



Θέματα Προγραμματισμού Διαδικτύου ~ CSS ~

Στελιος Σφακιανάκης
Εαρινό 2020



Εισαγωγή στο CSS ("Cascading Style Sheets")



Γιατί CSS;

- Η HTML χρησιμοποιείται για να περιγράψει τη δομή της σελίδας, δηλ. τις παραγράφους, κεφαλίδες, εικόνες, λίστες, κλπ
- Η HTML δεν (πρέπει να) περιγράφει τη μορφοποίηση δηλ. το πώς φαίνεται στον browser
 - Σε κάποιες περιπτώσεις ο browser έχει έναν default τρόπο που απεικονίζει κάποια tags (π.χ. λίστες) αλλά αυτό μπορεί να διαφέρει ανάλογα τον browser ή την συσκευή (π.χ. κινητά και ταμπλέτες)
- Θέλουμε επομένως να διαχωρίσουμε το **περιεχόμενο** από την **εμφάνιση!**
- Για το περιεχόμενο χρησιμοποιούμε HTML και για την εμφάνιση/μορφοποίηση τα "στυλ" (*stylesheets*)!



Παράδειγμα στο w3schools

https://www.w3schools.com/css/css_intro.asp

Αλλάζοντας μόνο το stylesheet, το ίδιο περιεχόμενο φαίνεται εντελώς διαφορετικά!

Welcome to My Homepage

Use the menu to select different Stylesheets

Stylesheet 1

Stylesheet 2

Stylesheet 3

Stylesheet 4

No Stylesheet

Same Page Different Stylesheets

This is a demonstration of how different stylesheets can change the layout of your HTML page. You can change the layout of this page by selecting different stylesheets in the menu, or by selecting one of the following links: [Stylesheet1](#), [Stylesheet2](#), [Stylesheet3](#), [Stylesheet4](#).

No Styles

This page uses DIV elements to group different sections of the HTML page. Click here to see how the page looks like with no stylesheet: [No Stylesheet](#).

Side-Bar

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.



CSS

- Οι εντολές CSS ορίζουν πώς μορφοποιείται το περιεχόμενο
 - Είναι μια "δηλωτική" γλώσσα προγ/μου, παντοδύναμη, αλλά ..πολύπλοκη!
- Επειδή διαχωρίζεται από το περιεχόμενο (HTML) το CSS ενός ιστοτόπου μπορεί να γραφεί από άλλη ομάδα (π.χ. από τους graphic designers)
- Οι εντολές CSS μπορεί να είναι σε τρία σημεία:
 - Σε ένα "εξωτερικό" αρχείο στο οποίο κάνει αναφορά η HTML σελίδα (external CSS)
 - Μέσα στην HTML σελίδα (Internal CSS)
 - Ως style attribute σε tags της HTML σελίδας (inline CSS)



Ορισμός CSS

- Εξωτερικό CSS: βάζουμε ένα **link** tag στο **<head>** του HTML εγγράφου με τη διεύθυνση από την οποία ο browser θα "κατεβάσει" το CSS αρχείο, π.χ.

```
<link rel="stylesheet" type="text/css" href="mysite.css">
```

- Εσωτερικό CSS: βάζουμε τις εντολές σε ένα **<style>** tag μέσα στο **<head>**, π.χ.

```
<style>
```

```
    body { font-family: serif; }
```

```
</style>
```

- Inline CSS: βάζουμε τις εντολές (που αφορούν ένα συγκεκριμένο tag) στην style attribute του tag:

```
<p style="text-indent: 15px">...
```



Συνήθως το αποφεύγουμε!



Σύνταξη εντολών CSS

- **Γενική δομή:**

```
επιλογέας {  
    ιδιότητα-1: τιμή-1 ;  
    ιδιότητα-2: τιμή-2 ;  
    ...  
}
```

```
/* Παράδειγμα: */  
p {  
    color: red;  
}
```

- Ο "επιλογέας" (selector) καθορίζει σε ποιο (ή ποια) tags στο HTML έγγραφο θα εφαρμοστούν οι κανόνες που βρίσκονται μέσα στις αγκύλες
- Οι κανόνες μέσα στις αγκύλες είναι χωρισμένοι με ερωτηματικό, και αποτελούνται από ζευγάρια ονόματος ιδιότητας και τιμής.
- Σχόλια μπαίνουν μέσα σε /* και */ (όπως στις C, C++, και Java)



Επιλογείς

- Η πιο απλή περίπτωση είναι να δώσουμε το όνομα ενός tag όπως στο προηγούμενο παράδειγμα:

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

- Το παραπάνω σημαίνει ότι κάθε παράγραφος (δηλ. ότι κείμενο περιέχεται μέσα σε **<p>**) θα έχει κόκκινο χρώμα γραμματοσειράς.
- Αν δώσουμε παραπάνω από ένα tags, π.χ. **p em { ... }** τότε οι κανόνες εφαρμόζονται σε όλα τα **** που περιέχονται μέσα σε **<p>**.



