



Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης



code4sp  
coding for social promotion



CITIZENS  
IN POWER



Center for Social  
Innovation

CD  
DR

ZAUG  
gGmbH





# ΠΛΕΓΜΑ CSS (GRID)



Με συγχρηματοδότηση από το  
πρόγραμμα «Erasmus+»  
της Ευρωπαϊκής Ένωσης

# ΤΙ ΕΙΝΑΙ ΤΟ ΠΛΕΓΜΑ CSS;

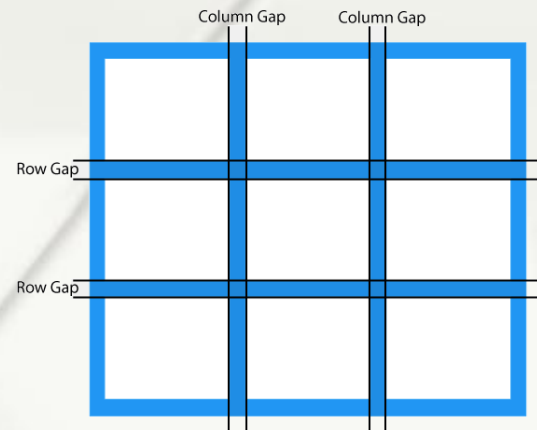
- Η ιδιότητα διάταξης πλέγματος CSS (**grid layout**) ή απλά **ιδιότητα πλέγματος** παρέχει τη δυνατότητα δημιουργίας πλέγματος για το σχεδιασμό ιστοσελίδων άμεσης ανταπόκρισης με πιο εύκολο και αποδοτικό τρόπο μεταξύ προγραμμάτων περιήγησης. Υπάρχουν άλλοι μέθοδοι για τον έλεγχο του πλέγματος μιας ιστοσελίδας, όπως οι πίνακες (tables), το μοντέλου κουτιού (box model) και το κουτί flex (flex box) της CSS.
- Στην αρχή θα πρέπει να ορίσετε ένα στοιχείο δοχείου (container element) όπως δηλαδή ένα **πλέγμα** με προβολή **display: grid**, να ορίσετε το μέγεθος των στηλών και των γραμμών με τις ιδιότητες **grid-template-columns** και **grid-template-rows** αντίστοιχα, και στη συνέχεια να τοποθετήσετε τα θυγατρικά στοιχεία (child elements) μέσα στο πλέγμα με **στήλες πλέγματος** και **γραμμές πλέγματος**.





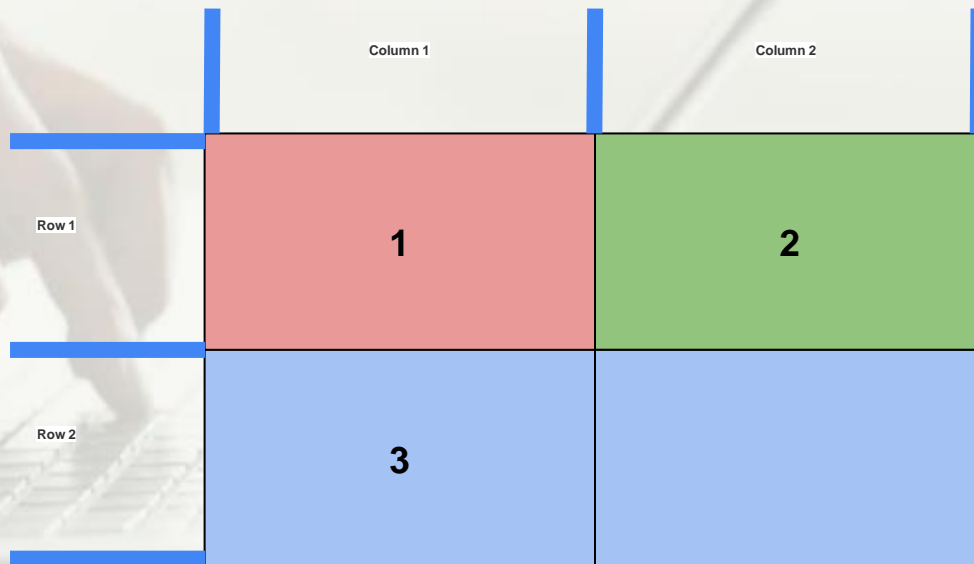
# ΤΙ ΕΙΝΑΙ ΕΝΑ ΠΛΕΓΜΑ **CSS** **(GRID);**

- Για ένα πλέγμα CSS Grid που βασίζεται σε μια προοπτική γραμμών και στηλών, θα πρέπει να χρησιμοποιήσουμε κάποια συγκεκριμένα βήματα.
- Μπορούμε να χρησιμοποιήσουμε το παράδειγμα της επόμενης επιφάνειας για να κατασκευάσουμε ένα απλό πλέγμα, και να ανακαλύψουμε τους διαφορετικούς τρόπους με τους οποίους μπορούμε να το κάνουμε.



