Το μοντέλο ευέλικτης διάταξης (Flexbox)

Το **μοντέλο ευέλικτης διάταξης** (**Flexbox**) είναι ένα μοντέλο πλαισίου CSS, βελτιστοποιημένο για το σχεδιασμό μιας εύχρηστης διεπαφής χρήστη. Παρέχει έναν αποτελεσματικό τρόπο διάταξης και στοίχισης των στοιχείων ενός περιέκτη, καθώς και της κατανομής του ελεύθερου χώρου ανάμεσα σε αυτά, ακόμη και στις περιπτώσεις που είναι άγνωστο το μέγεθος τους ή/και μεταβάλλεται δυναμικά (σε αυτό οφείλει και τον όρο ευέλικτο – Flex). Αυτά τα χαρακτηριστικά το καθιστούν ιδανικό για τη σχεδίαση ιστοσελίδων που θέλουμε να προσαρμόζονται κατάλληλα όταν προβάλλονται σε διάφορα μεγέθη οθονών (σε συσκευές από έξυπνα κινητά και τάμπλετ, έως λάπτοπ, υπολογιστές και τηλεοράσεις). Ένας εύκαμπτος (**Flex**) περιέκτης (container) μπορεί να αλλάξει τις διαστάσεις (πλάτος, ύψος) των παιδιών του (των στοιχείων δηλαδή που περιέχει), ακόμη και τη σειρά τους ώστε να γεμίσει τον διαθέσιμο χώρο που διατίθεται στην οθόνη προβολής της ιστοσελίδας. Όταν ο διαθέσιμος χώρος είναι υπερβολικός, τα μεγεθύνει, ενώ όταν υπάρχει έλλειψη τα συρρικνώνει για να αποφεύγει την υπερχείλιση. Ένα άλλο σημαντικό στοιχείο είναι ότι αδιαφορεί για την κατεύθυνση που διατάσσονται τα στοιχεία από τη φύση τους (κατακόρυφα τα μπλοκ στοιχεία και οριζόντια τα στοιχεία γραμμής). Στη συνέχεια θα γνωρίσουμε το μοντέλο ευέλικτης διάταξης (Flexbox), που κάνει ευκολότερο το σχεδιασμό της ιστοσελίδας μας ώστε να είναι προσαρμοστική σε διάφορα μεγέθη εικόνας, χωρίς να είναι απαραίτητη η χρήση float και positioning.

Το πρώτο πράγμα που πρέπει να κάνουμε είναι να κατασκευάσουμε έναν περιέκτη (container) και να τοποθετήσουμε μέσα σε αυτόν τα στοιχεία που θέλουμε να διατάξουμε. Ο περιέκτης γίνεται εύκαμπτος (flexible) αν χρησιμοποιήσουμε στο στυλ του την **ιδιότητα display με τιμή flex**. Τα στοιχεία που περιέχονται μέσα σε έναν εύκαμπτο περιέκτη γίνονται **εύκαμπτα στοιχεία** (**flexible items**).

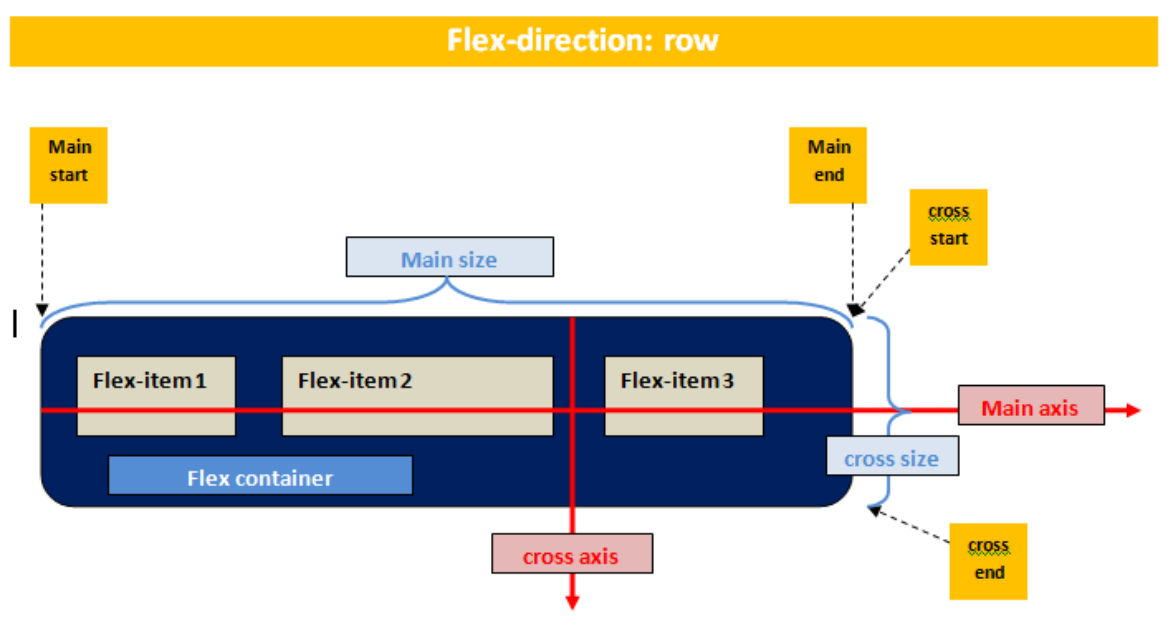
Πριν ξεκινήσουμε να αναλύουμε πως φτιάχνουμε ένα μοντέλο ευέλικτης διάταξης (Flexbox), θα διευκρινίσουμε ορισμένες έννοιες που αναφέρονται σε αυτό, όπως:

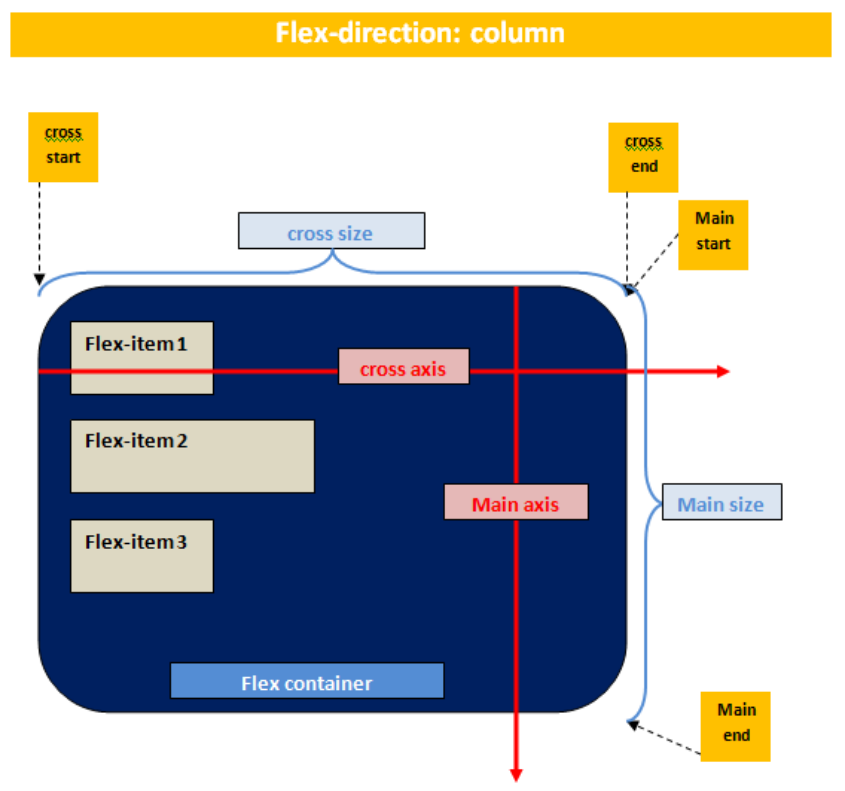
- **main axis** και **cross axis**
- **main size** και **cross size**
- **main start** και **main end**
- **cross start** και **cross end**

Το μοντέλο ευέλικτης διάταξης (Flexbox) χρησιμοποιεί μια σειρά από ιδιότητες (CSS properties). Ορισμένες από αυτές αναφέρονται στον περιέκτη (flex container) και ορισμένες στα παιδιά του (flex items). Βασικές έννοιες στο μοντέλο είναι αυτές, της **κατεύθυνσης** της ροής των στοιχείων του περιέκτη (flex-direction) και των **αξόνων**, του κύριου και του κάθετου σε αυτόν (**main axis** και **cross axis)**. Ποιος είναι κύριος και ποιος κάθετος άξονας εξαρτάται από την κατεύθυνση της ροής των στοιχείων (flex-direction). Ο κύριος άξονας (main axis) είναι ο πρωτεύων άξονας κατά μήκος του οποίου διατάσσονται τα flex items. Ταυτίζεται, κάθε φορά, με την κατεύθυνση της ροής των στοιχείων, ενώ ο κάθετος άξονας (cross axis) είναι κάθετος σε αυτή. Αν δηλαδή, η κατεύθυνση της ροής των στοιχείων είναι οριζόντια, ο main axis είναι οριζόντιος και ο cross axis είναι κατακόρυφος (τιμή που έχει η ιδιότητα flex-direction είναι: flex-direction: row;). Αν η κατεύθυνση της ροής των στοιχείων είναι κατακόρυφη, ο main axis είναι κατακόρυφος και ο cross axis είναι οριζόντιος (τιμή που έχει η ιδιότητα flex-direction είναι: flex-direction: column;). Τα flex items τοποθετούνται στον περιέκτη (container), ξεκινώντας από την αρχή του κυρίου άξονα (main start) και πηγαίνουν προς το τέλος του κυρίου άξονα (main end). Οι γραμμές που γεμίζουν με τα flex items, ξεκινούν από την αρχή του κάθετου άξονα

