

ΜΑΘΗΜΑ 8^ο - Σύνταξη της HTML

Σύνταξη της HTML (12') <https://youtu.be/B6P8rHtSiU0>

Θα δούμε τώρα τα κύρια στοιχεία της σύνταξης ενός html εγγράφου. Σε ένα html έγγραφο ανακατεύουμε περιεχόμενο κείμενα και λοιπά που θα δει ο χρήστης καθώς και εντολές html. Ας δούμε αυτές οι εντολές html πώς συντάσσονται, ποιοι είναι οι κανόνες σύνταξης ενός html εγγράφου. Ας πάμε να δούμε το βασικό στοιχείο ενός html εγγράφου που στην ουσία ορίζεται από ένα συνδυασμό μιας ετικέτας που μπορεί να πάρει και κάποιες παραμέτρους, του περιεχομένου και μιας άλλης ετικέτας που είναι η ίδια ετικέτα με το χαρακτηριστικό slash (/), αυτό εδώ το σύμβολο που είναι το σύμβολο τερματισμού της ετικέτας. Ένα ζευγάρι λοιπόν, ετικέτα και τερματική ετικέτα μαζί με κάποιο περιεχόμενο ορίζουν ένα στοιχείο html. Το στοιχείο λοιπόν <html> είναι ο συνδυασμός ετικέτα με παραμέτρους, περιεχόμενο και τερματική ετικέτα.

Ας δούμε ένα παράδειγμα.

Στην ετικέτα <p> paragraph παράγραφος με παράμετρο id=p1 βλέπω ότι ανάμεσα στην ετικέτα την ίδια και την παράμετρο id υπάρχει ένα κενό. Η id ακολουθείται από = και την τιμή που πρέπει να πάρει, θα δούμε και θα εξηγήσουμε τι σημαίνει id, κλείνουμε την ετικέτα. Αυτή λοιπόν είναι η ετικέτα ορισμού αυτού του στοιχείου που είναι στοιχείο τύπου paragraph p, ακολουθεί κείμενο, κάποιες λέξεις οι οποίες λέξεις θα εμφανιστούν στο χρήστη και στη συνέχεια υπάρχει η ετικέτα τερματισμού. Μέσα σε μία ετικέτα σε ένα στοιχείο html που ορίζεται από δύο ετικέτες μπορεί να υπάρχει ένα ή περισσότερα άλλα στοιχεία. Άρα τελικά να ορίσω μία ιεραρχία στοιχείων html. Επίσης η τιμή των παραμέτρων της ετικέτας μπορεί να εισάγεται είτε σε εισαγωγικά είτε χωρίς εισαγωγικά. Η html5 δεν είναι αυστηρή σε αυτό το σημείο.

Τέλος σημειώνουμε ότι τα ονόματα των ετικετών όπως εδώ είναι το <p> δεν έχει σημασία αν είναι κεφαλαία ελληνικά αν και συνηθίζουμε να χρησιμοποιούμε μικρά σε μικρούς χαρακτήρες όπως εδώ το p.

Παράδειγμα.

Τα στοιχεία παρατίθενται ιεραρχικά. Εδώ έχω την ετικέτα p, η οποία ορίζει ένα στοιχείο τύπου p που αρχίζει εδώ και τερματίζει στην ετικέτα τερματισμού (slash) /p. Η ετικέτα p όμως περιέχει μέσα της και κείμενο my dog is και τα λοιπά αλλά και μία άλλη ετικέτα την ετικέτα Strong που περικλείει ένα κομμάτι του κειμένου. Η strong ορίζει ένα νέο στοιχείο που είναι τύπου Strong που εμπεριέχεται στο στοιχείο p. Ιδιαίτερη προσοχή χρειάζεται στην ιεράρχη των στοιχείων.

Ένα άλλο παράδειγμα.

Εδώ αρχίζω την ετικέτα p, γράφω ένα κείμενο, ανοίγω την ετικέτα strong ένα νέο στοιχείο και μετά αντί να τερματίσω πρώτα τη strong και μετά την p το κάνω ανάποδα. Αυτό είναι λάθος γιατί ο συντακτικός αναλυτής της html στον browser θα μπερδευτεί, θα λέει που αρχίζει και που τελειώνει η κάθε ετικέτα.

Δύο είναι τα χαρακτηριστικά στοιχεία που έχουμε: Είναι τα λεγόμενα block elements στοιχεία block, τα οποία όταν καλώ και δημιουργώ ένα τέτοιο στοιχείο, ξέρω ότι όταν θα εμφανιστεί το στοιχείο αυτό θα ξεκινάει από νέα γραμμή. Τέτοιο είναι η παράγραφος. Όταν ορίζω μία παράγραφο το p δηλαδή, ξέρω ότι το περιεχόμενο της θα εμφανιστεί θα αρχίζει από μία νέα γραμμή. Το ίδιο και όταν τελειώνει το επόμενο αυτό που ακολουθεί θα έρχεται μετά από μία νέα γραμμή. Αυτά λέγονται block elements γιατί δημιουργούν ένα block περιεχομένου ένα block κειμένου για παράδειγμα.

