer">
67         <div class="copyright">&copy; Coding Factory - All rights reserved</div>
68         <div class="social">
69             <span>Follow us: </span>
70             <i class="social-text fa-brands fa-square-facebook"></i>
71             <i class="social-text fa-brands fa-linkedin"></i>
72         </div>
73     </div>
74 </div>
75 </body>
76 </html>
```

- Στο τέλος έχουμε τον footer, που περιέχει δύο <div> με το copyright και τα social



Home Page – CSS (1)

```
1  * {
2    box-sizing: border-box;
3    margin: 0;
4    padding: 0;
5  }
6
7  .container {
8    width: 98%;
9    margin-left: auto;
10   margin-right: auto;
11 }
12
13 .header {
14   background-color: #762124;
15   overflow: hidden;
16 }
17
18 .logo {
19   float: left;
20 }
21
22 .logo > img {
23   width: auto;
24   height: 120px;
25   vertical-align: middle;
26 }
```

- Δίνουμε styling στα UI element, τόσο για το layout όσο και για τα στοιχεία όπως το `img` που είναι άμεσος απόγονος του `.logo`
- Θέλουμε το `.logo` που περιέχει το `image` να είναι στοιχισμένο αριστερά (`float: left`) κάθετα μέσα στον `header` (`vertical-align: middle`, μπορούμε και εφαρμόζουμε `vertical-align` γιατί το `image` είναι `inline-block` στοιχείο)
- Στο `ui` για κάθετη στοίχιση δίνουμε `height` όσο και του `header` καθώς και `display: flex`
- Πρέπει να κάνουμε τον `header overflow: hidden` γιατί περιέχει στοιχεία `float`



Home Page – CSS (2)

```
28 .nav {
29     float: left;
30     width: 700px;
31 }
32
33 .nav > ul {
34     height: 120px;
35     list-style: none;
36     display: flex;
37     justify-content: space-around;
38     align-items: center;
39 }
40
41 .nav a {
42     display: block;
43     color: white;
44     text-decoration: none;
45     font-size: 1.2rem;
46     padding: 5px 10px;
47 }
48
49 .nav a:hover {
50     background-color: rgb(22, 152, 109);
51 }
```

- Το ίδιο και εδώ δίνουμε styling στο `` που υλοποιεί το Menu