(cross start) και συνεχίζουν προς το τέλος του κάθετου άξονα (cross end). Όταν η κατεύθυνση της ροής των στοιχείων είναι οριζόντια (flex-direction: row;), το κύριο μέγεθος (main size) είναι το εύρος του περιέκτη (δηλαδή όσο έχει ορισθεί η ιδιότητα width στον περιέκτη) και το cross size είναι το ύψος του περιέκτη (δηλαδή όσο έχει ορισθεί η ιδιότητα height στον περιέκτη). Όταν η κατεύθυνση της ροής των στοιχείων είναι κατακόρυφη (flex-direction: column;), το κύριο μέγεθος (main size) είναι το ύψος του περιέκτη (δηλαδή όσο έχει ορισθεί η ιδιότητα height στον περιέκτη) και το cross size είναι το εύρος του περιέκτη (δηλαδή όσο έχει ορισθεί η ιδιότητα width στον περιέκτη).

Τα σχήματα που ακολουθούν δίνουν μια εικόνα όσων αναλύσαμε παραπάνω:





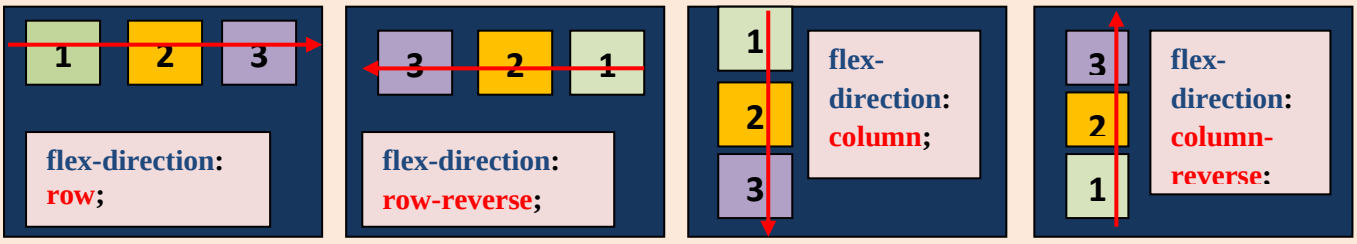
Οι βασικές ιδιότητες ενός εύκαμπτου περιέκτη (flex container), οι οποίες καθορίζουν τη διάταξη των στοιχείων που περιέχει, είναι οι ακόλουθες:

- display
- flex-direction
- flex-wrap
- justify-content
- align-items
- align-content

Η ιδιότητα display, όταν λάβει την τιμή flex (**display: flex;**) δημιουργεί ένα εύκαμπτο περιβάλλον για όλα τα παιδιά του περιέκτη.

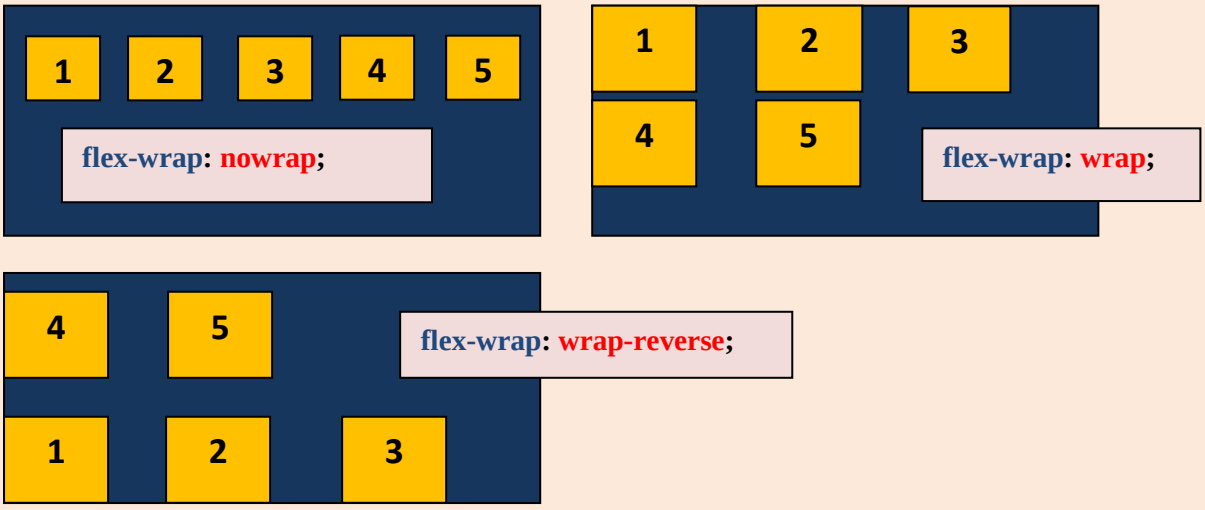
Η ιδιότητα flex-direction καθορίζει την **κατεύθυνση της ροής των στοιχείων** που περιέχει ο περιέκτης. Επίσης, προσδιορίζει ποιος θα είναι ο κύριος άξονας (main axis) και κατά συνέπεια και τον κάθετο άξονα (cross axis). Οι τιμές που λαμβάνει είναι οι ακόλουθες:

Τιμή	Περιγραφή
row	Τα στοιχεία στοιχίζονται σε σειρά με κατεύθυνση από αριστερά προς τα δεξιά (left to right). **Προκαθορισμένη τιμή.**
row-reverse	Τα στοιχεία στοιχίζονται σε σειρά με κατεύθυνση από δεξιά προς τα αριστερά (right to left).
column	Τα στοιχεία στοιχίζονται σε στήλη με κατεύθυνση από πάνω προς τα κάτω (top to bottom).
column-reverse	Τα στοιχεία στοιχίζονται σε στήλη με κατεύθυνση από κάτω προς τα πάνω (bottom to top).



Η ιδιότητα flex-wrap καθορίζει αν θα γίνεται **αναδίπλωση των στοιχείων** που περιέχει ο περιέκτης και με ποιο τρόπο. Οι τιμές που λαμβάνει είναι οι ακόλουθες:

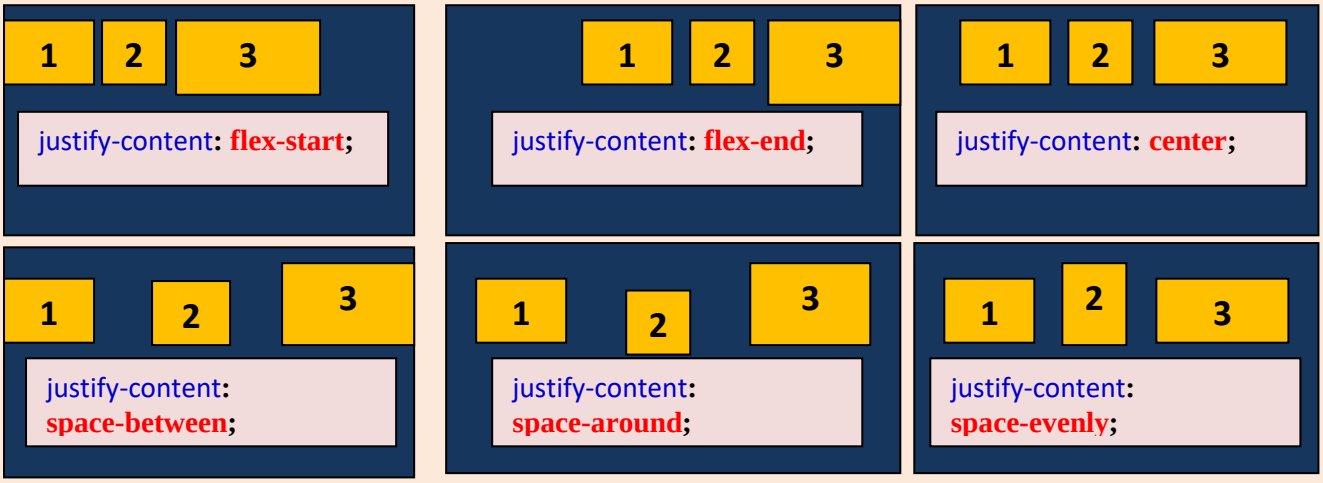
Τιμή	Περιγραφή
nowrap	Τα στοιχεία δεν αναδιπλώνονται αλλά προσπαθούν να μείνουν σε μια γραμμή, αλλάζουν οι διαστάσεις τους ώστε να καλύπτουν το διαθέσιμο εύρος του παραθύρου. **Προκαθορισμένη τιμή**.
wrap	Τα στοιχεία αναδιπλώνονται, σε περίπτωση που δεν χωράνε στο διαθέσιμο εύρος του παραθύρου και συνεχίζουν στην από κάτω σειρά.
wrap -reverse	Τα στοιχεία αναδιπλώνονται και συνεχίζουν στην από πάνω σειρά (τα τελευταία στοιχεία που περισσεύουν πάνε από πάνω και τα πρώτα που χωράνε προωθούνται στην από κάτω σειρά).



