

ΓΚ802 - CSS Μέρος 1

2022

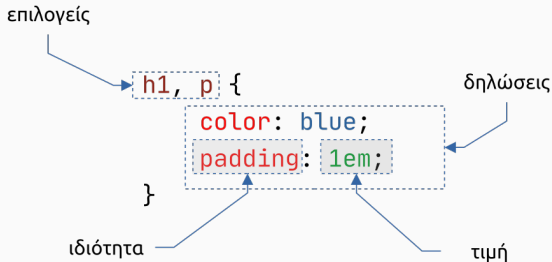
Τμ. ΗΜ&ΤΥ - Πανεπιστήμιο Πατρών

CSS

Η CSS είναι γλώσσα περιγραφής της παρουσίασης κειμένων (στυλ, layout κλπ.)

- εφαρμόζεται συνήθως σε αρχεία κειμένου που είναι γραμμένα σε κάποια γλώσσα όπως HTML, XML, SVG.
- η παρουσίαση ενός κειμένου σημαίνει τη μετατροπή του σε μορφή που να είναι χρήσιμη για τους χρήστες, π.χ. σε μια οθόνη.

Ένας κανόνας CSS (CSS rule) αποτελείται από:



Σχήμα 1: Ένας τυπικός κανόνας CSS.

Τρόποι εφαρμογής

- εξωτερικό stylesheet

```
<link rel="stylesheet" href="filename.css">
```

- εσωτερικό stylesheet

```
<head>
```

```
  <style>
```

```
    h1 {
```

```
      color: blue;
```

```
      border: 1px solid black;
```

```
    }
```

```
    p { color: red; }
```

```
  </style>
```

- inline stylesheet

```
<p style="color: red;">Red text</p>
```

Άλλες μορφές CSS statement

Οι κανόνες CSS είναι η πιο συνηθισμένη μορφή. Άλλο είδος CSS statement είναι:

- **at-rules:** Αρχίζουν με @ και δηλώνουν μεταδεδομένα ή άλλες πληροφορίες:
- **@import** φορτώνει κανόνες από εξωτερικό αρχείο
π.χ. `@import url("page.css");`
- **@media** εφαρμόζει το στυλ ανάλογα το μέσο προβολής
π.χ. `@media screen and (min-width: 900px){... }`
- **@supports** αν ο φυλλ. υποστηρίζει κάποια τεχνολογία
π.χ. `@supports not (display: grid){... }`
- **@font-face** καθορίζει μια γραμματοσειρά

Παράδειγμα με media query.

Η δήλωση θα ισχύσει μόνο όταν η διαθέσιμη οριζόντια διάσταση (viewport) είναι από 801px και πάνω.

```
@media (min-width: 801px) {  
  body {  
    margin: 0 auto;  
    width: 800px;  
  }  
}
```

CSS Selectors

Οι επιλογείς μας επιτρέπουν να επιλέξουμε σε ποια στοιχεία θα εφαρμογοστούν οι κανόνες μας.

Οι βασικοί επιλογείς

- universal/καθολικός (συμβολίζεται με *),
- type/τύπου,
- class/κλάσης,
- id και
- attribute selector/γνωρίσματος

μπορούν να συνδυαστούν σε πιο σύνθετους.

Spec: CSS Selectors Level 3

Type selector

Οι επιλογείς τύπου εφαρμόζονται σε όλα τα στοιχεία κάποιου τύπου:

```
p, h1, ul {  
  color: red;  
}
```

Θα επιλεγούν όλα τα p, όλα τα h1 και όλα τα ul.

```
<h1>Choose wisely</h1>
```

Αποτέλεσμα:

Choose wisely

Class selector

Επιλέγει όλα τα στοιχεία που στην ιδιότητα class τους έχουν τον επιλογέα:

```
<li class="first fancy-item">Oranges</li>
```

```
<li class="fancy-item">Apples</li>
```

```
.first {  
  font-weight: bold;  
}
```

Αποτέλεσμα:

- **Oranges**
- Apples

Το στοιχείο li έχει δύο κλάσεις, τη first και τη fancy-item, χωρισμένες με κενό.

Id selector

Επιλέγει όλα τα στοιχεία που έχουν στην ιδιότητα id τους τον επιλογέα:

```
#polite {
```

```
    font-weight: bold;
```

```
}
```

```
<p id="polite">Hello</p>
```

Είναι το `*`.

Επιλέγει όλα τα στοιχεία. Είναι δηλαδή το ίδιο με τον επιλογή τύπου, απλά επιλέγεται οποιοσδήποτε τύπος. Πολλές φορές εννοείται, για παράδειγμα

```
.warning { background-color: red; }
```

ισοδυναμεί με

```
*.warning { background-color: red; }
```

Ο επιλογέας αυτός μπορεί να είναι αργός.

