



#### ΠΕΡΙΕΧΟΜΕΝΑ:

1. Εισαγωγή στο grid
  1. Ορολογία και αυτόματη τοποθέτηση
2. Σύντμηση grid
3. Μήκη στο grid
  1. Επαναλαμβανόμενα tracks
4. Tool στον Google Chrome

Γιώργος Τ.

Χρυσός Χορηγός Μαθήματος

## ΜΑΘΗΜΑ 8.1: GRID: ΠΕΡΙΟΧΕΣ

### 1. Εισαγωγή στο Grid

#### CSS Grid Layout Module:

- Οργανώνει το χώρο της σελίδας σε ένα **grid (=πλέγμα)**
- Μπορούμε να τοποθετήσουμε στοιχεία μέσα στα κελιά του πλέγματος.

#### Πως δουλεύει;

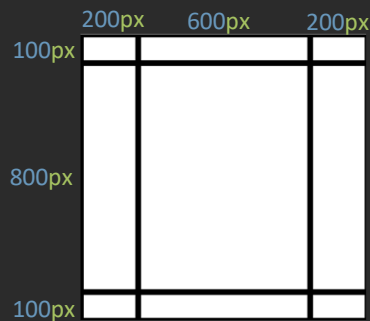
- Θέτουμε το **display** property σε ένα στοιχείο (π.χ. το body) ίσο με **grid**

```
body {
  display: grid;
}
```

- το στοιχείο γίνεται ένα πλέγμα και λέμε ότι είναι **grid container**.
- Καθορίζουμε γραμμές και στήλες του πλέγματος με τα properties **grid-template-rows** και **grid-template-columns** (για κάθε γραμμή ή στήλη καθορίζουμε το μήκος της)

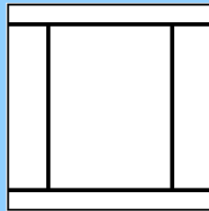
```
body {
  ...
  grid-template-rows: 100px 800px 100px;
  grid-template-columns: 200px 600px 200px;
}
```

- Έτσι έχει οριστεί ένα πλέγμα 3 γραμμές x 3 στήλες



- «Βαφτίζουμε» (με ονόματα της αρεσκείας μας), ανά γραμμές, μέσα σε εισαγωγικά, τα κελιά του πίνακα με το property **grid-template-areas**.
  - Σημαντικό: Τα κελιά που έχουν το ίδιο όνομα θα ενοποιηθούν.

```
body {
  ...
  grid-template-areas:
    "header header header"
    "sidebar-contents main sidebar-ads"
    "footer footer footer";
}
```

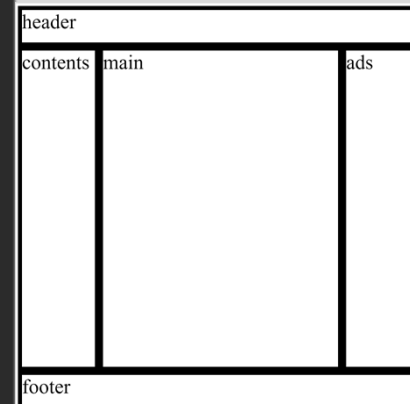


- Αναθέτουμε κάθε παιδί του στοιχείου body (στην HTML) σε μια περιοχή.
  - Όρίζοντας το property: grid-area ίσο με το όνομα που θέσαμε στα (ενοποιημένα) κελιά.
  - Κάθε παιδί λέμε ότι είναι ένα **grid item**.