Υπάρχει και μια ιδιότητα σύντμησης των δύο παραπάνω, η flex-flow (π.χ. flex-flow: row wrap; flex-flow: column wrap;).

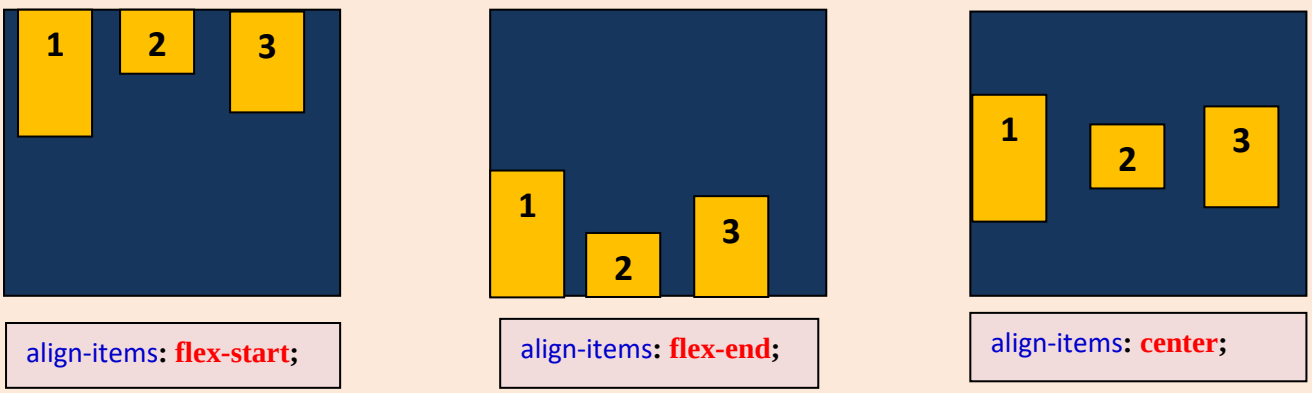
Η **στοίχιση κατά μήκος του κύριου άξονα**, καθορίζεται από την ιδιότητα justify-content. Προσδιορίζει την κατανομή του ελεύθερου χώρου ανάμεσα στα flex items.

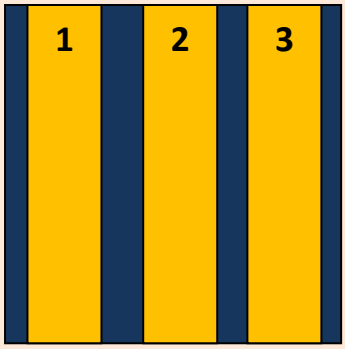
Τιμή	Περιγραφή
flex-start	Τα στοιχεία στοιχίζονται στην αρχή του main axis (main start). **Προκαθορισμένη τιμή.**
flex-end	Τα στοιχεία στοιχίζονται στο τέλος του main axis (main end).
center	Τα στοιχεία στοιχίζονται στο κέντρο του main axis.
space-between	Τα στοιχεία στοιχίζονται εξίσου στον main axis, με το πρώτο στοιχείο στο main start και το τελευταίο στο main end. Το διαθέσιμο κενό μοιράζεται εξίσου ανάμεσα στα στοιχεία.
space-around	Τα στοιχεία στοιχίζονται στον main axis με το διαθέσιμο κενό να μοιράζεται εξίσου ανάμεσα στα στοιχεία. Το κενό ανάμεσα σε δύο στοιχεία είναι διπλάσιο από το κενό μεταξύ του πρώτου ή του τελευταίου στοιχείου με τις πλευρές του περιέκτη (πλευρικά κενά), γιατί περιέχει δύο κενά, ένα από το αριστερό και ένα από το δεξιό στοιχείο.
space-evenly	Η διαφορά με την παραπάνω τιμή είναι ότι όλα τα κενά και τα δύο πλευρικά κενά είναι ίσα με τα κενά μεταξύ των στοιχείων.



