

ΜΕΡΟΣ 3 – Reusable Components

Κανόνες που μπορώ να χρησιμοποιήσω ξανά και ξανά

Αμέσως μετά την ενότητα **HERO** ακολουθούν κι άλλες ενότητες, με όλες αυτές να μοιράζονται κάποια κοινά χαρακτηριστικά. Παρατηρείστε ότι το κουτί που τις φιλοξενεί απλώνει από το ένα άκρο του παραθύρου έως το άλλο, το περιεχόμενο όμως κάθεται στο κέντρο του κουτιού καταλαμβάνοντας μια έκταση σταθερού πλάτους κι αυτό για όλες τις ενότητες. Επίσης το περιεχόμενο είναι στημένο σε στήλες δύο ή περισσότερες.

Για να δώσω αυτή τη μορφή σε κάθε ενότητα θα πρέπει για κάθε μια απ' αυτές να επαναλάβω την ίδια διαδικασία που ακολούθησα για να στήσω αφ' ενός τα περιεχόμενα της hero section σε στήλες και αφ' ετέρου να τα τοποθετήσω στο κέντρο της σελίδας σε μια περιοχή σταθερού πλάτους.

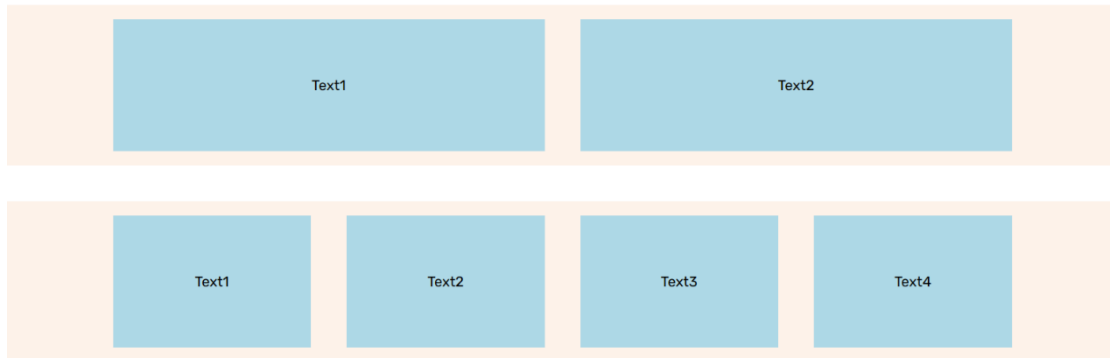
Η διαδικασία είναι η εξής:

1. Ξεκινώ εισάγοντας στον κώδικα της σελίδας το κουτί που θα φιλοξενήσει τα περιεχόμενα της ενότητας δηλ. την διπλή ετικέτα `<section></section>`. Το κουτί αυτό απλώνει από το ένα άκρο της οθόνης μέχρι το άλλο.
2. Μέσα στην διπλή ετικέτα `<section></section>` ρίχνω τη διπλή ετικέτα `<div></div>` η οποία δημιουργεί ένα νέο κουτί μέσα στο προηγούμενο. Μαζεύω το πλάτος του κουτιού και το στοιχίζω στο κέντρο. Μέσα σ' αυτή ρίχνω τα περιεχόμενα της ενότητας. Επειδή το κουτί της **div** που τα φιλοξενεί έχει συγκεκριμένο πλάτος και κάθεται στο κέντρο της σελίδας, έτσι τα περιεχόμενα θα απλώνουν στο πλάτος του κουτιού και θα είναι στοιχισμένα στο κέντρο.
3. Αν θέλω να χωρίσω τα περιεχόμενα σε στήλες, τότε μέσα στο προηγούμενο κουτί θα πρέπει να εισάγω τόσες **div** όσες οι στήλες και σε κάθε **div** να βάλω το περιεχόμενο της αντίστοιχης στήλης. Στην πρώτη **div** της πρώτης στήλης, στην δεύτερη **div** της δεύτερης κ.ο.κ. Στη συνέχεια εφαρμόζοντας τις ιδιότητες του **grid** στην **εξωτερική div** στήνω τις **εσωτερικές div** μαζί με τα περιεχόμενά τους σε στήλες.

Αυτή η διαδικασία πρέπει να επαναλαμβάνεται για κάθε ενότητα.

Ας τη δούμε ξανά σε ένα παράδειγμα.

