



CSS Grid Layout

Αθανάσιος Ανδρούτσος



Grid Layout (1)

- Το CSS Grid Layout ή CSS Grid είναι ένας two-dimensional grid-based layout manager που παρέχει ένα μηχανισμό διάταξης του περιεχομένου σε δύο διαστάσεις, παρόμοιο με το μηχανισμό των πινάκων
- Πρόκειται για καινούργιο μηχανισμό διαχείρισης του περιεχομένου (CSS Grid Layout Module Level 1, W3C Recommendation, 12/2020) που έχει περισσότερα πλεονεκτήματα σε σχέση με το Flexbox



CSS Grid (1)

- Δίνοντας στο **display** property ενός container box την τιμή **grid** (**display: grid**) προστάζουμε τον browser ότι όλο το **περιεχόμενο του container box** θα πρέπει να γίνει **render με το CSS Grid Layout model**
- Μέσα στον container μπορούμε να ορίσουμε τον αριθμό των γραμμών και των στηλών καθώς και τις διαστάσεις τους ως **fixed lengths**, ποσοστό ή να προσαρμόζονται στα περιεχόμενα



CSS Grid (2)

- Τα περιεχόμενα του grid μπορούμε να τα τοποθετήσουμε με βάση την γραμμή έναρξης και τέλους τόσο για τις γραμμές όσο και για τις στήλες
- Μπορούμε επίσης να στοιχίσουμε καθώς και να ορίσουμε το spacing μεταξύ γραμμών ή στηλών
- Το grid είναι responsive με media queries



Flexbox vs Grid (1)

Προγραμματισμός στο Web



- *Flexbox* στο επάνω μέρος
- *Grid* στο κάτω μέρος

- Μπορούμε να σκεφτόμαστε το Flexbox ως one-dimensional μιας και παρότι μπορούμε να κάνουμε wrap, δεν μπορούμε να ελέγξουμε σε ποιο ακριβώς flex item θα γίνει το wrap
- Ενώ στο Grid, ως two-dimensional, μπορούμε να δηλώσουμε στον parent container τις διαστάσεις των στηλών και των γραμμών και να τοποθετήσουμε τα grid items στα κελιά ή τις περιοχές που θέλουμε



Οριζόντιο κάθετο κέντρο σελίδας Flexbox (HTML)

Προγραμματισμός στο Web

```
<> flexcenter.html M X
chapter8 > <> flexcenter.html > html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, in
7   <title>Flex Center</title>
8   <link rel="stylesheet" href="./css/gridcenter.css">
9 </head>
10 <body>
11   <div class="center">
12     <div class="hello-text">Hello</div>
13   </div>
14 </body>
15 </html>

# flexcenter.css M X
chapter8 > css > # flexcenter.css > .hello-text
1 .center {
2   display: flex;
3   justify-content: center;
4   align-items: center;
5   height: 95vh;
6 }
7
8 .hello-text {
9   border: 1px solid red;
10 }
```

- Θέλουμε το **Hello** να το στοιχίσουμε στο κέντρο της σελίδας. Με flexbox αν δώσουμε ύψος 95vh (viewport height). Πολλές φορές στους browsers ιδιαίτερα σε κινητές συσκευές στο 100vh συμπεριλαμβάνεται και το URL bar, οπότε για να στοιχίσουμε στο κάθετο κέντρο δίνουμε 95vh



Οριζόντιο κάθετο κέντρο σελίδας Grid (HTML)

Προγραμματισμός στο Web

flexcenter.html M

chapter8 > flexcenter.html > html > body > div.center

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, in
7   <title>Flex Center</title>
8   <link rel="stylesheet" href="./css/gridcenter.css">
9 </head>
10 <body>
11   <div class="center">
12     <div class="hello-text">Hello</div>
13   </div>
14 </body>
15 </html>
```

gridcenter.css

chapter8 > css > # gridcenter.css > ...

```
1 .center {
2   display: grid;
3   height: 95vh;
4   place-items: center;
5 }
6
7
```

- Με place items στοιχίζουνε οριζόντια και κάθετα



Flexbox (1)

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Flex</title>
8      <link rel="stylesheet" href="./css/flex.css">
9  </head>
10 <body>
11 <div class="container">
12     <div class="flex-item-1 border-dark">Flexitem-1</div>
13     <div class="flex-item-2 border-dark">Flexitem-2</div>
14     <div class="flex-item-3 border-dark">Flexitem-3</div>
15 </div>
16 </body>
17 </html>
```