Επιλογή στοιχείων με βάση κάποιο από τα γνώρισμά τους (attribute).

- **[attr]**: επιλέγει όλα τα στοιχεία που περιέχουν το γνώρισμα attr, όποια και αν είναι η τιμή του.
- **[attr=val]**: επιλέγει όλα τα στοιχεία που περιέχουν το γνώρισμα attr, και που η τιμή του είναι val.
- **[attr ~=val]**: επιλέγει όλα τα στοιχεία που περιέχουν το γνώρισμα attr, και που η τιμή τους είναι στη λίστα λέξεων που περιέχονται στο val (χωρισμένες με κενό).

Attribute selectors - Παράδειγμα

```
<li data-quantity="700g" data-vegetable="not-spicy spicy chili">Red
```

```
↪ pepper</li>
```

```
[data-vegetable~="spicy"] {
```

```
  color : red;
```

```
}
```

Αποτέλεσμα:

- Red pepper

Attribute selectors 2

- **[attr ^=val]**: επιλέγει όλα τα στοιχεία που περιέχουν το γνώρισμαattr, και που η τιμή τους αρχίζει με val.
- **[attr \$=val]**: επιλέγει όλα τα στοιχεία που περιέχουν το γνώρισμαattr, και που η τιμή τους τελειώνει σε val.
- **[attr *=val]**: επιλέγει όλα τα στοιχεία που περιέχουν το γνώρισμαattr, και που η τιμή τους περιέχει το substring val.
- **[attr |=val]**: επιλέγει όλα τα στοιχεία που περιέχουν το γνώρισμαattr, και που η τιμή τους είναι ακριβώς val ή αρχίζει με val- (δηλ. val που ακολουθείται από -).

Οι ψευδοκλάσεις αρχίζουν με : και μας επιτρέπουν να επιλέξουμε στοιχεία με άλλον τρόπο, πέρα από την αναζήτηση στο DOM.

Οι επιλογείς αυτοί αρχίζουν πάντα με : ακολουθούμενο από το όνομα της ψευδοκλάσης.

Οι ψευδοκλάσεις μπορούν μεταξύ άλλων να αφορούν τη δυναμική ή στατική κατάσταση ενός στοιχείου, τη θέση του μέσα στο DOM κλπ.

Δυναμικές ψευδοκλάσεις

Οι δυναμικές ψευδοκλάσεις είναι οι

- **:link**. Επιλέγει τους συνδέσμους που δεν έχει επισκεφθεί ο χρήστης του φυλλομετρητή
- **:visited**. Επιλέγει τους συνδέσμους που έχει επισκεφθεί ο χρήστης του φυλλομετρητή,
- **:hover**. Επιλέγει τον σύνδεσμο τον οποίο δείχνει ο χρήστης, πάνω από τον οποίο δηλαδή αιωρείται ο δείκτης της συσκευής κατάδειξης (π.χ. του ποντικιού),
- **:active**. Επιλέγει τον σύνδεσμο που ενεργοποιείται από τον χρήστη, δηλαδή τον υπερσύνδεσμο στον οποίο έχει κλικάρει ο χρήστης, για όσο διαρκεί το κλικ.
- **:focus**. Επιλέγει τον σύνδεσμο που είναι εστιασμένος, που μπορεί να έχει επιλεγεί με το πληκτρολόγιο ή το ποντίκι.

Δυναμικές ψευδοκλάσεις - Παράδειγμα

```
a:link { ... }           /* σύνδεσμοι που δεν έχει επισκεφθεί ο χρήστης */  
a:visited { ... }        /* σύνδεσμοι που έχει επισκεφθεί ο χρήστης */  
a:hover { ... }          /* σύνδεσμοι πάνω από τους οποίους αιωρείται ο δείκτης  
↪ του ποντικιού */  
a:active { ... }         /* ενεργοποιημένος σύνδεσμος */  
a:focus:hover { ... }    /* σύνδεσμοι που έχουν την εστίαση και που πάνω τους  
↪ αιωρείται ο δείκτης του ποντικιού */
```

Πχ.

```
a:hover { color: blue; }  
a:visited, a:link { color: black; }
```

Οι δομικές ψευδοκλάσεις επιλέγουν στοιχεία με βάση τη θέση τους στο DOM, με τρόπο που δεν παρέχεται από άλλους επιλογείς.

- **:root**. Επιλέγει το αρχικό στοιχείο, που σε μια σελίδα HTML είναι το στοιχείο `html`.
- **:empty**. Επιλέγει στοιχεία που δεν έχουν παιδιά.