- Δίνουμε flexbox που κάνει τα παιδιά του (menu items) flex items και τα στοιχίζει οριζόντια με `space-around` ενώ και κάθετα στο κέντρο. Δίνουμε styling και στα στοιχεία `<a>`
- Η βασική ρύθμιση είναι ότι κάνουμε και το `` **display: flex**. Όπως είπαμε το `display flex` μας δίνει τη δυνατότητα εύκολα να στοιχίσουμε οριζόντια και κάθετα
- Με **align-items** στοιχίζουμε κάθετα, και με **justify-content** ρυθμίσουμε την οριζόντια στοίχιση
- Με **justify-content: space-around** ορίζουμε τα περιεχόμενα του `` δηλαδή τα `` να στοιχηθούν με κενό αριστερά-δεξιά και ανάμεσά τους
- Το default flow στα flex boxes είναι το οριζόντια, οπότε τα `` θα τοποθετηθούν οριζόντια. Για να φύγουν τα bullets κάνουμε `list-style: none`; και για τα `<a>` κάνουμε `text-decoration: none` και λευκό χρώμα



Home Page – CSS (3)

Προγραμματισμός στο Web

```
53 .nav a:active {
54     background-color: rgb(172, 172, 231);
55 }
56
57 .home-img {
58     /* parent-child collapsing margins needs overflow:hidden to clear */
59     overflow: hidden;
60     height: calc(95vh - 120px);
61     background-image: url('../img/tep-aueb.jpg');
62     background-repeat: no-repeat;
63     background-size: cover;
64     background-position: center;
65 }
66
67 .home-title {
68     width: 50%;
69     color: #762124;
70     font-size: 40px;
71     font-weight: 600;
72     text-align: center;
73     margin-left: auto;
74     margin-right: auto;
75     margin-top: 100px;
76 }
77
78 .gap-bottom {
79     margin-bottom: 25px;
80 }
```

- Επειδή στο `.home-title` δίνουμε `margin-top` και αυτό επηρεάζει και το `margin-top` του `.home-img`, δίνουμε στο `.home-img` `overflow: hidden` ώστε να δημιουργήσει νέο Block Formatting Context
- Επίσης, παρατηρούμε το `height` του `.home-img` που ορίζεται ως η διαφορά του `95vh - 120px` (ο `header` είναι `120px` δεδομένου του `logo`)
- `95vh` είναι το 95% του `height` του `viewport` του `browser`



Home Page – CSS (4)