Flexbox (2)

```
1  .container {
2    display: flex;
3    width: 90%;
4    margin-left: auto;
5    margin-right: auto;
6    border: 1px solid black;
7    border-radius: 10px;
8    height: 10vh;
9  }
10
11  .border-dark {
12    border: 1px solid red;
13  }
14
15  .flex-item-1 {
16    flex: 0 0 100px;
17    /* Since flex grow is 0 and flex-shrink is 0
18     then 100px functions as fixed width */
19  }
20
21  .flex-item-2 {
22    flex: 1 0 500px;
23    /* Since flex-grow is 1, then flex-basis functions as minimum width
24     If there is available space item-2 will grow. */
25  }
26
27  .flex-item-3 {
```

- Το flex property παίρνει τρεις παραμέτρους: flex-grow, flex-shrink και flex-basis
- Αν έχουμε ένα flex item με το flex-grow να είναι 1 τότε το item θα προσπαθήσει να γεμίσει τον κενό χώρο
- Αν έχουμε περισσότερα από 1 flex-items τότε τα values προστίθενται και κάθε item λαμβάνει size: value / total value sum

127.0.0.1:5500/testbed/grid-flexbox/flex.html



Flexitem-1	Flexitem-2	Flexitem-3
------------	------------	------------



Flexbox (3)

- Αλλάξαμε το flex-basis σε 1000px και έχουμε overflow

```
1 .container {
2   display: flex;
3   width: 90%;
4   margin-left: auto;
5   margin-right: auto;
6   border: 1px solid black;
7   border-radius: 10px;
8   height: 10vh;
9 }
10
11 .border-dark {
12   border: 1px solid red;
13 }
14
15 .flex-item-1 {
16   flex: 0 0 100px;
17   /* Since flex grow is 0 and flex-shrink is 0
18   then 100px functions as fixed width */
19 }
20
21 .flex-item-2 {
22   flex: 1 0 500px;
23   /* Since flex-grow is 1, then flex-basis functions as minimum width
24   If there is available space item-2 will grow. */
25 }
26
27 .flex-item-3 {
28   flex: 0 0 1000px;
29   /*Default is no-wrap*/
30 }
```

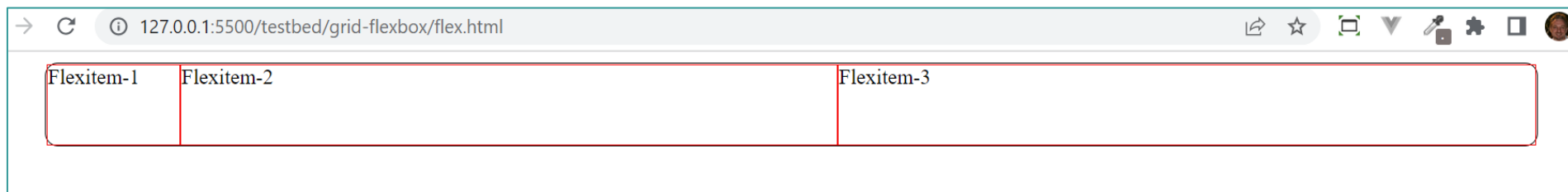
127.0.0.1:5500/testbed/grid-flexbox/flex.html

Flexitem-1	Flexitem-2	Flexitem-3



Flex-shrink

```
26  
27 .flex-item-3 {  
28     flex: 0 1 1000px;  
29     /*Default is no-wrap*/  
30 }
```



- Το Flex-shrink 1 κάνει shrink το size ώστε να μην υπάρχει overflow



Flex-wrap (1)

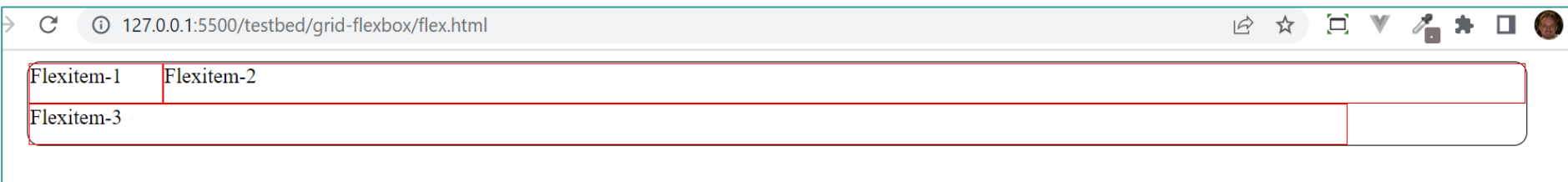
```
2   display: flex;
3   width: 90%;
4   flex-flow: row wrap;    /*wrap*/
5   margin-left: auto;
6   margin-right: auto;
7   border: 1px solid black;
8   border-radius: 10px;
9   height: 10vh;
10  }
11
12  .border-dark {
13    border: 1px solid red;
14  }
15
16  .flex-item-1 {
17    flex: 0 0 100px;
18    /* Since flex grow is 0 and flex-shrink is 0
19     then 100px functions as fixed width */
20  }
21
22  .flex-item-2 {
23    flex: 1 0 500px;
24    /* Since flex-grow is 1, then flex-basis functions as minimum width
25     If there is available space item-2 will grow. */
26  }
27
28  .flex-item-3 {
29    flex: 0 0 1000px;
30  }
```