Έστω ότι στη συνέχεια της σελίδας μας θέλουμε να δημιουργήσουμε τις επόμενες δύο ενότητες. Στην πρώτη ενότητα το περιεχόμενο θα πρέπει να μοιράζεται σε **2 στήλες** ενώ στην δεύτερη ενότητα σε **4**. Επίσης τα περιεχόμενα των ενότητων δεν θέλω να απλώνουν από το ένα άκρο της οθόνης έως το άλλο, αντίθετα θέλω να είναι περιορισμένα στο κέντρο της σελίδας και να καταλαμβάνουν μια έκταση πλάτους **1200px**.



Στο αρχείο **index.html** και μετά το **</section>** εκεί δηλαδή που τελειώνει η ενότητα **hero**, τοποθετώ δύο νέες διπλές ετικέτες **<section></section>** όπως φαίνεται πιο κάτω, μία για κάθε ενότητα. Στην πρώτη απ' αυτές δίνω το όνομα κλάσης **section-2** και στην δεύτερη το όνομα κλάσης **section-3**. Αυτές δημιουργούν μέσα στη σελίδα δύο νέα κουτιά τα οποία θα φιλοξενήσουν τα στοιχεία των δύο νέων ενότητων. Τα κουτιά αυτά απλώνουν από το ένα άκρο της σελίδας έως το άλλο.

```
index.html

<main>
  <section class="section-hero">
    .....
    .....
  </section>

  <section class="section-2">
  </section>

  <section class="section-3">
  </section>
</main>
</body>
</html>
```

Το περιεχόμενο της κάθε ενότητας δεν θέλω να απλώνει σε όλο το εύρος του κουτιού της **section** αντίθετα θέλω να κάθεσαι στο κέντρο του κουτιού και να καταλαμβάνει μια έκταση σταθερού πλάτους π.χ. **1200px**. Για να το πετύχω αυτό εισάγω μέσα σε κάθε διπλή ετικέτα **<section></section>** μια διπλή ετικέτα **<div></div>** όπως φαίνεται πιο κάτω, στην οποία δίνω το όνομα κλάσης **container**.

```
index.html

<section class="section-2">
  <div class="container">

  </div>
</section>

<section class="section-3">
  <div class="container">

  </div>
</section>
```

Η ετικέτα αυτή δημιουργεί ένα νέο κουτί μέσα στο κουτί της κάθε ενότητας, το οποίο καταλαμβάνει όλο το πλάτος του κουτιού της **section**. Με έναν κανόνα CSS στοχεύω επάνω στο κουτί **container** και μαζεύω το πλάτος του στα **1200px**. Επίσης το στοιχίζω στο κέντρο του κουτιού της **section**.

```
style.css

.container {
  max-width: 120rem;
  margin: 0 auto;
}
```

Με αυτό τον τρόπο ότι περιεχόμενο ρίξω μέσα στην **container div** θα απλώνει μέχρι τα **1200px** και θα εμφανίζεται κεντραρισμένο μέσα στο κουτί της ενότητας.

Αν τώρα το περιεχόμενο της ενότητας θέλω να είναι χωρισμένο σε στήλες, τότε μέσα στην **container div** θα πρέπει να τοποθετήσω τόσες διπλές ετικέτες **<div></div>** όσες και οι στήλες που θέλω. Στη συνέχεια θα γεμίσω κάθε μία div με το περιεχόμενο της αντίστοιχης στήλης. Την πρώτη **div** με το περιεχόμενο της πρώτης στήλης, τη δεύτερη με το περιεχόμενο της δεύτερης κοκ.

```

<section class="section-2">
  <div class="container">
    <div>Text1</div>
    <div>Text2</div>
  </div>
</section>

<section class="section-3">
  <div class="container">
    <div>Text1</div>
    <div>Text2</div>
    <div>Text3</div>
    <div>Text4</div>
  </div>
</section>

```

Το περιεχόμενο της ενότητας **section-2** θέλω να μοιράζεται σε δύο στήλες. Έτσι μέσα στην **container div** αυτής της ενότητας τοποθετώ τις δύο διπλές ετικέτες **div** που φαίνονται πιο πάνω. Μέσα στην ενότητα **section-3** χρειαζομαι τέσσερις στήλες. Έτσι μέσα στην **container div** αυτής της ενότητας τοποθετώ τις τέσσερις διπλές ετικέτες **div** που φαίνονται πιο πάνω. Μέσα σε κάθε **div** τοποθετώ υποδειγματικό περιεχόμενο τη φράση **Text** ακολουθούμενη από το νούμερο της στήλης. Κάθε μια απ' αυτές τις **div** με τη βοήθεια των CSS θα διαμορφωθεί στη συνέχεια σε στήλη.