Επιλογείς βάσει class

- Σχεδόν όλα τα tags μπορούν να έχουν ένα attribute που λέγεται class και που περιέχει τιμές που τους δίνουμε εμείς. Οι τιμές είναι αλφαριθμητικά (strings) και είναι χωρισμένες με κενούς χαρακτήρες (διαστήματα), π.χ. `<p class="perigrabi">...</p>`
 - Το "perigrabi" είναι το class name που δώσαμε και μπορούμε να έχουμε πολλά tags να έχουν το ίδιο
- Στο CSS μπορούμε να βάλουμε κανόνες μόνο για τα elements με αυτό το class ως εξής:
`.perigrabi { ... }` (Προσοχή: ξεκινάει με τελεία!)



Επιλογείς βάσει id

- Όλα τα tags μπορούν να έχουν ένα id attribute το οποίο παρέχει ένα αναγνωριστικό (identifier) που πρέπει να είναι μοναδικό για όλο το έγγραφο HTML
- Η τιμή του attribute μπορεί να είναι οτιδήποτε διαλέξουμε, ένα αλφαριθμητικό (string) χωρίς κενά (space, tab, κλπ)

```
<p id="my-para">...</p>
```

- Σε CSS βάζουμε κανόνες μόνο για τα element αυτό ως εξής:

```
#my-para { ... }
```



Συνδυασμοί selectors

- `p a.news {...}` : Όλα τα links (`a`) με class "news" που βρίσκονται μέσα ("έχουν πρόγονο") σε ένα `p`
- `ul#menu li.contacts a {...}` : Όλα τα links (`a`) που είναι μέσα σε `li` με class="contacts" τα οποία είναι στο `ul` με id="menu"
- "Combinators"
 - `ul > p {...}`: Το `p` που είναι κατευθείαν "παιδί" του `ul` (όχι απόγονος)
 - `ul ~ p {...}`: Όλα τα `p` που είναι ακολουθούν ένα `ul`
 - `ul + p {...}`: Το `p` που ακολουθεί αμέσως μετά από ένα `ul` (δηλ. η 1η παράγραφος μετά από `ul`)



"Ψευδοκλάσεις"

- Οι "pseudo-classes" επιτρέπουν να επιλέξουμε ένα element βάσει της κατάστασης του (π.χ. αν ένας σύνδεσμος οδηγεί σε σελίδα που την έχει επισκεφθεί ο χρήστης)

```
selector:pseudo-class {  
    property: value;  
}
```

- Παραδείγματα:
 - **:visited** για links που ο χρήστης έχει επισκεφθεί
 - **:hover** όταν ο χρήστης "περνάει" το mouse πάνω από το element
 - **:focus** όταν ένα element έχει το "focus" (π.χ. ο χρήστης έχει κάνει κλικ σε ένα στοιχείο ή το έχει επιλέξει με το tab)



CSS ιδιότητες (properties)

- Με τους selectors που είδαμε ήδη επιλέγουμε κάποια από τα elements του εγγράφου HTML και με τις ιδιότητες, που βάζουμε μέσα σε αγκύλες (`{...}`), ορίζουμε τις εντολές μορφοποίησης που θα εφαρμοστούν πάνω τους. Π.χ. :

`h1 {`

`font-size: 14pt;`

`font-family: Futura, sans-serif;`

`font-weight: bold;`

`}`

`p {`

`font-size: 12pt;`

`font-family: Garamond, serif;`

`}`



Ιδιότητες για το κείμενο

- **font-family**: Γραμματοσειρά (ή λίστα από γραμ/ρες σε σειρά προτίμησης χωρισμένες με κόμμα) που θα χρησιμοποιηθεί. Καλό είναι πάντα να βάζουμε τελευταία κάποια γενική, π.χ. **serif**, **sans-serif**, **monospace**
- **font-size**: μέγεθος γραμμάτων. Μονάδες: "**em**" (το πλάτος του κεφαλαίου M), "**pt**" (points, 1/72 της ίντσας), "**px**" (pixels), "**...%**" (ποσοστό, π.χ. **200%**, σε σχέση με το γύρω κείμενο)
- **font-weight**: μπορεί να είναι **normal**, **bold** ή σχετικά με το parent element: **bolder**, **lighter**
- **font-style**: μπορεί να είναι **italic**, **normal**, **oblique**
- **font-variant**: παραλλαγές του font π.χ. **small-caps**, **normal**, **oblique**
- Μπορούμε να τα συνδυάσουμε σε μια ιδιότητα: **font: 1em bold small-caps Georgia, serif;**



Κείμενο: Χρώμα

- **color**: με τιμή όνομα (π.χ. **red**), RGB value (π.χ. **rgb(243, 243, 160)**), ή δεκαεξαδική τιμή (π.χ. **#f5f5dc**)
- **background-color**: χρώμα φόντου

https://developer.mozilla.org/en-US/docs/Web/CSS/color_value

Χρώματα RGB ("red-green-blue")

