

CSS

Στο παρελθόν οι πίνακες αποτελούσαν βασικό εργαλείο για το σχεδιασμό της διάταξης («σκελετού») της σελίδας (page layout). Όμως, με τον ερχομό του CSS η χρήση τους αυτή σιγά σιγά εγκαταλείφθηκε εξαιτίας του ότι οι πίνακες γενικά φτιάχνουν πολύπλοκές -«πολυσύνθετες» σελίδες με πολλές ετικέτες κάνοντας αφενός τον σχεδιασμό της σελίδας δυσκολότερο και αφετέρου οι σελίδες δεν είναι φιλικές προς τις μηχανές αναζήτησης. Καθορίζουν το style, τη δομή και τη θέση των στοιχείων μέσα στο αρχείο.

Κατηγορίες Στοιχείων (Block level & Inline)

Inline στοιχεία (elements)

Μπορούν να περιέχουν μόνο κείμενο ή άλλα inline στοιχεία.

Κατά την στοιχειοθέτησή τους, τα περιεχόμενά τους δεν ξεκινούν σε νέα γραμμή. Δηλαδή δεν «χαλάνε» τη ροή του κειμένου (line breaks).

Τα inline στοιχεία «δεν έχουν» διαστάσεις μήκος (width) και ύψος (height). Τομήκος και το ύψος τους καθορίζεται αυτόματα (**auto**) από το μήκος και το ύψος του περιεχομένου τους. Παράδειγμα οι ετικέτες ****, **<i>**, ****, **
, **, **<u>**, **<a>**, ****, ****

Block level στοιχεία (elements)

Μπορούν να περιέχουν κείμενο, άλλα block level και **inline** στοιχεία.

Εξ' ορισμού τα block level στοιχεία ξεκινούν σε καινούργια γραμμή και όταν τελειώνουν αλλάζουν γραμμή (line breaks)

Έχουν μήκος εξ' ορισμού ίσο με όλο το μήκος (width 100%) του στοιχείου που τα περιέχει (parent element), ενώ το ύψος τους καθορίζεται αυτόματα από το ύψος του περιεχομένου τους (height auto). Παράδειγμα οι ετικέτες **<p>**, **<h1>**...**<h6>**, **<hr>**, **<pre>**, ****, ****, ****, **<table>**, **<tr>**, **<div>**

Προτεραιότητες

Εάν έχουν προσδιοριστεί περισσότερα του ενός στυλ στη σελίδα, η σειρά (προτεραιότητα) με την οποία αυτά θα εφαρμοστούν είναι η ακόλουθη:

1. Στυλ που προσδιορίζονται μέσα στην ετικέτα ενός HTML στοιχείου (inline style sheets)
2. Στυλ που προσδιορίζονται μέσα στο HEAD της σελίδας HTML (internal style sheets)
3. Εξωτερικά φύλλα μορφοποίησης (external style sheets)
4. Τα στυλ που ορίζει ο φυλλομετρητής (browser default styles)

Τρόποι εισαγωγής style sheet

- **Inline styles**

Καθορίζει το style για συγκεκριμένα elements
Χρήση της ιδιότητας "style" μέσα σε **ετικέτα**
Μπορεί να καθορίσει πολλά properties

Παράδειγμα: `<p style = "font-size: 20pt; color: #0000ff">This text has the <em>font-size</em> and <em>color</em> styles applied to it, making it 20pt. and blue.</p>`

- **Internal styles sheets**

Ορίζεται μέσα στο <head> με την **ετικέτα** <style>
Επηρεάζει τα elements στα οποία αναφέρεται

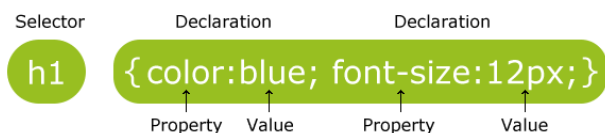
Παράδειγμα: <head>
 <style >
 h1 { font-family: arial, sans-serif }
 p { font-size: 14pt }
 </style>
 </head>

