Εισαγωγή στο CSS

Τι είναι το CSS (Cascading Style Sheets);

Το CSS είναι μια απλή γλώσσα που μας βοηθάει να ορίσουμε με σαφήνεια και ιδιαίτερη ευελιξία τον τρόπο με τον οποίο θα εμφανίζονται τα διάφορα στοιχεία στην ιστοσελίδα μας.

Ποια πλεονεκτήματα έχει η χρήση CSS;

- Πολύ μεγαλύτερη ευελιξία. Το CSS κατέστησε εφικτές μορφοποιήσεις οι οποίες ήταν αδύνατες ή πολύ δύσκολες με την κλασσική HTML.
- Ευκολότερη συντήρηση των ιστοσελίδων. Η εμφάνιση ενός ολόκληρου site μπορεί να ελέγχεται από ένα μόνο εξωτερικό αρχείο CSS. Έτσι, κάθε αλλαγή στο στυλ της ιστοσελίδας μπορεί να γίνεται με μια μοναδική αλλαγή σε αυτό το αρχείο, αντί για την επεξεργασία πολλών σημείων σε κάθε σελίδα που υπάρχει στο site.
- Μικρότερο μέγεθος αρχείου, δεδομένου ότι ο κάθε κανόνας μορφοποίησης γράφεται μόνο μια φορά και όχι σε κάθε σημείο που εφαρμόζεται.
- Καλύτερο SEO (Search engine optimization). Οι μηχανές αναζήτησης δεν «μπερδεύονται» ανάμεσα σε περιεχόμενο και τη μορφοποίηση του, αλλά έχουν πρόσβαση στο περιεχόμενο σκέτο, οπότε είναι πολύ ευκολότερο να το καταγράψουν και να το αρχειοθετήσουν (indexing).
- Γρηγορότερες σελίδες. Όταν χρησιμοποιούμε εξωτερικό αρχείο CSS (βλ. *Πως εισάγουμε CSS στη σελίδα μας;*), ο browser την πρώτη φορά που θα φορτώσει κάποια σελίδα του site μας το αποθηκεύει στην cache, οπότε δεν χρειάζεται να το κατεβάσει ξανά κάθε φορά που κατεβάζει ο χρήστης του κάποια άλλη σελίδα του site μας.

Βασικοί κανόνες σύνταξης

Σχόλια

Αν και η ανάγκη για σχόλια στο CSS δεν είναι τόσο επιτακτική όσο σε γλώσσες προγραμματισμού, μπορούμε να εισάγουμε σχόλια μέσα σε `/*...*/`
Παράδειγμα: `/*Αυτό είναι ένα σχόλιο*/`

CSS rules

Ένας «κανόνας» CSS αποτελείται από 2 μέρη: Τον επιλογέα (CSS selector) ο οποίος αφορά το ΤΙ θα μορφοποιηθεί και τις ιδιότητες (CSS properties) οι οποίες αφορούν το ΠΩΣ αυτό θα μορφοποιηθεί. Η σύνταξη είναι η εξής:

```
επιλογέας
{
      ιδιότητα1: τιμή1;
      ιδιότητα2: τιμή2;
      ιδιότητα3: τιμή3;
      ...
}
```

Απλοί επιλογείς CSS (CSS selectors)

Επιλέγουμε **ποια** στοιχεία θα μορφοποιήσουμε μέσω των επιλογέων CSS (CSS selectors), οι οποίοι ουσιαστικά αποτελούν κάποιους **«κανόνες»** ώστε να καταλάβει ο browser **που** θέλουμε να εφαρμοστούν τα όσα γράφουμε κάθε φορά. Υπάρχουν αρκετά περίπλοκοι CSS selectors στο specification του CSS, οι οποίοι προσφέρουν τεράστια ευελιξία, ωστόσο εδώ θα εξετάσουμε τους πιο βασικούς, και για λόγους απλούστευσης, και επειδή αρκετοί από τους πιο περίπλοκους δεν υποστηρίζονται από όλους τους browsers.

*****

Όταν ο επιλογέας είναι ένας χαρακτήρας αστερίσκου, τότε οι ιδιότητες που θα γράψουμε σε αυτόν τον κανόνα CSS εφαρμόζονται σε κάθε στοιχείο της σελίδας μας. Όπως είναι κατανοητό, συνήθως δεν είναι και πολύ χρήσιμος επιλογέας από μόνος του, και χρησιμοποιείται κυρίως σε συνδυασμό με άλλους.

στοιχείο

Όταν ο επιλογέας αποτελείται απλά από το όνομα ενός html tag, τότε οι ιδιότητες που θα γράψουμε σε αυτόν τον κανόνα CSS εφαρμόζονται σε κάθε τέτοιο στοιχείο html. Για παράδειγμα, ο επιλογέας `p` θα εφαρμοστεί σε οτιδήποτε στη σελίδα μας περιλαμβάνεται εντός των tags `<p>...</p>`, ο επιλογέας `table` θα εφαρμοστεί σε όλους τους πίνακες στη σελίδα μας, ο επιλογέας `img` θα αφορά όλες τις εικόνες στη σελίδα κοκ. Προφανώς όταν θέλουμε να εφαρμόσουμε κάποιες ιδιότητες CSS σε ολόκληρη τη σελίδα, χρησιμοποιούμε ως επιλογέα `body` μιας και όλο το ορατό τμήμα της σελίδας περιέχεται εντός των tags `<body>...</body>`.