000000	000033	000066	000099	0000CC	0000FF	996600	996633	996666	996699	9966CC	9966FF
003300	003333	003366	003399	0033CC	0033FF	999900	999933	999966	999999	9999CC	9999FF
006600	006633	006666	006699	0066CC	0066FF	99CC00	99CC33	99CC66	99CC99	99CCCC	99CCFF
009900	009933	009966	009999	0099CC	0099FF	99FF00	99FF33	99FF66	99FF99	99FFCC	99FFFF
00CC00	00CC33	00CC66	00CC99	00CCCC	00CCFF	CC0000	CC0033	CC0066	CC0099	CC00CC	CC00FF
00FF00	00FF33	00FF66	00FF99	00FFCC	00FFFF	CC3300	CC3333	CC3366	CC3399	CC33CC	CC33FF
330000	330033	330066	330099	3300CC	3300FF	CC6600	CC6633	CC6666	CC6699	CC66CC	CC66FF
333300	333333	333366	333399	3333CC	3333FF	CC9900	CC9933	CC9966	CC9999	CC99CC	CC99FF
336600	336633	336666	336699	3366CC	3366FF	CCCC00	CCCC33	CCCC66	CCCC99	CCCCCC	CCCCFF
339900	339933	339966	339999	3399CC	3399FF	CCFF00	CCFF33	CCFF66	CCFF99	CCFFCC	CCFFFF
33CC00	33CC33	33CC66	33CC99	33CCCC	33CCFF	FF0000	FF0033	FF0066	FF0099	FF00CC	FF00FF
33FF00	33FF33	33FF66	33FF99	33FFCC	33FFFF	FF3300	FF3333	FF3366	FF3399	FF33CC	FF33FF
660000	660033	660066	660099	6600CC	6600FF	FF6600	FF6633	FF6666	FF6699	FF66CC	FF66FF
663300	663333	663366	663399	6633CC	6633FF	FF9900	FF9933	FF9966	FF9999	FF99CC	FF99FF
666600	666633	666666	666699	6666CC	6666FF	FFCC00	FFCC33	FFCC66	FFCC99	FFCCCC	FFCCFF
669900	669933	669966	669999	6699CC	6699FF	FFFF00	FFFF33	FFFF66	FFFF99	FFFFCC	FFFFFF
66CC00	66CC33	66CC66	66CC99	66CCCC	66CCFF						
66FF00	66FF33	66FF66	66FF99	66FFCC	66FFFF						
990000	990033	990066	990099	9900CC	9900FF						
993300	993333	993366	993399	9933CC	9933FF						

- 6 δεκαεξαδικά ψηφία, τα 2 πρώτα για το κόκκινο, τα 2 επόμενα για πράσινο, και τα 2 τελευταία για το μπλε
- Με 2 16αδικά ψηφία έχουμε μέχρι 256 διαφορετικές αποχρώσεις, άρα συνολικά $256 \times 256 \times 256 = 16.777.216$ διαφορετικά χρώματα!



Κείμενο: πρόσθετα

- `text-align` (π.χ. `left`), `text-decoration` (π.χ. `underline`), `text-transform` (π.χ. `uppercase`), `text-indent`, `line-height`, `letter-spacing`, ...