- Με flex-flow: row wrap;
- Ή flex-flow: column wrap
- Τα items αυτόματα στοιχίζονται σε γραμμές



Flex-wrap (2)

```
2      display: flex;
3      width: 90%;
4      /* flex-flow: row wrap; */
5      flex-wrap: wrap;
6      margin-left: auto;
7      margin-right: auto;
8      border: 1px solid black;
9      border-radius: 10px;
10     height: 10vh;
11 }
12
```



- Το ίδιο γίνεται και με την ιδιότητα flex-wrap: wrap;



Grid Container

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>CSS Grid</title>
8  </head>
9  <body>
10     <div class="container">
11         <div class="grid-item-1">Grid-item-1</div>
12         <div class="grid-item-2">Grid-item-2</div>
13         <div class="grid-item-3">Grid-item-3</div>
14     </div>
15 </body>
16 </html>
```

- **Grid container.** Το UI element που κάνουμε display: grid



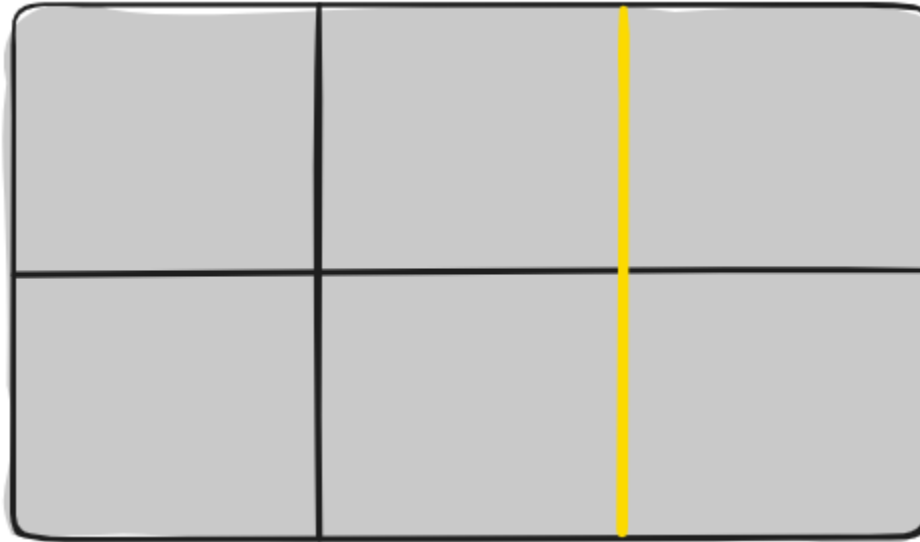
Grid Items

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>CSS Grid</title>
8  </head>
9  <body>
10     <div class="container">
11         <div class="grid-item-1">Grid-item-1</div>
12         <div class="grid-item-2">Grid-item-2</div>
13         <div class="grid-item-3">Grid-item-3</div>
14     </div>
15 </body>
16 </html>
```

- Grid items. Οι άμεσοι απόγονοι του Grid container



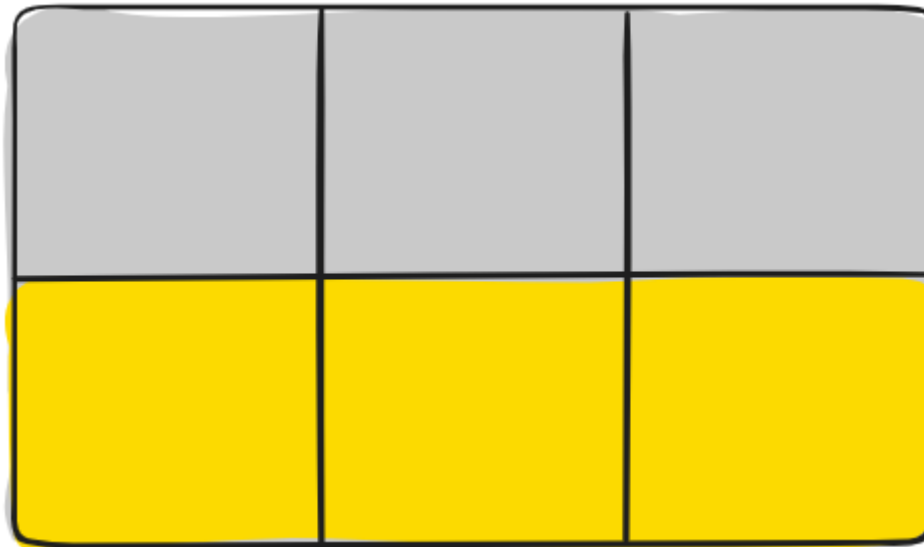
Grid Lines



- **Grid line.** row grid lines ή column grid lines. Οι γραμμές έχουν αρίθμηση ξεκινώντας από το 1. Στο παράδειγμα έχουμε 4 κάθετες και 3 οριζόντιες γραμμές