- External style sheets

Δημιουργία ξεχωριστού αρχείου css με το οποίο συνδέεται το html αρχείο
 Ιδανικό όταν το ίδιο style sheet εφαρμόζεται σε περισσότερα από ένα αντικείμενα

Παράδειγμα: <head>
 <title>External Style Sheets</title>
 <link rel = "stylesheet" type = "text/css" href = "styles.css">
 </head>

CSS σύνταξη



- όπου **Selector** ένα html tag, π.χ. body, p, h1,...
- όπου **Property** το χαρακτηριστικό (style attribute) που θέλουμε να αλλάξουμε, π.χ. color, font-family, text-align
- όπου **Value** μια τιμή για το Property

Παράδειγμα: h1 {color: blue; font-size= 16pt}

Ιδιότητες που αφορούν μορφοποίηση κειμένου

- color : Αφορά το χρώμα του κειμένου, αλλά αν δεν οριστεί χρώμα περιγράμματος (μέσω της ιδιότητας border-color), ο browser χρησιμοποιεί αυτό που ορίστηκε στην ιδιότητα color.
- font-size : Αφορά το μέγεθος της γραμματοσειράς.
- font-family : Η ιδιότητα αυτή μας επιτρέπει να ορίσουμε ένα πλήθος γραμματοσειρών που θα χρησιμοποιηθούν για το κείμενο, κατά σειρά προτίμησης.
- font-style : Αφορά μόνο την περίπτωση όπου το κείμενο θα είναι πλάγιο. Οι πιθανές τιμές της είναι normal, italic και oblique.
- font-weight : Αφορά το «βάρος» της γραμματοσειράς και στην πράξη χρησιμοποιείται για να ορίσει αν το κείμενο μας θα είναι έντονο ή όχι
- text-decoration : Μας επιτρέπει να εφαρμόσουμε στο κείμενό μας διάφορα εφέ, συμπεριλαμβανόμενης και της υπογράμμισης.
- text-align : Μας επιτρέπει να καθορίσουμε τη στοίχιση του κειμένου μας
- font : Μας επιτρέπει να καθορίσουμε αρκετές ιδιότητες κειμένου με μια μόνο

ιδιότητα (π.χ. τις font-size, font-family, font-weight, font-style κ.ά.)

Ιδιότητες που αφορούν το φόντο

- background-color : Χρησιμοποιείται για να ορίσει χρώμα φόντου στα στοιχεία που αφορά ο επιλογέας.
- background-image : Χρησιμοποιείται για να ορίσει μια εικόνα φόντου. Η τιμή που δέχεται είναι της μορφής url(διεύθυνση_εικόνας).
- background-repeat : Αφορά τον τρόπο που θα επαναλαμβάνεται η εικόνα που ορίσαμε στην παραπάνω ιδιότητα. Οι τιμές που δέχεται είναι: no-repeat, repeat-x, repeat-y, repeat.

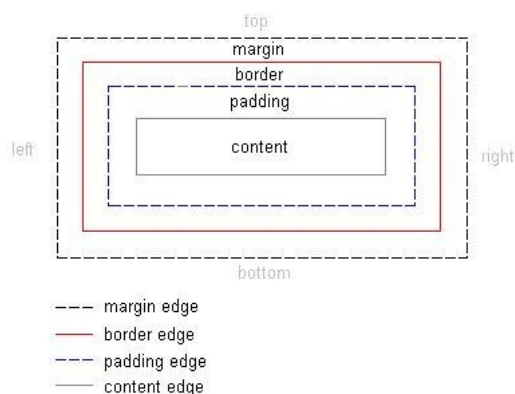
- background-position : Μας επιτρέπει να ορίσουμε τη θέση που τοποθετηθεί η εικόνα φόντου που ορίσαμε με την ιδιότητα background-image τόσο κάθετα (top, center, bottom), όσο και οριζόντια (left, center, right).
- background : Αποτελεί ουσιαστικά συντόμευση για να ορίσουμε με μία μόνο ιδιότητα όλες τις παραπάνω (όπως είναι η ιδιότητα font για τα χαρακτηριστικά κειμένου).

