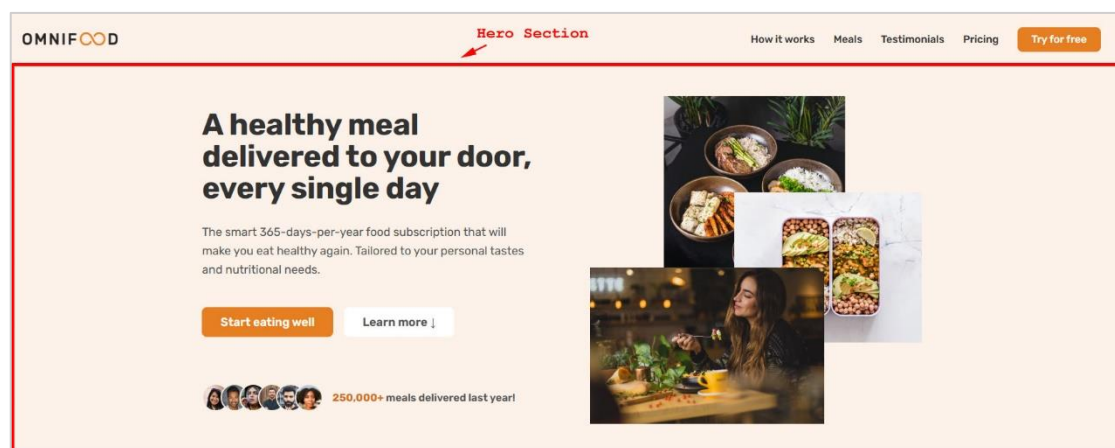


α ΜΕΡΟΣ 2 – HERO SECTION

Hero section είναι η ενότητα που εμφανίζεται στην οθόνη αμέσως μετά την κεφαλίδα δηλαδή αμέσως μετά το λογότυπο και το μενού. Αποτελεί το πρώτο πράγμα που βλέπουν οι επισκέπτες του ιστοτόπου.

Η περιοχή αυτή περιλαμβάνει συνήθως σύντομες και περιεκτικές πληροφορίες σχετικά με το τι προσφέρει ο δικτυακός σας τόπος, για ποιο λόγο οι χρήστες θα πρέπει να σας εμπιστευτούν καθώς και κάποιες ενέργειες που θα μπορούσαν να κάνουν στη συνέχεια.

Οι πληροφορίες αυτές στήνονται συνήθως με τη βοήθεια μιας επικεφαλίδας που τραβά την προσοχή ακολουθούμενης από το κατάλληλο κείμενο, μια ελκυστική εικόνα ή βίντεο σχετικά με το θέμα. Τα λεγόμενα **Call To Action buttons** (κουμπιά ενεργειών) καλό είναι να συμπεριλαμβάνονται.



ΚΑΤΑΣΚΕΥΗ ΤΗΣ ΕΝΟΤΗΤΑΣ

Μέσα στο αρχείο **Index.html** και ακριβώς κάτω από τη διπλή ετικέτα `<header></header>` τοποθετώ τη διπλή ετικέτα `<main></main>`. Αυτή συμπεριφέρεται ακριβώς όπως και η `<div></div>` - είναι block-level element - και παράγει μέσα στη σελίδα ένα κουτί το οποίο απλώνει σε όλο το πλάτος της **body**. Μέσα σ' αυτό το κουτί ρίχνω πάντα το κυρίως περιεχόμενο της σελίδας μου.

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Omnifood</title>
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link href="https://fonts.googleapis.com/css2?family=Rubik:wght@400;500;600;700&
    display=swap" rel="stylesheet">
  <link href="style.css" rel="stylesheet">
</head>
<body>
  <header>
    
    <nav>
      <ul>
        <li><a href="#">How it works</a></li>
        <li><a href="#">Meals</a></li>
        <li><a href="#">Testimonials</a></li>
        <li><a href="#">Pricing</a></li>
        <li><a href="#">Try for free</a></li>
      </ul>
    </nav>
  </header>
  <main>

  </main>
</body>
</html>

```

Στο εσωτερικό του κουτιού αυτού τοποθετώ τη διπλή ετικέτα `<section></section>`. Αυτή συμπεριφέρεται ακριβώς όπως και η `<div></div>` - είναι block-level element - και παράγει μέσα στη σελίδα ένα κουτί το οποίο απλώνει σε όλο το πλάτος της **main**. Επειδή το πλάτος της **main** είναι ίδιο με το πλάτος της **body** το κουτί της **section** απλώνει σε όλο το πλάτος της **body**. Μέσα σ' αυτό το κουτί θα ρίξω το περιεχόμενο της ενότητας **hero**.

Το περιεχόμενο της ιστοσελίδας αποτελείται από **πολλές ενότητες**. Για κάθε μία από αυτές τις ενότητες θα χρησιμοποιώ και ένα διαφορετικό ζευγάρι ετικετών `<section></section>`. Σε

κάθε ένα από αυτά τα ζευγάρια θα δίνω και ένα διαφορετικό όνομα με τη βοήθεια της παραμέτρου **class**. Ονομάζω την πρώτη ενότητα **section-hero**.