https://developer.mozilla.org/en-US/docs/Learn/CSS/Styling_text/Fundamentals



Μορφοποίηση Λιστών

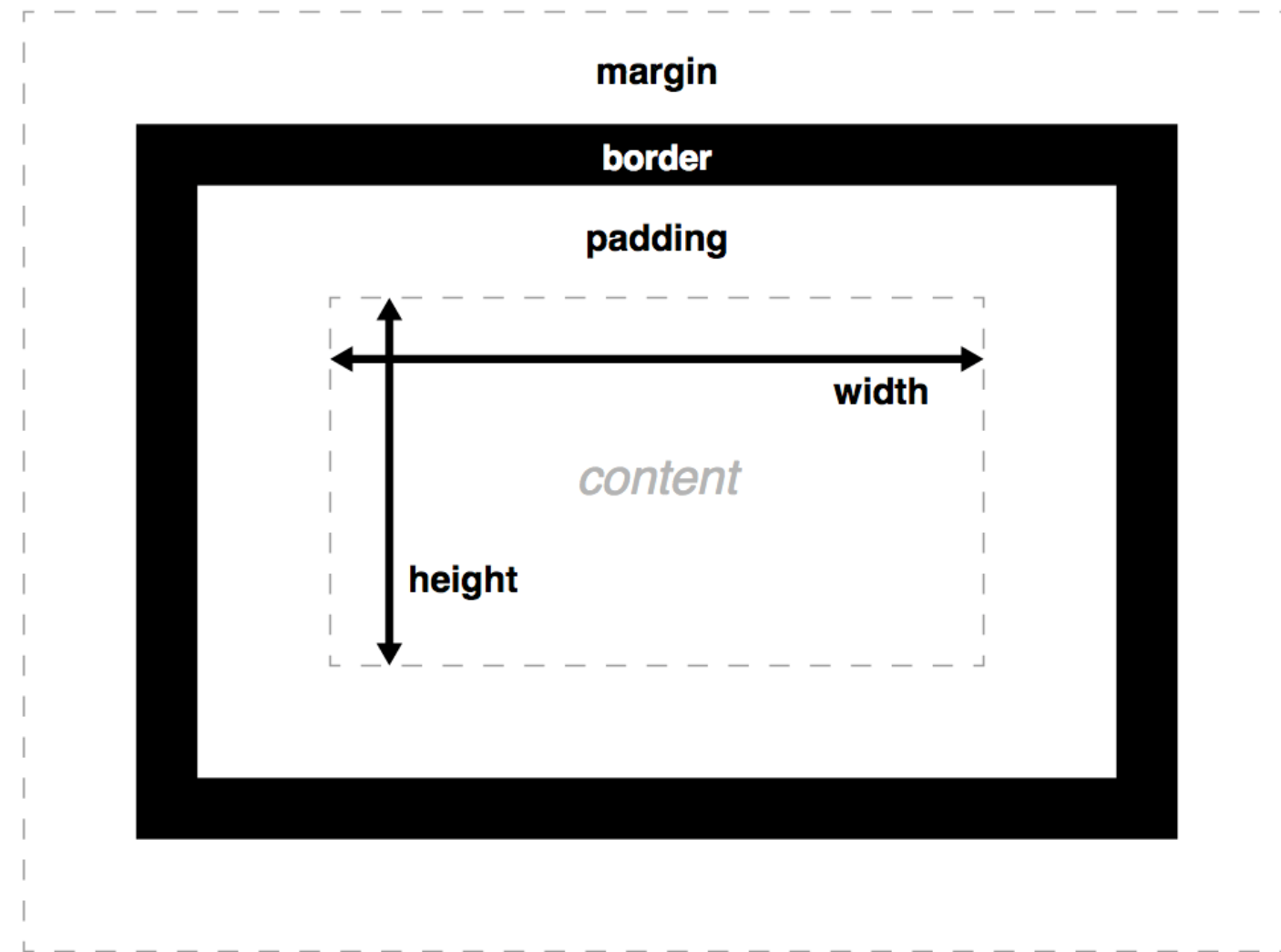
- **list-style-type**: Τί "κουκκίδα" θα χρησιμοποιηθεί. Ενδεικτικές τιμές: **none**, **disc**, **square**, **lower-alpha**, **upper-alpha**, **lower-greek**, **lower-latin**, **upper-latin**, κλπ
- **list-style-image**: Δίνεις τη δική σου εικόνα για να μπει ως "κουκκίδα", π.χ.

list-style-image: url("/images/img.png");



"Κουτιά" (Boxes)

- Κάθε element μέσα σε έγγραφο HTML έχει ένα νοητό ορθογώνιο παραλληλόγραμμο γύρω του που καθορίζει πώς τοποθετείται σε σχέση με τα υπόλοιπα περιεχόμενα
- Μπορούμε να ορίσουμε τις διαστάσεις του περιεχομένου (**width**, **height**) και το "πάχος" των **border**, **padding**, και **margin**

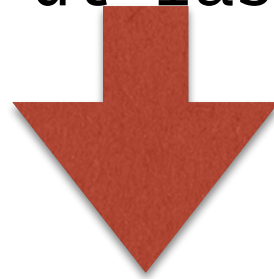




Παράδειγμα

```
<style>
p {
  padding: 10px;
  color: brown;
  background-color: orange;
  border: 2px solid green;
}
</style>
```