.όνομα_κλάσης

Όταν ο επιλογέας μας περιλαμβάνει μια τελεία (.) στην αρχή του, τότε ο browser ψάχνει όσα στοιχεία στη σελίδα μας περιλαμβάνουν την ιδιότητα *class* και εφαρμόζει τις ιδιότητες που θα γράψουμε στον κανόνα CSS αυτό σε οποιοδήποτε στοιχείο περιλαμβάνει την κλάση «όνομα_κλάσης» στην ιδιότητα class του. Φυσικά ως *όνομα_κλάσης* μπορούμε να γράψουμε οτιδήποτε αποτελείται από γράμματα, αριθμούς, παύλες και χαρακτήρες underscore (_) και να ξεκινάει με γράμμα. Αξίζει να σημειωθεί ότι μπορεί το ίδιο στοιχείο να ανήκει σε περισσότερες από μια κλάσεις, διαχωρισμένες με κενά μέσα στην *class* html attribute του. Πχ `<p class="emphasis bodytext">...</p>`.
Για παράδειγμα, ο παρακάτω κανόνας CSS:

```
.emphasis
{
      color: red;
}
```

θα κάνει κόκκινα τα γράμματα και στο στοιχείο `<p class="emphasis>blah blah</p>`, και στο στοιχείο `<div class="emphasis otherclass">blah blah</div>`αλλά **όχι** στο στοιχείο `<h1 class="otherclass">blah blah</h1>`.
Οι κλάσεις γενικά χρησιμοποιούνται όταν θέλουμε να **ομαδοποιήσουμε** κάποια στοιχεία html για τα οποία δεν μπορούμε να βρούμε κάποιον άλλο επιλογέα που να αφορά **όλα αυτά** και **μόνον αυτά**, οπότε τους προσδίδουμε μια συγκεκριμένη κλάση, ώστε να μπορούμε στο CSS μας να αναφερθούμε **μόνο σε αυτά** και να τα μορφοποιήσουμε.

στοιχείο.όνομα_κλάσης

Αποτελεί ουσιαστικά συνδυασμό των δύο παραπάνω επιλογέων. Εφαρμόζεται σε όσα στοιχεία αποτελούνται από το html tag *<στοιχείο>* και ανήκουν στην κλάση *όνομα_κλάσης*. Πχ ο επιλογέας `p.emphasis` εφαρμόζεται σε ο,τι περιέχεται σε tags της μορφής `<p class="emphasis">...</p>`. Ο επιλογέας αυτός είναι χρήσιμος όταν έχουμε πολλά **διαφορετικού τύπου** στοιχεία **με την ίδια κλάση** και επιθυμούμε να εφαρμόσουμε **διαφορετική μορφοποίηση** ανάλογα με τον **τύπο** του στοιχείου.

#όνομα_id

Όταν ο επιλογέας μας περιλαμβάνει ένα χαρακτήρα δίεσης (#) στην αρχή του, τότε ο browser εφαρμόζει τις ιδιότητες που θα γράψουμε στο στοιχείο το οποίο περιλαμβάνει την ιδιότητα id="όνομα_id". **Δεν πρέπει να υπάρχουν δύο (ή περισσότερα) στοιχεία στη σελίδα μας με το ίδιο id.** Τα ids διέπονται από τους ίδιους κανόνες ονοματολογίας με τις κλάσεις. Ουσιαστικά, ο,τι μπορούμε να κάνουμε με τα ids μπορούμε να το κάνουμε και με τη χρήση κλάσεων, απλά όταν το στοιχείο που θέλουμε να μορφοποιήσουμε είναι **μοναδικό**, είναι γενικά καλύτερο να χρησιμοποιούμε ids.

Ψευδό-κλάσεις και ψευδό-στοιχεία

Ορισμένες φορές χρησιμοποιούμε τις λεγόμενες ψευδό-κλάσεις (pseudo-classes) ή ψευδό-στοιχεία (pseudo-elements), τα οποία μας επιτρέπουν να επιλέγουμε στοιχεία τα οποία δεν αποτελούν html elements, αλλά κομμάτια τους ή συγκεκριμένες καταστάσεις τους. Ουσιαστικά αποτελούν κάποιες λέξεις-κλειδιά που γράφουμε μετά από έναν επιλογέα του τύπου στοιχείο και ξεκινούν με :. Αν όλα αυτά σας ακούγονται κάπως μπερδεμένα ή δυσνόητα, μην προβληματίζεστε, θα γίνουν πιο κατανοητά όταν εξετάσουμε τις συγκεκριμένες ψευδό-κλάσεις/στοιχεία που χρησιμοποιούνται συνηθέστερα παρακάτω.

a:link
Χρησιμοποιείται αποκλειστικά για στοιχεία a (Δηλαδή ως εξής: `a:link`) και αφορά τους συνδέσμους που ο χρήστης δεν έχει ακόμη επισκεφθεί.

a:visited
Επίσης χρησιμοποιείται αποκλειστικά για στοιχεία a (Δηλαδή ως εξής: `a:visited`) και αφορά τους συνδέσμους που ο χρήστης έχει επισκεφθεί.