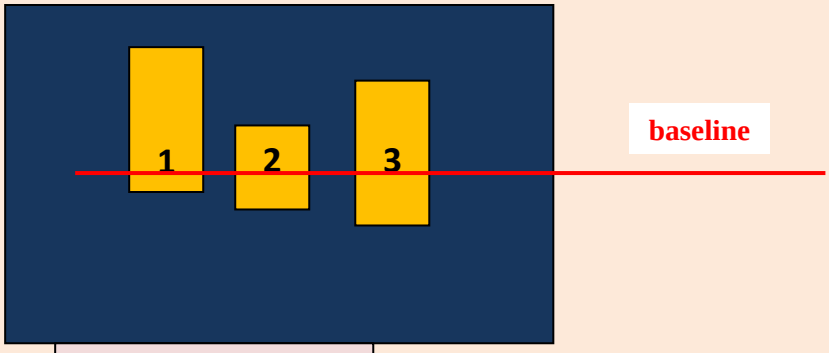
Η **στοίχιση κατά μήκος του κατακόρυφου άξονα** (cross axis), καθορίζεται από την ιδιότητα align-items.

Τιμή	Περιγραφή
flex-start	Τα στοιχεία στοιχίζονται στην αρχή του cross axis (cross start).
flex-end	Τα στοιχεία στοιχίζονται στο τέλος του cross axis (cross end).
center	Τα στοιχεία στοιχίζονται στο κέντρο του cross axis.
stretch	Τα στοιχεία εκτείνονται (τεντώνουν) για να καλύψουν όλο τον διαθέσιμο χώρο στην κατεύθυνση του cross axis. **Προκαθορισμένη τιμή.**
baseline	Τα στοιχεία στοιχίζονται όπως στοιχίζεται η baseline στο περιεχόμενο τους.



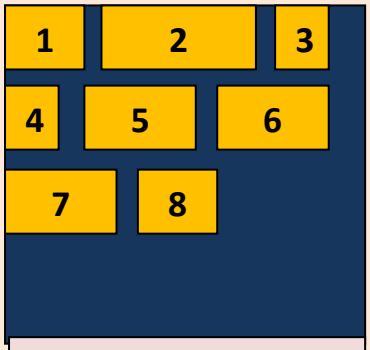


align-items: **stretch**;

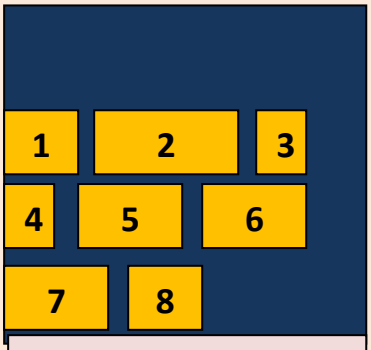


align-items: **baseline**;

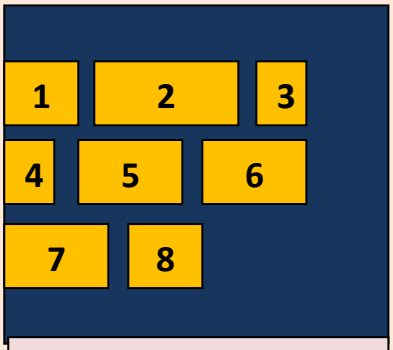
Η ιδιότητα align-content προσδιορίζει τη **στοίχιση μεταξύ των σειρών** που δημιουργούνται μέσα στον περιέκτη **κατά τον cross axis**. Λειτουργεί με το ίδιο τρόπο που λειτουργεί η ιδιότητα **justify-content** για τη στοίχιση των στοιχείων κατά τον main axis.



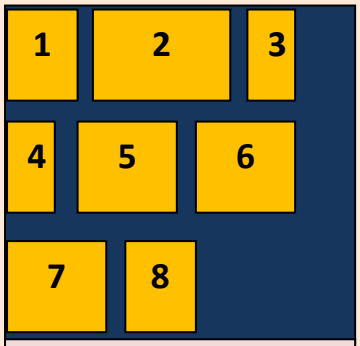
align-content: **flex-start**;



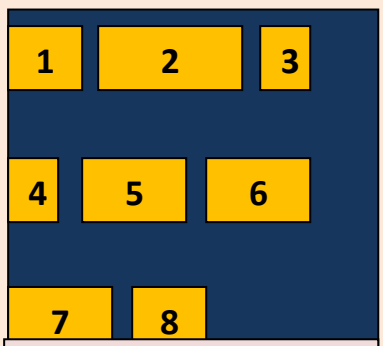
align-content: **flex-end**;



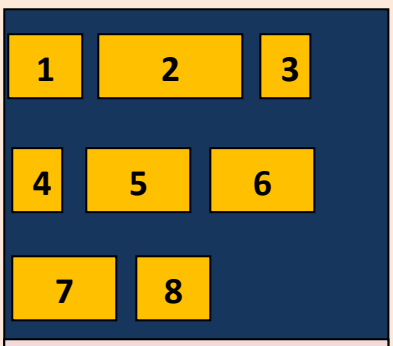
align-items: **center**;



align-items: **stretch**;