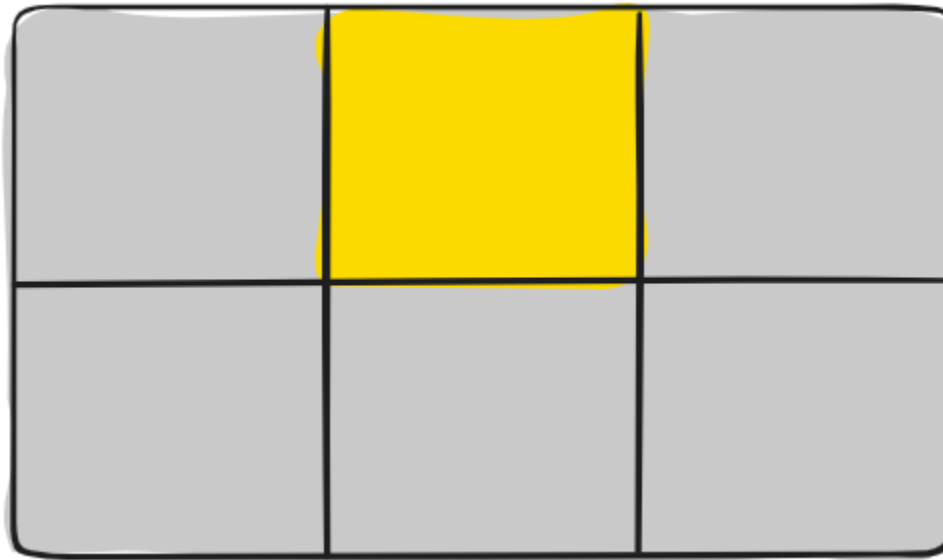
Grid Track



- Η περιοχή (οριζόντια ή κάθετη) ανάμεσα σε διαδοχικές γραμμές
- Είναι ουσιαστικά μία γραμμή (ή μία στήλη)



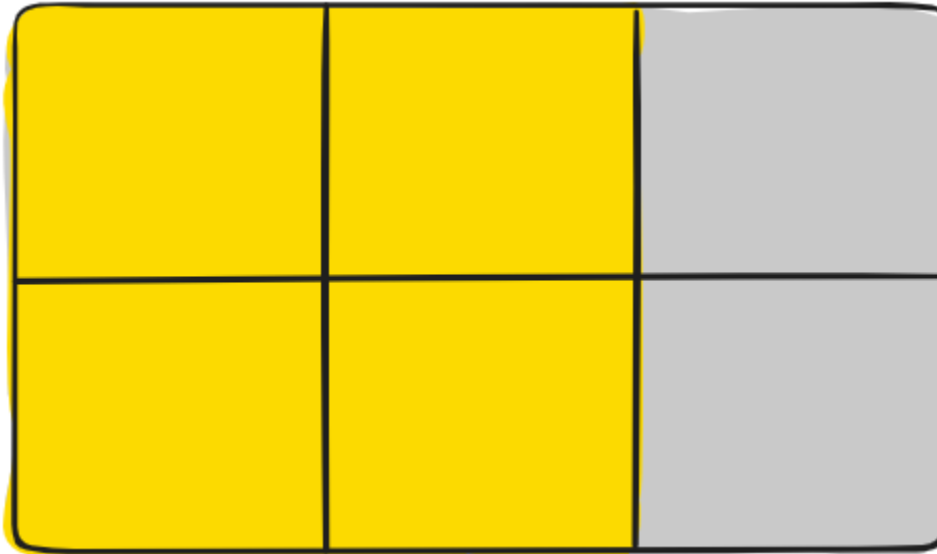
Grid Cell



- Η περιοχή στην τομή δύο Grid tracks ονομάζεται Grid Cell (κελί)



Grid Area



- Μία περιοχή που περιλαμβάνει δύο ή περισσότερα cells



Ιδιότητες στον Container

Προγραμματισμός στο Web

- Display: grid; Ορίζει έναν grid container
- **Grid-template-columns:** repeat(3, 1fr)
- **Grid-template-rows:** repeat(2, 100px)
- Τα δύο παραπάνω ορίζουν μία λίστα τιμών που ορίζουν γραμμές και στήλες. Συγκεκριμένα **3 στήλες και 2 γραμμές**. Οι τιμές μπορεί να είναι length σε px, percentage, auto ή ένα fraction (κλάσμα) του ελεύθερου χώρου του Grid Container
- Στο παραπάνω παράδειγμα, στις στήλες το πλάτος θα διαιρεθεί δια 3 και κάθε στήλη θα έχει 1/3 πλάτος, ενώ θα έχουμε 2 γραμμές με 100px η κάθε γραμμή