Υπάρχουν τώρα και άλλα στοιχεία που λέγονται inline elements, στοιχεία δηλαδή που εντάσσονται στο πλαίσιο μιας γραμμής και συνήθως αφορούν ένα μικρό τμήμα του κειμένου του περιεχομένου. Τέτοια είναι το strong, emphasis που θα δούμε αμέσως μετά και η ετικέτα <a> για anchor, για υπερσυνδέσμους και ούτω καθεξής. Θα πρέπει να προσέξω ότι μέσα σε ένα inline element δεν μπορώ να βάλω ένα block element, ένα στοιχείο block. Ένα block δεν χωράει μέσα σε ένα inline element.

Κάποια ιδιαιτερότητα παρουσιάζουν κάποιες ετικέτες κάποιες εντολές της html οι οποίες δεν χρειάζονται ετικέτα τερματισμού. Δηλαδή είναι αυτάρκες, τις λέμε κενές και κενά στοιχεία γιατί δεν έχουν περιεχόμενο. Τέτοιες ετικέτες ορίζουν το horizontal rule <hr>,
 Break, για image κλπ. Σε κάποιες προηγούμενες εκδόσεις της html αυτές οι ετικέτες έπρεπε να αυτοτερματιστούν άρα στο τέλος της ετικέτας έπρεπε να βάλω τον backslash υποδεικνύοντας ότι η ετικέτα τερματίζει στον εαυτό της. Αυτό πια η html5 δεν το απαιτεί.

Πώς συντάσσονται τα γνωρίσματα.

Έχω την ετικέτα `p` στην οποία βάζω ένα γνώρισμα. Το γνώρισμα είναι το γνώρισμα `id` το οποίο παίρνει τιμή αυτήν εδώ τη συμβολοσειρά. Κλείνω την ετικέτα, μετά είναι το περιεχόμενο της ετικέτας και εδώ είναι η ετικέτα τερματισμού. Όλο αυτό όπως είπαμε είναι ένα στοιχείο ενώ αυτό το τμήμα είναι το γνώρισμα της ετικέτας `p`. Παρατηρώ λοιπόν ότι το όνομα του γνωρίσματος χωρίζεται με κενό από το όνομα της ετικέτας ή από ένα προηγούμενο γνώρισμα. Θα σημειώσουμε εδώ ότι μία ετικέτα `html` μπορεί να έχει πολλά γνωρίσματα. Τα γνωρίσματα αυτά σε αντίθεση με άλλες γλώσσες μπορεί να θεωρήσει κανείς ότι όπως είναι σε μια συνάρτηση σε μία γλώσσα προγραμματισμού θα πρέπει να χωρίζονται με κενά με κόμματα, στην `html` χωρίζονται με κενά. Το όνομα του γνωρίσματος ακολουθείται από το σύμβολο `ίσον` και η τιμή του γνωρίσματος περιέχεται σε εισαγωγικά κάτι όμως που δεν είναι απαραίτητο. Είναι απαραίτητο να βάλω τα εισαγωγικά στην τιμή του γνωρίσματος όταν η τιμή αυτή περιέχει κενά γιατί όπως είδαμε το κενό είναι τερματικός διαχωριστικός χαρακτήρας μεταξύ των ορισμάτων και τιμών. Υπάρχουν επίσης μία ιδιαίτερη κατηγορία γνωρισμάτων που είναι λογικές μεταβλητές, είναι αληθή ή ψευδή και αυτά τα γνωρίσματα δεν παίρνουν `ίσον` κάποια τιμή.

Ένα παράδειγμα μιας ετικέτας `input` που παίρνει την τιμή `type` `ίσον` `text`, το πρώτο γνώρισμα, αυτή είναι η τιμή του γνωρίσματος κενό, δεύτερο γνώρισμα `disable`. Το `disable` όμως είναι μία λογική μεταβλητή η οποία δεν παίρνει τιμή.

Τα πιο συνηθισμένα γνωρίσματα που μπορεί να πάρει μία ετικέτα:

- Το πιο σημαντικό ίσως, είναι το γνώρισμα `id`. Το `id` παίρνει μία τιμή που εμείς του δίνουμε και είναι η μοναδική ταυτότητα αυτού του στοιχείου. Δεν μπορεί κάποιο άλλο στοιχείο στο ίδιο `html` document να έχει την ίδια τιμή. Η τιμή αυτή είναι η ταυτότητα του στοιχείου και αυτή χρησιμοποιείται για να αναφέρομαι σε αυτό το στοιχείο είτε σε κώδικα `JavaScript` είτε σε `stylesheets`, σε κανόνες μορφοποίησης εμφάνισης.
- Ένα δεύτερο γνώρισμα πολύ σημαντικό είναι το γνώρισμα `class`. `Class` = τιμή. Η `class` ορίζει μία ομάδα μία κλάση από στοιχεία που όλα μοιράζονται κάποιο `στυλ`. Μπορώ δηλαδή μέσω της κλάσης να ορίσω ένα `στυλ`, ένα τρόπο εμφάνισης όχι πια ενός στοιχείου όπως γίνεται με το `id` αλλά πολλών στοιχείων.