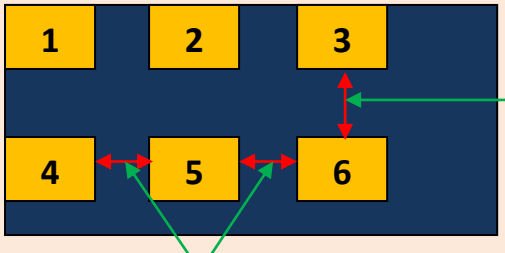


align-items:**space-between**;



align-items: **space-around**;

Η ιδιότητα gap (row-gap, column-gap) καθορίζει την **απόσταση** (το μέγεθος του κενού) **μεταξύ των στοιχείων**. Δεν εφαρμόζεται για την απόσταση με τις εξωτερικές πλευρές του πλαισίου του περιέκτη (outer edges).



column -gap: **20px**;

row-gap: **10px**;

Σύντμηση gap: **10px 20px**;

Εκτός από τις παραπάνω ιδιότητες, που αναφέρονται σε έναν εύκαμπτο περιέκτη, υπάρχουν και οι παρακάτω **ιδιότητες που αναφέρονται στα εύκαμπτα στοιχεία (flex items)** που περιέχει.

Η ιδιότητα align-self καθορίζει τη **στοίχιση ενός εύκαμπτου στοιχείου, επικαλύπτοντας την τιμή που έχει η ιδιότητα align-items του περιέκτη**. Λαμβάνει τις ίδιες δυνατές τιμές με την ιδιότητα align-items.

Η ιδιότητα order καθορίζει **τη σειρά ενός εύκαμπτου στοιχείου εντός του περιέκτη**. Προκαθορισμένη τιμή είναι το 0. Τα στοιχεία διατάσσονται με τη σειρά που τα έχουμε γράψει στο αρχείο και έχουν τιμή order ίση με 0. Αν δώσουμε σε κάποιο στοιχείο μια άλλη τιμή, εκτός του 0, τα στοιχεία θα τοποθετηθούν με βάση την τιμή που έχουν στην ιδιότητα order (σε αυτά που δεν έχει καθορισθεί τιμή θεωρείται το 0). Προηγείται το στοιχείο που έχει την μικρότερη τιμή στην ιδιότητα order και αυτά που έχουν την ίδια τιμή θα τοποθετηθούν με τη σειρά που έχουν γραφτεί στο αρχείο μας.

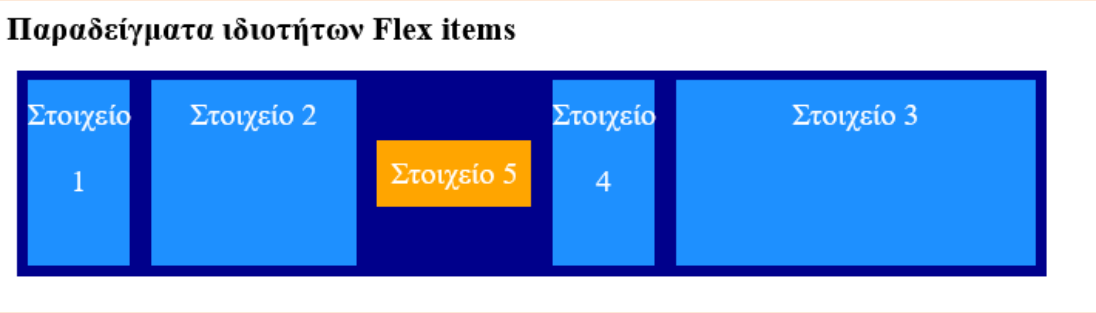
Η ιδιότητα flex-grow καθορίζει **πόσο θα αναπτυχθεί ένα εύκαμπτο στοιχείο σχετικά με τα υπόλοιπα στοιχεία** του εύκαμπτου περιέκτη. Προκαθορισμένη τιμή είναι το 0, που σημαίνει ότι δεν θα καταλάβει μεγαλύτερο χώρο όταν αυξηθεί ο διαθέσιμος χώρος στον περιέκτη. Τα στοιχεία μοιράζονται τον διαθέσιμο χώρο αναλογικά με τις τιμές που έχουν στην ιδιότητα flex-grow και παραμένουν σταθερά αν έχουν τιμή 0. Αν έχουν τιμή flex-grow: 1; Ο διαθέσιμος χώρος θα μοιρασθεί ισοδύναμα σε όλα τα flex items. Αν κάποιο στοιχείο έχει τιμή 2, θα λάβει διπλάσιο χώρο από τα υπόλοιπα στοιχεία που έχουν τιμή 1 κ.ο.κ.