Προγραμματισμός στο Web

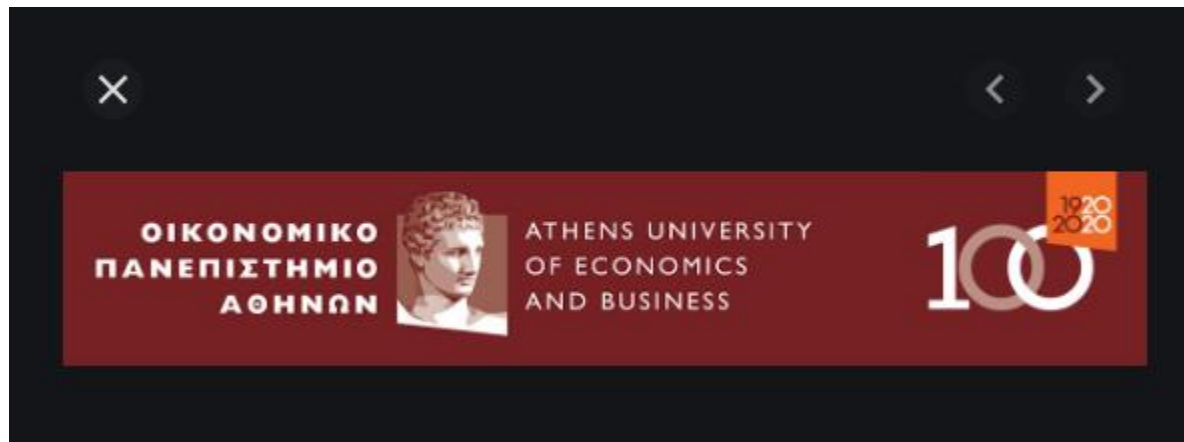
```
82 .main {
83     display: flex;
84     height: 400px;
85     flex-direction: column;
86     justify-content: flex-start;
87     align-items: center;
88 }
89
90
91 .footer {
92     display: flex;
93     height: 120px;
94     background-color: #762124;
95     justify-content: space-between;
96     align-items: center;
97     color: wheat;
98     padding: 0 10px;
99 }
100
101 .social-text {
102     vertical-align: middle;
103     color: blue;
104     font-size: 1.5rem;
105 }
106
107 .social>span {
108     vertical-align: middle;
109 }
```

- Η .main ορίζεται με display: flex και επομένως μπορούμε να στοιχίσουμε εύκολα οριζόντια και κάθετα
- Το ίδιο και ο footer είναι flexbox
- Μπορούμε επίσης δύο inline elements όπως είναι τα .social-text που είναι τα font awesome icons και το span που είναι το «Follow us» να τα στοιχίσουμε στο κάθετο κέντρο με vertical-align: middle



Header / Logo - image

- Το λογότυπο είναι μία οπτική αναπαράσταση, το σύμβολο της επιχείρησης / οργανισμού
- Λογότυπα μπορούμε να δημιουργήσουμε ή να επεξεργαστούμε με το Photoshop ή open source προγράμματα επεξεργασίας εικόνας όπως το GIMP
- Έστω ότι έχουμε ήδη κατεβάσει ή κατεβάζουμε από το google το παρακάτω logo του AUEB
- Στη συνέχεια θα επεξεργαστούμε το Logo aueb100 ώστε να του δώσουμε την απόχρωση #363636 στο background

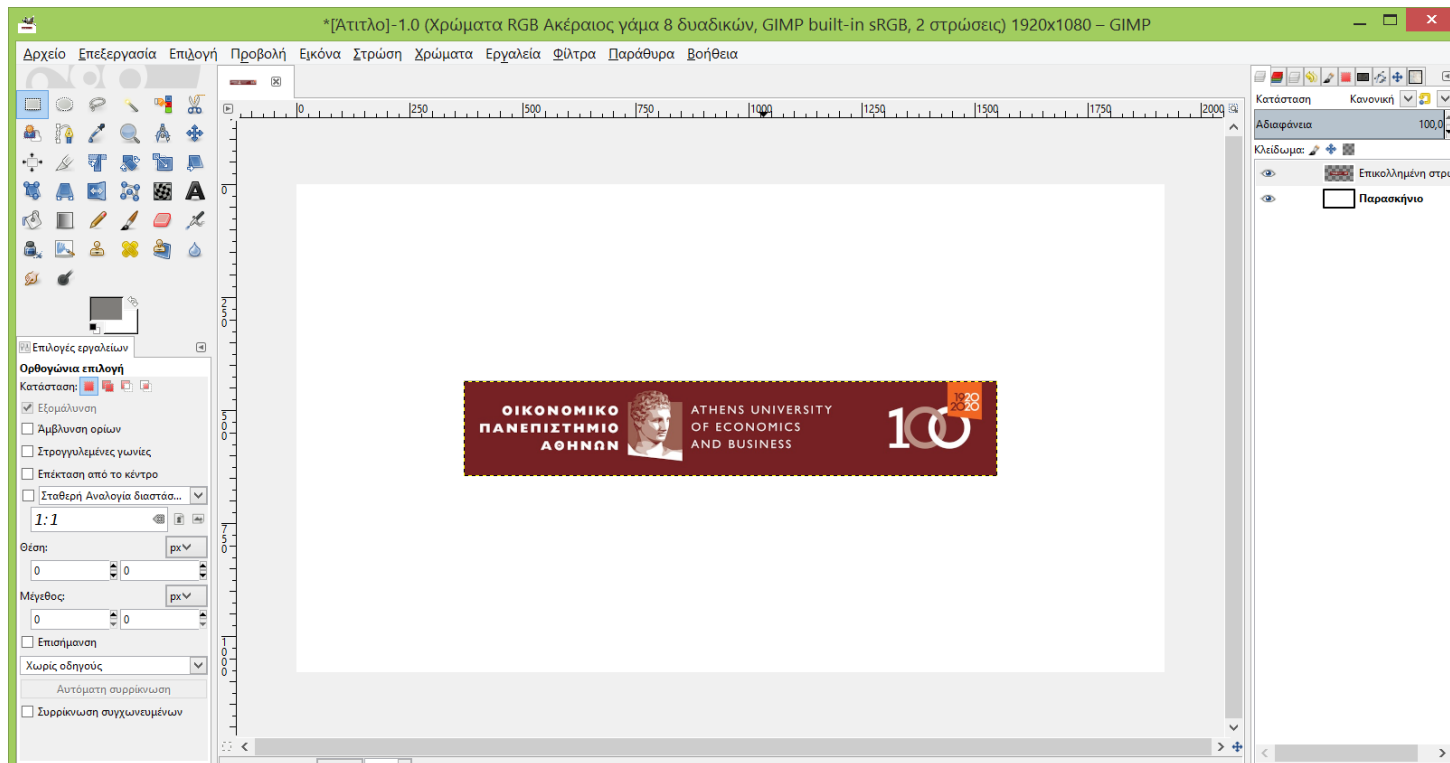




Εισαγωγή εικόνας στο GIMP

Προγραμματισμός στο Web

- Εισάγουμε με drag & drop την εικόνα στο GIMP

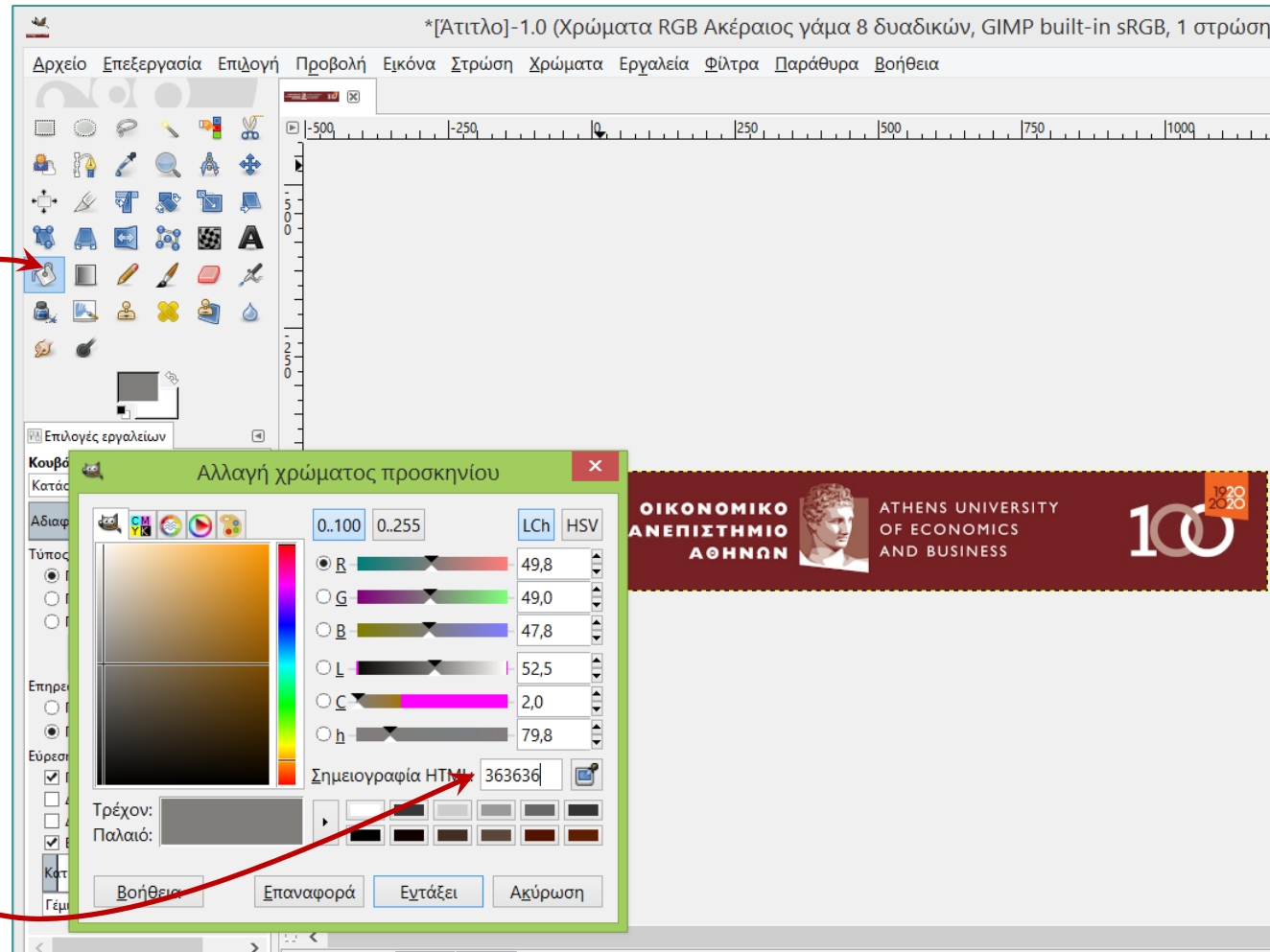




Επεξεργασία στο GIMP

Προγραμματισμός στο Web

- Επιλέγουμε από την εργαλειοθήκη τον κουβά γεμίσματος ενώ επίσης στο foreground color δίνουμε τον κωδικό του background-color από το css





Αποτέλεσμα

Προγραμματισμός στο Web

- Κάνουμε κλικ με τον κουβά γεμίσματος μέσα στην εικόνα και πάνω στο βασικό της χρώμα και επιτυγχάνουμε το παρακάτω αποτέλεσμα ώστε να μην ξεχωρίζει η εικόνα από το background-color του css
