

## Εργαστήριο #9

### Τι πρέπει να έχετε ολοκληρώσει από το προηγούμενο εργαστήριο.

- Το σημερινό εργαστήριο είναι ανεξάρτητο από τα προηγούμενα, επειδή όμως θα ζητηθεί να “ανακυκλώσετε” υλικό από εκείνα, **βεβαιωθείτε ότι έχετε ολοκληρώσει τις προηγούμενες ασκήσεις!**

### Οδηγίες

Στο σημερινό εργαστήριο συνεχίζεται η παρουσίαση των οδηγιών μορφοποίησης. Ειδικότερα, μαθαίνετε **βασικές τεχνικές τοποθέτησης (layout)** των διαφόρων στοιχείων HTML σε μια ιστοσελίδα.



Η κατασκευή μιας πραγματικής (επαγγελματικής) ιστοσελίδας απαιτεί συνδυασμένη γνώση γραφιστικής και μεθόδων αλληλεπίδρασης ανθρώπου-υπολογιστή. Στο εργαστήριο αυτά δεν καλύπτονται· απλά μαθαίνετε τα **τεχνικά εργαλεία** που επιτρέπουν την κατασκευή τέτοιων ιστοσελίδων!

⇒ 1. Το μοντέλο “κουτιού” (*box model*) κάθε στοιχείου HTML κατά το πρότυπο CSS είναι πολύ σημαντικό για την κατανόηση των μεθόδων τοποθέτησης: διαβάστε για αυτό στο **Παράρτημα Α!**

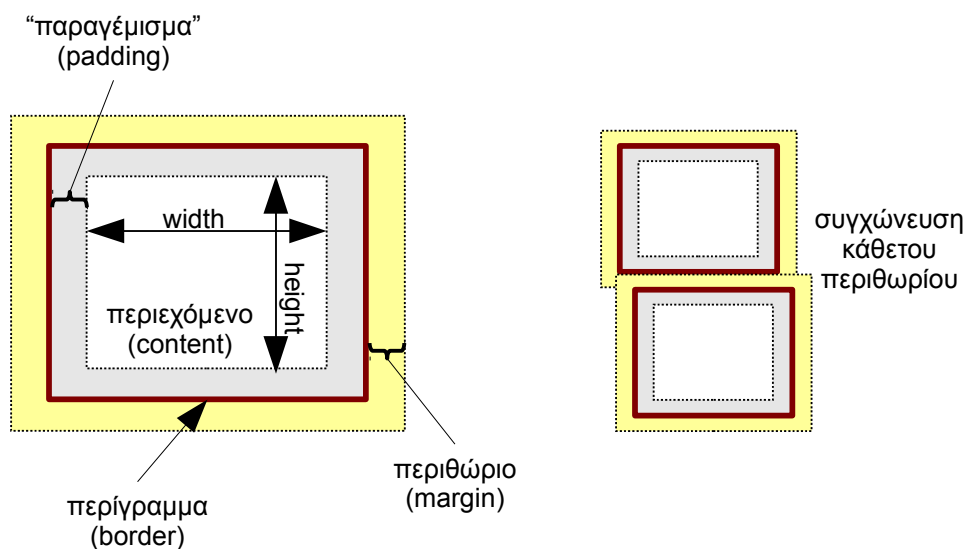
⇒ 2. Διαβάστε το **Παράρτημα Β** για να μάθετε για τις διάφορες μεθόδους τοποθέτησης (*positioning*) των στοιχείων HTML που παρέχει το πρότυπο CSS.

⇒ 3. Διαβάστε το **Παράρτημα Γ** για βασικές οδηγίες “στησίματος” ιστοσελίδων (*page layouts*). Στη συνέχεια, χρησιμοποιήστε το υλικό του 3<sup>ου</sup> εργαστηρίου:

- Δημιουργήστε ένα σταθερό σχήμα (*fixed layout*) για την αρχική σελίδα.
- Δημιουργήστε ένα ρευστό σχήμα (*fluid layout*) για τις υπόλοιπες ιστοσελίδες, με μια στήλη πλοήγησης αριστερά, μια στήλη κυρίως κειμένου, *header* και *footer*.

## Παράρτημα Α: CSS Box Model.

Όπως φαίνεται στο **σχήμα 1**, κάθε στοιχείο HTML περιβάλλεται κατά το πρότυπο CSS από το “παραγέμισμα” (**padding**), το περίγραμμα (**border**) και το περιθώριο (**margin**).