Ιδιότητες που αφορούν το περίγραμμα

- border-color : Ρυθμίζει το χρώμα περιγράμματος.
- border-width : Ρυθμίζει το πάχος του περιγράμματος σε κάποια από τις μονάδες μέτρησης που μπορούν να χρησιμοποιηθούν, π.χ. σε pixels.
- border-style : Ορίζει το στυλ του περιγράμματος. Οι τιμές που χρησιμοποιούνται συνήθως είναι : solid, dashed, dotted, double. Επίσης, και οι τιμές groove, ridge, inset, outset.
- border : Αποτελεί ουσιαστικά συντόμευση για να ορίσουμε με μία μόνο ιδιότητα όλες τις παραπάνω.

Τι κάνουμε αν θέλουμε να ορίσουμε διαφορετικό πάχος/χρώμα/στυλ περιγράμματος για το πάνω/δεξί/κάτω/αριστερό περίγραμμα; Κάθε ιδιότητα της μορφής border-κάτι αντιστοιχεί σε τέσσερις ιδιότητες της μορφής border-top-κάτι, border-right-κάτι, border-bottom-κάτι, border-left-κάτι. Π.χ. border-left-width: 1px; border-right-width: 2px;

Διάφορες χρήσιμες ιδιότητες

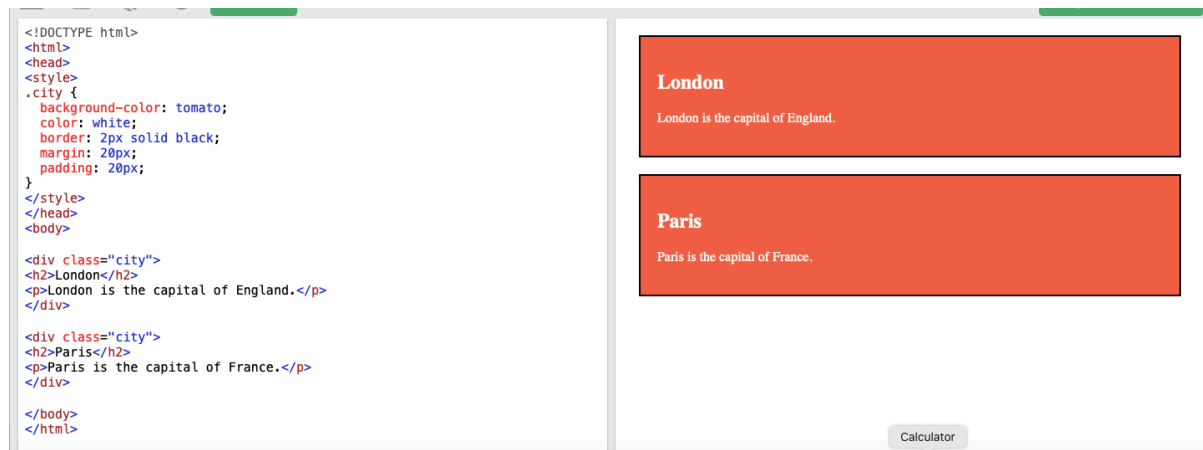


- padding : Ορίζει το κενό που θα υπάρχει μεταξύ των ορίων ενός στοιχείου και των περιεχομένων του. Μπορούμε ομοίως με το border, είτε να χρησιμοποιήσουμε τις ιδιότητες padding-top, padding-right, padding-bottom, padding-left, είτε να εισάγουμε και τις τέσσερις τιμές στην ιδιότητα padding με τη σειρά [top] [right] [bottom] [left], είτε [top,bottom] [right,left].
- margin : Το αντίθετο ουσιαστικά του padding. Ορίζει τον χώρο μεταξύ των ορίων ενός στοιχείου και όσων το περιβάλλουν. Και εδώ μπορούμε να χρησιμοποιήσουμε τις επιμέρους ιδιότητες margin-top, margin-right, margin-bottom, margin-left.
- width, height : Ορίζουν το πλάτος και το ύψος ενός στοιχείου, είτε σε ποσοστό (βάσει του στοιχείου που το περιέχει), είτε σε κάποια μονάδα μήκους.