στοιχείο:active
Αφορά τα στοιχεία τύπου <στοιχείο> τη στιγμή που ο χρήστης έχει πατημένο το ποντίκι πάνω σε αυτά. Πχ ο επιλογέας a:active εφαρμόζεται σε συνδέσμους την ώρα που ο χρήστης έχει πατημένο το ποντίκι πάνω τους.

στοιχείο:hover
Από τις πιο συχνά χρησιμοποιούμενες ψευδό-κλάσεις. Αφορά τα στοιχεία τύπου <στοιχείο> τη στιγμή που ο χρήστης έχει το δείκτη του ποντικιού πάνω σε κάποιο από αυτά (χωρίς να πατάει κάποιο πλήκτρο). Μπορεί να μας βοηθήσει να δημιουργήσουμε διάφορα όμορφα εφέ, τα οποία παλιότερα ήταν εφικτά μόνο με javascript.

στοιχείο:focus
Χρησιμοποιείται κυρίως για στοιχεία φορμών και εφαρμόζεται στα στοιχεία τύπου <στοιχείο> που εκείνη τη στιγμή έχουν «focus», παραδείγματος χάριν, ένα πεδίο κειμένου στο οποίο ο χρήστης έκανε κλικ για να εισάγει κείμενο.

στοιχείο:first-letter
Αφορά το πρώτο γράμμα του κειμένου εντός κάποιου στοιχείου τύπου <στοιχείο>. Το ψευδό-στοιχείο αυτό μπορεί να μας βοηθήσει να δημιουργήσουμε αρχιγράμματα. Πχ ο επιλογέας p:first-letter αφορά το πρώτο γράμμα κάθε παραγράφου.

στοιχείο:first-line
Ψευδό-στοιχείο παρόμοιο με το παραπάνω, μόνο που αντί να αφορά μόνο το πρώτο γράμμα του κειμένου μέσα στο στοιχείο τύπου <στοιχείο>, αφορά ολόκληρη την πρώτη γραμμή.

Αν και δεν έχει ιδιαίτερη πρακτική χρησιμότητα, αξίζει να σημειωθεί ότι τα 5 πρώτα θεωρούνται ψευδό-κλάσεις (pseudo-classes) ενώ τα 2 τελευταία ψευδό-στοιχεία (pseudo-elements)

Σύνθετοι επιλογείς CSS (CSS selectors)

Πολλές φορές μπορούμε να συνδυάσουμε σε έναν επιλογέα περισσότερους από έναν υπό-επιλογείς, βάσει συγκεκριμένων κανόνων σύνταξης, κάτι που μας προσφέρει μεγαλύτερη ευελιξία και εξοικονόμηση χρόνου. Οι πιο βασικοί και συχνά χρησιμοποιούμενοι τρόποι συνδυασμού επιλογέων παρουσιάζονται παρακάτω:

επιλογέας$_1$, επιλογέας$_2$, επιλογέας$_3$, ...
Οι ιδιότητες που θα γράψουμε σε αυτόν τον κανόνα CSS, θα εφαρμοστούν σε κάθε στοιχείο που πληροί τις προϋποθέσεις επιλογής είτε για τον επιλογέα$_1$, είτε για τον επιλογέα$_2$, είτε για τον επιλογέα$_3$ κοκ.

Παραδείγματος χάριν, ο επιλογέας `input[type="text"], textarea` θα εφαρμοστεί σε κάθε πεδίο κειμένου που υπάρχει στη σελίδα μας, είτε είναι για εισαγωγή κειμένου μιας γραμμής (`<input type="text" />`) είτε είναι πολλών γραμμών (`<textarea></textarea>`).

επιλογέας$_1$ επιλογέας$_2$ επιλογέας$_3$ ... επιλογέας$_n$
Οι ιδιότητες που θα γράψουμε σε αυτόν τον κανόνα CSS θα εφαρμοστούν σε στοιχεία που πληρούν τις προϋποθέσεις του επιλογέα$_n$ και επιπροσθέτως περιέχονται μέσα σε κάποιο στοιχείο που πληροί τις προϋποθέσεις του επιλογέα$_{n-1}$ το οποίο βρίσκεται μέσα σε κάποιο στοιχείο που πληροί τις προϋποθέσεις του επιλογέα$_{n-2}$ κοκ. Παραδείγματος χάριν ο επιλογέας p img θα εφαρμοστεί σε όσες εικόνες περιέχονται μέσα σε tags `<p>...</p>`.