Grid Template (1)

```
testbed > grid-flexbox > <> grid.html > ...
```

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=e
6   <meta name="viewport" content="width=device-widt
7   <title>CSS Grid</title>
8   <link rel="stylesheet" href="._css/grid.css">
9 </head>
10 <body>
11   <div class="container">
12     <div class="grid-item-1">Grid-item-1</div>
13     <div class="grid-item-2">Grid-item-2</div>
14     <div class="grid-item-3">Grid-item-3</div>
15     <div class="grid-item-4">Grid-item-4</div>
16     <div class="grid-item-5">Grid-item-5</div>
17     <div class="grid-item-6">Grid-item-6</div>
18   </div>
19 </body>
20 </html>
```

- Έχουμε 6 items ή μπορεί και περισσότερα, που θέλουμε να τακτοποιήσουμε σε γραμμές και στήλες ανά τρεις στήλες
- Οπότε θέλουμε 3 στήλες και αυτόματα να προστίθενται γραμμές



Grid template (2)

```
1  .container {  
2      display: grid;  
3      grid-template-columns: 1fr 2fr 1fr;  
4      grid-auto-rows: auto;  
5  }  
6  
7  div[class^="grid"] {  
8      border: 1px solid red;  
9  }
```

A screenshot of a web browser window. The address bar shows the URL '127.0.0.1:5500/testbed/grid-flexbox/grid.html'. The browser content displays a 2x3 grid of six items, each with a red border. The items are labeled 'Grid-item-1' through 'Grid-item-6'.

Grid-item-1	Grid-item-2	Grid-item-3
Grid-item-4	Grid-item-5	Grid-item-6

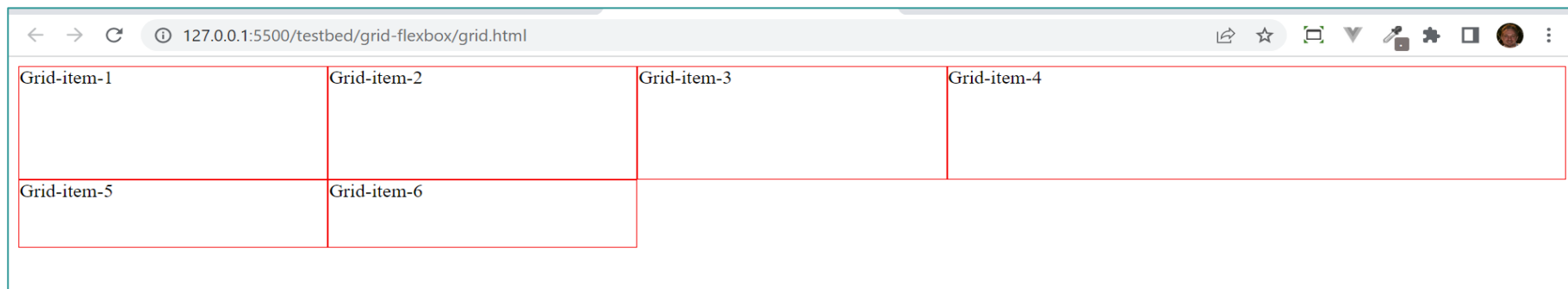
- Με `grid-auto-rows` δημιουργούνται δυναμικά καινούργιες γραμμές όταν έχουμε περισσότερα items από cells



Grid template (3)

```
testbed > grid-flexbox > css > # grid.css > ...
1  .container {
2      display: grid;
3      grid-template-columns: 1fr 1fr 1fr 2fr;
4      grid-template-rows: 100px;
5      grid-auto-rows: 60px;
6  }
7
8  div[class^="grid"] {
9      border: 1px solid red;
10 }
11
```

- Έχουμε 4 στήλες, όπου η κάθε στήλη καταλαμβάνει το $\frac{1}{4}$ του πλάτους της γραμμής
- Έχουμε μία γραμμή με 100px ύψος
- Και έχουμε δυναμικά επιπλέον γραμμές (ανάλογα πόσα στοιχεία έχουμε) όπου η κάθε γραμμή έχει ύψος 60px





Grid Areas

```
testbed > grid-flexbox > <> grid-areas.html > ...
```

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1">
7      <title>CSS Areas</title>
8      <link rel="stylesheet" href="./css/grid-areas.css">
9  </head>
10 <body>
11     <div class="container">
12         <div class="grid-item-1">Grid-item-1</div>
13         <div class="grid-item-2">Grid-item-2</div>
14         <div class="grid-item-3">Grid-item-3</div>
15         <div class="grid-item-4">Grid-item-4</div>
16     </div>
17 </body>
18 </html>
19
```