```
index.html

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Omnifood</title>
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link href="https://fonts.googleapis.com/css2?family=Rubik:wght@400;500;600;700&
    display=swap" rel="stylesheet">
  <link href="style.css" rel="stylesheet">
</head>
<body>
  <header>
    
    <nav>
      <ul>
        <li><a href="#">How it works</a></li>
        <li><a href="#">Meals</a></li>
        <li><a href="#">Testimonials</a></li>
        <li><a href="#">Pricing</a></li>
        <li><a href="#">Try for free</a></li>
      </ul>
    </nav>
  </header>
  <main>
    <section class="section-hero">

    </section>
  </main>
</body>
</html>
```

Μέσα στο κουτί αυτό θα ρίξω το περιεχόμενο της ενότητας **hero**.

ΠΡΟΣΘΗΚΗ ΠΕΡΙΕΧΟΜΕΝΩΝ

Παρατηρώ ότι το περιεχόμενο είναι διατεταγμένο σε δύο στήλες. Στην αριστερή στήλη περιέχονται τα κείμενα στη δεξιά στήλη βρίσκεται η εικόνα. Όταν δουλεύω με στήλες μια καλή πρακτική είναι να κλείνω το περιεχόμενο κάθε στήλης σε μια ξεχωριστή ετικέτα `<div></div>`.

Στο εσωτερικό της διπλής ετικέτας **section** τοποθετώ δύο ζεύγη ετικετών `<div></div>`. Στην πρώτη δίνω το όνομα **hero-text-box** στη δεύτερη το όνομα **hero-img-box**. Μέσα στην πρώτη **div** εισάγω τα κείμενα (μια επικεφαλίδα επιπέδου 1, μία παράγραφο και τα κουμπιά) που θέλω να φαίνονται στην αριστερή στήλη και μέσα στην δεύτερη **div** την εικόνα. Ακολουθούν οι σχετικοί κώδικες.

```
index.html

<main>
  <section class="section-hero">
    <div class="hero-text-box">
      <h1> A healthy meal delivered to your door, every single day </h1>
      <p>
        The smart 365-days-per-year food subscription that will make you
        eat healthy again. Tailored to your personal tastes and
        nutritional needs.
      </p>
      <a href="#">Start eating well</a>
      <a href="#">Learn more &darr;</a>
    </div>
    <div class="hero-img-box">
      
    </div>
  </section>
</main>
```

Προσέξτε ότι μέσα στην πρώτη **div** και ακριβώς κάτω απ' την παράγραφο, στη θέση των δύο κουμπιών, τοποθετήθηκαν δύο σύνδεσμοι που φέρουν το κείμενο των κουμπιών. Στην συνέχεια και με τη βοήθεια των κανόνων CSS θα τους δώσουμε τη μορφή κουμπιών.

Τέλος **καλό είναι να αποδίδω ονόματα κλάσης** στα διάφορα **στοιχεία της σελίδας** ώστε να μπορώ να στοχεύσω άμεσα σ' αυτά για να τα μορφοποιήσω, δίχως να χρειάζεται να χρησιμοποιώ σύνθετους επιλογείς. Σ' αυτό το σκεπτικό θα δώσω στην ετικέτα **h1** το όνομα κλάσης **heading-primary**, στην παράγραφο που ακολουθεί το όνομα κλάσης **hero-description** και στις ετικέτες **a** το όνομα κλάσης **btn** και στην εικόνα το όνομα κλάσης **hero-img**, όπως φαίνεται στη συνέχεια.

```
<main>
  <section class="section-hero">
    <div class="hero-text-box">
      <h1 class="heading-primary">
        A healthy meal delivered to your door, every single day
      </h1>
      <p class="hero-description">
        The smart 365-days-per-year food subscription that will make you
        eat healthy again. Tailored to your personal tastes and
        nutritional needs.
      </p>
      <a href="#" class="btn">Start eating well</a>
      <a href="#" class="btn">Learn more &darr;</a>
    </div>
    <div class="hero-img-box">
      
    </div>
  </section>
</main>
```

ΔΗΜΙΟΥΡΓΙΑ ΣΤΗΛΩΝ ΜΕ ΧΡΗΣΗ ΤΟΥ GRID

Παρατηρώ ότι το περιεχόμενο είναι διατεταγμένο σε **2 στήλες του ίδιου πλάτους**. Στην αριστερή στήλη περιέχονται τα κείμενα στη δεξιά στήλη βρίσκεται η εικόνα. Για να κόψω στήλες μέσα στο κουτί **section-hero** στοχεύω στο κουτί με τη βοήθεια ενός κανόνα και αλλάζω την προεπιλεγμένη τιμή της ιδιότητας **display** αυτού από **block** σε **grid** (πλέγμα). Στη συνέχεια χρησιμοποιώντας την ιδιότητα **grid-template-columns** με τιμή **1fr 1fr** δηλώνοντας ότι θέλω **2 στήλες** που η κάθε μία να καταλαμβάνει το ίδιο ποσοστό επί του συνολικού πλάτους του κουτιού. Άρα ζητώ να κόψει 2 στήλες του ίδιου πλάτους μέσα στο κουτί (**fr** από τη λέξη **fraction**).



Επειδή το στοιχείο **section-hero** φιλοξενεί στο εσωτερικό του **δύο στοιχεία** (child elements) - τις **δύο div** - τότε αυτές θα τοποθετηθούν στις δύο στήλες που δημιουργήθηκαν. Το πρώτο κατά σειρά στοιχείο (η πρώτη div με ότι αυτή περιέχει) στην πρώτη στήλη και το επόμενο στοιχείο (η δεύτερη div) στη δεύτερη στήλη.

Στο τέλος του αρχείου **style.css** προσθέτω τον επόμενο κανόνα:

```
style.css

/* HERO SECTION */

.section-hero {
  display: grid;
  grid-template-columns: 1fr 1fr;
}
```

Σώζω και βλέπω τις αλλαγές.

Η εικόνα είναι τεράστια και δεν αφήνει τις δύο στήλες να πάρουν ίδιο πλάτος. Για να το πετύχω αυτό στοχεύω στην εικόνα με τον επόμενο κανόνα:

```
style.css

/* HERO SECTION */

.section-hero {
  display: grid;
  grid-template-columns: 1fr 1fr;
}

.hero-img {
  width: 100%;
}
```

Σώζω και βλέπω τις αλλαγές. Η στήλη έρχεται στο 50% του πλάτους της section και η εικόνα καταλαμβάνει το 100% του πλάτους της στήλης.

Συνεχίζω προσθέτοντας στο κουτί της **section-hero** χρώμα φόντου **#fdf2e9**

```
/* HERO SECTION */

.section-hero {
  display: grid;
  grid-template-columns: 1fr 1fr;
  background-color: #fdf2e9;
}