Στο αρχείο **style.css** δημιουργώ νέο κανόνα μορφοποίησης που στοχεύει στα κουτιά των ενότητων **section-2** και **section-3** και δίνει στο φόντο τους χρώμα **#fdf2e9**, και απόσταση πάνω και κάτω από τα κουτιά **50px (5rem)**.

Επίσης στον κανόνα που πρόσθεσα πιο πριν και ο οποίος στοχεύει και μορφοποιεί όλα τα κουτιά που φέρουν την κλάση **container**, συμπληρώνω την ιδιότητα **background-color** με τιμή **#2E99BC** προκειμένου να δώσω ένα διαφορετικό χρώμα στο φόντο της **container div**. Μ' αυτό τον τρόπο θα κάνω το κουτί της **container div** να ξεχωρίσει από το κουτί της **section** και έτσι θα μπορέσω να αντιληφθώ το μέγεθός του.

```

.section-2, .section-3{
  background-color: #fdf2e9;
  margin: 5rem 0;
}

```

```
.container {
  background-color: #2E99BC;
  max-width: 120rem;
  margin: 0 auto;
}
```

Σώζω και βλέπω τις αλλαγές. Βλέπω τα κουτιά **section** σε μπεζ χρώμα να απλώνουν σε όλο το πλάτος της σελίδας, ενώ τα κουτιά **container** σε σκούρο μπλε να έχουν πλάτος **1200px** και να είναι στοιχισμένα στο κέντρο του μπεζ κουτιού (του κουτιού **section**).



Για να κόψω στήλες μέσα στα κουτιά **container** θα πρέπει να στοχεύσω σ' αυτά και αφενός να αλλάξω την τιμή της ιδιότητας **display** αυτών σε **grid**, αφ' ετέρου να προσθέσω την ιδιότητα **grid-template-columns** με τιμή το πλήθος των στηλών.

Ένας τρόπος να το πετύχω αυτό είναι να δώσω ένα ξεχωριστό όνομα κλάσης σε κάθε διαφορετικό κουτί container π.χ. container2, container3 κοκ. Στη συνέχεια να στοχεύσω στο αντίστοιχο κουτί με το όνομα και να ρυθμίσω το πλάτος του, να το φέρω στο κέντρο της σελίδας και αν χρειάζεται να κόψω τις απαραίτητες στήλες.

Μ' αυτό τον τρόπο όμως θα έχω πολλούς κανόνες με ίδιες μορφοποιήσεις, όπως π.χ. η ρύθμιση του πλάτους των κουτιών στα 1200px και η στοίχισή τους στο κέντρο της σελίδας.

Ένας καλύτερος τρόπος είναι ο εξής:

Φτιάχνω έναν κανόνα με όλες τις κοινές μορφοποιήσεις - ρύθμιση του πλάτους των κουτιών και στοίχισή τους στο κέντρο της σελίδας - και τον συνδυάζω με το όνομα κλάσης **container**. Πρόκειται για τον κανόνα που έχω γράψει ήδη και φαίνεται πιο κάτω.

```
style.css

.container {
  background-color: #2E99BC;
  max-width: 120rem;
  margin: 0 auto;
}
```

Συνδυάζω το όνομα κλάσης **container** με κάθε **div** (εντός της section) που φιλοξενεί το περιεχόμενο της ενότητας (όπως έχω κάνει ήδη).

```
<section class="section-2">
  <div class="container">
    <div>Text1</div>
    <div>Text2</div>
  </div>
</section>

<section class="section-3">
  <div class="container">
    <div>Text1</div>
    <div>Text2</div>
    <div>Text3</div>
    <div>Text4</div>
  </div>
</section>
```

Επιστρέφω στο αρχείο **style.css** και προσθέτω

- νέο κανόνα που στοχεύει σε κάθε στοιχείο που φέρει το όνομα κλάσης **grid-2-cols** ο οποίος δημιουργεί μέσα στο στοιχείο **2 στήλες**
- νέο κανόνα που στοχεύει σε κάθε στοιχείο που φέρει το όνομα κλάσης **grid-3-cols** ο οποίος δημιουργεί μέσα στο στοιχείο **3 στήλες**
- νέο κανόνα που στοχεύει σε κάθε στοιχείο που φέρει το όνομα κλάσης **grid-4-cols** ο οποίος δημιουργεί μέσα στο στοιχείο **4 στήλες** κ.ο.κ

```
.container {
  background-color: #2E99BC;
  max-width: 120rem;
  margin: 0 auto;
}

.grid-2-cols {
  display: grid;
  grid-template-columns: 1fr 1fr;
  gap: 9.6rem;
}
```

```

.grid-3-cols {
  display: grid;
  grid-template-columns: 1fr 1fr 1fr;
  gap: 9.6rem;
}

.grid-4-cols {
  display: grid;
  grid-template-columns: 1fr 1fr 1fr 1fr;
  gap: 9.6rem;
}