Properties για children

Προγραμματισμός στο Web

```
1  .container {
2    display: grid;
3    grid-template-columns: repeat(4, 1fr);
4    grid-auto-rows: auto;
5    grid-template-areas:
6      "header header header header"
7      "main main . sidebar"
8      "footer footer footer footer";
9  }
10
11  .grid-item-1 {
12    grid-area: header;
13  }
14
15  .grid-item-2 {
16    grid-area: main;
17  }
```

```
19  .grid-item-3 {
20    grid-area: sidebar;
21  }
22
23  .grid-item-4 {
24    grid-area: footer;
25  }
26
27
28  div[class^="grid"] {
29    border: 1px solid red;
30  }
```

- Με τα grid areas δίνουμε aliases (ψευδώνυμα) σε cells. Στη συνέχεια εκχωρούμε τα cells στα grid items με **grid-area**
- Εδώ ορίζουμε 4 στήλες και 3 γραμμές. Στη 2^η γραμμή οι 2 πρώτες στήλες είναι main, η 3^η στήλη είναι κενή (η τελεία σημαίνει κενό κελί) και η τέταρτη στήλη της 2^{ης} γραμμής είναι sidebar



Properties στα children - sizing

Προγραμματισμός στο Web

```
testbed > grid-flexbox > <> gridchildren-sizing.html > ...
```

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-sc
7      <title>CSS Areas</title>
8      <link rel="stylesheet" href="./css/gridchildren-sizing.css">
9  </head>
10 <body>
11     <div class="container">
12         <div class="grid-item-1">Grid-item-1</div>
13         <div class="grid-item-2">Grid-item-2</div>
14         <div class="grid-item-3">Grid-item-3</div>
15         <div class="grid-item-4">Grid-item-4</div>
16         <div class="grid-item-5">Grid-item-5</div>
17         <div class="grid-item-6">Grid-item-6</div>
18     </div>
19 </body>
20 </html>
21
```

- Έστω τα έξι items.
- Θα δώσουμε size όχι στον container αλλά στα ίδια τα items



Grid-column (1)

```
13  
14 .grid-item-1 {  
15     background-color: green;  
16     grid-column-start: 1;  
17     grid-column-end: 3;  
18 }  
19
```

```
.grid-item-1 {  
    background-color: green;  
    grid-column-start: 1;  
    grid-column-end: span 2;  
}
```

```
.grid-item-1 {  
    background-color: green;  
    grid-column: 1 / 3;  
}
```

- Όλα δίνουν το ίδιο αποτέλεσμα. Η αρίθμηση είναι των διαχωριστικών γραμμών μεταξύ γραμμών ή μεταξύ στηλών (αν έχουμε 3 στήλες είναι 4 γραμμές)

0/testbed/grid-flexbox/gridchildren-sizing.html



Grid-item-1

Grid-item-2



Grid-column (2)

```
testbed / grid-flexbox / gridchildren-sizing.html / ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>CSS Areas</title>
8      <link rel="stylesheet" href="./css/gridchildren-sizing.css">
9  </head>
10 <body>
11     <div class="container">
12         <div class="grid-item-1">Grid-item-1</div>
13         <div class="grid-item-2">Grid-item-2</div>
14         <div class="grid-item-3">Grid-item-3</div>
15         <div class="grid-item-4">Grid-item-4</div>
16         <div class="grid-item-5">Grid-item-5</div>
17         <div class="grid-item-6">Grid-item-6</div>
18         <div class="grid-item-4">Grid-item-7</div>
19         <div class="grid-item-5">Grid-item-8</div>
20         <div class="grid-item-6">Grid-item-9</div>
21     </div>
22 </body>
23 </html>
```



Grid-column (3)

```
1  .container {
2    width: 800px;
3    margin-left: auto;
4    margin-right: auto;
5    display: grid;
6    grid-template-columns: 1fr 1fr 1fr 1fr 1fr;
7    grid-template-rows: repeat(2, 100px);
8
9  }
10
11  div[class^="grid"] {
12    border: 1px solid red;
13  }
14
15  .grid-item-1 {
16    background-color: green;
17    grid-column: 1 / 3;
18    justify-self: center;
19    align-self: center;
20  }
```

- Ορίζουμε πέντε στήλες και δύο γραμμές
- Ορίζουμε το κελί `grid-item-1` να έχει μέγεθος δύο στήλες (από τη διαχωριστική γραμμή 1 έως την 3)
- Ορίζουμε επίσης στοίχιση στο οριζόντιο και το κάθετο κέντρο



Grid-column (4)

```
.grid-item-1 {  
    background-color: ■ green;  
    grid-column: 1 / 3;  
    justify-self: center;  
    align-self: center;  
}
```

/testbed/grid-flexbox/gridchildren-sizing.html



Grid-item-1

Grid-item-2

Grid-item-3

Grid-item-4

Grid-item-5

Grid-item-6

Grid-item-7

Grid-item-8

Grid-item-9



E-shop (1)

Προγραμματισμός στο Web