.hero-img {
  width: 100%;
}
```

Σώζω και βλέπω τις αλλαγές.

Δημιουργώ νέο κανόνα που στοχεύει στην ετικέτα **h1** και επιβάλλει τις ακόλουθες μορφοποιήσεις.

- Μέγεθος γραμμάτων **52px** δηλ. **5.2rem**
- Ένταση γραμμάτων στο **700**
- Απόσταση γραμμών μεταξύ τους **1.05**
- Χρώμα γραμμάτων **#333**
- Απόσταση χαρακτήρων **-0.5px** προκειμένου να έρθουν πιο κοντά
- Απόσταση επικεφαλίδας από το στοιχείο που βρίσκεται ακριβώς από κάτω **32px** δηλ. **3.2rem**

```
.heading-primary {  
    font-size: 5.2rem;  
    font-weight: 700;  
    line-height: 1.05;  
    color: #333;  
    letter-spacing: -0.5px;  
    margin-bottom: 3.2rem;  
}
```

Προσέξτε ότι στοχεύω μέσω της κλάσης **heading-primary** που έδωσα προηγουμένως. Σώζω και βλέπω τις αλλαγές.

Δημιουργώ νέο κανόνα που στοχεύει στην ετικέτα **p** και επιβάλλει τις ακόλουθες μορφοποιήσεις

- Μέγεθος γραμμάτων **20px** δηλ. **2rem**
- Απόσταση γραμμών μεταξύ τους **1.6**
- Απόσταση επικεφαλίδας από το στοιχείο που βρίσκεται ακριβώς από κάτω **48px** δηλ. **4.8rem**

```
.hero-description {  
    font-size: 2rem;  
    line-height: 1.6;  
    margin-bottom: 4.8rem;  
}
```

Προσέξτε ότι στοχεύω μέσω της κλάσης **hero-description** που έδωσα προηγουμένως. Σώζω και βλέπω τις αλλαγές.

ΜΟΡΦΟΠΟΙΗΣΗ ΤΩΝ ΚΟΥΜΠΙΩΝ

Δημιουργώ νέο κανόνα που στοχεύει στα κουμπιά που έχουν τη κλάση **btn** και επιβάλλει τις πιο κάτω μορφοποιήσεις. Προσέξτε ότι χρησιμοποιώ ένα επιλογέα που στοχεύει και στις δύο καταστάσεις του κουμπιού (κατάσταση **link** & κατάσταση **visited**) και τους δίνει την ίδια μορφοποίηση. Στην κατάσταση **link** βρίσκεται ένα κουμπί όσο ο χρήστης δεν το επισκέπτεται. Στην κατάσταση **visited** περνά το κουμπί αφού ο χρήστης το επισκεφθεί. Για να δώσω την ίδια μορφοποίηση και στις δύο καταστάσεις χρησιμοποιώ τον σύνθετο επιλογέα:

.btn:link, .btn:visited

Ακολουθούν οι μορφοποιήσεις:

- Μέγεθος γραμμάτων 20px (**2rem**)
- Απομάκρυνση της υπογράμμισης
- Χρώμα φόντου **#e67e22**
- Χρώμα γραμμάτων άσπρο **#fff**
- Αλλαγή της εμφάνισης του στοιχείου από inline σε **inline-block** (για να μπορέσουν να επιβληθούν σωστά οι εσωτερικές αποστάσεις – **padding**s που θα ορίσω στη συνέχεια)
- Απόσταση κειμένου από την πάνω και κάτω πλευρά 16px (**1.6rem**) και από την αριστερή και δεξιά πλευρά 32px (**3.2rem**)
- Στρογγυλεμένες άκρες στα **9px**
- Ένταση γραμμάτων **600**

style.css

```
.btn:link, .btn:visited {  
    font-size: 2rem;  
    text-decoration: none;  
    background-color: #e67e22;  
    color: #fff;  
    display: inline-block;  
    padding: 1.6rem 3.2rem;  
    border-radius: 9px;  
    font-weight: 600;  
}
```

Προσέξτε όμως ότι στην τελική σελίδα θέλω οι μορφοποιήσεις των δύο κουμπιών να είναι ίδιες εκτός από το χρώμα του φόντου και το χρώμα της γραμματοσειράς, που στο δεξί κουμπί θέλω να διαφοροποιούνται.

Ένας εύκολος τρόπος να το κάνω αυτό είναι ο εξής:

Αρχικά στον πιο πάνω κανόνα που είναι κοινός και για τα δύο κουμπιά κρατώ μόνο εκείνες τις μορφοποιήσεις που θέλω να είναι **κοινές** και για τα δύο κουμπιά και διαγράφω τις μορφοποιήσεις που διαφοροποιούνται από το ένα κουμπί στο άλλο.

Εδώ κρατώ όλες τις μορφοποιήσεις εκτός από την **background-color** που δίνει χρώμα στο φόντο του κουμπιού και την **color** που δίνει χρώμα στα γράμματα. Ο κανόνας παίρνει την πιο κάτω μορφή.

```
style.css

.btn:link, .btn:visited {
    font-size: 2rem;
    text-decoration: none;
    display: inline-block;
    padding: 1.6rem 3.2rem;
    border-radius: 9px;
    font-weight: 600;
}
```

Στη συνέχεια πηγαίνω στο αρχείο **index.html** , εντοπίζω τον κώδικα των κουμπιών

```
index.html

<a href="#" class="btn">Start eating well</a>
<a href="#" class="btn">Learn more &darr;</a>
```

και προσθέτω σε κάθε ένα απ' αυτά ένα **δεύτερο όνομα** μέσω της παραμέτρου **class**. Στο αριστερό κουμπί δίνω το όνομα **btn-left** στο δεξί το όνομα **btn-right** όπως φαίνεται πιο κάτω.