# Προοπτική Γραμμών και Στηλών

- Όπως μπορείτε να παρατηρήσετε στο παράδειγμα εδώ, έχουμε ένα σχεδιασμό στη βάση **γραμμών** και **στηλών**. Στο σχεδιασμό αυτό έχουμε δύο στήλες και δυο γραμμές, τις οποίες μπορούμε να δημιουργήσουμε όταν προσθέτουμε το κουτί 1 & 2 στον ίδιο τελεστή div σε μια συνάρτηση. Στη συνέχεια ορίζουμε τις απόλυτες μέγιστες ή ελάχιστες τιμές πλάτους των κουτιών που θέλουμε, ενώ γράφουμε το κουτί 3 έξω από τον προηγούμενο τελεστή div, ο οποίος θα πρέπει να γραφτεί με τις ίδιες εντολές της γλώσσας CSS. Εάν χρησιμοποιήσετε την ιδιότητα **CSS grid**, θα είναι ακόμα πιο εύκολο.

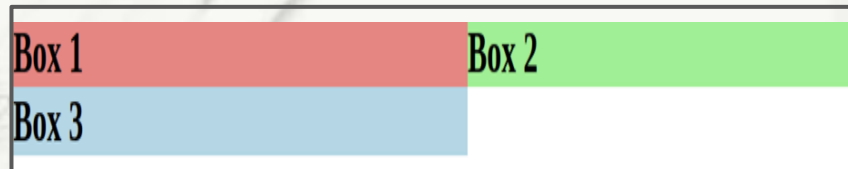




# Προοπτική Γραμμών και Στηλών

- Εάν χρησιμοποιήσετε τις ακόλουθες εντολές σε ένα αρχείο CSS (**file**) θα έχετε το αποτέλεσμα που βλέπετε στα δεξιά. Using the following commands on the **CSS file** will give the result on the left.  
Με την ιδιότητα **grid-template-columns** μπορείτε να ορίσετε τις τιμές ανάλογα με το πόσο μεγάλες θέλετε να είναι οι στήλες. Στην περίπτωση πιο κάτω θα πρέπει να περάσετε δύο τιμές για να δημιουργήσετε δύο στήλες με πλάτος **500px**, έστω και αν υπάρχει επιπλέον χώρος στα αριστερά. **can takes more than value based on how mush columns you want to have, in this case i just pass on it two values and you can see that it make just two columns with 500px of width even if there's an extra space at the left.**

```
.container {  
  display: grid;  
  grid-template-columns: 500px 500px;  
}
```



Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης



# Προοπτική Γραμμών και Στηλών

- Με την εντολή **grid-template -rows**, μπορούμε να αυξήσουμε το ύψος των γραμμών όπως φαίνεται στο πιο κάτω παράδειγμα. **Also their is grid-template -rows, and us you can see below it makes the heights more bigger and this is because I had given 300px to each row.**

```
.container {  
  display: grid;  
  grid-template-columns: 500px 500px;  
  grid-template-rows: 300px 300px;  
}
```



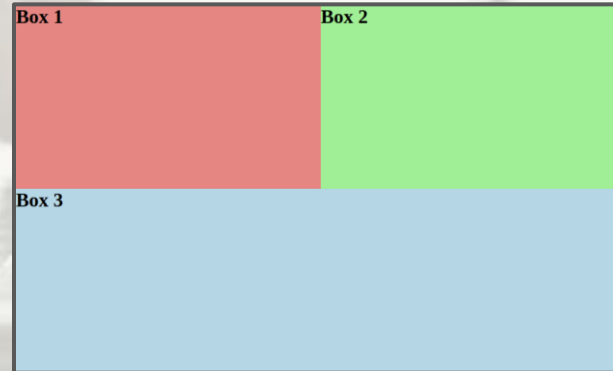
Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης



# Προοπτική Γραμμών και Στηλών

- Στο στάδιο αυτό, μπορούμε να εκτείνουμε το **κουτί 3** πέραν του κενού χώρου που βρίσκεται κάτω από το **κουτί 2**, χρησιμοποιώντας την εντολή **grid-column-start / end**, η οποία μας επιτρέπει να επιλέξουμε το αρχικό και τελικό σημείο του χώρου που θέλουμε να καταλαμβάνει το **κουτί 3**.

```
.container {  
  display: grid;  
  grid-template-columns: 500px 500px;  
  grid-template-rows: 300px 300px;  
}  
  
.box3 {  
  grid-column-start: 1;  
  grid-column-end: 3;  
}
```