Άλλα ενδιαφέροντα γνωρίσματα είναι:

- Το γνώρισμα `hidden` που είναι λογική μεταβλητή και ορίζει αν το στοιχείο αυτό θα εμφανίζεται ή όχι.
- `Lang` είναι ένα γνώρισμα που παίρνει τιμή τη γλώσσα, με κάποια κωδικοποίηση γλωσσών θα το δούμε αμέσως μετά για το συγκεκριμένο στοιχείο άρα σε ποια γλώσσα είναι το κείμενο που περιέχει το συγκεκριμένο στοιχείο.
- Τέλος στο `style` που χρησιμοποιείται για `inline styling` που θα τα δούμε στο κεφάλαιο της `css`. Υπάρχουν βέβαια και πολλά άλλα.

Παράδειγμα: Η χρήση των ειδικών χαρακτήρων ως απλό κείμενο

Πώς να χειριστούμε τους ειδικούς χαρακτήρες σαν απλό κείμενο μέσα στην `html`. Θέλουμε να γράψουμε το εξής κείμενο και να το εμφανίσουμε στο χρήστη. Στην `html` για παράγραφο χρησιμοποιούμε το μικρότερο `p` το μεγαλύτερο, την ετικέτα μικρότερο ή μεγαλύτερο. Αν θέλουμε αυτό να το εμφανίσουμε στο χρήστη αυτό δεν είναι δυνατόν γιατί το μικρότερο `p` μεγαλύτερο είναι η ίδια εντολή της `html`. Συνεπώς ο συντακτικός αναλυτής θα καταλάβει ότι αυτή είναι μια εντολή και όχι `text`. Πώς μπορώ να το παρουσιάσω σαν `text`; Ο μόνος τρόπος για να το κάνω είναι να χρησιμοποιήσω κάποιους ειδικούς συμβολισμούς που εισάγονται με το σύμβολο `ampersand` και ακολουθούν μία λεκτική περιγραφή του συμβόλου. Το μικρότερο είναι `ampersand less than` ερωτηματικό(<). Το μεγαλύτερο `ampersand gt greater than` ερωτηματικό (>). Το `quote` που επίσης ένας χαρακτήρας που χρησιμοποιείται στις εντολές `html` `ampersand quot` ερωτηματικό("). Το μονό εισαγωγικό με `'`. Ένας ειδικός χαρακτήρας είναι το `non breaking space` `nbsp` που μου επιτρέπει να εισάγω ένα κενό. Αυτό γιατί στην `html` όπως θα δούμε στη συνέχεια το κενό το χειρίζεται η `html` με έναν ιδιαίτερο τρόπο άρα όταν θέλω να βάλω πολλά κενά θα πρέπει να το πω με αυτό τον ειδικό τρόπο.

Η `html` τους χαρακτήρες που δεν τυπώνονται που μπορεί να είναι το κενό, `new line`, το `tab` και τα λοιπά, μία ακολουθία τέτοιων χαρακτήρων την χειρίζεται σαν να είναι ένας χαρακτήρας. Για παράδειγμα αν γράψουμε, το καλοκαίρι κάνει ζέστη και το καλοκαίρι κενό κενό κενό `Tab` και τα λοιπά κάνει `newline` κενό κενό κενό `tab` ζέστη που θα το εμφανίζαμε στο `text editor` έτσι, στο `browser` η `html` θα το παρουσιάσει ακριβώς με τον ίδιο τρόπο. Αυτό γιατί αυτή την ακολουθία των λευκών χαρακτήρων, των κενών χαρακτήρων θα το αντικαταστήσει με ένα `Space`. Και το ίδιο και αυτήν εδώ την ακολουθία.

Άρα οι μη εκτυπώσιμοι χαρακτήρες μετατρέπονται σε ένα κενό χαρακτήρα από το συντακτικό αναλυτή. Τώρα υπάρχουν περιπτώσεις που θέλουμε να υπάρχουν πολλά κενά. Τι κάνουμε τότε; Χρησιμοποιούμε τον ειδικό χαρακτήρα που μόλις είδαμε. Το `no brake space` το `ampersand nbsp` ερωτηματικό ().

Πως μπορώ μέσα σε ένα κώδικα html να εισάγω σχόλια:

Εδώ η html χρησιμοποιεί έναν λίγο πιο σύνθετο τρόπο σε αντίθεση με άλλες γλώσσες. Ξεκινάει με το μικρότερο που είναι το σύμβολο για όλες τις ετικέτες στις εντολές της και μετά ακολουθεί το θαυμαστικό και παύλα-παύλα και ξέρουμε λοιπόν ότι όλο αυτό που ακολουθεί είναι ένα σχόλιο δηλαδή δεν θα εμφανιστεί στο χρήστη και για να τερματίσει η περιοχή του σχολίου θα έχουμε την ακολουθία παύλα-παύλα μεγαλύτερο. Είναι στην ουσία σαν να είναι μία εντολή που αυτο-τερματίζεται, η οποία όμως ξεκινάει με αυτούς τους χαρακτήρες και τερματίζει με αυτούς τους χαρακτήρες.