



#### ΠΕΡΙΕΧΟΜΕΝΑ:

1. Κενά μεταξύ tracks
2. Στοίχιση
  1. Οριζόντια στοίχιση
  2. Κάθετη στοίχιση
  3. Στοίχιση item
3. order και z-index
4. Ασκήσεις

**row-gap:** Κενό μεταξύ γραμμών

τιμές	Μήκος		
default	0		
Στοιχεία	grid container	Κληρονόμηση	Όχι

- Αντίστοιχα ορίζονται:
  - column-gap:** το οποίο ορίζει το κενό μεταξύ στηλών.
  - gap:** σύντμηση των δύο (σύνταξη gap row-gap column-gap)

### Παρατήρηση:

- Σε παλιότερους browsers τα properties γράφονται ως **grid-row-gap**, **grid-column-gap**, **grid-gap** (έγινε μετονομασία τους σε νεότερη έκδοση της CSS)

### Παράδειγμα 1: gap

```
.container {
  display: grid;
  grid-template-rows: repeat(3,100px);
  grid-template-columns: repeat(3,100px);
}
div div {
  border: 1px black solid;
}
#container1 {
  row-gap: 2px;
  column-gap: 10px;
}
#container2 {
  gap: 10px;
}
```

row-gap, column-gap

1	2	3
4	5	6
7	8	9

### Σημείωση:

- Δεν είναι δυνατόν να «στοχεύσουμε» κάποιο συγκεκριμένο κενό (π.χ. αυτό μεταξύ 2<sup>ης</sup> και 3<sup>ης</sup> γραμμής)
- Οπότε σαν workaround, μπορούμε να στοχεύσουμε τα στοιχεία των κελιών που μας ενδιαφέρουν και να θέσουμε σε αυτά θετικά ή αρνητικά margins.

### Παράδειγμα 2: gap-part-columns

```
.container {
  display: grid;
  grid-template-rows: repeat(4,100px);
  grid-template-columns: repeat(4,100px);
  gap: 30px;
}
div div {
  border: 1px black solid;
}
#container1 :nth-child(4n+2){
  margin-left: 40px;
}
#container2 :nth-child(n+5) {
  margin-bottom: -20px;
}
#container2 :nth-child(n+9) {
  margin-bottom: 0;
}
```

2nd column

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

2nd row

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

**justify-items:** Οριζόντια στοίχιση των items στο κελί τους

τιμές	start, end, center, stretch, normal, κ.α.		
default	normal		
Στοιχεία	grid container	Κληρονóμηση	Όχι

- normal: Αντίστοιχο με stretch (με κάποιες εκκεντρικές εξαιρέσεις)
- start, end, center:
  - Δεν γίνεται stretch, αλλά γίνεται fit στο περιεχόμενο και έπειτα αντίστοιχη τοποθέτηση στην αρχή, κέντρο, τέλος

**justify-content:** Οριζόντια στοίχιση του grid στον container

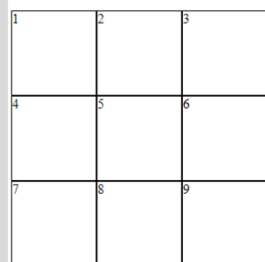
τιμές	start, end, center, stretch, space-around, space-between, space-evenly		
default	normal		
Στοιχεία	grid container	Κληρονóμηση	Όχι

- (Εφόσον ο container έχει width που είναι μεγαλύτερο από το πλάτος του grid)
- start, end, center: Στοιχίση του grid στον container στην αρχή, τέλος, κέντρο αντίστοιχα.
- Υπόλοιπες ιδιότητες: Βλέπε σχήμα (κάτω)

### Παράδειγμα 3: justify-items

```
.container {
  display: grid;
  grid-template-rows:
    repeat(3,100px);
  grid-template-columns:
    repeat(3,100px);
}
div div {
  border: 1px black solid;
}
#container1 {
  justify-items: stretch;
}
...
```

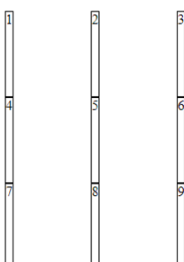
justify-items: stretch



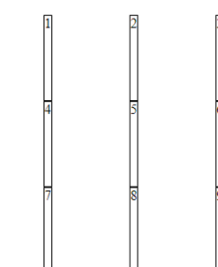
justify-items: left



justify-items: center



justify-items: right



### Παράδειγμα 4: justify-content

```
.container {
  display: grid;
  grid-template-rows:
    repeat(3,100px);
  grid-template-columns:
    repeat(3,100px);
  width: 400px;
  height: 400px;
  border: 2px darkgray solid;
}
div div {
  border: 1px black solid;
  background-color: lightgrey;
}
#container1 {
  justify-content: start;
}
...
```