Σχήμα 1

Παρατηρήστε ότι:

1. Οι διαστάσεις (width x height) ενός στοιχείου καθορίζονται από τις διαστάσεις του περιεχομένου, **δεν προσμετράται δηλαδή ο χώρος που προσθέτουν τα padding, border, margin**.
2. Στην εξ'ορισμού (default) τοποθέτηση των στοιχείων, **το κάθετο περιθώριο συγχωνεύεται** (σχήμα 1).
3. Για κάθε στοιχείο HTML, ο browser έχει προκαθορισμένα padding και margin, τα οποία μπορείτε να αλλάξετε μέσω κανόνων CSS.

### Μέγεθος του περιθωρίου (margin)

Μπορείτε να ρυθμίσετε το μέγεθος του περιθωρίου ενός στοιχείου σε σταθερό νούμερο, σε διάφορες μονάδες μέτρησης: π.χ. σε pixels (**px**) ή σε σχέση με το μέγεθος της γραμματοσειράς (**em**). Επίσης μπορείτε να καθορίσετε το μέγεθος ως ποσοστό (**%**) του πλάτους (width) του στοιχείου-πατέρα. Τέλος, μπορείτε να δώσετε την τιμή **auto**, για την οποία θα μιλήσουμε αργότερα.

Μπορείτε να δώσετε ξεχωριστές τιμές για τα 4 περιθώρια π.χ.:

```
body {  
  margin-top: 1em;  
  margin-right: 2em;  
  margin-bottom: 3em;  
  margin-left: 2em;  
}
```

Εναλλακτικά μπορείτε να ορίσετε όλα τα περιθώρια με μία ιδιότητα:

```
body { margin: 2em } /* all 2em */  
body { margin: 1em 2em } /* top&bottom 1em, right&left 2em */  
body { margin: 1em 2em 3em } /* top 1em, right&left 2em,  
                               bottom 3em */  
body { margin: 1em 2em 3em 4em } /* top 1em, right 2em,  
                                   bottom 3em, left 4em */
```

## Μέγεθος του “παραγεμίσματος” (padding)

Το “παραγέμισμα” (padding) δημιουργεί χώρο μεταξύ του περιεχομένου και του περιγράμματος. Εμφανίζει το ίδιο χρώμα φόντου με το περιεχόμενο. Ρυθμίζετε το μέγεθος του “παραγεμίσματος” ακριβώς όπως με το περίγραμμα: οι αντίστοιχες ιδιότητες ονομάζονται **padding-top**, **padding-right**, **padding-bottom**, **padding-left** και **padding**.

## Ιδιότητες του περιγράμματος (border)

Για το περίγραμμα ενός στοιχείου HTML μπορείτε να ρυθμίσετε:

- Το πάχος (**width**) του περιγράμματος σε κάποιο νούμερο (px, em, κλπ) ή στις τιμές [*thin, medium, thick*], μέσω της συγκεντρωτικής ιδιότητας **border-width** (όπως ακριβώς το παράδειγμα με την ιδιότητα margin) ή των ξεχωριστών **border-top-width**, **border-right-width**, **border-bottom-width**, και **border-left-width**. Παράδειγμα:

```
h2 { border-width: 1px 2px 1px 2px; }
```

- Το στυλ (**style**) του περιγράμματος, σε κάποιο από τα [*none, solid, hidden, dotted, double, dashed, groove, ridge, inset, outset*]. Η συγκεντρωτική ιδιότητα ονομάζεται **border-style** και οι ανεξάρτητες **border-top-style**, **border-right-style**, **border-bottom-style**, **border-left-style**. Παράδειγμα:

```
p { border-style: solid dotted; }
```

- Το χρώμα (**color**) του περιγράμματος μέσω της συγκεντρωτικής ιδιότητας **border-color** ή των αντίστοιχων ανεξάρτητων **border-top-color**, **border-right-color**, **border-bottom-color**, **border-left-color**. Παράδειγμα:

```
p { border-color: #7f0000; }
```

Για το περίγραμμα, υπάρχει και η ιδιότητα **border** που επιτρέπει να θέσουμε όλες τις ιδιότητες μαζί, στη σειρά *πάχος – στυλ – χρώμα*:

```
p { border: 2px solid red; }
```

(η ιδιότητα *border* δεν επιτρέπει να θέσουμε ανεξάρτητες τιμές για τα 4 περιγράμματα)

## Παράρτημα Β: CSS και μέθοδοι τοποθέτησης (positioning).

Μέχρι τώρα έχουμε δει τη μέθοδο τοποθέτησης που χρησιμοποιεί εξ'ορισμού ένας browser: ονομάζεται **static** και τοποθετεί τα στοιχεία HTML στην “κανονική ροή” (normal flow) τους.