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do  
eiusmod tempor incididunt ut labore et ....</p>
```



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.



More goodies..

- **display**: πώς εμφανίζεται ένα element σε σχέση με τα υπόλοιπα. Π.χ. αν βάλουμε **none** τότε δεν φαίνεται καθόλου! Άλλες συχνές τιμές είναι το **block** (κάθετη τοποθέτηση) και το **inline** (οριζόντια τοποθέτηση)
- **background-image**: ορίζω μια εικόνα ως φόντο ενός element
- Υπάρχουν και πολλά περισσότερα να κάνει κανείς με το CSS (π.χ. layouts, positioning, etc.)



- Παράδειγμα από w3schools : CSS Navigation bar (menu) https://www.w3schools.com/Css/css_navbar.asp



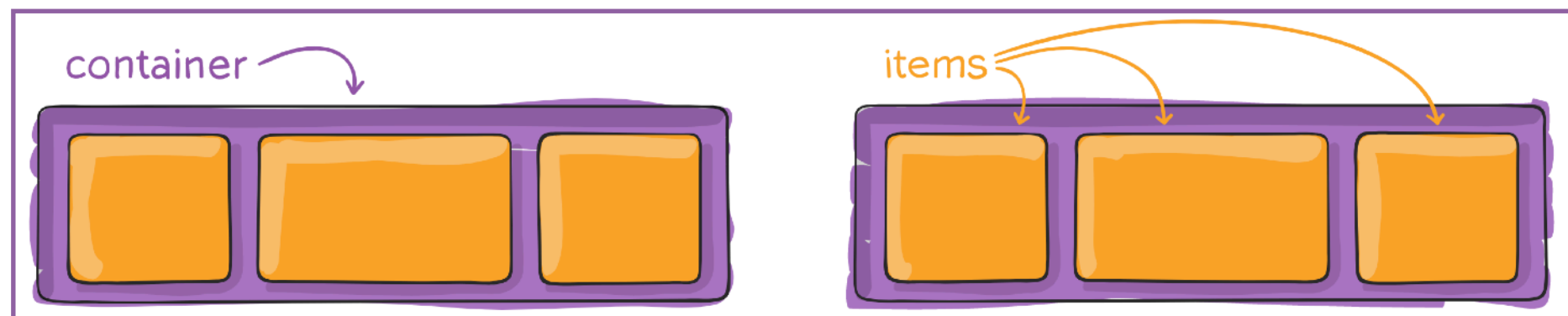
Flexbox

- Ευέλικτη διάταξη στοιχείων (layout) μέσα σε σελίδες HTML, παράδειγμα:
 - Ευκολα διατάσσουμε στοιχεία σε μια σειρά
 - Ή σε μια στήλη
 - Ή στο κέντρο
 - Κλπ
- Κατάλληλο για να κάνουμε layouts που παραδοσιακά ήταν δύσκολα



Flexbox - Δομή

- Γενικά με το flexbox τοποθετούμε κάποια elements (**div**, **a**, **ul**, ό,τι θέλουμε) σε μια διάταξη γραμμής ή στήλης
- Τα elements αυτά ("**items**") πρέπει να είναι μέσα σε "**container**" δηλ. να είναι "παιδιά" σε ένα άλλο element π.χ. **div**
- Οι βασικές εντολές του flexbox εφαρμόζονται στο container element
- Εκτός από τη κύρια διάταξη γραμμής ή στήλης ("**main axis**") υπάρχει και η κάθετη της ("**cross axis**")



<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>



👉 Βασικές εντολές για τον container

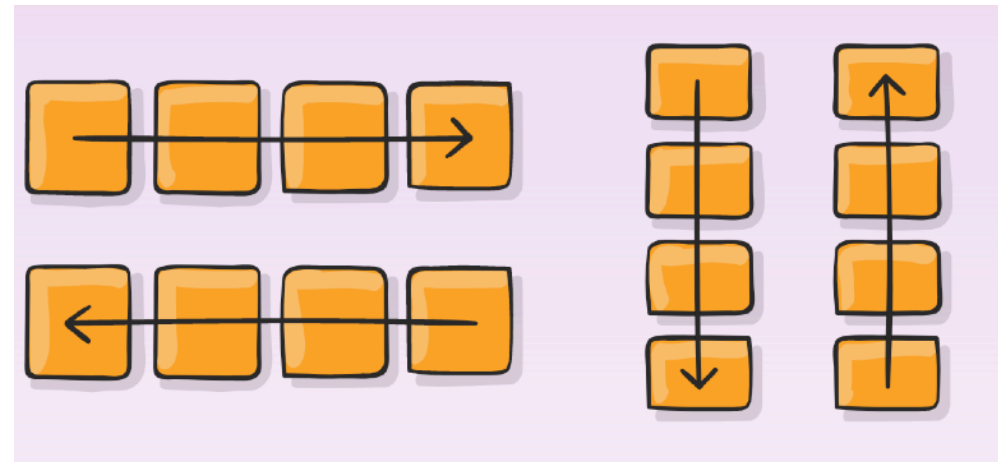
- Για να εφαρμόσουμε το flexbox βάζουμε πάντα τον κανόνα **display: flex;** στο container element!
- Επιπρόσθετα, μπορούμε να βάλουμε κανόνες για:
 - **flex-direction**, για να δηλώσουμε αν τα items θα είναι διαταγμένα σε γραμμή ή στήλη
 - **flex-wrap**: αν τα items θα κάνουν “wrap” αν δεν χωράνε (στη γραμμή ή στη στήλη με βάση το direction)
 - **justify-content**: πώς θα είναι στοιχισμένα τα items (π.χ. σε ίση απόσταση το ένα από το άλλο)
 - **align-items**: πώς θα είναι στοιχισμένα τα items στο “**cross axis**”
- Παράδειγμα:

```
.container {  
  display: flex;  
  flex-direction: row;  
  flex-wrap: wrap;  
  justify-content: space-around;  
  align-items: center;  
}
```



flex-direction:

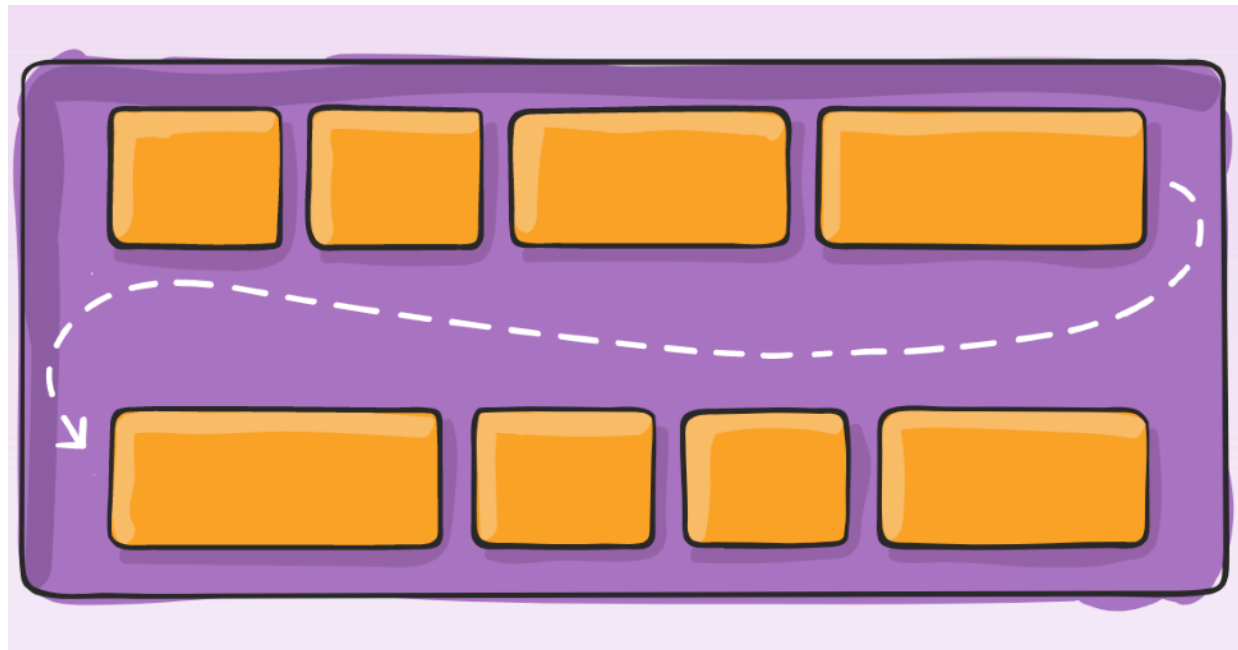
Κατεύθυνση των items



- Με το **flex-direction** ορίζουμε ποιο θα είναι το **main axis** δηλ. αν θα είναι τα στοιχεία σε γραμμή ή σε στήλη και επίσης την κατεύθυνση τους. Το default είναι το **row**
 - Π.χ. **flex-direction: column**
- Τέσσερις βασικές επιλογές:
 - **row** : διάταξη σε μια γραμμή όπως πάει το κείμενο (απο αριστερά στα δεξιά, *συνήθως*)
 - **column**: διάταξη σε στήλη (από πάνω προς τα κάτω, *συνήθως*)
 - **row-reverse** : σε γραμμή αλλά με ανάποδη σειρά δηλ. το 1ο item μπαίνει τελευταίο κλπ
 - **column-reverse** : σε στήλη αλλά με ανάποδη σειρά



flex-wrap: “Αναδίπλωση”

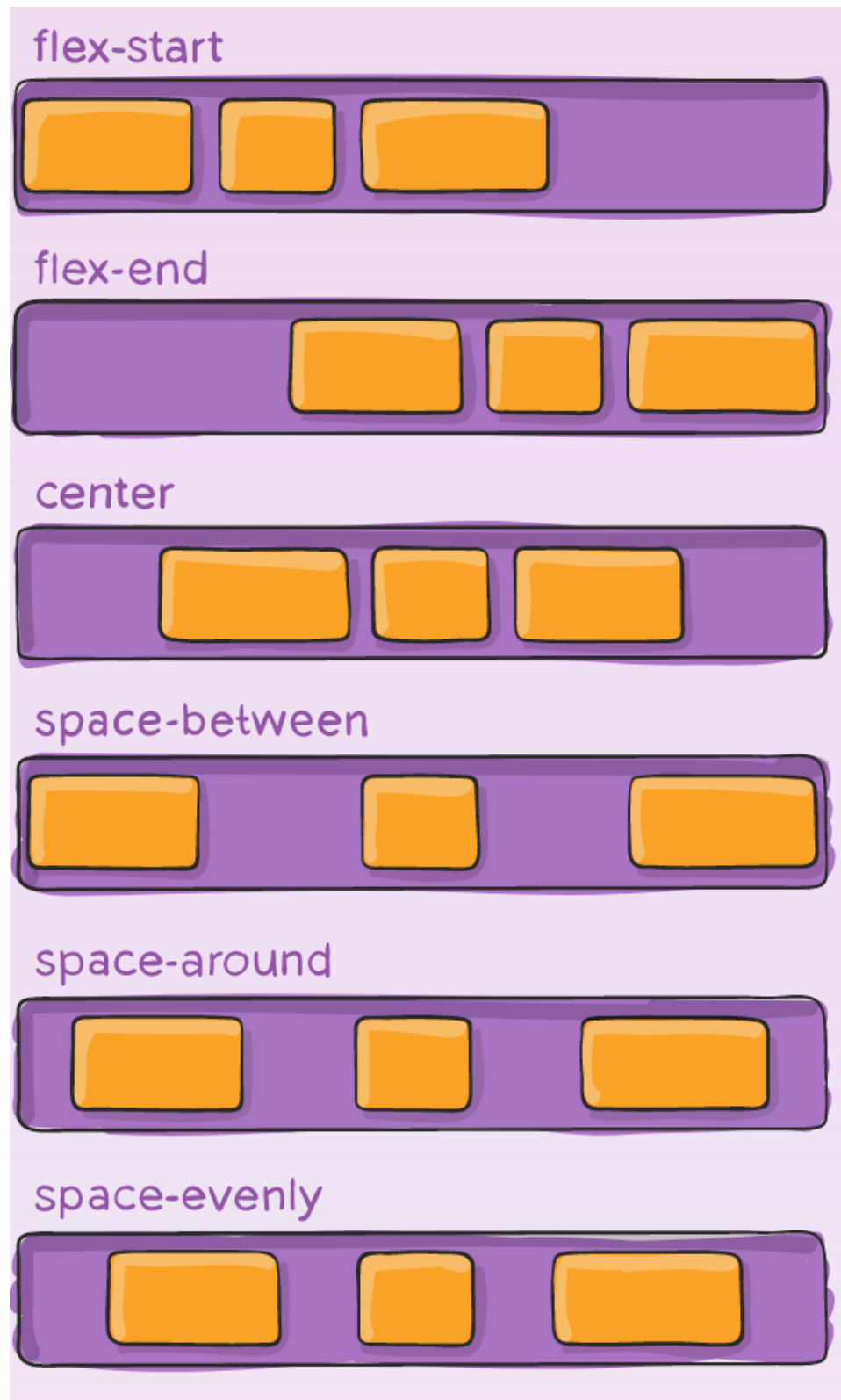


- Με το **flex-wrap** ορίζουμε αν τα items θα αναδιπλώνονται σε μια νέα γραμμή (ή στήλη) στην περίπτωση που δεν χωράνε. Το default είναι το **nowrap**
- Π.χ. **flex-wrap: wrap**



justify-content:

Διάταξη των items στο main axis

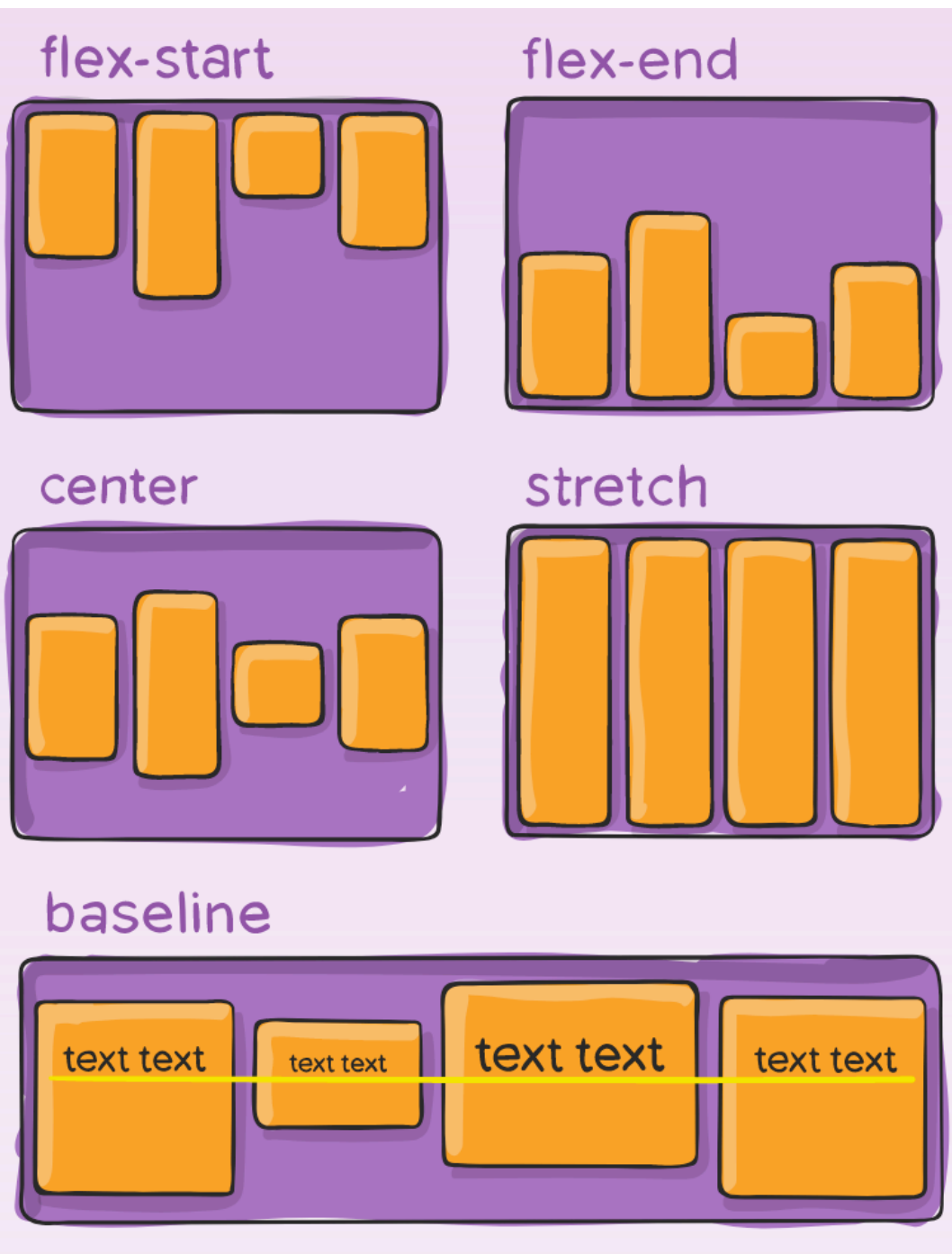


- Με το **justify-content** ορίζουμε τη στοίχιση των items στο “**main axis**”
- Οι πιο βασικές τιμές που μπορούμε να δώσουμε είναι:
 - **flex-start** : στην αρχή του axis
 - **flex-end**: στο τέλος του axis
 - **center** : στη μέση του axis
 - **space-between**: ο διαθέσιμος κενός χώρος μοιράζεται ανάμεσα στα items
 - **space-around**: ο διαθέσιμος κενός χώρος μοιράζεται ανάμεσα στα items αλλά και πριν το 1ο item και μετά το τελευταίο.
- Το default είναι το **flex-start**.



align-items:

Διάταξη των items στο cross axis



- Με το **align-items** ορίζουμε τη στοίχιση των items στο “**cross axis**”
 - Π.χ. Αν το **flex-direction** είναι **row**, δηλ. διάταξη κατα πλάτος, τότε το **align-items** καθορίζει διάταξη ως προς το ύψος.
- Οι πιο βασικές τιμές που μπορούμε να δώσουμε είναι:
 - **flex-start** : στην αρχή του axis
 - **flex-end**: στο τέλος του axis
 - **center** : στη μέση του axis
 - **stretch**: τα items “τεντώνονται” να καλύψουν όλο το χώρο
- Το default είναι το **stretch**.