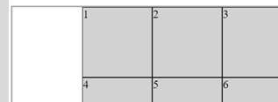
justify-content: start



justify-content: center



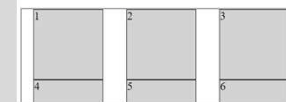
justify-content: end



justify-content: normal



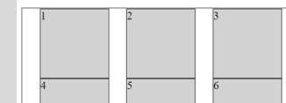
justify-content: space-around



justify-content: space-between



justify-content: space-evenly



**align-items:** Κάθετη στοίχιση των items στο κελί τους

τιμές	start, end, center, stretch, normal, κ.α.		
default	normal		
Στοιχεία	grid container	Κληρονόμηση	Όχι

**align-content:** Κάθετη στοίχιση του grid στον container

τιμές	start, end, center, stretch, space-around, space-between, space-evenly		
default	normal		
Στοιχεία	grid container	Κληρονόμηση	Όχι

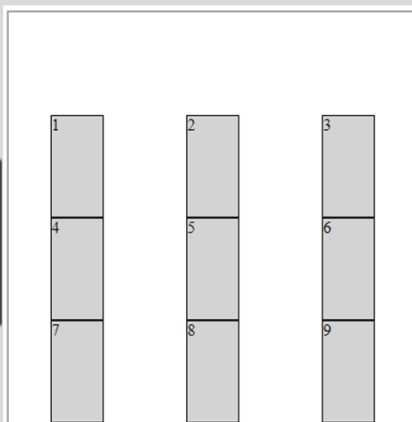
#### Παρατήρηση:

- Οι τιμές είναι αντίστοιχες στη λειτουργία με αυτές που είδαμε στα properties: align-items και align-content

#### Παράδειγμα 5: align-items

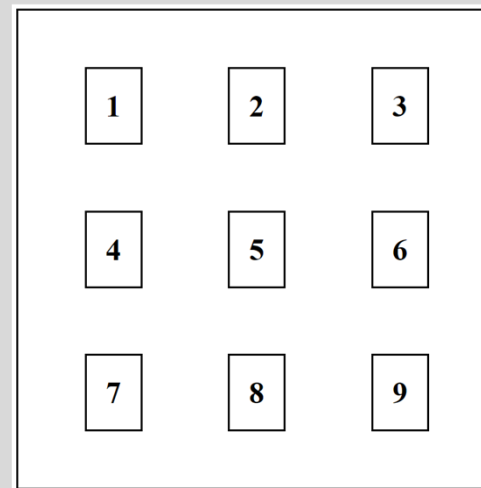
```
.container {
  display: grid;
  grid-template-rows: repeat(3,100px);
  grid-template-columns: repeat(3,100px);
  width: 400px;
  height: 400px;
  border: 2px darkgray solid;
  justify-items: center;
  justify-content: space-around;
  align-items: normal;
  align-content: end;
}
```

```
div div {
  ...
  width: 50px;
}
```



#### Άσκηση 1:

Ξεκινώντας από το αρχείο exercise01-initial.html και προσθέτοντας κανόνες μόνο στον container, προσαρμόστε την εμφάνιση ώστε να φαίνεται όπως η ακόλουθη:



Υπενθύμιση:

- Οι ακόλουθες ιδιότητες του container στοιχίζουν όλα τα στοιχεία στα κελιά τους:
  - justify-items**: Οριζόντια στοίχιση items
  - align-items**: Κάθετη στοίχιση items
- Μπορούμε να υπερβούμε αυτές τις τιμές για κάποιο συγκεκριμένο grid item με τα ακόλουθα properties:

justify-self: Οριζόντια στοίχιση item			
τιμές	start, end, center, stretch, normal, κ.α.		
default	normal		
Στοιχεία	grid item	Κληρονόμηση	Όχι

align-self: Κάθετη στοίχιση item			
τιμές	start, end, center, stretch, normal, κ.α.		
default	normal		
Στοιχεία	grid container	Κληρονόμηση	Όχι