```
index.html

<a href="#" class="btn btn-left">Start eating well</a>
<a href="#" class="btn btn-right">Learn more &darr;</a>
```

Με αυτό τον τρόπο έχω τη δυνατότητα να στοχεύσω στο κάθε κουμπί με δύο διαφορετικούς κανόνες. Ο πρώτος θα εντοπίζει όλα τα κουμπιά που φέρουν το όνομα **btn** και θα τα μορφοποιεί με τον ίδιο τρόπο. Είναι αυτός που γράψαμε πιο πάνω που σχηματίζει το κουτί και μορφοποιεί τα γράμματα δίνοντάς τους συγκεκριμένο μέγεθος, ένταση και απομακρύνει την υπογράμμιση .

Στη συνέχεια νέος κανόνας θα στοχεύει μόνο στο κουμπί **btn-left** και θα προσθέτει τις ακόλουθες μορφοποιήσεις:

- Χρώμα φόντου **#e67e22**
- Χρώμα γραμμάτων άσπρο **#fff**
- Απόσταση από το δεξί κουμπί **16px (1.6rem)**

Επιστρέφω στο αρχείο **style.css** και κάτω από τον προηγούμενο κανόνα προσθέτω

```
style.css

.btn-left:link, .btn-left:visited {
    background-color: #e67e22;
    color: #fff;
    margin-right: 1.6rem;
}
```

Επίσης νέος κανόνας θα στοχεύει μόνο στο κουμπί **btn-right** και θα προσθέτει τις ακόλουθες μορφοποιήσεις:

- Χρώμα φόντου άσπρο **#fff**
- Χρώμα γραμμάτων **#555**

Στο αρχείο **style.css** και κάτω από τον προηγούμενο κανόνα προσθέτω

```
style.css

.btn-right:link, .btn-right:visited {
    background-color: #fff;
    color: #555;
}
```

Σώζω και βλέπω τις αλλαγές.

Όταν ο χρήστης περνά με το ποντίκι πάνω από τα κουμπιά ή όταν πατά επάνω τους με το ποντίκι θέλω να αλλάζουν τα χρώματα στο υπόβαθρο, στο δε δεξί κουμπί να εμφανίζεται ένα περίγραμμα.

Στο αρχείο **style.css** και κάτω από τον τελευταίο κανόνα δημιουργώ νέο που στοχεύει στις καταστάσεις **hover & active** του **αριστερού κουμπιού** και επιβάλλει την πιο κάτω μορφοποίηση.

```
style.css
.btn-left:hover, .btn-left:active {
    background-color: #cf711f;
}
```

Στο αρχείο **style.css** και κάτω από τον τελευταίο κανόνα δημιουργώ νέο που στοχεύει στις καταστάσεις **hover & active** του **δεξιού κουμπιού** και επιβάλλει την πιο κάτω μορφοποίηση.

```
style.css
.btn-right:hover, .btn-right:active {
    background-color: #fdf2e9;
    box-shadow: inset 0 0 0 3px #fff;
}
```

Σώζω και βλέπω τις αλλαγές.

Τέλος προσθέτοντας την ιδιότητα **transition** με τιμή **all 0.3s** στον γενικό κανόνα **.btn:link**, **.btn:visited** καταφέρνω να δώσω μια ομαλή μετάβαση στην κατάσταση **hover** των κουμπιών η οποία θα διαρκεί 0.3 sec (βλέπε πιο κάτω με κόκκινο χρώμα).

```
style.css
.btn:link, .btn:visited {
    font-size: 2rem;
    text-decoration: none;
    background-color: #e67e22;
    color: #fff;
    display: inline-block;
    padding: 1.6rem 3.2rem;
    border-radius: 9px;
    font-weight: 600;
    transition: all 0.3s;
}
```

ΜΑΖΕΥΟΝΤΑΣ ΤΟ ΠΛΑΤΟΣ ΤΩΝ ΠΕΡΙΕΧΟΜΕΝΩΝ

Παρατηρώ ότι τα περιεχόμενα απλώνουν σε όλο το εύρος της ενότητας **section**. Αντίθετα θέλω το πλάτος του περιεχομένου να είναι μικρότερο από το πλάτος της **section**, επίσης να είναι το ίδιο για τα περιεχόμενα όλων των ενοτήτων που απαρτίζουν τη σελίδα και τέλος να είναι κεντραρισμένο μέσα στο κουτί της **section**. Επίσης θέλω το πλάτος της περιοχής του κειμένου να είναι προσαρμοστικό όταν περνάω σε μικρότερες συσκευές (να μαζεύει ανάλογα το πλάτος της συσκευής μέσω της οποίας βλέπω τη σελίδα και να μη χρειάζεται να κάνω χρήση κάποιας μπάρας ολίσθησης για να το δω).

Για να το πετύχω αυτό χρησιμοποιώ το εξής τρικ.

Κλείνω όλο το περιεχόμενο της **section** σε μια διπλή ετικέτα **div** στην οποία δίνω ένα όνομα εδώ **hero** όπως φαίνεται πιο κάτω.

```
index.html

<main>
  <section class="section-hero">
    <div class="hero">
      <div class="hero-text-box">
        <h1 class="heading-primary">
          A healthy meal delivered to your door, every single day
        </h1>
        <p class="hero-description">
          The smart 365-days-per-year food subscription that will make you
          eat healthy again. Tailored to your personal tastes and
          nutritional needs.
        </p>
        <a href="#" class="btn">Start eating well</a>
        <a href="#" class="btn">Learn more &darr;</a>
      </div>
      <div class="hero-img-box">
        
      </div>
    </div>
  </section>
</main>
```

Επιστρέφω στο αρχείο **style.css** και από τον πιο κάτω κανόνα απομακρύνω τις δύο πρώτες γραμμές που δημιουργούν τις στήλες

```
style.css
/* HERO SECTION */
.section-hero {
    display: grid;
    grid-template-columns: 1fr 1fr;
    background-color: #fdf2e9;
}
```

και τις μεταφέρω σε έναν νέο κανόνα που δημιουργώ ακριβώς από κάτω και στοχεύει στην **div** με το όνομα κλάσης **hero**.