Τα στοιχεία HTML εμφανίζονται σε δύο βασικές κατηγορίες:

- Ως στοιχεία **inline**, που παρατίθενται στην “κανονική ροή” το ένα δίπλα στο άλλο, συγκροτώντας το περιεχόμενο της ιστοσελίδας. Τέτοια στοιχεία είναι το απλό κείμενο, οι σύνδεσμοι (<a>), οι εικόνες (<img>), κλπ.
- Ως **block** στοιχεία, τα οποία μπορούν να περιέχουν άλλα block ή inline στοιχεία. Στην “κανονική ροή” εμφάνισης, ο browser τοποθετεί κάθε block στοιχείο μόνο του, χωρίς άλλα block στοιχεία αριστερά και δεξιά του. Τέτοια στοιχεία είναι π.χ. οι παράγραφοι (<p>) και οι επικεφαλίδες (<h...>).

### Τα στοιχεία div και span

Εκτός από τα “κανονικά” HTML στοιχεία, υπάρχουν δύο στοιχεία “γενικής χρήσης” που παρέχονται για να βοηθήσουν στο “στήσιμο” της ιστοσελίδας. Τα στοιχεία αυτά δεν έχουν σημασιολογική ερμηνεία (semantics) και θα πρέπει να τα χρησιμοποιούμε μόνο όταν δεν μπορούμε να χρησιμοποιήσουμε τα “κανονικά”:

- Το block στοιχείο **div**.
- Το inline στοιχείο **span**.

Συχνά, για να επιλέξουμε στοιχεία div ή span, χρησιμοποιούμε κλάσεις ή ids (βλ. προηγούμενο εργαστήριο).

## Εναλλακτικές Μέθοδοι Τοποθέτησης

Το πρότυπο CSS, εκτός από τη μέθοδο **static**, παρέχει τις εξής εναλλακτικές μεθόδους τοποθέτησης, οι οποίες ενεργοποιούνται μέσω της ιδιότητας **position**:

### A) Τοποθέτηση *relative*.

- Πρώτα υπολογίζεται η θέση του στοιχείου με την “κανονική ροή” και μετά το στοιχείο μετατοπίζεται **σε σχέση με την κανονική του θέση**.
- Ο χώρος που έπιανε το στοιχείο στην κανονική ροή **δεν καλύπτεται από άλλο στοιχείο**.
- Στη νέα του θέση, το στοιχείο **μπορεί να υπερκαλύπτει** (overlap) άλλα γειτονικά στοιχεία.

Σύμφωνα με τα προηγούμενα, η τοποθέτηση *relative* έχει περιορισμένη χρησιμότητα: σχεδόν πάντα χρησιμοποιείται για άλλον σκοπό (βλ. επόμενη μέθοδο τοποθέτησης).

Στο επόμενη παράδειγμα, ένα στοιχείο `p` με `id` “testpar” τοποθετείται 40 pixels δεξιά και 10 pixels πιο κάτω από την κανονική του θέση:

```
p#testpar { position: relative; top: 10px; left: 40px; }
```

### B) Τοποθέτηση *absolute*.

- Εδώ το στοιχείο βγαίνει τελείως έξω από την “κανονική ροή”: δεν εξαρτάται από την “κανονική” του θέση ούτε κρατείται κενός ο χώρος που έπιανε αρχικά. Η θέση του καθορίζεται από (κάποιες από) τις ιδιότητες **top**, **right**, **bottom** και **left** σε σχέση με το **περιβάλλον στοιχείο αναφοράς**.
- Εξ'ορισμού, το περιβάλλον στοιχείο αναφοράς (πού είναι το αρχικό σημείο, στο οποίο προστίθενται τα `left`, `top` κλπ) είναι **η συνολική ιστοσελίδα** (στοιχείο `body`). Αν θέλετε να έχετε ως αναφορά ένα άλλο στοιχείο – πατέρα του τρέχοντος, **τοποθετήστε τον πατέρα με τη μέθοδο *relative*** (χωρίς να προσδιορίζετε μέγεθος μετατόπισης για τον πατέρα). Το ίδιο συμβαίνει αν ο πατέρας τοποθετηθεί ως ***absolute***.
- Τα περιθώρια ενός στοιχείου με τοποθέτηση *absolute* **δεν συγχωνεύονται ποτέ**.