Παράδειγμα 6: self

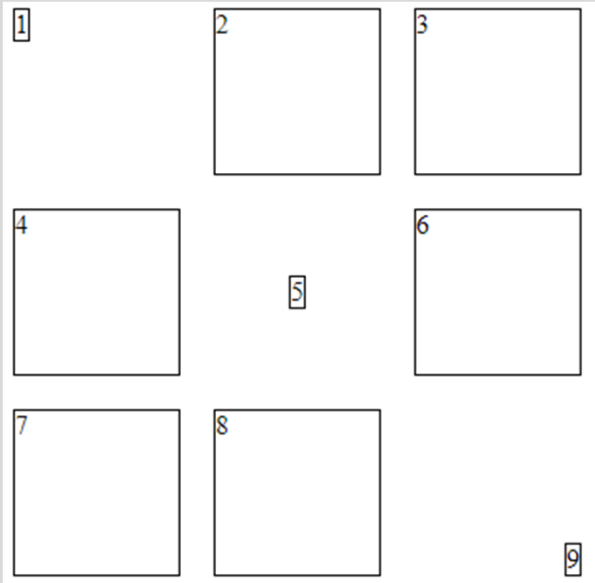
```
...
<div class="item">5</div>
...

.item {
  justify-self: center;
  align-self: center;
}
```

1	2	3
4	5	6
7	8	9

Άσκηση 2:

Ξεκινώντας από το αρχείο exercise02-initial.html τροποποιήστε την εμφάνιση ώστε να ομοιάζει με την ακόλουθη:



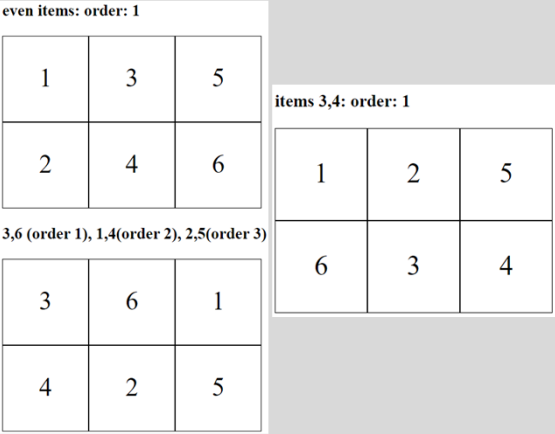
order: Σειρά Εμφάνισης grid items

τιμές	Αριθμός		
default	0		
Στοιχεία	grid items	Κληρονόμηση	Όχι

- (Ίδια συμπεριφορά με flex) Ο αριθμός καθορίζει τη σειρά εμφάνισης των items
  - Αν δύο η περισσότερα items έχουν τον ίδιο αριθμό, τότε η σειρά τους θα είναι ίδια με την html

Παράδειγμα 7: order

```
div#container1 div:nth-child(even) {
  order: 1;
}
div#container2 div:nth-child(3n) {
  order: 1;
}
div#container2 div:nth-child(3n+1) {
  order: 2;
}
div#container2 div:nth-child(3n+2) {
  order: 3;
}
div#container3 div:nth-child(4),
div#container3 div:nth-child(3){
  order: 1;
}
```



z-index: Όσο πιο μεγάλο, τόσο «πιο πάνω»

τιμές	auto, number		
default	auto		
Στοιχεία	grid items	Κληρονόμηση	Όχι

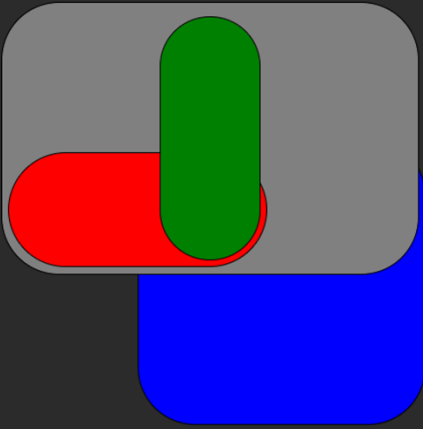
- Είναι δυνατόν να επικαλύπτονται items (εφσον τα βάλουμε σε επικαλυπτόμενα κελιά)
- Καθορίζουμε πιο θα είναι πιο πάνω (όπως στο positioning) με την:

Παράδειγμα 8: z-index

```
<div class="container">
  <div id="item1"></div>
  <div id="item2"></div>
  <div id="item3"></div>
  <div id="item4"></div>
</div>
```

```
.container {
  display: grid;
  grid-template-rows:
    repeat(3, 100px);
  grid-template-columns:
    repeat(3, 100px);
}
```

```
#item1 {
  grid-area: 2/2/4/4;
  background-color: blue;
}
#item2 {
  grid-area: 2/1/3/3;
  background-color: red;
  z-index: 1;
  margin: 10px;
}
#item3 {
  grid-area: 1/2/3/3;
  background-color: green;
  z-index: 2;
  margin: 15px;
}
#item4 {
  grid-area: 1/1/3/4;
  background-color: gray;
  margin: 5px;
}
```