```
style.css
/* HERO SECTION */
.section-hero {
    background-color: #fdf2e9;
}

.hero {
    display: grid;
    grid-template-columns: 1fr 1fr;
}
```

Μ' αυτό τον τρόπο έχω δημιουργήσει ένα νέο κουτί μέσα στο κουτί της **section** με το όνομα **hero**. Το κουτί αυτό είναι τώρα χωρισμένο σε δύο στήλες. Μαζεύω το πλάτος του στα **1300px (130rem)** και το **στοιχίζω στο κέντρο**.

Επίσης προσθέτω σε όλη τη **section** απόσταση περιεχομένου από το πάνω μέρος του κουτιού (**padding-top**) **48px(4.8rem)** και από το κάτω μέρος του κουτιού (**padding-bottom**) **96px (9.6rem)**.

```
style.css
/* HERO SECTION */
.section-hero {
    background-color: #fdf2e9;
    padding: 4.8rem 0 9.6rem 0;
}
```

```
.hero {  
  max-width: 130rem;  
  margin: 0 auto;  
  display: grid;  
  grid-template-columns: 1fr 1fr;  
}
```

Προσέξτε ότι αντί της **width** γίνεται χρήση της **max-width**. Η ιδιότητα αυτή επιτρέπει στο κουτί να προσαρμόζει το πλάτος του στο πλάτος μικρότερων συσκευών όταν ο χρήστης περιηγείται στη σελίδα από μικρότερες συσκευές (με πλάτος κάτω από 1300px – 130rem).

Τέλος προσθέτω την ιδιότητα **align-items** του grid με τιμή **center** για να στοιχίσω στο κέντρο του κατακόρυφου άξονα τα στοιχεία των στηλών και την ιδιότητα **gap** με τιμή **96px (9.6rem)** με την οποία προσθέτω απόσταση **96px** μεταξύ των στηλών.

```
style.css  
  
.hero {  
  max-width: 130rem;  
  margin: 0 auto;  
  display: grid;  
  grid-template-columns: 1fr 1fr;  
  gap: 9.6rem;  
  align-items: center;  
}
```

Σώζω και βλέπω τις αλλαγές. Αν προσπαθήσω να μαζέψω το πλάτος της οθόνης θα δω ότι η σελίδα προσαρμόζεται.

Ακολουθούν οι μορφοποιήσεις της ενότητας HERO έως εδώ.

style.css

```
/* HERO SECTION */

.section-hero {
    background-color: #fdf2e9;
    padding: 4.8rem 0 9.6rem 0;
}

.hero {
    max-width: 1300px;
    margin: 0 auto;
    display: grid;
    grid-template-columns: 1fr 1fr;
    gap: 9.6rem;
    align-items: center;
}

.hero-img {
    width: 100%;
}

.heading-primary {
    font-size: 5.2rem;
    font-weight: 700;
    line-height: 1.05;
    color: #333;
    letter-spacing: -0.5px;
    margin-bottom: 3.2rem;
}

.hero-description {
    font-size: 2rem;
    line-height: 1.6;
    margin-bottom: 4.8rem;
}
```

```
.btn:link, .btn:visited {
    font-size: 2rem;
    text-decoration: none;
    display: inline-block;
    padding: 1.6rem 3.2rem;
    border-radius: 9px;
    font-weight: 600;
    transition: all 0.3s;
}

.btn-left:link, .btn-left:visited {
    background-color: #e67e22;
    color: #fff;
    margin-right: 1.6rem
}

.btn-right:link, .btn-right:visited {
    background-color: #fff;
    color: #555;
}

.btn-left:hover, .btn-left:active {
    background-color: #cf711f;
}

.btn-right:hover, .btn-right:active {
    background-color: #fdf2e9;
    box-shadow: inset 0 0 3px #fff;
}
```

ΠΡΟΣΘΗΚΗ ΤΩΝ ΕΙΚΟΝΩΝ ΚΑΤΩ ΑΠΟ ΤΑ ΚΟΥΜΠΙΑ

Στην αριστερή στήλη και κάτω από τα κουμπιά θα προσθέσω τις εικόνες που ακολουθούν μαζί με το συνοδευτικό κείμενο.

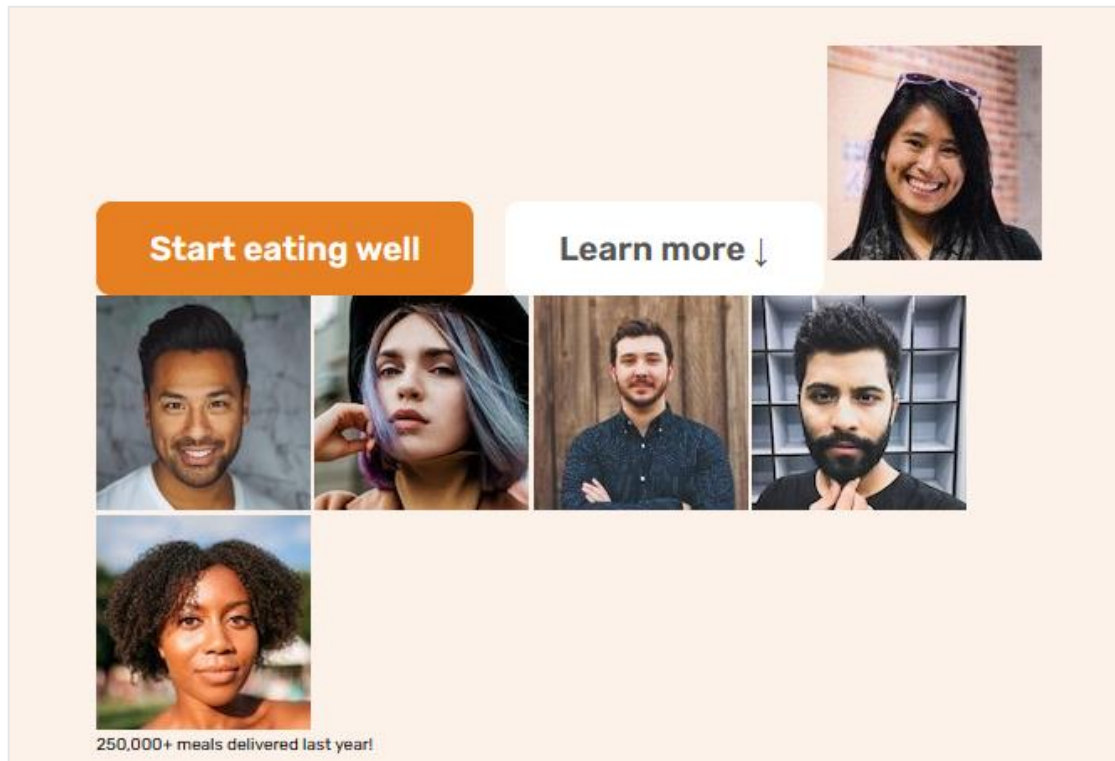


Μέσα στον κώδικα της αριστερής στήλης και ακριβώς κάτω απ' τον κώδικα των κουμπιών εισάγω τις 6 εικόνες και μία παράγραφο με το κείμενο :

index.html