Μια πολύ χρήσιμη κατηγορία ψευδοκλάσεων είναι αυτές που επιλέγουν στοιχεία με βάση τη θέση τους στο DOM, με πιο χαρακτηριστικό τον επιλογέα **:nth-child($An+B$)**. Τα A και B είναι σταθερές που δίνονται από το χρήστη. Επιλέγονται όλα τα στοιχεία που βρίσκονται στη θέση i που υπολογίζεται από την έκφραση $i = An + B$, για μη αρνητικό ακέραιο n . Οι θέσεις των στοιχείων ξεκινούν από το 1.

Δομικές ψευδοκλάσεις - Παράδειγμα

ul :nth-child(3n) - επιλέγει κάθε τρίτο li που είναι απόγονος ενός ul:

li:nth-child(even) /* το 2ο, 4ο, 6ο, κλπ στοιχείο*/

li:nth-child(10n-1) /* το 9ο, 19ο, 29ο, κλπ στοιχεία */

li:nth-child(10n+9) /* το ίδιο */

li:nth-child(10n+-1) /* λάθος σύνταξη, αγνοείται */

Η επιλογή γίνεται στο σύνολο των στοιχείων που είναι γείτονες.

1. Ποια στοιχεία θα επιλεγούν με τον επιλογέα `:nth-child(-3n+1)` ;
2. Γράψτε έναν επιλογέα που να επιλέγει όλα τα στοιχεία σε ζυγές θέσεις και έναν που να επιλέγει αυτά σε μονές θέσεις.

Οι ψευδοκλάσεις είναι:

| | | | |
|-----------------------|-----------------------|------------------------|--------------------------|
| <i>:active</i> | <i>:link</i> | <i>:last-of-type</i> | <i>:read-only</i> |
| <i>:disabled</i> | <i>:default</i> | <i>:not</i> | <i>:root</i> |
| <i>:first-child</i> | <i>:enabled</i> | <i>:nth-child</i> | <i>:visited</i> |
| <i>:focus-within</i> | <i>:fullscreen</i> | <i>:nth-of-type</i> | <i>:nth-last-of-type</i> |
| <i>:invalid</i> | <i>:indeterminate</i> | <i>:out-of-range</i> | <i>:only-of-type</i> |
| <i>:left</i> | <i>:last-child</i> | <i>:right</i> | <i>:read-write</i> |
| <i>:checked</i> | <i>:matches()</i> | <i>:valid</i> | <i>:required</i> |
| <i>:empty</i> | <i>:dir</i> | <i>:nth-last-child</i> | <i>:scope</i> |
| <i>:first-of-type</i> | <i>:first</i> | <i>:only-child</i> | <i>:target</i> |
| <i>:hover</i> | <i>:focus</i> | <i>:optional</i> | |
| <i>:lang</i> | <i>:in-range</i> | | |

Συμπληρωματικά με τον επιλογέα τύπου, έχουμε και τον επιλογέα ψευδο-στοιχείου (pseudo-element), με τον οποίο μπορούμε να επιλέξουμε ένα *τμήμα* του στοιχείου.

Για παράδειγμα, με το ψευδο-στοιχείο **::first-line** μπορούμε να επιλέξουμε την πρώτη γραμμή μιας παραγράφου.

Άλλα ενδιαφέροντα ψευδο-στοιχεία είναι τα **::before** και **::after**, που εισάγουν περιεχόμενο στην αρχή ή στο τέλος του επιλεγμένου στοιχείου:

```
/* Εισάγει τη λέξη "Προσοχή: " στην αρχή κάθε στοιχείου p με κλάση .warning
↪ */
p.warning::before {
    content: "Προσοχή: "
}
```


Μπορούμε να συνδυάσουμε επιλογείς για να κατασκευάσουμε πιο σύνθετους:

- $A\ B$: Επιλέγει τα B που είναι απόγονοι των A^*
- $A>B$: Επιλέγει τα B που είναι άμεσοι απόγονοι των A
- $A\sim B$: Επιλέγει τα B που είναι γείτονες των A
- $A+B$: Επιλέγει τα B που είναι άμεσοι γείτονες των A

* delimiter είναι το space.

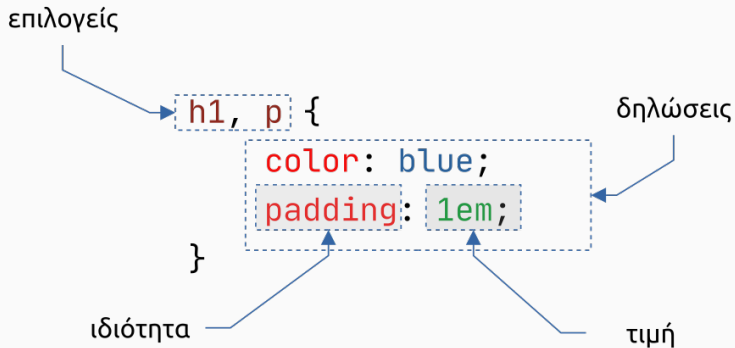
Μπορούμε να εφαρμόσουμε έναν κανόνα σε πολλούς επιλογείς, αρκεί να τους χωρίσουμε με κόμμα.

```
h1, h2, h3, h4, h5, h6 {  
  font-family: helvetica, 'sans serif';  
}
```

Αν η λίστα περιέχει άκυρο επιλογήα (συντακτικό σφάλμα), αγνοείται όλη η λίστα.