```

Ανάλογα με το πλήθος των στηλών που χρειαζόμαστε μέσα στο κουτί συνδυάζω μ' αυτό και το σχετικό όνομα κλάσης όπως φαίνεται πιο κάτω.

index.html
<pre> <section class="section-2"> <div class="container grid-2-cols"> <div>Text1</div> <div>Text2</div> </div> </section> <section class="section-3"> <div class="container grid-4-cols"> <div>Text1</div> <div>Text2</div> <div>Text3</div> <div>Text4</div> </div> </section> </pre>

Έτσι στην ενότητα **section-2** το περιεχόμενο θα κατανεμηθεί σε **2 στήλες**, ενώ στην ενότητα **section-3** θα κατανεμηθεί σε **4 στήλες**.

Προσέξτε ότι μέσω της ιδιότητας **gap** προσθέτω μια απόσταση μεταξύ των στηλών που θα δημιουργηθούν της τάξης των **96px**. Η απόσταση αυτή καλό είναι να είναι κοινή (η ίδια) μεταξύ όλων των στηλών που θα δημιουργήσω μέσα στη σελίδα μου.

Για να δω τα αποτελέσματα στο αρχείο **style.css** δίνω όγκο στις στήλες μέσω του επόμενου κανόνα.

```
style.css
.section-2 div div, .section-3 div div {
    font-size: 20px;
    background-color: lightblue;
    padding: 80px 50px;
    text-align: center;
}
```

Ο επιλογέας στοχεύει και μορφοποιεί όλες τις ετικέτες **div** που βρίσκονται μέσα σε μια άλλη **div** εντός του στοιχείου **section-2**. Επίσης στοχεύει και μορφοποιεί όλες τις ετικέτες **div** που βρίσκονται μέσα σε μια άλλη **div** εντός του στοιχείου **section-3**.

Ακολουθεί η τελική μορφή του **style.css** όπου έχει προστεθεί ένα **padding 20px** στις **sections** για να μπορέσω να διακρίνω πιο εύκολα τα περιεχόμενα που βρίσκονται σε στήλες. Τέλος αφαιρώ το χρώμα φόντου από τον κανόνα **container**.

```
style.css
.section-2, .section-3{
    background-color: #fdf2e9;
    margin: 5rem 0;
    padding: 20px;
}

.container {
    width: 120rem;
    margin: 0 auto;
}

.grid-2-cols {
    display: grid;
    grid-template-columns: 1fr 1fr;
    gap: 9.6rem;
}

.grid-3-cols {
    display: grid;
    grid-template-columns: 1fr 1fr 1fr;
    gap: 9.6rem;
}
```

```
.grid-4-cols {  
    display: grid;  
    grid-template-columns: 1fr 1fr 1fr 1fr;  
    gap: 9.6rem;  
}  
  
.section-2 div div, .section-3 div div {  
    font-size: 20px;  
    background-color: lightblue;  
    padding: 80px 50px;  
    text-align: center;  
}
```

Παρατηρείστε ότι οι κανόνες **.grid-2-cols**, **.grid-3-cols**, **.grid-4-cols** μοιράζονται κοινές μορφοποιήσεις. Ένας καλύτερος τρόπος είναι να απομακρύνω αυτές τις μορφοποιήσεις από τους πιο πάνω κανόνες και να τις συνδυάσω με έναν νέο κανόνα όπως φαίνεται πιο κάτω. Μπορώ να τον συνδυάσω με το όνομα κλάσης **grid**.

```
style.css  
  
.grid {  
    display: grid;  
    gap: 9.6rem;  
}  
  
.grid-2-cols {  
    grid-template-columns: 1fr 1fr;  
}  
  
.grid-3-cols {  
    grid-template-columns: 1fr 1fr 1fr;  
}  
  
.grid-4-cols {  
    grid-template-columns: 1fr 1fr 1fr 1fr;  
}
```