```
<main>
  <section class="section-hero">
    <div class="hero">
      <div class="hero-text-box">
        <h1 class="heading-primary">
          A healthy meal delivered to your door, every single day
        </h1>
        <p class="hero-description">
          The smart 365-days-per-year food subscription that will make you
          eat healthy again. Tailored to your personal tastes and
          nutritional needs.
        </p>
        <a href="#" class="btn">Start eating well</a>
        <a href="#" class="btn">Learn more &darr;</a>
        
        
        
        
        
        
        <p>250,000+ meals delivered last year!</p>
      </div>
      <div class="hero-img-box">
        
      </div>
    </div>
  </section>
</main>
```

Σώζω και βλέπω τις αλλαγές. Παρατηρώ ότι το περιεχόμενο που πρόσθεσα ξεκινά να προβάλλεται δίπλα στα κουμπιά.



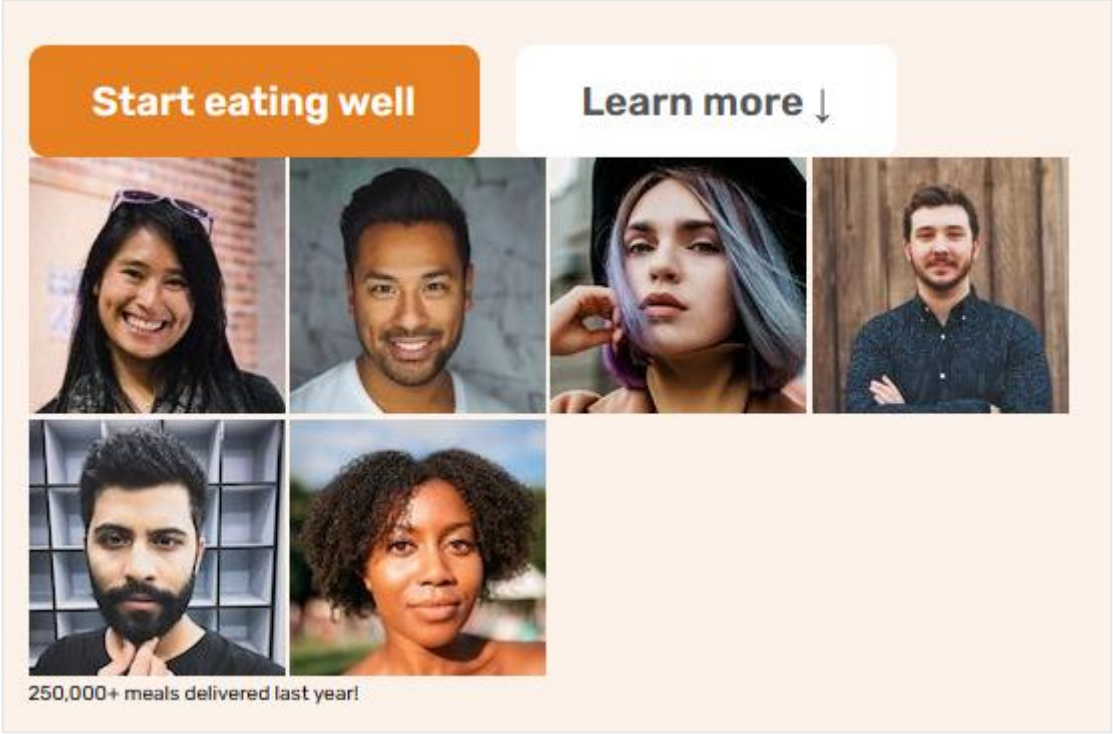
Για να κάτσει όλο το περιεχόμενο κάτω από τα κουμπιά το κλείνω μέσα σε μια διπλή ετικέτα **div** στην οποία δίνω το όνομα **delivered-meals**.

```
index.html

<main>
  <section class="section-hero">
    <div class="hero">
      <div class="hero-text-box">
        <h1 class="heading-primary">
          A healthy meal delivered to your door, every single day
        </h1>
        <p class="hero-description">
          The smart 365-days-per-year food subscription that will make you
          eat healthy again. Tailored to your personal tastes and
          nutritional needs.
        </p>
        <a href="#" class="btn">Start eating well</a>
        <a href="#" class="btn">Learn more &darr;</a>
      </div>
    </div>
  </section>
</main>
```

```
<div class="delivered-meals">
  
  
  
  
  
  
  <p>250,000+ meals delivered last year!</p>
</div>
</div>
<div class="hero-img-box">
  