- float : Δέχεται τις τιμές right, left και none και επιτρέπει τα στοιχεία που περιβάλλουν το στοιχείο εκείνο στο οποίο εφαρμόζουμε την ιδιότητα αυτή να «ρέουν» τριγύρω του. Μια συνηθισμένη χρήση της ιδιότητας είναι στις εικόνες που συνοδεύουν ένα άρθρο.

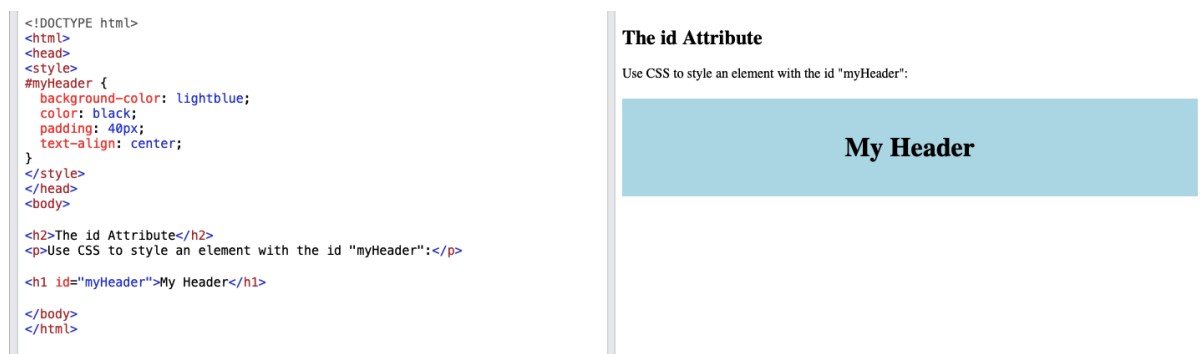
Οι ιδιότητες Class και Id (Ομαδοποίηση στοιχείων)

Μερικές φορές θέλουμε να εφαρμόσουμε ένα ειδικό στυλ (style) σ' ένα συγκεκριμένο στοιχείο (element) ή σε μια συγκεκριμένη ομάδα στοιχείων (group of elements). Θα δούμε το πώς μπορούμε να χρησιμοποιήσουμε το class και το id για να ορίσουμε ιδιότητες για επιλεγμένα στοιχεία.

Έτσι, επιπλέον της δυνατότητας ορισμού ενός στυλ για ένα HTML στοιχείο, το CSS μας επιτρέπει να ορίσουμε τη δική μας κλάση για να μορφοποιήσουμε ένα ή περισσότερα HTML στοιχεία υπό τη μορφή του ".class". Χρησιμοποιεί το class attribute του HTML element, και ορίζεται με μια "."



Με την ιδιότητα **id** προσδίδουμε στα στοιχεία της σελίδας μας ένα **μοναδικό** όνομα το οποίο το χρησιμοποιούμε για να αναφερόμαστε στα στοιχεία αυτά είτε μέσα από αρχεία script είτε μέσα από CSS στυλ. Δεν επιτρέπεται να υπάρχουν στην ίδια σελίδα δύο ή περισσότερα στοιχεία με το ίδιο όνομα. Ορίζεται με ένα **"#"**.



Ομαδοποίηση των Στοιχείων με Span και Div

Τα στοιχεία `<span>` και `<div>` χρησιμοποιούνται για να ομαδοποιήσουν και να δομήσουν ένα έγγραφο και συνδυάζονται συχνά με τις ιδιότητες class και id.

Θα δούμε από κοντά τη χρήση των `<span>` και `<div>` καθώς αυτά τα δύο στοιχεία της HTML έχουν μεγάλη σπουδαιότητα για τα CSS.