Πως εισάγουμε CSS στη σελίδα μας;

Μπορούμε να εισάγουμε CSS στη σελίδα μας με τρεις διαφορετικούς τρόπους, αναλόγως την περίπτωση. Οι τρόποι αυτοί, κατά σειρά φθίνουσα προτεραιότητας, είναι:

- **Inline CSS:** Αν επιθυμούμε να μορφοποιήσουμε ένα στοιχείο μόνο, και δεν πρόκειται να χρειαστούμε αυτό το είδος μορφοποίησης για τίποτε άλλο στο site, μπορούμε να γράψουμε «χύμα» (δηλ. χωρίς το κομμάτι του CSS selector και χωρίς αγκύλες) CSS properties μέσα στο attribute `style`, το οποίο το διαθέτει σχεδόν κάθε στοιχείο html. Παραδείγματος χάριν, για να κάνουμε μια συγκεκριμένη παράγραφο κόκκινη, μπορούμε να προσθέσουμε `style="color:red;"` στο `<p>` tag (ολοκληρωμένα: `<p style="color:red;">`). Αξίζει να σημειωθεί, ότι αν κάποιες από τις ιδιότητες που θα γράψουμε εντός της attribute style ενός στοιχείου «συγκρούονται» με κανόνες CSS που έχουν οριστεί για αυτό αλλού και το αφορούν, τότε ο browser θα επιλέξει να εφαρμόσει αυτά που γράψαμε εντός του attribute style, ως πιο συγκεκριμένα για το στοιχείο αυτό.
- **CSS για μια συγκεκριμένη σελίδα:** Πολλές φορές, μπορεί να θέλουμε να εφαρμόσουμε κάποιους κανόνες CSS μόνο για τη συγκεκριμένη σελίδα και όχι για όλο το site (ή να διαφοροποιήσουμε σε κάποιες ιδιότητες τους υπάρχοντες). Ένας τρόπος να το κάνουμε αυτό, για να μην δημιουργήσουμε ξεχωριστό αρχείο CSS είναι να εισάγουμε εντός των tags `<head>...</head>` της σελίδας τους κανόνες CSS μας μέσα σε `<style>` tags (με την html attribute `type` τους σε `text/css`). Παραδείγματος χάριν, για να κάνουμε το χρώμα γραμμάτων μιας συγκεκριμένης σελίδας γκρι, θα γράφαμε στο head της:

```
<style type="text/css">
body
{
      color:gray;
}
</style>
```

  Αξίζει να σημειωθεί ότι συνήθως όταν ξεκινάμε την ανάπτυξη ενός site, είναι πιο βολικό να χρησιμοποιήσουμε αυτό τον τρόπο αρχικά, μιας και πρέπει να επεξεργαζόμαστε μόνο ένα αρχείο αντί για δύο. Όταν τελειώνουμε την πρώτη σελίδα, συνήθως μεταφέρουμε το CSS που έχουμε γράψει σε εξωτερικό αρχείο (βλ. παρακάτω) ώστε να μπορούμε να χρησιμοποιήσουμε τους ίδιους κανόνες CSS και στις άλλες σελίδες του site μας, χωρίς να πρέπει φυσικά να τους κάνουμε copy-paste σε κάθε σελίδα.
- **Εξωτερικό αρχείο CSS:** Η πιο «σωστή» χρήση του CSS και αυτή που θα έπρεπε να είναι η πρώτη λύση στην οποία θα καταφύγουμε, είναι η χρήση εξωτερικού αρχείου CSS. Για να τη χρησιμοποιήσουμε, γράφουμε τους κανόνες CSS μας σε ένα αρχείο με επέκταση css (πχ main.css) και στο head της κάθε σελίδας του site μας γράφουμε **`<link href="main.css" type="text/css" />`** (αν το όνομα του αρχείου CSS είναι main.css, αλλιώς προφανώς γράφουμε το σωστό όνομα αρχείου). Αξίζει να σημειωθεί ότι μπορούμε να έχουμε πολλά αρχεία css στην ίδια σελίδα, και μεγαλύτερη προτεραιότητα έχει πάντα αυτό που έχει γραφτεί τελευταίο.

Βασικές ιδιότητες CSS (CSS properties)

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. Τα χρώματα μπορούν να εισαχθούν είτε σε μορφή RGB (πχ `color: rgb(255,128,30);`), είτε σε μορφή hex (πχ `color: #ff801e;`) είτε με τη μορφή κάποιου keyword (πχ `color:orange;`). Για περισσότερες πληροφορίες όσον αφορά τον τρόπο ορισμού των χρωμάτων στο CSS (και την html) μπορείτε να δείτε εδώ.

font-size
Αφορά το μέγεθος της γραμματοσειράς. Οι τιμές που δέχεται μπορούν να είναι εκφρασμένες σε ένα μεγάλο πλήθος μονάδων μεγέθους, από τις οποίες οι πιο ευρέως διαδεδομένες είναι τα pixels (πχ `font-size: 12px;`) και οι στιγμές (πχ `font-size:10pt`). Περισσότερες πληροφορίες εδώ.