</div>
</div>
</section>
</main>
```

Σώζω και βλέπω τις αλλαγές.



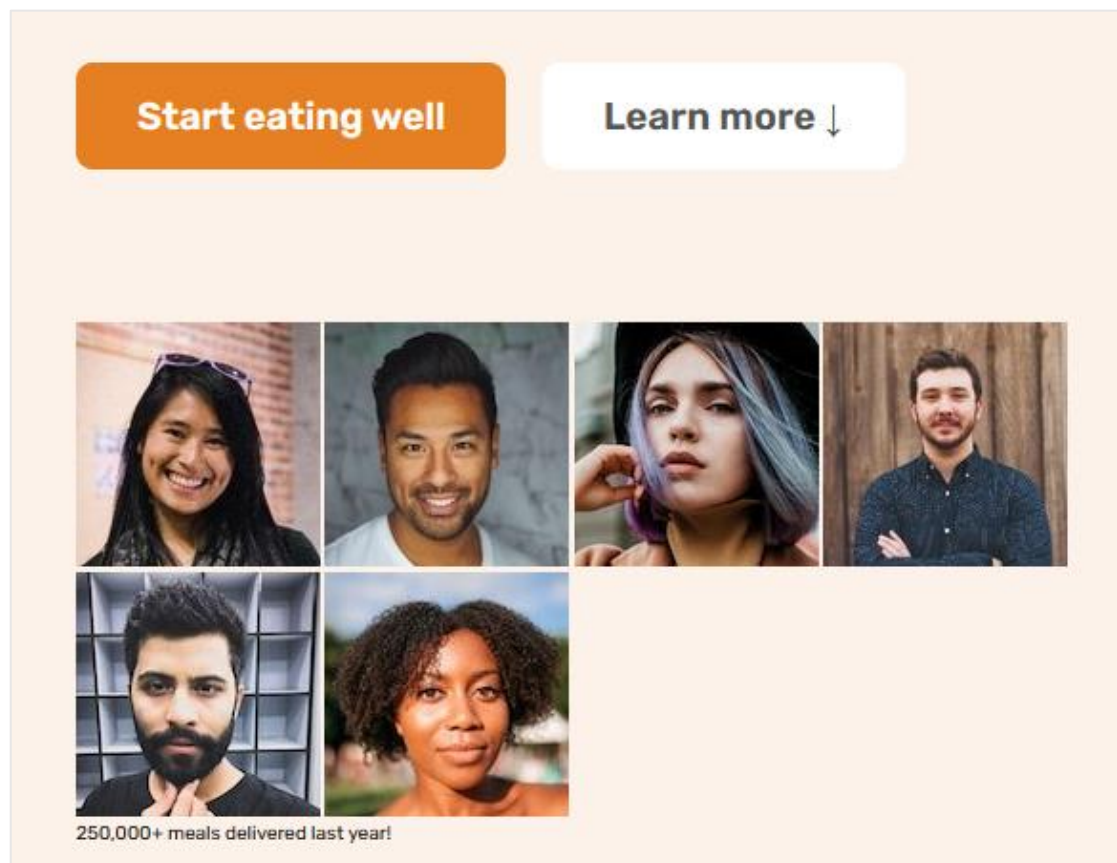
Η ετικέτα `div` είναι block-level στοιχείο που σημαίνει ότι κλείνει τα περιεχόμενα σε ένα κουτί που απλώνει σε όλη την αριστερή στήλη. Έτσι τα περιεχόμενα που προσθέσαμε πριν αναγκάζονται να πέσουν κάτω από τα κουμπιά.

Στο τέλος του αρχείου style.css προσθέτω νέο κανόνα που στοχεύει στο στοιχείο delivered-meals και του δίνει επάνω απόσταση 80px (8rem).

```
style.css

.delivered-meals {
  margin-top: 8rem;
}
```

Σώζω και βλέπω τις αλλαγές. Παρατηρήστε ότι τα περιεχόμενα δεν κολλάνε πλέον στα κουμπιά.



Δίνω στην **παράγραφο** το όνομα κλάσης **delivered-text**.

index.html

```
<main>
  <section class="section-hero">
    <div class="hero">
      <div class="hero-text-box">
        <h1 class="heading-primary">
          A healthy meal delivered to your door, every single day
        </h1>
        <p class="hero-description">
          The smart 365-days-per-year food subscription that will make you
          eat healthy again. Tailored to your personal tastes and
          nutritional needs.
        </p>
        <a href="#" class="btn">Start eating well</a>
        <a href="#" class="btn">Learn more &darr;</a>
        <div class="delivered-meals">
          
          
          
          
          
          
          <p class="delivered-text">250,000+ meals delivered last year!</p>
        </div>
      </div>
      <div class="hero-img-box">
        
      </div>
    </div>
  </section>
</main>
```

Και στοχεύω επάνω της με τον επόμενο κανόνα δίνοντας:

- Μέγεθος γραμμάτων 18px (1.8rem)
- Ένταση γραμμάτων 600

style.css

```
.delivered-text {  
    font-size: 1.8rem;  
    font-weight: 600;  
}
```

Σώζω και βλέπω ότι το κείμενο της παραγράφου μεγάλωσε.

250,000+ meals delivered last year!

Για να δώσω διαφορετικό χρώμα στο **250,000** θα πρέπει μέσα στο αρχείο **index.html** να το περικλείσω στο εσωτερικό της ετικέτας `<span></span>` όπως φαίνεται στη συνέχεια

index.css

```
<p class="delivered-text"><span>250,000+</span> meals delivered last year!</p>
```

Και να στοχεύσω στην ετικέτα με τον επόμενο κανόνα που πρέπει να τοποθετήσω μέσα στο αρχείο style.css.

style.css

```
.delivered-text span {  
    color: #cf711f;  
    font-weight: 700;  
}
```

Σώζω και βλέπω τις αλλαγές.

250,000+ meals delivered last year!

Για να φέρω τις εικόνες να κάτσουν η μία δίπλα στην άλλη θα τις κλείσω όλες μέσα σε μια νέα **div** με το όνομα **delivered-meals** την οποία θα μετατρέψω σε **flexbox**. Επίσης θα μικρύνω τις διαστάσεις των εικόνων στα 48px πλάτος και 48px ύψος.

index.html

```
<main>
  <section class="section-hero">
    <div class="hero">
      <div class="hero-text-box">
        <h1 class="heading-primary">
          A healthy meal delivered to your door, every single day
        </h1>
        <p class="hero-description">
          The smart 365-days-per-year food subscription that will make you
          eat healthy again. Tailored to your personal tastes and
          nutritional needs.
        </p>
        <a href="#" class="btn">Start eating well</a>
        <a href="#" class="btn">Learn more &darr;</a>
        <div class="delivered-meals">
          <div class="delivered-meals">
            