Σε ποια στοιχεία θα εφαρμοστεί το παρακάτω στυλ;

```
table td, table th {  
    border: 1px solid black; padding: 0.5em 0.5em 0.4em;  
}
```



Σχήμα 2: Ένας τυπικός κανόνας CSS.

- Προκαθορισμένες λέξεις: {align: **left**}, {**display**: inline} κλπ.
- Διαστάσεις: επί τοις εκατό, αριθμοί χωρίς/με μονάδες,
- Ειδικοί τύποι: χρώμα, εικόνα
- Συναρτήσεις (url(), rgb() ...)
- Αλφαριθμητικά: με εισαγωγικά, χωρίς εισαγωγικά, και τα δύο.

Τα πίξελ (px) είναι **απόλυτες** τιμές, θα μένουν ίδιες ανεξάρτητα από άλλες τιμές.

Άλλες απόλυτες τιμές:

- q, mm, cm, in: Quarter millimeter, milimeter, centimeter, inch.
- pt, pc: point (1/72 της ίντσας), pica (12 points).

Μονάδες και τιμές 3

- Τα **em** είναι σχετική τιμή, με βάση το μέγεθος της γραμματοσειράς του τρέχοντος στοιχείου.
- Το προκαθορισμένο font-size, πριν εφαρμοστεί άλλη CSS, είναι 16 pixel (1em=16px, 2em=32px κ.ο.κ). Ωστόσο, το μέγεθος της γραμματοσειράς κληρονομείται από το γονικό στοιχείο (inherited).
- Η τιμή της μονάδας **rem** της εξαρτάται μόνο από το μέγεθος της γραμματοσειράς του **:root** (δηλ. του στοιχείου html), και όχι του γονικού στοιχείου (δηλ. δεν κληρονομείται).
- Άλλες μονάδες είναι το **ex** (το ύψος του x), **ch** (πλάτος του 0), και τα πιο χρήσιμα **vw**, **vh** (1vh = 1% του ύψους του viewport).
- Το % ορίζει μια τιμή ως ποσοστό της τιμής του γονικού στοιχείου.

- Μετασχηματισμοί γεωμετρίας (`skew()`, `scale()`, ...)
- Μαθηματικές συναρτήσεις (`calc()`, `max()`, ...)
- Φίλτρα (`blur()`, `drop-shadow()`, ...)
- Χρώματα (`rgb()`, `color()`, ...)
- Εικόνες (`linear-gradient()`, `image()`, ...)
- Γραμματοσειρές, Σχήματα

- `p { background: url('myimage.png'); }`
- `div { width: calc((90vw - 20px) + (100% / 9)); }`
- `span { color: rgb(255, 0, 0); }`
- ...

CSS Tricks: A Complete Guide to CSS Functions

Χρωματικό σύστημα μεγέθους 24 bit. Τα χρώματα εκφράζονται σαν συνδυασμοί **R**ed, **G**reen, **B**lue.

- **keywords:** 165 ονοματισμένα χρώματα.

```
p { background-color: red; }
```

- δεκαεξαδικές τιμές:

```
p {background-color: #ff0000}
```

- rgb(r, g, b):

```
p {background-color: rgb(255, 0, 0)}
```

- rgba(r, g, b, a): αδιαφάνεια (1=καθόλου διαφανές, 0=τελείως διαφανές)

```
p {background-color: rgb(255, 0, 0, 0.7)}
```

Όταν πολλοί κανόνες επιλέγουν την ίδια ιδιότητα ενός στοιχείου, τότε υπερισχύει ένας μόνο κανόνας.

Από την πιο σημαντική προς τη λιγότερο σημαντική κατηγορία:

- **προέλευση του κανόνα** όπου δηλώνεται η τιμή και αν η τιμή έχει **σημανθεί ως σημαντική ή όχι (!important):**

```
p { background-color: red; border: none !important;}
```

- **ειδικότητα - specificity**, πόσο στοχευμένος είναι ο κανόνας

```
inline > id > class > element
```

- **σειρά εμφάνισης - source order** στον κώδικα: ο πιο πρόσφατος κανόνας θα υπερισχύσει του παλιότερου.
- αν η τιμή κληρονομείται ή όχι (**inherited**)

Ένας κανόνας μπορεί να υπάρχει στις εξής τρεις προελεύσεις:

1. Στο λεγόμενο **στυλ φυλλομετρητή** (*user agent style*)
2. Στο **στυλ χρήστη** (*user defined style*)
3. Στα στυλ που παρέχει ο δημιουργός της ιστοσελίδας, με τρεις τρόπους.