Επιστρέφοντας στο **index.html** για να χωρίσω το περιεχόμενο της πρώτης **div** σε 2 στήλες θα πρέπει να τη συνδυάσω και με τις δύο κλάσεις **grid** και **grid-2-cols**. Για να χωρίσω το περιεχόμενο της επόμενης **div** σε 4 στήλες θα πρέπει να τη συνδυάσω και με τις δύο κλάσεις **grid** και **grid-4-cols** όπως φαίνεται πιο κάτω.

```
index.html

<section class="section-2">
  <div class="container grid grid-2-cols">
    <div>Text1</div>
    <div>Text2</div>
  </div>
</section>

<section class="section-3">
  <div class="container grid grid-4-cols">
    <div>Text1</div>
    <div>Text2</div>
    <div>Text3</div>
    <div>Text4</div>
  </div>
</section>
```

Ακολουθεί η τελική μορφή του **style.css**

```
style.css

.section-2, .section-3{
  background-color: #fdf2e9;
  margin: 5rem 0;
  padding: 20px;
}

.container {
  max-width: 120rem;
  margin: 0 auto;
}
```

```

.grid {
  display: grid;
  gap: 9.6rem;
}

.grid-2-cols {
  grid-template-columns: 1fr 1fr;
}

.grid-3-cols {
  grid-template-columns: 1fr 1fr 1fr;
}

.grid-4-cols {
  grid-template-columns: 1fr 1fr 1fr 1fr;
}

.section-2 div div, .section-3 div div {
  font-size: 20px;
  background-color: lightblue;
  padding: 80px 50px;
  text-align: center;
}

```

Αφού σώσω και δω το τελικό αποτέλεσμα, διαγράψω μέσα από το αρχείο **index.html** τους κώδικες των δύο ενότητων

```

index.html

<section class="section-2">
  <div class="container grid grid-2-cols">
    <div>Text1</div>
    <div>Text2</div>
  </div>
</section>

<section class="section-3">
  <div class="container grid grid-4-cols">
    <div>Text1</div>
    <div>Text2</div>

```

```

</div>Text3</div>
</div>Text4</div>
</div>
</section>

```

Και μέσα από το style.css τον πρώτο και τελευταίο κανόνα

```

style.css

.section-2, .section-3 {
    background-color: #fdf2e9;
    margin: 5rem 0;
    padding: 20px;
}

.container {
    max-width: 120rem;
    margin: 0 auto;
}

.grid {
    display: grid;
    gap: 9.6rem;
}

.grid-2-cols {
    grid-template-columns: 1fr 1fr;
}

.grid-3-cols {
    grid-template-columns: 1fr 1fr 1fr;
}

.grid-4-cols {
    grid-template-columns: 1fr 1fr 1fr 1fr;
}

.section-2 div div, .section-3 div div {
    font-size: 20px;
    background-color: lightblue;
    padding: 80px 50px;
    text-align: center;
}

```

Τοποθετώ τους κανόνες που απομένουν κάτω από τον τίτλο **REUSABLE COMPONENTS** (επαναχρησιμοποιούμενα στοιχεία) όπως φαίνεται πιο κάτω – μιας και πρόκειται για κανόνες που θα τους χρησιμοποιήσω για να διαμορφώσω τη διάταξη κάθε μιας από τις ενότητες που θα προσθέσω στη συνέχεια.

```
style.css

/* REUSABLE COMPONENTS */

.container {
    max-width: 120rem;
    margin: 0 auto;
}

.grid {
    display: grid;
    gap: 9.6rem;
}

.grid-2-cols {
    grid-template-columns: 1fr 1fr;
}

.grid-3-cols {
    grid-template-columns: 1fr 1fr 1fr;
}

.grid-4-cols {
    grid-template-columns: 1fr 1fr 1fr 1fr;
}
```

Τέλος στον κανόνα που στοχεύει στη κλάση **grid** θα αντικαταστήσω την ιδιότητα

```
gap: 9.6rem;
```

η οποία ορίζει την απόσταση μεταξύ των στηλών του πλέγματος στα 96px και μεταξύ των γραμμών πάλι στα 96px, με τις πιο κάτω που δίνουν μικρότερη απόσταση μεταξύ των στηλών του πλέγματος 64px και λίγο πιο μεγάλη 96px μεταξύ των γραμμών.

```
column-gap: 6.4rem;
row-gap: 9.6rem;
```

Έτσι οι κανόνες διαμορφώνονται ως εξής:

```
style.css

/* REUSABLE COMPONENTS */

.container {
    max-width: 120rem;
    margin: 0 auto;
}

.grid {
    display: grid;
    column-gap: 6.4rem;
    row-gap: 9.6rem;
}

.grid-2-cols {
    grid-template-columns: 1fr 1fr;
}

.grid-3-cols {
    grid-template-columns: 1fr 1fr 1fr;
}

.grid-4-cols {
    grid-template-columns: 1fr 1fr 1fr 1fr;
}
```