Ομαδοποίηση με το Span

Το στοιχείο `<span>` δεν επιφέρει καμιά οπτική (visual) αλλαγή στην σελίδα μας. Μπορεί να περιέχει μόνο inline στοιχεία και κείμενο. Πρέπει πάντα να κλείνει με `</span>` Χρησιμοποιείται για να ομαδοποιήσουμε συνήθως inline στοιχεία ή να διαχωρίσουμε συγκεκριμένο τμήμα κειμένου έτσι ώστε να μπορούμε μετά να τα/ο μορφοποιήσουμε χρησιμοποιώντας CSS. Το στοιχείο `<span>` είναι inline σε αντίθεση με το `<div>` που είναι block level.

Ακολουθεί ένα παράδειγμα :

`<p>Στην Περιφέρεια Δυτικής Μακεδονίας ανήκουν οι Νομοί Γρεβενών, Καστοριάς, Κοζάνης και Φλώρινας.</p>`

Ας υποθέσουμε ότι θέλουμε στο παραπάνω κομμάτι κειμένου να εμφανίζονται οι τέσσερις νομοί της Δυτικής Μακεδονίας με κόκκινο χρώμα (red). Για να γίνει αυτό, μπορούμε να απομονώσουμε τους νομούς με το `<span>` και σε κάθε span να αντιστοιχίσουμε μια τάξη (class), στην οποία μπορούμε να προσδώσουμε ένα χρώμα με το style sheet.

<p> Στην Περιφέρεια Δυτικής Μακεδονίας ανήκουν οι Νομοί

 Γρεβενών,

 Καστοριάς,

 Κοζάνης και

 Φλώρινας.

</p>

Το CSS θα είναι το εξής :

```
span.nomos {
```

```
color:red;
```

```
}
```

```
<!DOCTYPE html>
<html>
<head>
<style>
.nomos {
color:red;
}
</style>
</head>
<body>

<p> Στην Περιφέρεια Δυτικής Μακεδονίας ανήκουν οι Νομοί
<span class="nomos"> Γρεβενών</span>,
<span class="nomos"> Καστοριάς</span>,
<span class="nomos"> Κοζάνης</span> και
<span class="nomos"> Φλώρινας</span>.
</p>

</body>
</html>
```

Στην Περιφέρεια Δυτικής Μακεδονίας ανήκουν οι Νομοί Γρεβενών, Καστοριάς, Κοζάνης και Φλώρινας.

Η Ετικέτα <div>

Η ετικέτα <div> ορίζει μια ενότητα /τμήμα (division) στην σελίδα μας. Είναι block level στοιχείο και μπορεί να περιέχει block level και inline στοιχεία. Μπορεί να περιέχει όλες τις ετικέτες που μπαίνουν στο <body> όπως για παράδειγμα <div>, <p>, <h1>,,,<table>,,<a> κτλ

Πρέπει πάντα να κλείνει με </div>. Χρησιμοποιείται για να ομαδοποιήσουμε συνήθως block level (και inline) στοιχεία έτσι ώστε να μπορούμε μετά να τα μορφοποιήσουμε χρησιμοποιώντας CSS.

Η <div> μοιάζει με την παράγραφο μόνο που «διαχωρίζει» την σελίδα σε μεγαλύτερες ενότητες/τμήματα. Μας δίνει την δυνατότητα να ονομάσουμε (συνήθως μέσω των ιδιοτήτων id, class και style) και να μορφοποιήσουμε μέσω CSS συγκεκριμένες ενότητες στη σελίδα μας. Οι ετικέτες <div> με την βοήθεια του CSS αποτελούν σήμερα τα κύρια εργαλεία για το σχεδιασμό της διάταξης («σκελετού») της σελίδας (page layout).

```
<div style="background-color:black;color:white;padding:20px;">
```

```
<h2>London</h2>
```

```
<p>London is the capital city of England. It is the most populous city in the
United Kingdom, with a metropolitan area of over 13 million inhabitants.</p>
```

```
</div>
```