Η ιδιότητα flex-shrink καθορίζει **πόσο θα συρρικνωθεί ένα εύκαμπτο στοιχείο σχετικά με τα υπόλοιπα στοιχεία** του εύκαμπτου περιέκτη. Προκαθορισμένη τιμή είναι το 1.

Η ιδιότητα flex-basis καθορίζει την **αρχική διάσταση ενός εύκαμπτου στοιχείου πριν κατανεμηθεί ο υπολειπόμενος διαθέσιμος χώρος**. Λαμβάνει τιμές μέτρησης μήκους (π.χ. 20%, 3rem) ή λέξεις κλειδιά, όπως: **auto** (σύμφωνα με την τιμή που έχει η ιδιότητα flex-grow) και **content** (σύμφωνα με το περιεχόμενο του στοιχείου). Αν λάβει την τιμή 0, ο επιπλέον χώρος γύρω από το περιεχόμενο δεν λαμβάνεται υπόψη.

Τέλος,, η ιδιότητα flex αποτελεί σύντμηση των ιδιοτήτων flex-grow, flex-shrink, και flex-basis.

Στο παράδειγμα που ακολουθεί, φτιάχνουμε ένα στοιχείο div ως περιέκτη και εσωκλείουμε σε αυτό πέντε στοιχεία παραγράφου με διαφορετικό αναγνωριστικό σε κάθε ένα από αυτά.



Το αρχείο που ακολουθεί περιέχει την κατασκευή και το στυλ των στοιχείων αυτών. Προσδιορίζοντας τη ιδιότητα display: flex; τα στοιχεία μας παρατίθενται σε μια σειρά και όχι σε νέα σειρά το καθένα (παρότι είναι στοιχεία παραγράφου, δηλαδή στοιχεία μπλοκ). Επίσης, παρατηρούμε ότι τα στοιχεία δεν εμφανίζονται με τη σειρά που έχουν γραφεί στο αρχείο γιατί έχει καθορισθεί τιμή στην ιδιότητα order. Πρώτα θα εμφανισθούν τα στοιχεία 1, 2 και 5 που έχουν ως τιμή το 0 (στα στοιχεία 2 και 5 δεν έχει ορισθεί τιμή στην ιδιότητα order, άρα θα λάβουν την προκαθορισμένη τιμή 0). Μεταξύ τους ισχύει η σειρά που έχουν γραφεί στο αρχείο μας. Ακολουθεί το στοιχείο 4 που έχει τιμή 2 και τελευταίο το στοιχείο 3 που έχει τη μεγαλύτερη τιμή, το 3.

Το στοιχείο 2 έχει ως αρχική διάσταση 200 px γιατί έχει ορισθεί τιμή στην ιδιότητα flex-basis (#item2 {flex-basis: 200px;}). Το στοιχείο 3 αναπτύσσεται 3πλάσια από τα υπόλοιπα γιατί έχει ορισθεί τιμή στην ιδιότητα flex-grow το 3 (#item3 {order: 3; flex-grow: 3;}). Το στοιχείο 5 έχει διαφορετική στοίχιση γιατί έχει ορισθεί τιμή στην ιδιότητα align-self με τιμή center. Επίσης, έχει άλλο χρώμα στο υπόβαθρο και επειδή έχει flex: 0 0 150px (ισοδυναμεί με flex-grow:0; flex-shrink:0; flex-basis: 150px;) δεν θα αυξηθεί ούτε θα συρρικνωθεί περαιτέρω και η αρχική του διάσταση θα είναι 150px (#item5 {background-color: orange; align-self: center; flex: 0 0 150px}).

Ακολουθεί ολόκληρο το αρχείο για τη δραστηριότητα μας.

```
<!DOCTYPE html>
<html>
<head>
<style>
 .container {
  background-color: darkblue;
  color: white;
  width: 1000px;
  height: 200px;
  margin: 10px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
  display: flex;
}
p {
  background-color: DodgerBlue;
  color: white;
  width: 100px;
  margin: 10px;
  text-align: center;
  line-height: 65px;
  font-size: 30px;
}
#item1 {order: 0;}
#item2 {flex-basis: 200px;}
#item3 {order: 3; flex-grow: 3;}
#item4 {order: 2; flex-shrink: 0;}
#item5 {background-color: orange; align-self: center; flex: 0 0 150px}
</style>
</head>
<body>
<h1>Παραδείγματα ιδιοτήτων Flex items</h1>
<div class="container">
  <p id="item1">Στοιχείο 1</p>
  <p id="item2">Στοιχείο 2</p>
  <p id="item3">Στοιχείο 3</p>
  <p id="item4">Στοιχείο 4</p>
  <p id="item5">Στοιχείο 5</p>
</div>
</body>
</html>
```