#### Παράδειγμα 1: grid

```
<body>
<header>header</header>
<main>main</main>
<aside id="aside-left">
  contents
</aside>
<aside id="aside-right">
  ads
</aside>
<footer>footer</footer>
</body>
```

```
header {
  grid-area: header;
}
#aside-left {
  grid-area: sidebar-contents;
}
#aside-right {
  grid-area: sidebar-ads;
}
main {
  grid-area: main;
}
footer {
  grid-area: footer;
}
```



## ΜΑΘΗΜΑ 8.1: GRID: ΠΕΡΙΟΧΕΣ

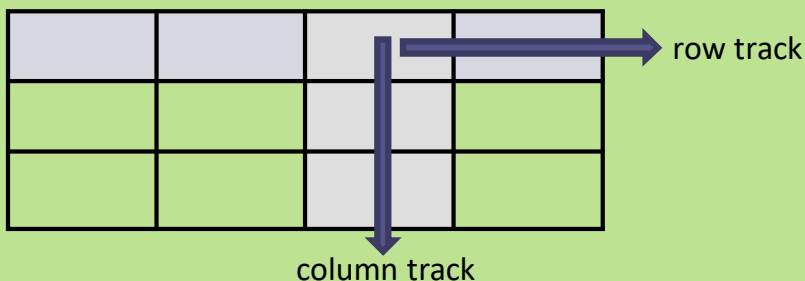
### 1.1. Ορολογία και αυτόματη τοποθέτηση

#### Ορολογία στο grid:

- Το πλέγμα αποτελείται από **κελιά (cells)**. Πολλά κελιά μαζί ορίζουν μια **περιοχή (grid area)**. Ένα κελί είναι και αυτό περιοχή.



- Τόσο οι γραμμές όσο και οι στήλες από κελιά, ονομάζονται **tracks** (διαχωρίζονται σε **row tracks** και **column tracks**)



- Οι γραμμές που ορίζουν το πλέγμα (οριζόντιες και κάθετες) καλούνται **grid lines** (τα οποία αριθμούνται αυτόματα με ευθεία και αντίστροφη αρίθμηση ως εξής):



#### Σημείωση:

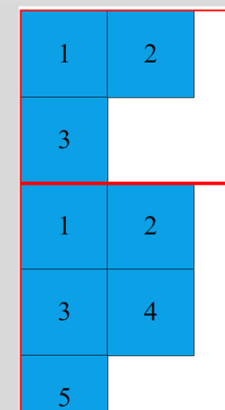
- Αν δεν τοποθετήσουμε ένα grid item σε κάποιο κελί ρητώς (με την grid-area), τότε αυτό θα τοποθετηθεί στο πρώτο διαθέσιμο κελί
- Αν δεν υπάρχει διαθέσιμο κελί, τότε θα δημιουργηθεί νέο (σε νέα γραμμή).
- Σε επόμενο μάθημα θα μάθουμε πολύ περισσότερες λεπτομέρειες για την **«αυτόματη τοποθέτηση»**

#### Παράδειγμα 2: auto-placement

```
<div class="container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
</div>
<div class="container">
  <div>1</div>
  <div>2</div>
  <div>3</div>
  <div>4</div>
  <div>5</div>
</div>

.container {
  outline: 2px red solid;
  display: grid;
  grid-template-rows: 100px 100px;
  grid-template-columns: 100px 100px;
}

.container > * {
  background-color: #0ca0e7;
}
```

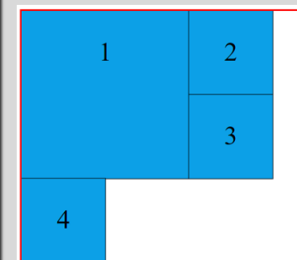


#### Παράδειγμα 3: auto-placement2

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

.container {
  display: grid;
  grid-template-rows: 100px 100px 100px;
  grid-template-columns: 100px 100px 100px;
  grid-template-areas:
    "area1 area1 area2"
    "area1 area1 area2";
}