font-family
Η ιδιότητα αυτή μας επιτρέπει να ορίσουμε ένα πλήθος γραμματοσειρών που θα χρησιμοποιηθούν για το κείμενο, κατά σειρά προτίμησης. Ουσιαστικά μέσω αυτής ορίζουμε τη γραμματοσειρά του κειμένου, απλά μας επιτρέπει να ορίσουμε και εναλλακτικές επιλογές, ώστε αν η γραμματοσειρά που ορίσαμε δεν υπάρχει στον υπολογιστή του χρήστη, να μην μας καταστρέψει ο browser την εμφάνιση της σελίδας επιλέγοντας όποια αυτός νομίζει, απλά την επόμενη επιλογή μας. Αν καμία από όσες γραμματοσειρές ορίσαμε δεν υπάρχει στον υπολογιστή του χρήστη, τότε ο browser επιλέγει κάποια που να ανήκει στη γενική οικογένεια γραμματοσειρών (generic font-family) που θα ορίσουμε στο τέλος με το κατάλληλο keyword. Τέτοια keywords
είναι τα εξής:

- serif: Γραμματοσειρές με «πατούρες» όπως η Georgia.
- sans-serif: Γραμματοσειρές χωρίς «πατούρες» όπως η Trebuchet MS.
- monospace: Γραμματοσειρές όπου το κάθε γράμμα καταλαμβάνει ίσο πλάτος με τα υπόλοιπα, όπως η Courier New.
- cursive: Καλλιγραφικές γραμματοσειρές όπως η *Monotype Corsiva*
- fantasy: «Διακοσμητικές» γραμματοσειρές.

Οι τελευταίες 2 γενικές οικογένειες γραμματοσειρών χρησιμοποιούνται πολύ σπάνια.
Παράδειγμα χρήσης της ιδιότητας: `font-family: Calibri, Trebuchet MS, Verdana, sans-serif;`

font-style
Σε αντίθεση με αυτό που θα περίμενε κανείς από μια ιδιότητα με αυτό το όνομα, δηλαδή την εφαρμογή πολλών και διαφόρων «εφέ» στο κείμενο, στην πραγματικότητα αφορά μόνο την περίπτωση όπου το κείμενο θα είναι *πλάγιο*. Για άλλα εφέ, χρησιμοποιούνται άλλες ιδιότητες, οι οποίες θα εξεταστούν παρακάτω. Οι πιθανές τιμές της είναι *normal*, *italic* και *oblique*. Οι δύο τελευταίες συνήθως κάνουν το ίδιο πράγμα (και είναι προτιμότερη η χρήση της *italic* μιας και υποστηρίζεται και από Internet Explorer 3, ενώ η υποστήριξη για την *oblique* ξεκινά από Internet Explorer 4), ενώ είναι προφανής η λειτουργία της τιμής *normal*.

font-weight
Αφορά το «βάρος» της γραμματοσειράς και στην πράξη χρησιμοποιείται για να ορίσει αν το κείμενο μας θα είναι **έντονο** ή όχι, μιας και οι περισσότερες γραμματοσειρές που χρησιμοποιούνται στο web διατίθενται μόνο σε δύο βάρη: Κανονικό και έντονο, σε αντίθεση με πιο εξειδικευμένες γραμματοσειρές που χρησιμοποιούνται από γραφίστες, οι οποίες πολλές φορές διατίθενται σε διάφορα βάρη. Οπότε οι τιμές που συνήθως χρησιμοποιούνται σε αυτή την ιδιότητα είναι οι *normal* και *bold*, η λειτουργία των οποίων είναι προφανής.

text-decoration
Μας επιτρέπει να εφαρμόσουμε στο κείμενο μας διάφορα εφέ, συμπεριλαμβανόμενης και της υπογράμμισης. Οι τιμές που δέχεται είναι οι εξής:

- **none:** Καμία διακόσμηση
- **underline:** Υπογράμμιση

- **overline:** Γραμμή πάνω από το κείμενο (ουσιαστικά το αντίθετο της υπογράμμισης)
- **line-through:** Διαγράμμιση
- **blink:** Κάνει το κείμενο να αναβοσβήνει. Δεν πρέπει να χρησιμοποιείται, παρά μόνο σε εξαιρετικά σπάνιες περιπτώσεις, μιας και η εμπειρία έχει δείξει ότι είναι εξαιρετικά κουραστικό για τον αναγνώστη και αφαιρεί επαγγελματικότητα από το design της ιστοσελίδας.

Οι παραπάνω τιμές μπορούν να χρησιμοποιηθούν και συνδυαστικά, όταν επιθυμούμε να εφαρμόσουμε πάνω από ένα τέτοιο εφέ στο κείμενο μας. Παραδείγματος χάριν `text-decoration: underline overline;`

text-align
Μας επιτρέπει να καθορίσουμε τη στοίχιση του κειμένου μας. Όπως θα περίμενε κανείς, οι πιθανές τιμές είναι left, center, right και justify.