Άσκηση 3 (μέρος της σελίδας του youtube.com)

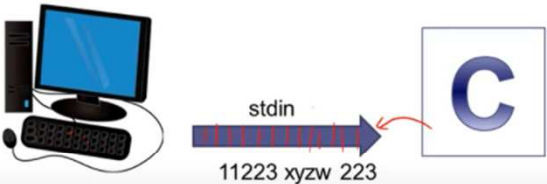
Δημήτρης Ψούνης, Η Γλώσσα C, Μάθημα 15: Συναρτήσεις Εισόδου

www.psounis.gr

## A. Ροές

### 2. Η ροή προκαθορισμένης εισόδου stdin

- Καλό είναι να σκεφτόμαστε την stdin σαν μια «σωλήνα» η οποία συνδέει το πληκτρολόγιο με το πρόγραμμά μας



11223 xyzw 223

- Π.χ. Η συνάρτηση scanf που μάθαμε, διαβάζει τόσους χαρακτήρες από την stdin, όσοι καθορίζονται από τον προσδιοριστή που έχουμε θέσει σαν 1ο όρισμα. Αν ο χρήστης έχει εισάγει περισσότερους χαρακτήρες τότε αυτοί παραμένουν στην stdin για επόμενο διάβασμα από κάποια εντολή διαχείρισης της προκαθορισμένης εισόδου.
- Θα δούμε και άλλες συναρτήσεις που διαχειρίζονται την προκαθορισμένη είσοδο.

4:27 / 6:56

Η ΓΛΩΣΣΑ C - ΜΑΘΗΜΑ 15 - ΣΥΝΑΡΤΗΣΕΙΣ ΕΙΣΟΔΟΥ - ΘΕΩΡΙΑ 1 από 5

5.693 προβολές · 26 Φεβ 2016

56 ΔΕΝ ΜΟΥ ΑΡΕΣΕΙ ΚΟΙΝΟΠΟΙΗΣΗ ΑΠΟΘΗΚΕΥΣΗ ...

ΜΑΘΗΜΑ 1.2 - ΤΥΦΛΗ ΑΝΑΖΗΤΗΣΗ - ΘΕΩΡΙΑ 3 από 3

Δημήτρης Ψούνης  
3,7 χιλ. προβολές · πριν από 6 έτη

19:49

PROLOG

ΜΑΘΗΜΑ 2 ΤΡΟΠΟΣ ΕΚΤΕΛΕΣΗΣ ΕΝΟΣ ΠΡΟΓΡΑΜΜΑΤΟΣ

17:02

Μίξη - Δημήτρης Ψούνης

YouTube

ΜΑΘΗΜΑ 15 ΣΥΝΑΡΤΗΣΕΙΣ ΕΙΣΟΔΟΥ

((.))

ΜΑΘΗΜΑ 2.2 ΚΑΤΗΓΟΡΗΜΑΤΙΚΗ ΛΟΓΙΚΗ...

Δημήτρης Ψούνης  
1,8 χιλ. προβολές · πριν από 6 έτη

12:48

/\* Grapsimo

ΜΑΘΗΜΑ 20

ΘΕΩΡΙΑ 4/8

fwrite(data, sizeof

13:24

fclose(fp);

for (i=0; i<100; i++)

{

puts(s[i]);

Η ΓΛΩΣΣΑ C - ΜΑΘΗΜΑ 20 - ΑΡΧΕΙΑ - Θεωρία 4 από 8 - Δυναμικά Αρχεία

Δημήτρης Ψούνης  
2,9 χιλ. προβολές · πριν από 2 έτη

Η ΓΛΩΣΣΑ C - ΜΑΘΗΜΑ 16 - ΣΥΝΑΡΤΗΣΕΙΣ ΕΞΟΔΟΥ - Θεωρία ...

Δημήτρης Ψούνης  
2.4 νιλ. προβολές · πριν από 6 έτη

Ξεκινώντας από το exercise03-initial.html και χρησιμοποιώντας grid για το βασικό layout κατασκευάστε αντίστοιχη εμφάνιση με αυτή που φαίνεται στην οθόνη (αντιγραφή του layout του youtube.com).

- Μόνος περιορισμός να μην πειράξετε την HTML και τον κανόνα για τα min-width, max-width του container. Σε όλα τα υπόλοιπα χρησιμοποιήστε όποιο εργαλείο κρίνετε σκόπιμο.