.placed {
  grid-area: area1;
}
```



## ΜΑΘΗΜΑ 8.1: GRID: ΠΕΡΙΟΧΕΣ

### 2. Σύντμηση grid

- Η **ιδιότητα-σύντμηση grid** προσφέρει σύντμηση για τις grid-template-areas, grid-template-rows και grid-template-columns

**grid:** σύντμηση grid-template-rows, -columns, -areas

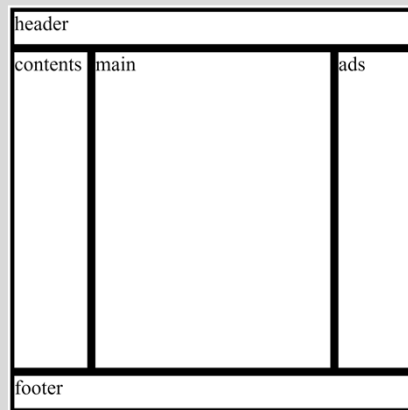
τιμές	"row-names1" row-len1 ... / col-len-1, ...		
default	συνδυασμός τιμών σε properties (στις 3 τιμές: none)		
Στοιχεία	grid container	Κληρονόμηση	Όχι

#### Παράδειγμα 4: grid-shorthand

```
body {
  display: grid;
  grid-template-rows: 100px 800px 100px;
  grid-template-columns: 200px 600px 200px;
  grid-template-areas:
    "header header header"
    "sidebar-contents main sidebar-ads"
    "footer footer footer";
}
```

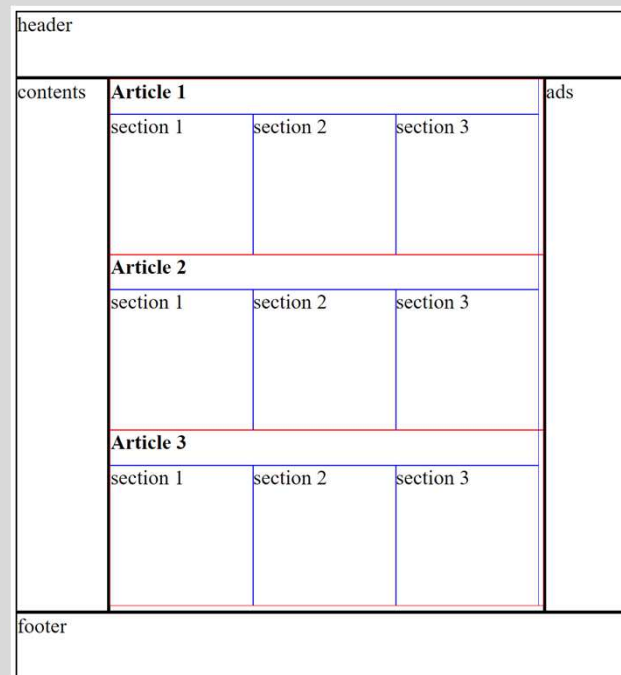
Είναι ισοδύναμο με το:

```
body {
  display: grid;
  grid:
    "header header header" 100px
    "sidebar-contents main sidebar-ads" 800px
    "footer footer footer" 100px
    /
    200px 600px 200px;
}
```



#### Άσκηση 1:

- Ξεκινώντας από το αρχείο exercise01-initial.html επιτυχετε την ακόλουθη διεύθυνση των άρθρων της σελίδας.



- Τα άρθρα πιάνουν το 33% του διαθέσιμου ύψους
- Οι ενότητες πιάνουν το 33% του διαθέσιμου πλάτους.

## ΜΑΘΗΜΑ 8.1: GRID: ΠΕΡΙΟΧΕΣ

### 3. Μήκη στο grid

- Το μήκος στα properties `grid-template-rows` και `grid-template-columns` μπορεί να είναι:
  - Συνηθισμένες Μετρήσεις Μήκους
  - Ποσοστά (επί του συνολικού μήκους)
  - Κλασματικά Μέρη (fr(actional) units)
    - π.χ. αν καθορίσουμε ότι το ένα track είναι 1fr και ένα άλλο είναι 2fr, τότε το 1<sup>ο</sup> θα λάβει το  $1/(1+2)=1/3$  του διαθέσιμου χώρου και το 2<sup>ο</sup> θα λάβει τα  $2/(1+2)=2/3$  του διαθέσιμου χώρου

- min-content
  - Ελάχιστο δυνατό, χωρίς να γίνεται overflow (π.χ. μήκος της μέγιστης λέξης, ή εικόνας)
- max-content
  - Μέγιστο δυνατό (π.χ. μία παράγραφος δεν θα γίνεται wrap)
- minmax(a, b)
  - a: ελάχιστη τιμή (μήκος)
  - b: μέγιστη τιμή (μήκος)
- fit-content(x)
  - $\min(\max \text{ size}, \max(\min \text{ size}, x))$  ☺

#### Παράδειγμα 5: grid-fractional-units