Ιδιότητες που αφορούν το φόντο

background-color
Χρησιμοποιείται για να ορίσει χρώμα φόντου στα στοιχεία που αφορά ο επιλογέας. Το χρώμα μπορεί να γραφεί σε οποιαδήποτε από τις μορφές που περιγράφηκαν για την ιδιότητα color.

background-image
Χρησιμοποιείται για να ορίσει μια εικόνα φόντου. Η τιμή που δέχεται είναι της μορφής *url(διεύθυνση_εικόνας)*, παραδείγματος χάριν `background-image:url(http://www.e-steki.gr/logo.gif);`. Γενικά όταν εισάγουμε μια διεύθυνση URL ως τιμή (ή τμήμα τιμής) κάποιας ιδιότητας CSS, πάντα τη γράφουμε εντός των παρενθέσεων του `url()`. Φυσικά μπορούμε να εισάγουμε και σχετικά URLs, πχ `background-image:url(logo.gif);`.

background-repeat
Αφορά τον τρόπο που θα επαναλαμβάνεται η εικόνα που ορίσαμε στην παραπάνω ιδιότητα. Οι τιμές που δέχεται είναι:

- **no-repeat:** Καμία επανάληψη
- **repeat-x:** Η εικόνα επαναλαμβάνεται στον οριζόντιο άξονα
- **repeat-y:** Η εικόνα επαναλαμβάνεται στον κάθετο άξονα
- **repeat:** Η εικόνα επαναλαμβάνεται και οριζόντια και κάθετα (η τιμή *repeat* είναι και η προεπιλεγμένη τιμή της ιδιότητας *background-repeat*).

background-position
Η ιδιότητα αυτή μας επιτρέπει να ορίσουμε τη θέση που θα τοποθετηθεί η εικόνα φόντου που ορίσαμε με την ιδιότητα *background-image* τόσο κάθετα, όσο και οριζόντια. Συνήθως ως τιμές της χρησιμοποιούμε κάποια keywords, τα οποία είναι τα εξής:

- **left, center, right** όσον αφορά την οριζόντια θέση της εικόνας
- **top, center, bottom** όσον αφορά την κάθετη θέση της εικόνας

Τα παραπάνω μπορούμε να τα χρησιμοποιήσουμε και συνδυαστικά (και συνήθως αυτό γίνεται), όπως στα παρακάτω παραδείγματα:

```
background-position: left bottom;
background-position: top right;
background-position: center top;
```

ή μπορούμε να ορίσουμε την τοποθέτηση μόνο κατά τον ένα άξονα (για τον άλλο υποτίθεται η προεπιλεγμένη τιμή, η οποία είναι συνήθως top για τον κάθετο άξονα και left για τον οριζόντιο), όπως στα παρακάτω παραδείγματα:

```
background-position: left;
background-position: bottom;
```

Αντί για τα παραπάνω keywords, μπορούμε να ορίσουμε τη θέση της εικόνας φόντου και αριθμητικά, είτε με ποσοστό ή με κάποια μονάδα μεγέθους, όπως στα παρακάτω παραδείγματα:

```
background-position: 10% 25%;
background-position: 60px 100px;
```

αν τέτοιες τιμές χρησιμοποιούνται πολύ πιο σπάνια.
Η ιδιότητα αυτή χρησιμοποιείται συνήθως όταν έχουμε ορίσει να μην επαναλαμβάνεται η εικόνα φόντου (μέσω της ιδιότητας *background-repeat*). Κάποιοι browsers υποστηρίζουν τη χωριστή δήλωση θέσης της εικόνας στον κάθετο και τον οριζόντιο άξονα, μέσω δύο ιδιοτήτων με όνομα *background-position-x* και *background-position-y*, αλλά καλύτερα να αποφεύγεται η χρήση αυτών των ιδιοτήτων, μιας και δεν υποστηρίζονται από όλους τους browsers.

Ιδιότητες που αφορούν το περίγραμμα

border-color
Ρυθμίζει το χρώμα περιγράμματος. Το χρώμα μπορεί να γραφεί σε οποιαδήποτε από τις μορφές που περιγράφηκαν για την ιδιότητα color. Αν δεν οριστεί αυτή η ιδιότητα, χρησιμοποιείται το χρώμα που ορίστηκε στην ιδιότητα color.

border-width

Ρυθμίζει το πάχος του περιγράμματος σε κάποια από τις μονάδες μέτρησης που μπορούν να χρησιμοποιηθούν στο CSS, συνηθέστερα σε pixels.
Παράδειγμα: border-width: 10px;

border-style
Ορίζει το στυλ του περιγράμματος. Οι τιμές που χρησιμοποιούνται συνήθως για την ιδιότητα αυτή είναι οι εξής:

- **solid:** «Συμπαγές» περίγραμμα, δηλαδή χωρίς κάποια διακόσμηση, μια ενιαία γραμμή.
- **dashed:** Περίγραμμα που αποτελείται από παύλες.
- **dotted:** Περίγραμμα που αποτελείται από τελείες. Αξίζει να σημειωθεί ότι τα αποτελέσματα της χρήσης της τιμής αυτής διαφέρουν ελαφρώς ανά τους browsers. Για παράδειγμα ο Internet Explorer εμφανίζει τα dotted περιγράμματα ως αποτελούμενα από μικρούς κύκλους, ενώ ο Mozilla Firefox τα εμφανίζει ως αποτελούμενα από μικρά τετράγωνα.
- **double:** Σαν την τιμή *solid*, μόνο που δημιουργεί δύο περιγράμματα, η απόσταση και το πάχος των οποίων δεν μπορούν να ρυθμιστούν χωριστά αλλά ρυθμίζονται αυτόματα ώστε το πάχος τους να είναι συνολικά όσο η τιμή που ορίσαμε στην ιδιότητα *border-width*. Συνήθως χρησιμοποιείται σε συνδυασμό με πάχος περιγράμματος *3px*, για το οποίο δίνει ένα αρκετά καλαίσθητο αποτέλεσμα.

Τι κάνουμε αν θέλουμε να ορίσουμε διαφορετικό πάχος/χρώμα/στύλ περιγράμματος για το πάνω/δεξί/κάτω/αριστερό περίγραμμα;
Κάθε μια από τις ιδιότητες που αναφέρθηκαν παραπάνω, αποτελείται από τέσσερις επιμέρους ιδιότητες, οι οποίες μας επιτρέπουν να ορίσουμε διαφορετικές τιμές για την ιδιότητα αυτή όσον αφορά 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. Επίσης, κάθε ιδιότητα από αυτές που αναφέρθηκαν παραπάνω δέχεται και τιμές της μορφής *[top] [right] [bottom] [left]* (ένας εύκολος μνημονικός κανόνας για τη σειρά των τιμών είναι ότι η σειρά αυτή είναι σύμφωνη με την πορεία των δεικτών του ρολογιού) ή της μορφής *[top,bottom] [right,left]* αντί για μόνο μια τιμή που θα εφαρμοστεί και για τις τέσσερις πλευρές του περιγράμματος. Για παράδειγμα, για να ορίσουμε ένα γκρι συμπαγές περίγραμμα με πάχος αριστερά 1px, δεξιά 2px, πάνω 3px και κάτω 4px θα μπορούσαμε να γράψουμε είτε:

```
border-left-width: 1px;
border-right-width: 2px;
border-top-width: 3px;
border-bottom-width: 4px;
border-style: gray;
border-style: solid;
```

είτε:

```
border-width: 3px 2px 4px 1px;
border-style: gray;
```

Διάφορες χρήσιμες ιδιότητες

padding
Ορίζει το κενό που θα υπάρχει μεταξύ των ορίων ενός στοιχείου και των περιεχομένων του. Είναι πολύ σημαντικό να ορίζουμε padding σε στοιχεία στα οποία έχουμε ορίσει κάποιο περίγραμμα, ώστε να μην «κολλάνε» τα περιεχόμενα τους με το περίγραμμα, κάτι που φαίνεται ιδιαίτερα άσχημο και ερασιτεχνικό. Επίσης, καλό είναι να μην είστε ιδιαίτερα φειδωλοί με το padding. Περισσότερο padding δίνει πιο επαγγελματική εμφάνιση (μέχρι κάποιων ορίων φυσικά). Για να ορίσουμε διαφορετικό padding ανά πλευρά, 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]*. Παραδείγματα:

```
padding: 8px;
padding: 2px 6px;
padding: 0px 6px 6px 6px;
```

margin
Το αντίθετο ουσιαστικά του padding. Ορίζει τον χώρο μεταξύ των ορίων ενός στοιχείου και όσων το περιβάλλουν. Είναι ιδιαίτερα σημαντικό να ορίζουμε margin σε εικόνες, ώστε να έχουν απόσταση από τα περιεχόμενα τους, μιας και είναι ιδιαίτερα αντιαισθητικό να «κολλάνε» με το κείμενο. Φυσικά και εδώ μπορούμε να χρησιμοποιήσουμε είτε τις ιδιότητες *margin-top, margin-right, margin-bottom, margin-left* για να ορίσουμε διαφορετικές τιμές margin ανά πλευρά, είτε να εισάγουμε τις τέσσερις διαφορετικές τιμές margin με τη σειρά *[top] [right] [bottom] [left]*, ή *[top,bottom] [right,left]*. Παραδείγματα:

```
margin: 4px;
margin: 2px 4px;
margin: 0px 8px 2px 2px;
```

width, height
Όπως είναι προφανές, οι δύο αυτές ιδιότητες ορίζουν το πλάτος και το ύψος ενός στοιχείου, είτε σε ποσοστό (το οποίο υπολογίζεται βάσει του στοιχείου που το περιέχει), είτε σε κάποια μονάδα μήκους. Παραδείγματα:

```
width: 100px;
width: 90%;
height: 300px;
height: 100%;
```