Η μέθοδος τοποθέτησης ***absolute*** χρησιμοποιείται συχνά για το “στήσιμο” ιστοσελίδων, όπου ανεξάρτητα τμήματα κειμένου ή εικόνες πρέπει να πάνε σε συγκεκριμένη θέση (βλ. και Παράστημα Γ).

### Γ) Τοποθέτηση *fixed*.

Παρόμοια τοποθέτηση με τη μέθοδο *absolute*, με τη διαφορά ότι το σημείο αναφοράς είναι πάντα το παράθυρο του browser: αυτό έχει ως συνέπεια το στοιχείο να είναι **ακίνητο κατά το scrolling**.

Η μέθοδος τοποθέτησης ***fixed*** χρησιμοποιείται σε ειδικές μόνο περιπτώσεις, π.χ. για να κρατάει την κορυφή μιας ιστοσελίδας (header) ακίνητη, ανεξάρτητα από το scrolling του χρήστη.

## Δ) Η χρήση των floats.

Αν και δεν αποτελούν μία ακόμα μέθοδο τοποθέτησης (positioning), τα **floats** χρησιμοποιούνται ευρέως για το στήσιμο ιστοσελίδων!

Σε προηγούμενο εργαστήριο χρησιμοποιήσαμε την ιδιότητα **float** για να μετακινήσουμε εικόνες αριστερά ή δεξιά του κειμένου. Το ίδιο μπορεί να γίνει με κάθε στοιχείο HTML, αρκεί να ορίσουμε το μέγεθός του (οι εικόνες έχουν εξ'ορισμού μέγεθος, ενώ π.χ. μια παράγραφος όχι – απλώνεται όσο περισσότερο γίνεται).

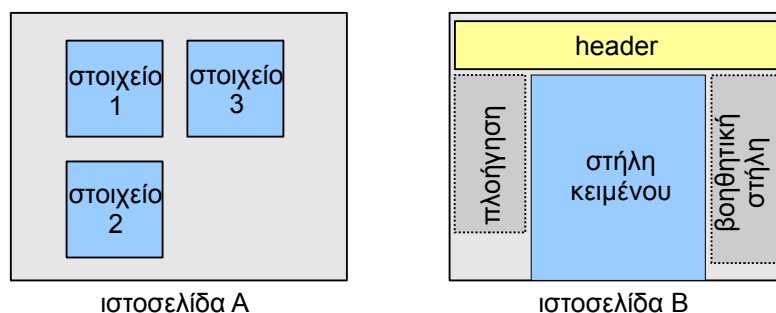
```
p#test { float: left; width:200px; }
```

Float γίνεται αριστερά ή δεξιά: ένα στοιχείο float αριστερά (δεξιά) βγαίνει από την “κανονική ροή” και μετατοπίζεται όσο αριστερότερα (δεξιότερα) είναι δυνατόν. Τα επόμενα στοιχεία “ρέουν” στα δεξιά (αριστερά) του στοιχείου float. Τα περιθώρια ενός στοιχείου float δεν συγχωνεύονται ποτέ.

Η ιδιότητα **clear** με τιμές [*left, right, both*] περιγράφει ποια πλευρά ενός στοιχείου δεν μπορεί να είναι γειτονική με ένα προηγούμενο στοιχείο float. Και η ιδιότητα αυτή χρειάζεται για το στήσιμο ιστοσελίδων (βλ. Παράρτημα Γ).

## Παράρτημα Γ: Βασικές τεχνικές “στησίματος” ιστοσελίδων.

Όσο κι αν ακούγεται απίστευτο, το πρότυπο CSS δεν έχει καταλήξει (ακόμα) στον οριστικό τρόπο “στησίματος” ιστοσελίδων. Έτσι, πρέπει να αρκεστούμε στα εργαλεία που είδαμε στα Παραρτήματα Α και Β. Οι τεχνικές που χρησιμοποιούνται είναι πάρα πολλές: εδώ παρουσιάζονται κάποιες βασικές παραλλαγές μόνο!



Σχήμα 2

Πάρα πολλές ιστοσελίδες έχουν μία από τις δύο μορφές του σχήματος 2:

- Η πρώτη μορφή (A) εμφανίζεται σε αρχικές ιστοσελίδες. Οι σελίδες αυτές είναι το σημείο εισόδου προς άλλες σελίδες περιεχομένου. Εδώ μας ενδιαφέρει κυρίως η στυλιστική αισθητική κι όχι το περιεχόμενο. Πρέπει να έχουμε απόλυτο έλεγχο της εμφάνισης (συντάσσεται σε επίπεδο pixel). Στην ιστοσελίδα εμφανίζονται μεμονωμένα στοιχεία: εικόνες, πολύ σύντομο

κείμενο, logos κ.οκ. Οι σελίδες αυτού του τύπου πρέπει να δείχνουν όλα τα στοιχεία “σε μία οθόνη” και να μην απαιτούν scrolling.