Βασικές εντολές για τα items

- Η βασικότερη όλων είναι το **flex-grow**: Με αυτό καθορίζουμε αν το item μπορεί να “μεγαλώνει” και να καλύπτει τον διαθέσιμο χώρο και πόσο από το διαθέσιμο χώρο του ανατίθεται
- Η τιμή για το **flex-grow** είναι ένας ακέραιος, που δηλώνει την αναλογία του διαθέσιμου χώρου που θα καλύψει το συγκεκριμένο item. Παράδειγμα:
 - Αν όλα τα items έχουν **flex-grow: 1** τότε ο διαθέσιμος χώρος θα μοιραστεί ίσα σε όλα τα items
 - Αν ένα item έχει **flex-grow: 2** και όλα τα υπόλοιπα έχουν **flex-grow: 1**, τότε το συγκεκριμένο θα πάρει το διπλάσιο διαθέσιμο χώρο από τα υπόλοιπα
- Η default τιμή του είναι το 0 που σημαίνει ότι το item δεν “μεγαλώνει” / επεκτείνεται ακόμα και αν υπάρχει διαθέσιμος χώρος



Για περισσότερα...

- Παραδείγματα Flexbox από w3schools https://www.w3schools.com/Css/css3_flexbox.asp
- Παραδείγματα layouts με το Flexbox: <https://tobiasahlin.com/blog/common-flexbox-patterns/>
- Ένα παράξενο trick με το Flexbox: <https://css-tricks.com/the-peculiar-magic-of-flexbox-and-auto-margins/>