```
<div class="container">
  <div>1</div>
  <div>2</div>
  ...
  <div>9</div>
</div>
```

```
.container {
  display: grid;
  grid-template-rows: 3fr 200px 1fr;
  grid-template-columns: 4fr 200px 1fr;
  height: 100vh;
}
.container > * {
  background-color: #0ca0e7;
  outline: 1px black solid;
  text-align: center;
  padding-top: 30px;
}
```

1	2	3
4	5	6
7	8	9

#### Παράδειγμα 6: fit-content

```
.container {
  outline: 1px black solid;
  display: grid;
  grid-template-rows: 200px;
  grid-template-columns:
    1fr fit-content(50px)
    fit-content(100px);
  overflow: hidden;
}
```

>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ab, aliquid amet at beatae et eum in ipsa labore laudantium nobis odit, optio quaerat, qui ratione sequi tempore vel veniam voluptatum.	>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam aliquid, beatae deserunt dolores dolorum	>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Dolor eaque enim, et expedita quod voluptas. Ipsa nobis officiis sapiente sit soluta vero?	>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Asperiores consequuntur dolor eveniet, molestias nisi numquam obcaecati placeat qui repellendus ut? Architecto eaque eligendi eum laudantium maiores quos ratione. Doloremque, ducimus!
---	--	--	---

#### Παράδειγμα 7: minmax

```
.container {
  outline: 1px black solid;
  display: grid;
  grid-template-rows: 200px;
  grid-template-columns:
    minmax(100px, 200px)
    1fr 1fr;
  overflow: hidden;
}
```

>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ab, aliquid amet at beatae et eum in ipsa labore laudantium nobis odit, optio quaerat, qui ratione sequi tempore vel veniam voluptatum.	>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam aliquid, beatae deserunt dolores dolorum eaque enim hic id, in modi necessitatibus officia omnis optio quae qui ratione voluptatem, voluptates voluptatum.	>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Dolor eaque enim, et expedita quod voluptas. Ipsa nobis officiis sapiente sit soluta vero? Accusantium blanditiis cum dolor eveniet libero minima suscipit?	>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Dolor sit amet, consectetur adipiscing elit. Dolor sit amet, consectetur adipiscing elit. Dolor sit amet, consectetur adipiscing elit. Dolor sit amet, consectetur adipiscing elit.
---	--	---	---

## ΜΑΘΗΜΑ 8.1: GRID: ΠΕΡΙΟΧΕΣ

### 3.1. Επαναλαμβανόμενα tracks

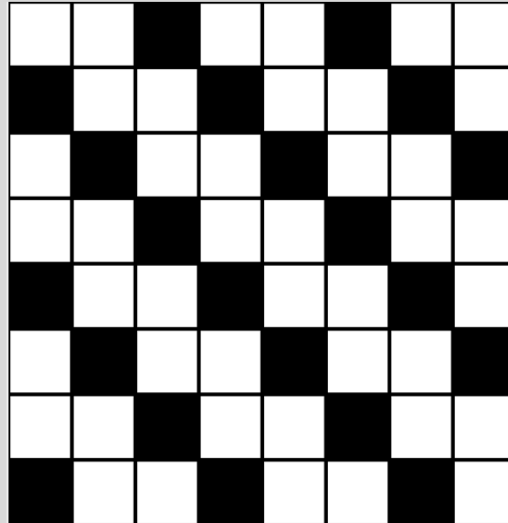
Στα grid-template-rows και grid-template-columns:

- Μπορούμε να ορίσουμε επαναλαμβανόμενα μήκη με την:
  - repeat(N, len)**
    - Ορίζει N tracks, το καθένα με μήκος len
- Μπορούμε να ορίσουμε και πολλά μήκη που θα επαναλαμβάνονται N φορές
  - πχ. **repeat(N, len1 len2 ...)** παράγει len1, len2,..., len1, len2, ...
- και μπορεί να συνδυαστεί με μήκη
  - πχ. **20px repeat(3, 30px) 10px** παράγει 20px 30px 30px 30px 10px

- Η repeat, μπορεί εναλλακτικά με το πλήθος tracks να συνταχθεί ως:
  - repeat(auto-fill, len)**
    - Γεμίζει το διαθέσιμο χώρο με tracks μεγέθους len.
  - (Συνήθως το θέτουμε για να ορίσουμε τις στήλες)
    - Αν δεν επαρκεί ο διαθέσιμος χώρος για να χωρέσουν όλα τα items, τότε δημιουργείται νέα γραμμή που μπαίνουν τα υπόλοιπα items κ.ο.κ.

#### Παράδειγμα 8: grid-repeating