Υπάρχουν και οι ιδιότητες *min-width, min-height, max-width, max-height* οι οποίες ορίζουν τα όρια στα οποία μπορούν να κινούνται οι διαστάσεις ενός στοιχείου, όταν δεν θέλουμε να ορίσουμε συγκεκριμένες διαστάσεις, αλλά καλό είναι να μην χρησιμοποιούνται ακόμα, όσο βολικές κι αν φαίνονται, μιας και δυστυχώς η υποστήριξη τους από τους υπάρχοντες browsers είναι περιορισμένη.

float
Η ιδιότητα αυτή είναι ανεκτίμητης χρησιμότητας στο όταν χρησιμοποιούμε CSS και για το layout της σελίδας μας, αν και είναι η δυσκολότερη στην κατανόηση από όσες αναφέρθηκαν. 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. Πολλές φορές χρησιμοποιούμε μετά από floated στοιχεία κάποιο στοιχείο (συνήθως ένα άδειο div) με την ιδιότητα `clear: both;` για να «καθαρίσει» τα floats για το μετέπειτα περιεχόμενο. Μπορείτε να διαβάσετε περισσότερα για τα floats εδώ και εδώ.

Προτεραιότητα

Πολλές φορές για το ίδιο στοιχείο μπορεί να ταιριάζουν αρκετοί κανόνες CSS, ορισμένοι εκ των οποίων να περιέχουν ιδιότητες που «συγκρούονται». Για παράδειγμα, έστω ότι έχουμε στο CSS μας τους παρακάτω κανόνες:

```
.testclass
{
      color: red;
```

```
}

div
{
      color: gray;
}
```
και στην html μας έχουμε κάποιο στοιχείο `<div class="testclass">test text</div>`. Το κείμενο του θα είναι κόκκινο ή γκρι;
Απάντηση: Το κείμενο θα είναι κόκκινο, διότι ο επιλογέας .testclass είναι πιο «συγκεκριμένος» απ'ότι ο επιλογέας div.

Ωστόσο, αν στο CSS μας είχαμε τους κανόνες:
```
.testclass
{
      color:red;
}

.testclass2
{
      color: gray;
}
```
και στην html μας είχαμε το στοιχείο `<div class="testclass testclass2">test text</div>`, τότε το κείμενο θα ήταν γκρι, διότι η δήλωση της κλάσης testclass2 διαβάστηκε τελευταία από τον browser, οπότε έχει μεγαλύτερη προτεραιότητα.
Γενικά οι κανόνες προτεραιότητας στο CSS είναι αρκετά περίπλοκοι και δεν μπορούν να αναλυθούν εκτενώς εδώ. Οι βασικότεροι είναι οι εξής:

- Οι ιδιότητες CSS που βρίσκονται μέσα στην ιδιότητα style ενός στοιχείου έχουν πάντα μεγαλύτερη προτεραιότητα από οποιεσδήποτε άλλες (εκτός από όσες χρησιμοποιούν *!important*, το οποίο θα αναλυθεί παρακάτω)
- Οι κανόνες CSS που βρίσκονται μέσα στα tags `<style>...</style>` έχουν μεγαλύτερη προτεραιότητα από αυτούς που βρίσκονται σε ένα εξωτερικό αρχείο CSS.
- Όταν δύο ή περισσότερα αρχεία CSS συμπεριλαμβάνονται στην ίδια ιστοσελίδα, οι κανόνες αυτού που γράφτηκε τελευταίο στη σελίδα έχουν μεγαλύτερη προτεραιότητα από αυτούς του πρώτου.
- Ένας κανόνας CSS έχει μεγαλύτερη προτεραιότητα από όσους κανόνες CSS βρίσκονται γραμμένοι πιο πάνω από αυτόν αν δεν συντρέχει άλλος λόγος για να μετρήσει παραπάνω κάποιος από αυτούς.
- Οι κανόνες με επιλογέα του τύπου *.όνομα_κλάσης* έχουν μεγαλύτερη προτεραιότητα από κανόνες με επιλογέα του τύπου *στοιχείο*, μιας και είναι πιο συγκεκριμένοι.
- Οι επιλογείς του τύπου *#όνομα_id* έχουν μεγαλύτερη προτεραιότητα από επιλογείς του τύπου *.όνομα_κλάσης*, μιας και είναι πιο συγκεκριμένοι.

Σε κάθε περίπτωση, αν επιθυμούμε μια ιδιότητα ενός κανόνα να έχει μεγαλύτερη προτεραιότητα από αυτή που της αντιστοιχεί, μπορούμε πριν το ; να γράψουμε `!important`. Ωστόσο γενικά πρέπει να χρησιμοποιείται αυτή η τεχνική **με φειδώ** μιας και μπορεί να οδηγήσει σε μπερδεμένο κώδικα CSS. Παραδείγματος χάριν, στο πρώτο παράδειγμα αν είχαμε γράψει στο CSS μας:
```
.testclass
{
      color: red;
}

div
{
      color: gray !important;
}
```
τότε τα γράμματα στο στοιχείο `<div class="testclass">test text</div>` θα ήταν γκρι, αντί για κόκκινα.