- Επιλέγουμε Αρχείο / Εξαγωγή ως και επιλέγουμε κατάληξη .png και δίνουμε ένα όνομα αρχείου (π.χ. aueb100b.png). Αποθηκεύουμε μέσα στο φάκελο *img* του project μας





Semantic Elements

← → ↻ html.spec.whatwg.org/multipage/sections.html

HTML

Living Standard — Last Updated 18 November 2022

[← 4 The elements of HTML](#) — [Table of Contents](#) — [4.4 Grouping content](#) →

4.3 Sections

4.3.1 The **body** element

4.3.2 The **article** element

4.3.3 The **section** element

4.3.4 The **nav** element

4.3.5 The **aside** element

4.3.6 The **h1**, **h2**, **h3**, **h4**, **h5**, and **h6** elements

4.3.7 The **hgroup** element

4.3.8 The **header** element

4.3.9 The **footer** element

4.3.10 The **address** element

- Πρόκειται τεχνικά για `<div>` αλλά με semantics για screen readers, και Google
- Τα πιο σημαντικά είναι τα `<header>`, `<nav>`, `<main>`, `<footer>`, `<section>`, `<aside>`, `<article>`
- Σε μία σελίδα υπάρχει ένας Header, ένα `<nav>` ένα `<main>` και ένα `<footer>`
- Ενώ μπορεί να περιέχονται πολλά `<section>`, `<article>` που είναι αυτόνομα κομμάτι περιεχομένου



Footer - HTML

```
90
91 .footer {
92     display: flex;
93     height: 120px;
94     background-color: #762124;
95     justify-content: space-between;
96     align-items: center;
97     color: wheat;
98     padding: 0 10px;
99 }
100
101 .social-text {
102     vertical-align: middle;
103     color: blue;
104     font-size: 1.5rem;
105 }
106
107 .social>span {
108     vertical-align: middle;
109 }
110
```

- Ο footer αποτελείται από δύο `<div>` το *copyright* και το *social*
- Το *social* περιέχει ένα `` element για μορφοποίηση του κειμένου και δύο *fa-brands icons*. Έχουμε κάνει `<link>` στο *html* το *CDN*
- Παρατηρήστε το σύμβολο του *copyright ©* Στην *HTML* σελίδα, είναι *HTML Entity*

```
div class="footer">
  <div class="copyright">&copy;