- Η δεύτερη μορφή (B) αντιπροσωπεύει σελίδες κειμένου. Είναι διατεταγμένη σε στήλες: μία κεντρική με το περιεχόμενο κείμενο, μία δεύτερη στήλη με μενού πλοήγησης και ενδεχομένως μια τρίτη στήλη με βοηθητικούς συνδέσμους και μικρά κομμάτια κειμένου. Σχεδόν πάντα υπάρχει ένα τμήμα στην κορυφή (header) και ένα αντίστοιχο στο τέλος (footer). Στις σελίδες αυτού του τύπου μας ενδιαφέρει πιο πολύ η ευχρηστία. Ο λεπτομερής έλεγχος θέσης των στοιχείων δεν είναι αναγκαίος, ενώ το scrolling για την εμφάνιση περισσότερου κειμένου είναι σύνηθες.

### A) Σταθερό σχήμα (fixed layout).

Σύμφωνα με την τεχνική αυτή, θεωρούμε ότι όλα τα βασικά block στοιχεία της ιστοσελίδας (συνήθως χρησιμοποιούνται στοιχεία **div**) έχουν σταθερό και προκαθορισμένο μέγεθος (**width**, **height**, σε pixels ή %) και θέση. Η τοποθέτηση είναι **absolute** και οι ιδιότητες **left**, **top** χρησιμοποιούνται για κάθε στοιχείο.

Τα πάντα βρίσκονται μέσα σε ένα στοιχείο **div** (“**main container**”), το οποίο έχει τοποθέτηση **relative** (ώστε να αποτελεί το περιβάλλον στοιχείο αναφοράς) και **σταθερό μέγεθος** (συνήθίζεται να είναι 760 ή 960 pixels). Η δήλωση των περιθωρίων για το στοιχείο αυτό είναι:

```
#main-container { margin: 0 auto; }
```

κάτι που σημαίνει “χωρίς περιθώριο πάνω και κάτω, κεντραρισμένο αριστερα-δεξιά”.

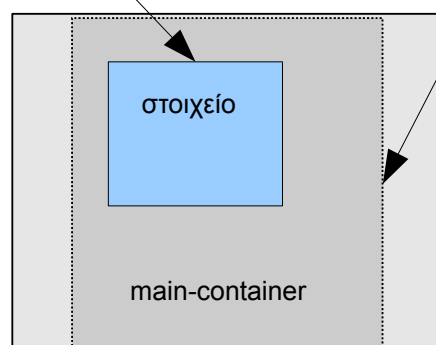
Το περιβάλλον αυτό στοιχείο προσφέρει πάντα σταθερό συνολικό μέγεθος και αποτελεί το πλαίσιο αναφοράς για τα μεγέθη των στοιχείων μέσα σε αυτό.

Επιπλέον, επειδή κάθε browser έχει τη δική του ιδέα για τα εξ'ορισμού margins και paddings κάθε στοιχείου (ακόμα και του body), συχνά φροντίζουμε πριν από κάθε άλλο να εξουδετερώσουμε τα προκαθορισμένα αυτά νούμερα:

```
* { margin: 0; padding: 0; }
```

position: absolute;  
top: ...px; left: ...px;  
width: ...px; height: ...px;

margin: 0 auto;  
position: relative;  
width: ...px;