```
e-commerce.html U X
testbed > grid-flexbox > e-commerce.html > html > body > div.product-grid > div.product
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7 </head>
8 <link rel="stylesheet" href="./e-commerce.css">
9 <body>
10
11   <div class="product-grid">
12     <div class="product">
13       
14       <h3>Camera 1</h3>
15       <p>€89.99</p>
16       <button>Add to Cart</button>
17     </div>
18     <div class="product">
19       
20       <h3>Camera 2</h3>
21       <p>€99.99</p>
22       <button>Add to Cart</button>
23     </div>
24     <div class="product">
25       
26       <h3>Camera 3</h3>
27       <p>€124.99</p>
28       <button>Add to Cart</button>
29     </div>
30     <div class="product">
31       
32       <h3>Camera 1</h3>
33       <p>€89.99</p>
34       <button>Add to Cart</button>
35     </div>
```

```
36
37   <div class="product">
38     
39     <h3>Camera 2</h3>
40     <p>€99.99</p>
41     <button>Add to Cart</button>
42   </div>
43   <div class="product">
44     
45     <h3>Camera 3</h3>
46     <p>€124.99</p>
47     <button>Add to Cart</button>
48   </div>
49   <div class="product">
50     
51     <h3>Camera 1</h3>
52     <p>€89.99</p>
53     <button>Add to Cart</button>
54   </div>
55   <div class="product">
56     
57     <h3>Camera 2</h3>
58     <p>€99.99</p>
59     <button>Add to Cart</button>
60   </div>
61   <!-- Repeat the product structure for additional products -->
62 </div>
63
64 </body>
65 </html>
```

- Έστω μία ακολουθία από products



E-shop (2)

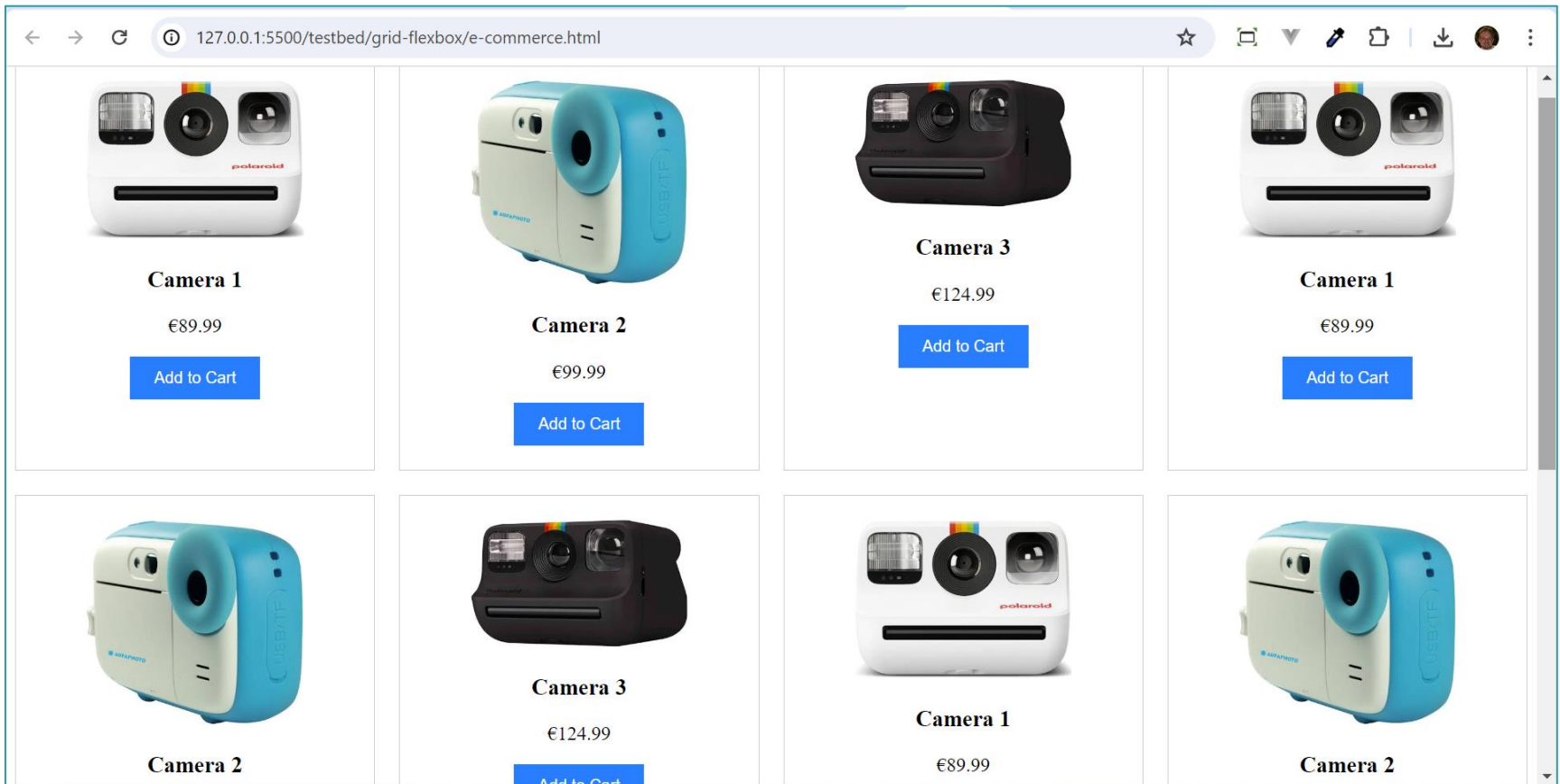
```
1 .product-grid {
2   display: grid;
3   grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