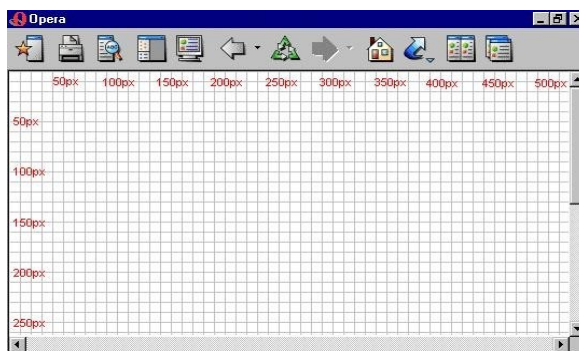
London

London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.

Standing on the River Thames, London has been a major settlement for two millennia, its history going back to its founding by the Romans, who named it Londinium.

CSS-P (Positioning) πρότυπο

Με την ιδιότητα `position` των CSS, μπορούμε να τοποθετήσουμε ένα στοιχείο ακριβώς εκεί που θέλουμε να βρίσκεται μέσα στην ιστοσελίδα. Μαζί με την ιδιότητα `float`, η ιδιότητα `position` μάς δίνει πολλές δυνατότητες για να δημιουργήσουμε μια προηγμένη και ακριβή διάταξη (layout). Για να μπορέσουμε να κατανοήσουμε την ιδιότητα `position`, θα πρέπει να φανταστούμε ένα παράθυρο φυλλομετρητή ως ένα σύστημα συντεταγμένων :



Το πρότυπο αυτό παρέχει ιδιότητες (`top`, `bottom`, `left`, και `right`) που προσδιορίζουν την τοποθέτηση ενός στοιχείου. Η τιμή της `position` μπορεί να είναι:

- **Static** - Τα HTML στοιχεία τοποθετούνται by default σε static μορφή. Ένα static στοιχείο τοποθετείται πάντα σύμφωνα με την κανονική ροή της σελίδας. Δεν έχουν επίδραση οι ιδιότητες `top`, `bottom`, `left`, και `right`.
- **Fixed** - Ένα στοιχείο με `fixed position` τοποθετείται σε σταθερή θέση σχετικά με το browser window και τις ιδιότητες `top`, `bottom`, `left`, και `right`. Δεν θα μετακινηθεί ακόμη και εάν το παράθυρο γίνει scrolled.
- **Inherit** - Προσδιορίζει ότι η τιμή της ιδιότητας `position` του HTML στοιχείου θα κληρονομηθεί από το στοιχείο πατέρα.
- **Relative** - Για να τοποθετήσουμε ένα στοιχείο σε σχετική θέση (relative positioning), θα πρέπει να δώσουμε την τιμή `relative` στην ιδιότητα `position`. Η θέση ενός στοιχείου που τοποθετείται σχετικά υπολογίζεται από την αρχική θέση που είχε στο έγγραφο. Αυτό σημαίνει ότι μπορούμε να μετακινήσουμε το στοιχείο προς τα δεξιά (`right`), προς τα αριστερά (`left`), προς τα πάνω (`up`) ή προς τα κάτω (`down`).
- **Absolute**

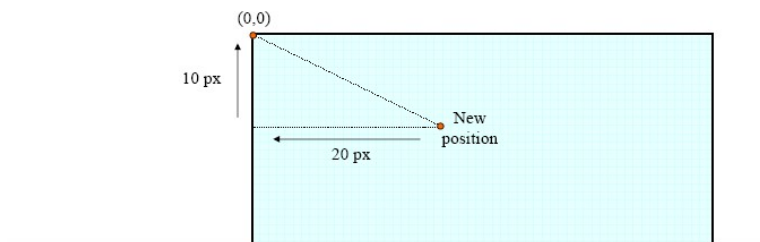
Θέση αντικειμένου

• `position: absolute`

Τοποθετεί ένα αντικείμενο σε μία συγκεκριμένη θέση που προσδιορίζεται από τα χαρακτηριστικά `top` και `left`.