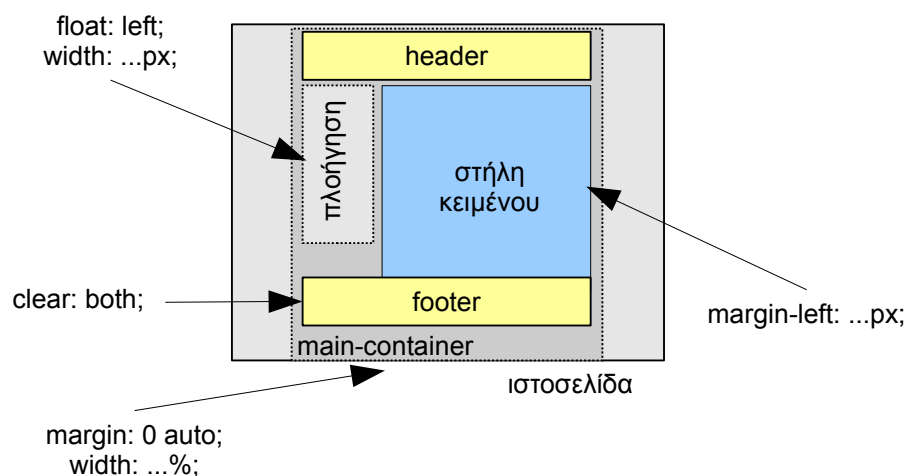
ιστοσελίδα

Σχήμα 3

Συνοπτικά η ιδέα φαίνεται στο σχήμα 3. Το σταθερό σχήμα μας επιτρέπει τον απόλυτο έλεγχο τοποθέτησης και είναι κατάλληλο για αρχικές ιστοσελίδες. Από την άλλη πλευρά, προσαρμόζεται πολύ δύσκολα σε οθόνες με διαφορετικές διαστάσεις, ιδίως σε πολύ μικρές συσκευές.

## B) Ρευστό σχήμα (fluid layout).

Στο σχήμα αυτό χρησιμοποιούνται float στοιχεία (**div**) και τα μεγέθη προσαρμόζονται στο τρέχον μέγεθος του παραθύρου. Δεν έχετε τον απόλυτο έλεγχο τοποθέτησης και τα διάφορα στοιχεία παρουσιάζονται ως στήλες περιεχομένου. Το σχήμα είναι ιδιαίτερα κατάλληλο για ιστοσελίδες “κειμένου”.



Σχήμα 4

Το σχήμα 4 δείχνει πώς μπορεί να κατασκευαστεί ένα σχήμα με 2 στήλες, header και footer.

- Η στήλη πλοήγησης είναι **float:left** και έχει σταθερό μέγεθος. Η προσαρμοστικότητα του σχήματος οφείλεται στο πλάτος της κύριας στήλης κειμένου που μπορεί να αυξομειώνεται.
- Δώστε το κατάλληλο αριστερό περιθώριο στο κυρίως κείμενο (**left-margin**) για να το εμποδίσετε να τυλιχτεί κάτω από τη στήλη πλοήγησης.
- Εφόσον υπάρχουν στοιχεία float, είναι σημαντικό το στοιχείο footer να έχει την ιδιότητα **clear: both** για να αποφύγει πιθανή τοποθέτηση στα δεξιά του στοιχείου float.
- Το ίδιο σχήμα επεκτείνεται στις 3 στήλες, με το αντίστοιχο στοιχείο float δεξιά.
- Μπορείτε να περιορίσετε την κύρια στήλη με τις ιδιότητες ελάχιστου ή/και μέγιστου μήκους (**min-width** και **max-width** αντίστοιχα).

Μια δεύτερη παραλλαγή χρησιμοποιεί το float:left **για όλες τις στήλες** (στοιχεία **div**), με μέγεθος σε ποσοστά (%). Στην περίπτωση αυτή θα πρέπει να θυμάστε ότι το περιβάλλον στοιχείο, αν περιέχει μόνο floats, δεν τα περικλείει στο ύψος! Αυτό ίσως δημιουργήσει προβλήματα, όπως π.χ. όταν περιμένετε ότι το χρώμα φόντου του



περιβάλλοντος στοιχείου θα φαίνεται έως το τέλος των στηλών που περικλείει (αυτό δεν θα συμβεί!). Υπάρχουν διάφορες λύσεις για το θέμα: ψάξτε στο διαδίκτυο για “css containing floats”.

### Μια σημασιολογική προσθήκη...

Στις διάφορες μεθόδους τοποθέτησης, τα στοιχεία που χρησιμοποιούνται σχεδόν αποκλειστικά είναι τύπου **div**. Η νεώτερη έκδοση της HTML όμως, προσφέρει τα παρακάτω στοιχεία με σημασιολογική ερμηνεία:

- **nav**, για στοιχεία που περιέχουν συνδέσμους πλοήγησης στην σελίδα/site
- **header**, για στοιχεία στην κορυφή της σελίδας και
- **footer**, για τα στοιχεία που βρίσκονται στο τέλος.

Αν και η υποστήριξη από όλους τους browsers δεν είναι ακόμα πλήρης (<http://caniuse.com/#feat=html5semantic>), μπορείτε να τα χρησιμοποιήσετε αντί των γενικών στοιχείων div!