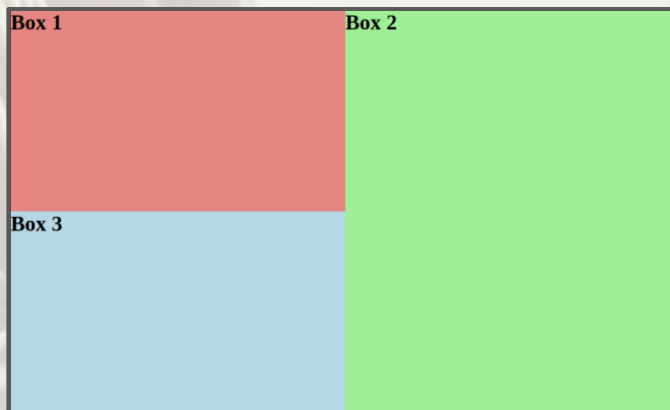




# Προοπτική Γραμμών και Στηλών

- Το ίδιο μπορούμε να εφαρμόσουμε και με τις γραμμές, χρησιμοποιώντας την εντολή `grid-row`, όπως μπορείτε να παρατηρήσετε στο παράδειγμα πιο κάτω.

```
.container {  
  display: grid;  
  grid-template-columns: 500px 500px;  
  grid-template-rows: 300px 300px;  
}  
  
.box1 {  
  grid-row: 1/2;  
}  
  
.box2 {  
  grid-row: 1/3;  
}
```



# Προοπτική Γραμμών και Στηλών

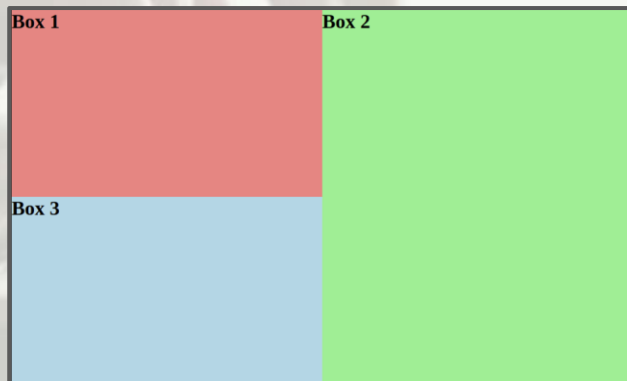
- Για διευκολύνουμε ακόμα περισσότερο τη διαδικασία μπορούμε να χρησιμοποιήσουμε την εντολή **grid-template-areas**, η οποία μπορεί να μας δώσει το ίδιο αποτέλεσμα αν ορίσουμε τα ονόματα των χώρων των θυγατρικών στοιχείων ....and as you can see it give us the same result, by declaring the areas names in the children's and use it in a visual way in the container.

```
.container {
  display: grid;
  grid-template-columns: 500px 500px;
  grid-template-rows: 300px 300px;
  grid-template-areas:
    "box1 box2"
    "box3 box2";
}

.box1 {
  grid-area: box1;
}

.box2 {
  grid-area: box2;
}

.box3 {
  grid-area: box3;
}
```





# Προοπτική Γραμμών και Στηλών

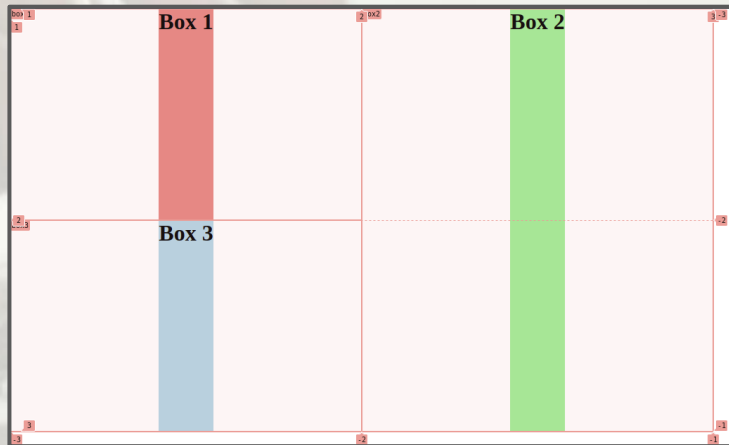
- Μπορούμε επίσης να αλλάξουμε όλες τις θέσεις των θυγατρικών στοιχείων χρησιμοποιώντας την εντολή **justify-items** με τις επιλογές τοποθέτησης **center**, **start**, **end** και **stretch**

```
.container {
  display: grid;
  grid-template-columns: 500px 500px;
  grid-template-rows: 300px 300px;
  grid-template-areas:
    "box1 box2"
    "box3 box2";
  justify-items: center;
}

.box1 {
  grid-area: box1;
}

.box2 {
  grid-area: box2;
}

.box3 {
  grid-area: box3;
}
```





# Προοπτική Γραμμών και Στηλών

- Μπορούμε επίσης να χρησιμοποιήσουμε την προηγούμενη εντολή για ένα συγκεκριμένο στοιχείο, απλά χρησιμοποιώντας την εντολές **align-self** ή **justify-self** στο θυγατρικό στοιχείο που θέλουμε αυτό να εφαρμοστεί.

```
.container {  
  display: grid;  
  grid-template-columns: 500px 500px;  
  grid-template-rows: 300px 300px;  
  grid-template-areas:  
    "box1 box2"  
    "box3 box2";  
}  
  
.box1 {  
  grid-area: box1;  
  align-self: center;  
}  
  
.box2 {  
  grid-area: box2;  
  justify-self: center;  
}  
  
.box3 {  
  grid-area: box3;  
}
```