- **Εξωτερικό αρχείο CSS.:**

```
<link rel="stylesheet" href="filename-url.css">
```

- **Εσωτερικό CSS (ή CSS σελίδας):**

```
<head>
```

```
  <style> p { color: darkgray; } </style>
```

- **Ενσωματωμένο ή inline στυλ:**

```
<p style="color: darkgray"> ... </p>
```

Το εξωτερικό CSS είναι η προτιμώμενη πρακτική.

Όλοι οι κανόνες CSS είναι είτε μη σημαντικοί (normal) είτε σημαντικοί (important). Σημαντικοί είναι αυτοί που η τιμή τους τελειώνει με !important, για παράδειγμα:

```
span .serious {  
    color: red !important;  
}
```

Μια σημαντική (!important) ιδιότητα έχει αυτομάτως μεγαλύτερο βάρος, δηλ. ανεξάρτητα σε ποια πηγή τη συναντήσαμε, η τιμή της θα επικρατήσει.

Ωστόσο αν η ίδια ιδιότητα έχει σημανθεί σαν σημαντική σε περισσότερες από μια πηγές, η προτεραιότητα επικράτησης θα **αντιστραφεί**.

| Σειρά προτεραιότητας | Πηγή κανόνων |
|----------------------|--------------------------------------|
| 1 | Δηλώσεις μετάβασης (transition) |
| 2 | !important στο στυλ του φυλλομετρητή |
| 3 | !important στο στυλ του χρήστη |
| 4 | !important στο στυλ του δημιουργού |
| 5 | Animation |
| 6 | Στο στυλ του δημιουργού |
| 7 | Στο στυλ του χρήστη |
| 8 | Στο στυλ του φυλλομετρητή |

Ερώτηση:

Γιατί το στυλ χρήστη έχει μικρότερη προτεραιότητα από το στυλ δημιουργού, παρόλο που το πρώτο είναι ρητή ενέργεια του χρήστη;

Αν η προέλευση δυο ή περισσότερων κανόνων είναι η ίδια, τότε θα επικρατήσει αυτός με τον πιο συγκεκριμένο επιλογέα.

Η ειδικότητα ενός επιλογέα συντίθεται από τρία συστατικά στοιχεία:

- το πλήθος των **επιλογέων id**,
- το πλήθος των **επιλογέων κλάσης** (π.χ. `.first`), των **επιλογέων ιδιότητας** (π.χ. `[type=button]`), και των **επιλογέων ψευδοκλάσης** (π.χ. `:link`),
- το πλήθος των **επιλογέων τύπου** (π.χ. `p`) και των **επιλογέων ψευδοστοιχείων** (όπως το `::first-line`).

Παράδειγμα

- `#friendly` -> ειδικότητα (1, 0, 0)
- `#friendly #animal` -> ειδικότητα (2, 0, 0)
- `p.first em` -> ειδικότητα (0, 1, 2)

Επικρατεί πάντα ο επιλογέας που έχει τη μεγαλύτερη τιμή από αριστερά προς τα δεξιά, δηλ. η ειδικότητα (1, 0, 0), είναι πιο ισχυρή από την (0, 1, 2).

Σειρά εμφάνισης - Source order

Αν η προέλευση δυο ή περισσότερων κανόνων είναι η ίδια και οι επιλογείς έχουν την ίδια ειδικότητα, τότε θα επικρατήσει η πιο πρόσφατη δήλωση, δηλ. αυτή που συνάντησε τελευταία ο φυλλομετρητής όταν φόρτωσε την ιστοσελίδα:

```
/* Τελικά θα έχουμε κίτρινα γράμματα σε μαύρο φόντο */
```

```
p {  
    color: yellow;  
    background-color: navy;  
}  
  
p {  
    background-color: black;  
}
```

Μερικές από τις ιδιότητες κληρονομούνται, άλλες όχι.