```



HTML Entities

- Reserved (π.χ. `>`, `<` -- `<`, `>`) , Invisible (whitespace -- ` ` –non breaking space) και Special characters (copyright, ampersand, euro -- `©`, `&`, `&euro`)
- Τα Entity names ξεκινάνε με `&` , ενώ τα αντίστοιχα Entity numbers ξεκινάνε με `&#`



Menu Sticky

```
13  .header {  
14      background-color: #762124;  
15      overflow: hidden;  
16      position: sticky;  
17      top: 0px;  
18  }
```

- Για να σταθεροποιήσουμε τον header και να είναι πάντα στο πάνω μέρος της οθόνης ανεξάρτητα από το scrolling ορίζουμε το position να είναι fixed και η απόσταση από το πάνω μέρος του container να είναι 0
- Το position όπως και το float βγάζει το στοιχείο από το normal flow. Το top: 0 το τοποθετεί 0px στο από το πάνω μέρος του viewport



Γενικές Σχεδιαστικές Αρχές

Προγραμματισμός στο Web

- Γενικές Σχεδιαστικές Αρχές
 - Γνωστική Ψυχολογία
 - Σχεδιασμός Διεπαφών Ανθρώπου Υπολογιστή –
- Βασικά Design Heuristics
 - 10 κανόνες Nielsen



Feedback

- Οι χρήστες πρέπει πάντα να είναι ενημερωμένοι για το τι συμβαίνει, μέσω κατάλληλης ανάδρασης και μέσα σε εύλογο χρονικό διάστημα
- Το σημαντικότερο σχεδιαστικό χαρακτηριστικό!
- Θυμηθείτε λάθος σχεδιαστικές επιλογές: Τηλεφωνικές συσκευές χωρίς οθόνη!
- Σκεφτείτε, λάθος σχεδιασμός:
 - Buttons χωρίς ένδειξη ότι πατήθηκαν
 - Buttons χωρίς ένδειξη ότι προχωράει η επεξεργασία



Digital Experience and Real World

Προγραμματισμός στο Web

- Γλώσσα κατανοητή με βάση το κοινό
 - Π.χ. εφαρμογή για παιδιά δημοτικού
- Συνάφεια με τον πραγματικό κόσμο
 - AR (Augmented Reality, βλ. IKEA)
 - <https://www.wired.com/story/ikea-place-ar-kit-augmented-reality/>
- Παράδειγμα: Καλάθι αγορών (όπως στο σούπερ μάρκετ)



Control of the System

Προγραμματισμός στο Web

- Επιλογές αναίρεσης πράξεων
- Για παράδειγμα σε κάθε βήμα μιας συναλλαγής, μπορεί να υπάρχει Undo / Cancel
- Αλλαγές: Ποσότητας, αφαίρεση προϊόντος από καλάθι, κλπ.