```
div {
  border: 5px black solid;
  font-size: 16px;
}
body {
  display: grid;
  width: 80%;
  margin: auto;
  height: 100vh;
  min-width: 1000px;
  min-height: 600px;
  grid-template-rows: repeat(8, 1fr);
  grid-template-columns: repeat(8, 1fr);
}
div:nth-of-type(3n) {
  background-color: black;
}
```



#### Παράδειγμα 9: auto-fill

```
.container {
  outline: 1px black solid;
  display: grid;
  grid-template-rows: 100px;
  grid-template-columns:
    repeat(auto-fill, 100px);
  overflow: hidden;
}
```

1	2	3	
4	5	6	
7	8	9	
10	11		

1	2	3	4	
5	6	7	8	
9	10	11		

1	2	3	4	5	
6	7	8	9	10	
11					

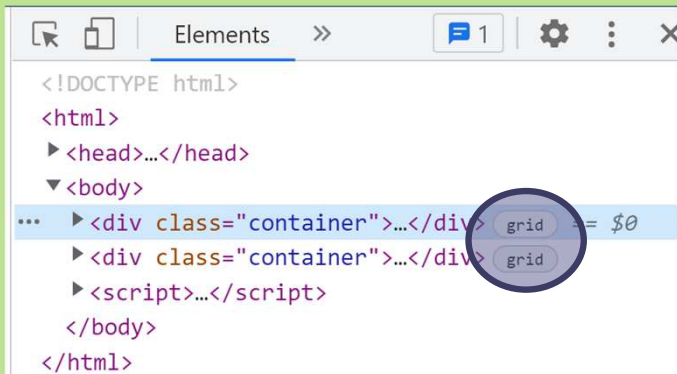
#### Παρατήρηση:

- Τις λεπτομέρειες για τις νέες γραμμές που κατασκευάζονται μπορούμε να τις καθορίσουμε με άλλα properties που θα δούμε σε επόμενο μάθημα (αυτόματη τοποθέτηση)