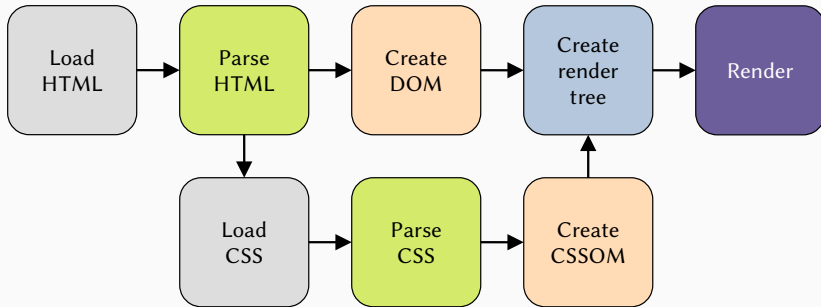
Για παράδειγμα, η τιμή της ιδιότητας `font-family` είναι κληρονομούμενη, ενώ η τιμή της `margin` όχι.

Αυτό σημαίνει πως αν αλλάξουμε την τιμή της `font-family` π.χ. στο `<body>` θα κληρονομηθεί και από τα απόγονα στοιχεία, αλλά όχι και η τιμή του `margin`.

Για κάθε ιδιότητα, μπορούμε να αλλάξουμε τη συμπεριφορά αυτή με τις τιμές:

- `p { padding : inherit }`: η τιμή της ιδιότητας θα κληρονομηθεί από το γονικό στοιχείο
- `p { padding : initial }`: παίρνει την προκαθορισμένη τιμή (default), όπως ορίζεται στην προδιαγραφή.
- `p { padding : unset }`: η τιμή επιστρέφει στην φυσική της κατάσταση (αν κληρονομείται, τότε θα κληρονομηθεί, αλλιώς γίνεται initial).

DOM parsing και CSS

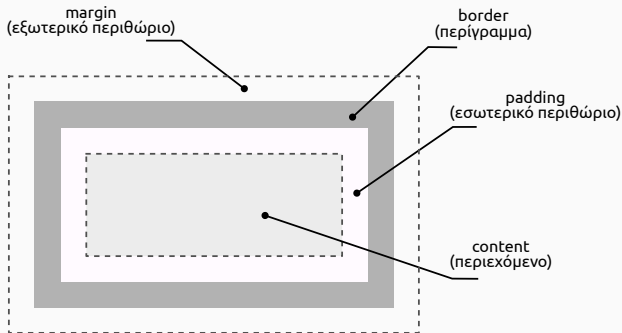


Σχήμα 3: DOM parsing και CSS

Το μοντέλο Box

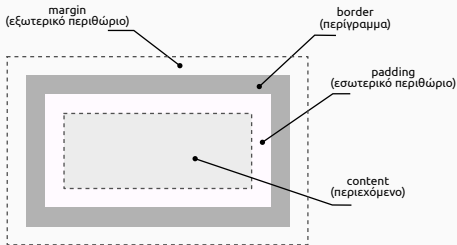
Το μοντέλο Box

- Κάθε στοιχείο HTML εμφανίζεται σαν ένα ορθογώνιο παραλληλόγραμμο κουτί.
- Κάθε κουτί αποτελείται από τέσσερις περιοχές.



Σχήμα 4: Το μοντέλο Box.

Το μοντέλο Box 2

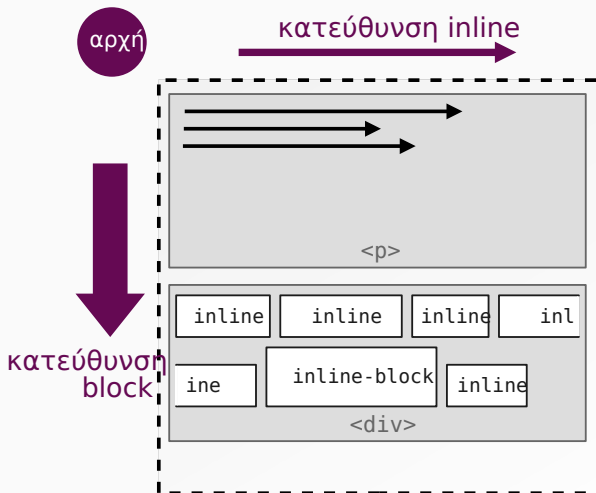


Σχήμα 5: Το μοντέλο Box.

- Προκαθορισμένη συμπεριφορά: `box-sizing: content-box;` - σημαίνει πως ο ορισμός του μεγέθους του στοιχείου (πχ. με `width:`, `height:`) ορίζει το μέγεθος της περιοχής content.
- Αν `box-sizing: border-box;` τότε οι `width:`, `height:` κλπ. αφορούν το μέγεθος του κουτιού του περιγράμματος.

- Περιεχόμενο: `width`, `min-width`, `max-width`, `height`, `min-height`, `max-height`.
- Εσωτ. περιθώριο: `padding-top`, `padding-right`, `padding-bottom`, `padding-left`, `padding`.
- Περίγραμμα: `border`, `border-width`. Το υπόβαθρο του στοιχείου (`background-color`, `background-image`) φτάνει μέχρι και το περίγραμμα.
- Εξωτ. περιθώριο: `margin-top`, `margin-right`, `margin-bottom`, `margin-left`, `margin`.