Συνέπεια – Συμμόρφωση έναντι προτύπων

Προγραμματισμός στο Web

- Ομοιότητα χρήσης / συμβόλων κλπ. σχεδιαστικών αντικειμένων
- Γειτνίαση (Βασίζεται στον συνειρμό)
- Conventions αγοράς / πολιτισμών / ανταγωνιστών (π.χ. καλαθάκι για online αγορές)
- Responsive Design



Error Prevention (Πρόληψη σφαλμάτων)

Προγραμματισμός στο Web

- Επιβεβαιώσεις για επικίνδυνες πράξεις (π.χ. ακύρωση, διαγραφή)
- Γενικά οτιδήποτε αποτελεί μη-αναστρέψιμη πράξη θα πρέπει να υπάρχει προειδοποίηση



Κατανοητά artifacts

Προγραμματισμός στο Web

- Learnability / usability
 - Δεν βασιζόμαστε στη μνήμη των χρηστών αλλά στην διεπαφή
 - Αναγνωρίζουμε τη χρήση. Θυμηθείτε λάθος σχεδιαστικές επιλογές
 - Πόρτες με δύο φύλλα. Ποιο φύλλο ανοίγει προς ποια κατεύθυνση;;;
 - Βρύσες χωρίς πόμολο. Πως ανοίγει;;;



Flexibility & Efficiency (Ευελιξία και αποδοτικότητα)

Προγραμματισμός στο Web

- Διαφορετικοί τρόποι για να εκτελεστεί μία πράξη ανάλογα με το επίπεδο των χρηστών
- Το σύστημα πρέπει να μπορεί να εξυπηρετεί άπειρους και έμπειρους χρήστες
- Accelerators για έμπειρους χρήστες
 - π.χ. One-click checkout



Aesthetic & Minimal Design

Προγραμματισμός στο Web

- Κείμενο που να είναι ευκρινές, χωρίς περιττές πληροφορίες, χωρίς ενοχλητικές εικόνες, χωρίς περιττές πράξεις
- *Signal-to-noise ratio*
 - Signals. Οι πράξεις που θέλουμε να κάνουμε
 - Noise. Οτιδήποτε αποσπά την προσοχή μας
- **Προτροπή!** Βασικό χαρακτηριστικό του καλού design. Τέχνη, Μήνυμα, Μάρκετινγκ, Γνωστική Ψυχολογία (π.χ. μπάλα -> παιχνίδι, κλπ. συνειρμοί)
 - Θέλουμε ένα **λογικό** Signal-to-noise ration



Βοήθεια

- Ιδανικά το σύστημα πρέπει να μπορεί να χρησιμοποιηθεί χωρίς βοήθεια. Όταν είναι απαραίτητο η βοήθεια πρέπει να είναι εύκολα αναζητήσιμη, εστιασμένη σε αυτό που κάνει ο χρήστης, να απαριθμεί συγκεκριμένα βήματα και να μην είναι πολύ μεγάλη
- FAQs, Glossary, API Documentation, κλπ.



Περιεχόμενο

Προγραμματισμός στο Web

- Αρχιτεκτονική περιεχομένου
 - Card sorting
 - <https://www.nngroup.com/articles/card-sorting-definition/>
- Mapping & Triggers
 - Buttons – Touchpoints - Καθαρά μηνύματα



Άσκηση 1

Προγραμματισμός στο Web

ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ | ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

100 Home We Educate We Innovate We Are



All you need is Love, Passion, Design & Software and ... "Web Programming"

Become the Web Programmer you meant to be with a deep knowledge in Web technologies and methods for empowering your career

Why Web Development Course

The program "Development of Web applications with HTML / CSS / JavaScript / jQuery / AJAX and Bootstrap" is designed for people who have no knowledge or little knowledge of programming on the Web. It starts with presenting techniques for designing and developing web pages and applications with HTML / CSS / JavaScript languages. The JavaScript language will be presented in detail as it is the basis for more advanced topics that will be presented later such as jQuery / AJAX / JSON / XML and Bootstrap. There will also be Web Servers topics where learners will install and use Apache Tomcat as well as Web Hosting and file transfer and management issues using tools such as FileZilla. Also included is the presentation of UI & UX design techniques (User Interface & User Experience Design) so that learners are able to understand the visualization techniques of a web page, as well as the presentation of GitHub and Git, in order for learners to gradually create a portfolio for their personal or professional development in the labor market

© Web Development Course, AUEB. All rights reserved, 2020

Follow-us:  

- Δημιουργήστε τη σελίδα αριστερά
- Χρησιμοποιήστε το GIMP για να δώσετε στην εικόνα μικρότερη φωτεινότητα



Άσκηση 2

- Μέσα σε ένα main div, με float ορίστε μία photo gallery όπως ορίζεται στη διαφάνεια 33