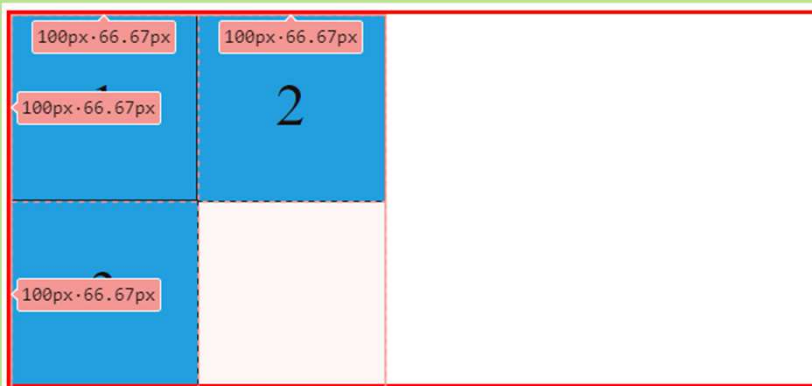
## ΜΑΘΗΜΑ 8.1: GRID: ΠΕΡΙΟΧΕΣ

### 4. Tool στον Google Chrome

- **Ο Google Chrome** έχει ένα ειδικό εργαλείο για το Grid.
- (Δεξί κλικ και έπειτα Inspect στη σελίδα μας για να εμφανιστούν τα developer tools)
- Στον κώδικα HTML έχει εμφανιστεί ένα κουμπάκι “grid” στα HTML στοιχεία που είναι grid containers:

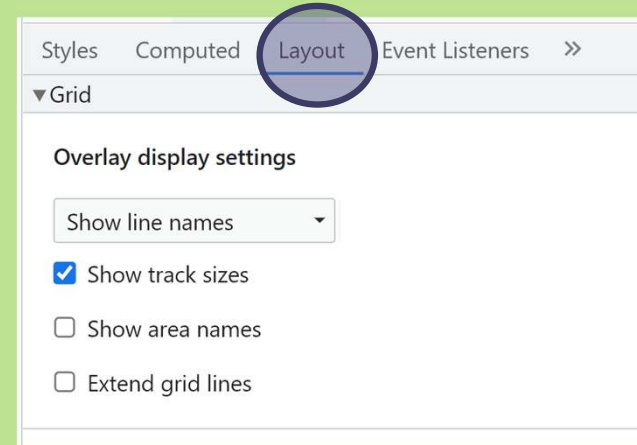


- Έπειτα εμφανίζονται στη σελίδα τα μήκη που αφορούν κάθε κελί του grid (πλάτος - ύψος):

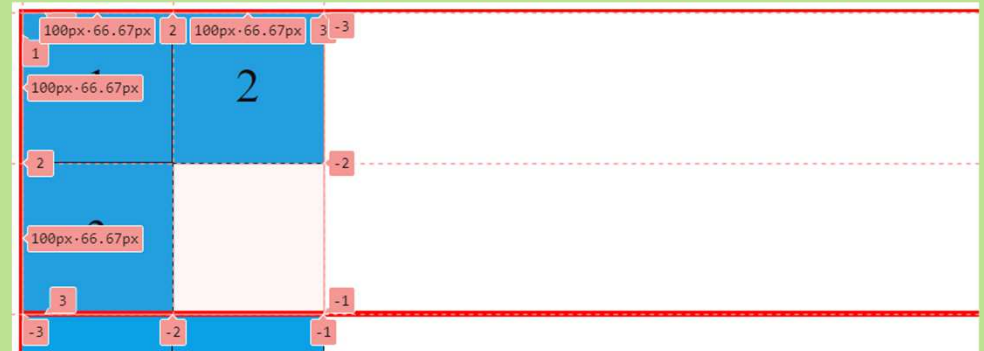


#### Περαιτέρω επιλογές:

- Κάνουμε κλικ στο tab “Layout” των Developer Tools:



- Όπου μπορούμε (επιλέγοντας τα αντίστοιχα κουτιά) να εμφανίσουμε τα ονόματα των περιοχών (και των γραμμών - επόμενο μάθημα) κ.α.:



#### Σημείωση:

- Ο ίδιος τρόπος εργασίας μπορεί να γίνει και για Flex containers (scroll down to «Layout»)