Η κανονική ροή

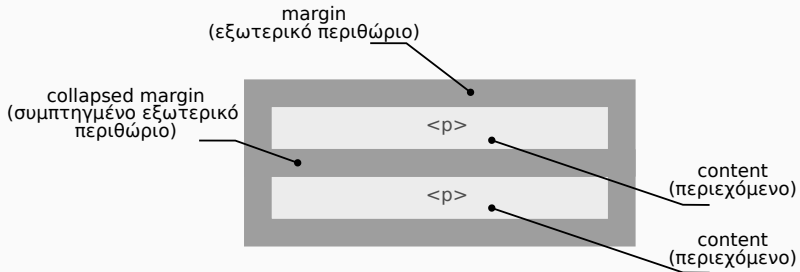


Σχήμα 6: Η κανονική ροή είναι ο προκαθορισμένος τρόπος διάταξης στοιχείων στην οθόνη.

Τα στοιχεία **block**

- καταλαμβάνουν το 100% του πλάτους του προγόνου τους και όσο ύψος χρειάζονται για να δείξουν το περιεχόμενό τους,
- τοποθετούνται από πάνω προς τα κάτω (block direction) και κάθε στοιχείο block είναι μόνο του στην οριζόντια διάσταση (δηλ. πριν και μετά έχουμε αλλαγή γραμμής),
- ρ, div ...
- η απόσταση μεταξύ δύο διαδοχικών κουτιών καθορίζεται από το **margin** τους
- Ισχύει πάντα μόνο το μεγαλύτερο από τα δύο **margin**, το άλλο αγνοείται (**margin collapse**)

Margin collapse



Τα στοιχεία inline

- καταλαμβάνουν όσο πλάτος και ύψος χρειάζονται για να δείξουν το περιεχόμενό τους,
- τοποθετούνται το ένα μετά το άλλο στην οριζόντια διάσταση (inline direction), μέχρι να εξαντληθεί το διαθέσιμο πλάτος, έπειτα αναδιπλώνονται σε νέα γραμμή.

Το μέγεθος ενός στοιχείου μπορεί να εξαρτάται από το μέγεθος της περιοχής που βρίσκεται (το λεγόμενο **containing block**)

- Το **containing block** είναι συνήθως η περιοχή content του κοντινότερου προγόνου που είναι block.

Η ιδιότητα `display`

- `display: block`: Το στοιχείο θα γίνει block.
- `display: inline`: Το στοιχείο θα γίνει inline.
- `display: inline-block`: Το στοιχείο θα γίνει inline, αλλά θα γίνουν σεβαστά όλα τα margin / padding (στο inline δεν γίνονται όλα σεβαστά).
- `display: none`: Το στοιχείο δε θα προβληθεί στην οθόνη (αφαιρείται από το render tree).

Επιλογή γραμματοσειράς

Με την ιδιότητα `font-family` ορίζουμε μία ή περισσότερες γραμματοσειρές για το ίδιο στοιχείο.

Η ιδιότητα είναι κληρονομούμενη.

Δεν μπορούμε να ξέρουμε ποιες γραμματοσειρές θα είναι διαθέσιμες στους φυλλομετρητές των τελικών χρηστών.

```
/* Αν δεν βρεθεί η Helvetica, θα εφαρμοστεί η Verdana, αλλιώς η προκαθορισμένη  
↪ sans-serif στο φυλλομετρητή */  
body { font-family: Helvetica, Verdana, sans-serif; }
```


Επιλογή γραμματοσειράς 2

Επιτρέπεται να δηλώσουμε και κάποιον από τους πέντε γενικούς τύπους γραμματοσειρών

- serif
- sans-serif
- monospace
- cursive
- fantasy

Μπορούμε να ορίσουμε γραμματοσειρά που βρίσκεται στο διαδίκτυο, π.χ με τη Google Fonts, την Font Library, (κυρίως με άδεια CC BY-SA), τη Font Squirrel κ.ά.

Επιλογή γραμματοσειράς 3

Μερικές χρήσιμες ιδιότητες που μπορούν να χρησιμοποιηθούν σε συνάρτηση με τις γραμματοσειρές είναι:

- **font-size.** Καθορίζει το μέγεθος της γραμματοσειράς.
- **font-style.** `normal`, `italic`, `oblique`
- **font-weight.** Είτε αριθμητικά, από 100, 200, ... ως 900, είτε σε σχέση με το τρέχον βάρος της γραμματοσειράς ως `lighter` ή `bolder`, είτε τέλος με τις λέξεις `normal` και `bold` που ισοδυναμούν με 400 και 700 αντίστοιχα.
- **text-decoration.** Προσθέτει μια οριζόντια γραμμή στο κείμενο. Δυνατές τιμές είναι οι `none`, `underline` (υπογράμμιση), `overline` (υπεργράμμιση), `line-through` (διαγράμμιση) και μάλιστα μπορεί να πάρει και δύο τιμές ταυτόχρονα, π.χ.
`{text-decoration: underline overline;}`.