`p { position: absolute; left: 20px; top: 10px }`

Η παράγραφος τοποθετείται 20 pixel δεξιότερα από το αριστερό άκρο της σελίδας και 10 pixel πιο κάτω από το πάνω άκρο της σελίδας.



Η Ιδιότητα Clear

Η ιδιότητα clear χρησιμοποιείται για να ελέγχει το πώς θα συμπεριφέρονται τα επόμενα στοιχεία των στοιχείων που επιπλέουν. Εξ ορισμού, τα επόμενα στοιχεία μετακινούνται για να γεμίσουν τον διαθέσιμο χώρο ο οποίος θα ελευθερωθεί όταν ένα πλαίσιο (box) τοποθετείται σε μια πλευρά.

Η ιδιότητα clear μπορεί να λάβει μια από τις τιμές left, right, both ή none.

```
<div id="picture">

</div>
<h1>Bill Gates</h1>
<p class="floatstop">causas naturales et antecedentes, idcirco etiam nostrarum voluntatum...</p>
```

Για να εμποδίσουμε το κείμενο από το να πάει δίπλα στην εικόνα, μπορούμε να προσθέσουμε τα εξής στον κώδικα των CSS :

```
#picture {
float:left; width: 100px;
}
.floatstop {
clear:both;
}
```

Σημασιολογικά Στοιχεία (HTML5)



Ετικέτα	Περιγραφή
<code><article></code>	Defines independent, self-contained content
<code><aside></code>	Defines content aside from the page content
<code><details></code>	Defines additional details that the user can view or hide
<code><figcaption></code>	Defines a caption for a <code><figure></code> element
<code><figure></code>	Specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.

<code><footer></code>	Defines a footer for a document or section
<code><header></code>	Specifies a header for a document or section
<code><main></code>	Specifies the main content of a document
<code><mark></code>	Defines marked/highlighted text
<code><nav></code>	Defines navigation links
<code><section></code>	Defines a section in a document
<code><summary></code>	Defines a visible heading for a <code><details></code> element
<code><time></code>	Defines a date/time

Ασκήσεις-Παραδείγματα

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <title>Example of CSS Embedded Style Sheet</title>
6   <style>
7     body { background-color: YellowGreen; }
8     p { color: #fff; }
9   </style>
10 </head>
11 <body>
12   <h1>This is a heading</h1>
13   <p>This is a paragraph of text.</p>
14 </body>
15 </html>

```

Στο δεύτερο παράδειγμα χρησιμοποιούμε το CSS για να μορφοποιήσουμε κάθε ομάδα με τα χρώματά της (πράσινο, κόκκινο και κίτρινο).

Εδώ βλέπουμε το αποτέλεσμα στο browser:

Χρώματα ομάδων

Η **ΑΕΚ**, ο **ΠΑΟ** και ο **Ολυμπιακός** είναι ομάδες. Η **ΑΕΚ** φοράει κιτρινόμαυρα, ο **Παναθηναϊκός** πράσινο άσπρο και ο **Ολυμπιακός** κόκκινο άσπρο.

ΑΕΚ

Κείμενο για την ΑΕΚ

ΠΑΟ

Κείμενο για τον ΠΑΟ

ΟΣΦΠ

Κείμενο για τον Ολυμπιακό

Και εδώ βλέπουμε τον κώδικα:

```

1  <!DOCTYPE html><html><head><style>
2
3  div {width: 400px;}
4
5  .kitrino {color: yellow; background-color: black;}
6  .prasino {color: white; background-color: RGB(4, 113, 58);}
7  .kokkino {color: white; background-color: RGB(227, 6, 19);}
8
9  </style></head>
10
11 <body><h1>Χρώματα ομάδων</h1>
12
13 <div>Η <span class = "kitrino">ΑΕΚ</span>,
14 ο <span class = "prasino">ΠΑΟ</span> και ο
15 <span class = "kokkino">Ολυμπιακός</span> είναι ομάδες.
16 Η <span class = "kitrino">ΑΕΚ</span> φοράει κιτρινόμαυρα,
17 ο <span class = "prasino">Παναθηναϊκός</span> πράσινο άσπρο
18 και ο <span class = "kokkino">Ολυμπιακός</span>
19 κόκκινο άσπρο.</div>
20
21 <h2 class = "kitrino">ΑΕΚ</h2>
22 <p class = "kitrino">Κείμενο για την ΑΕΚ</p>
23 <h2 class = "prasino">ΠΑΟ</h2>
24 <p class = "prasino">Κείμενο για τον ΠΑΟ</p>
25 <h2 class = "kokkino">ΟΣΦΠ</h2>
26 <p class = "kokkino">Κείμενο για τον Ολυμπιακό</p>
27 </body></html>

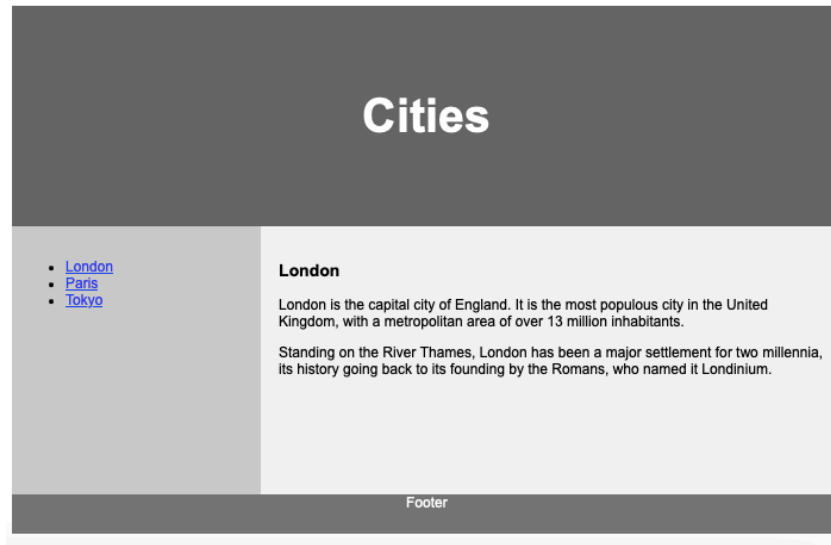
```

(* Προς αποφυγή παρεξηγήσεων: οι ομάδες τοποθετήθηκαν με βάση τη σειρά που τερμάτισαν στο πρωτάθλημα ποδοσφαίρου 2022-2023)

CSS Layout Float

In this example, we have created a header, two columns/boxes and a footer. On smaller screens, the columns will stack on top of each other.

Resize the browser window to see the responsive effect (you will learn more about this in our next chapter - HTML Responsive.)



```
<!DOCTYPE html>
<html>
<head>
<style>
.all-browsers {
margin: 8px;
padding: 5px;
background-color: lightgray;
}

.all-browsers > h1, .browser {
margin: 10px;
padding: 5px;
}

.browser {
background: white;
}

.browser > h2, p {
margin: 4px;
font-size: 90%;
}
</style>
</head>
<body>
<article class="all-browsers">
<h1>Most Popular Browsers</h1>
<article class="browser">
<h2>Google Chrome</h2>
<p>Google Chrome is a web browser developed by Google, released in 2008.</p>
</article>
<article class="browser">
<h2>Mozilla Firefox</h2>
<p>Mozilla Firefox is an open-source web browser developed by Mozilla. </p>
</article>
</article>
</body>
</html>
```

Most Popular Browsers

Google Chrome

Google Chrome is a web browser developed by Google, released in 2008. Chrome is the world's most popular web browser today!

Mozilla Firefox

Mozilla Firefox is an open-source web browser developed by Mozilla. Firefox has been the second most popular web browser since January, 2018.