4   grid-gap: 20px;
5 }
6
7 .product {
8   border: 1px solid #ccc;
9   padding: 20px;
10  text-align: center;
11 }
12
13 .product img {
14   width: 60%;
15   height: auto;
16 }
17
18 .product button {
19   background-color: #007bff;
20   color: #fff;
21   border: none;
22   padding: 10px 20px;
23   cursor: pointer;
24 }
25
26 /* Media query for smaller screens */
27 @media (max-width: 768px) {
28   .product-grid {
29     grid-template-columns: repeat(auto-fit, minmax(150px, 1fr));
30   }
31 }
```

- Με auto-fit δημιουργεί τόσο columns όσα χρειάζονται για να κάνει fit όλη τη γραμμή
- Ο ελάχιστος container είναι 250px αλλά μπορεί να γίνει expand (ως 1fr ώστε να καλυφθεί όλο το πλάτος της γραμμής) αν δεν χωράει και άλλο column



E-shop (3)

Προγραμματισμός στο Web



- Το αποτέλεσμα είναι ένα responsive grid



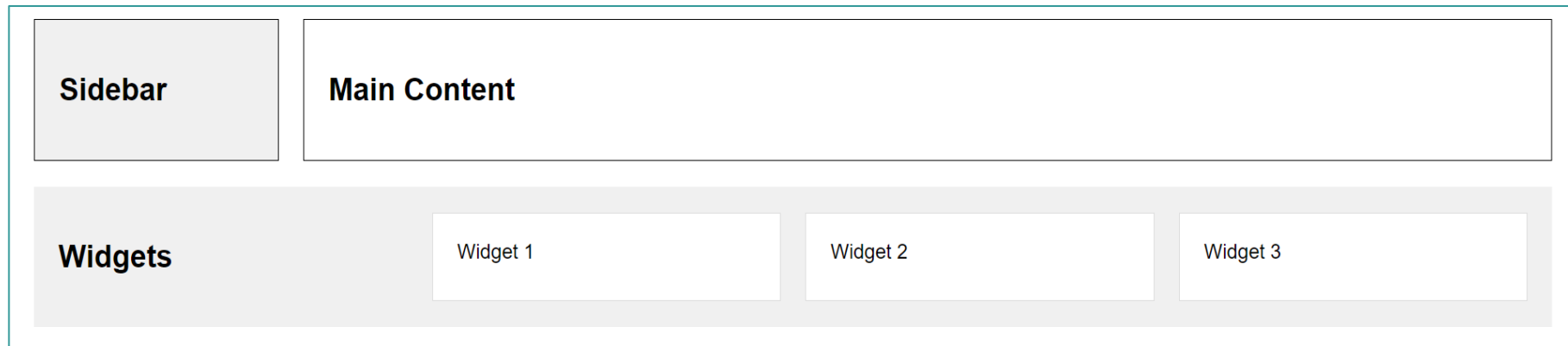
Εργασία (1)

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Dashboard Layout</title>
7   <link rel="stylesheet" href="./css//dashboard.css">
8 </head>
9 <body>
10  <div class="dashboard">
11    <div class="sidebar">
12      <h2>Sidebar</h2>
13      <!-- Sidebar content goes here -->
14    </div>
15    <div class="main-content">
16      <h2>Main Content</h2>
17      <!-- Main content goes here -->
18    </div>
19    <div class="widgets">
20      <h2>Widgets</h2>
21      <div class="widget">Widget 1</div>
22      <div class="widget">Widget 2</div>
23      <div class="widget">Widget 3</div>
24      <!-- Add more widgets as needed -->
25    </div>
26  </div>
27 </body>
28 </html>
```

- Έστω το
template
αριστερά



Εργασία (2)



- Με `display grid` δημιουργήστε το παραπάνω wireframe