Παράδειγμα επιλογής γραμματοσειράς

```
p {  
    text-decoration-line: underline;  
    text-decoration-style: dotted; /* ή double, dashed, wavy, solid */  
    text-decoration-color: blue;  
    /* ισοδυναμεί με */  
    text-decoration: underline dotted blue;  
}
```

Μπορούμε να καθορίσουμε για το υπόβαθρο ένα χρώμα, ή μια διαβάθμιση (gradient) ή μια εικόνα.

Ο καθορισμός του χρώματος γίνεται με την ιδιότητα `background-color`

- χρωματίζει το υπόβαθρο του κουτιού περιεχομένου και του εσωτερικού περιθωρίου

Διακόσμηση υποβάθρου 2

Πάνω από το χρώμα υποβάθρου, μπορούμε να ορίσουμε και μια εικόνα υποβάθρου, με την ιδιότητα `background-image`.

Η ιδιότητα αυτή δέχεται μία ή περισσότερες εικόνες, που ζωγραφίζονται η μία κάτω από την άλλη.

```
div {  
    background-image:  
        radial-gradient(rgba(255, 0, 0, 0.5), rgba(0, 255, 0, 0.5)),  
        url("https://openclipart.org/image/400px/335407");  
    width: 30vh;  
    height: 30vh;  
    border: 10px dotted red;  
    padding: 10px;  
    background-clip: border-box;  
    background-origin: content-box;  
}
```

Πολύ χρήσιμες είναι οι τιμές της ιδιότητας `background-repeat`, που καθορίζουν τί θα γίνει αν οι διαστάσεις του κουτιού είναι μεγαλύτερες από την εικόνα.

Η εικόνα μπορεί να επαναλαμβάνεται (`repeat`), που είναι η προκαθορισμένη συμπεριφορά, να επαναλαμβάνεται μόνο κάθετα ή οριζόντια (`repeat-y`, `repeat-x`) ή να μην επαναλαμβάνεται (`no-repeat`). Με τις τιμές `round` και `space` έχουμε ακόμη περισσότερο έλεγχο.

Οι μεταβάσεις (transitions) είναι ο πιο απλός τρόπος που έχουμε για κίνουμενα εφέ.

Όταν αλλάζουμε την τιμή μιας ιδιότητας, π.χ. με την ψευδοκλάση `:hover`, η μετάβαση από την αρχική στην τελική κατάσταση είναι ακαριαία.

Με την ιδιότητα `transition` μπορούμε να περιγράψουμε πιο ομαλές μεταβάσεις, ή, με άλλα λόγια, να αλλάξουμε τις τιμές τους σταδιακά και όχι απότομα.

Μεταβάσεις - Παράδειγμα

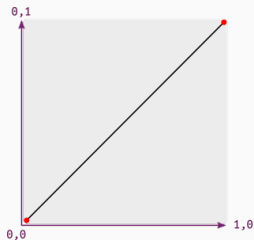
```
<a href="">Ενας σύνδεσμος</a>
```

```
a {  
    /* color: blue;  
    background-color: white; */  
    transition-property: background-color color ;  
    transition-duration: 500ms;  
    transition-timing-function: ease-in-out;  
    transition-delay: 0s;  
}
```

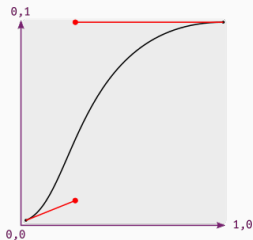
```
a:hover {  
    background-color: navy;  
    color: lightyellow;  
}
```


Μεταβάσεις 2

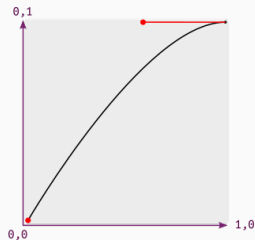
Η ιδιότητα `transition-timing-function` μπορεί να πάρει κάποιες έτοιμες συναρτήσεις, όπως `ease`, `ease-in`, `ease-out`, `ease-in-out`, `linear` κ.ά.



linear
`cubic-bezier(0, 0, 1, 1)`



ease
`cubic-bezier(0.25, 0.1, 0.25, 1)`



ease-out
`cubic-bezier(0, 0, 0.58, 1)`