            
            
            
            
            
          </div>
          <p class="delivered-text">250,000+ meals delivered last year!</p>
        </div>
      </div>
      <div class="hero-img-box">
        
      </div>
    </div>
  </section>
</main>
```

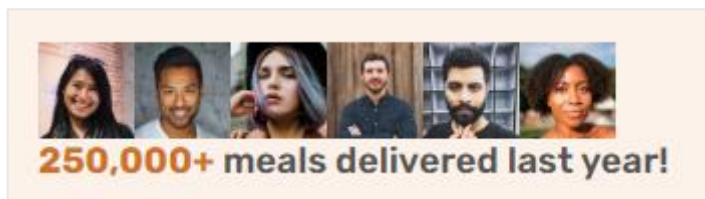
και θα προσθέσω τον επόμενο κανόνα στο αρχείο **style.css**

```
style.css

.delivered-imgs {
    display: flex;
}

.delivered-imgs img {
    height: 4.8rem;
    width: 4.8rem;
}
```

Σώζω και βλέπω τις αλλαγές.

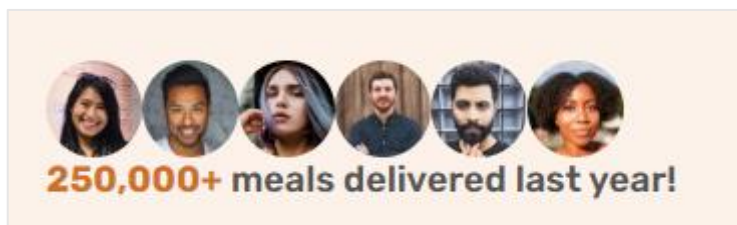


Μετατρέπω τις εικόνες από τετράγωνα σε στρογγυλές προσθέτοντας στον πιο πάνω κανόνα την ιδιότητα **border-radius** με τιμή **50%**.

```
style.css

.delivered-imgs img {
    height: 4.8rem;
    width: 4.8rem;
    border-radius: 50%;
}
```

Σώζω και βλέπω τις αλλαγές.

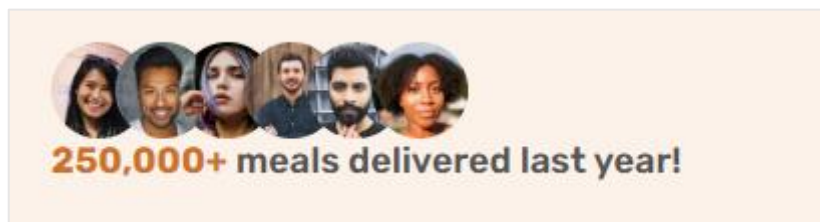


Για να φέρω τη μία πάνω στην άλλη μειώνω την αριστερή απόσταση αυτών (margin-left) κατά 16px (1.6rem) προσθέτοντας στον πιο πάνω κανόνα την ιδιότητα **margin-left: -1.6rem**

```
style.css

.delivered-imgs img {
    height: 4.8rem;
    width: 4.8rem;
    border-radius: 50%;
    margin-left: -1.6rem;
}
```

Σώζω και βλέπω τις αλλαγές.

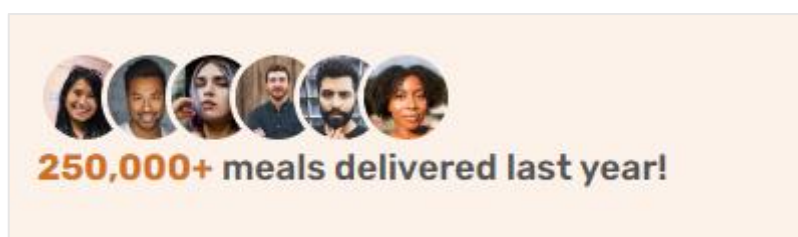


Βάζω περίγραμμα στις εικόνες προσθέτοντας στον προηγούμενο κανόνα την ιδιότητα border όπως φαίνεται πιο κάτω

```
style.css

.delivered-imgs img {
    height: 4.8rem;
    width: 4.8rem;
    border-radius: 50%;
    margin-left: -1.6rem;
    border: 3px solid #fdf2e9;
}
```

Σώζω και βλέπω τις αλλαγές.

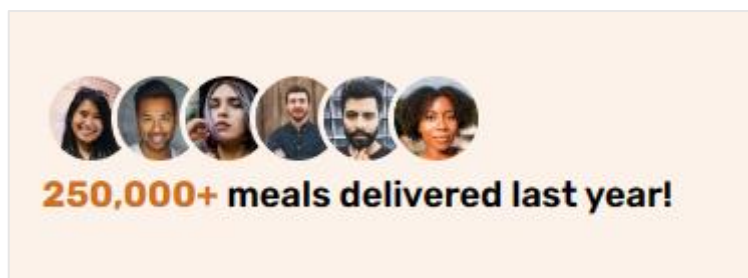


Παρατηρώ ότι η αριστερή εικόνα ξεφεύγει έξω από το κουτί εξαιτίας του αρνητικού margin που έχει επιβληθεί στα αριστερά όλων των εικόνων. Θα το επαναφέρω στο 0 μόνο για την αριστερή εικόνα μέσω του επόμενου κανόνα που προσθέτω στο τέλος του αρχείου style.css

```
style.css
.delivered-imgs img:first-child {
    margin-left: 0;
}
```

Με τον επιλογέα **.delivered-imgs img:first-child** επιλέγω μέσα από το στοιχείο **delivered-imgs** την πρώτη κατά σειρά εικόνα που είναι θεωρητικά το πρώτο του παιδί. Το κουτί της **delivered-imgs div** λειτουργεί ως γονέας με τις εικόνες που είναι στο εσωτερικό του να λειτουργούν ως άμεσα παιδιά αυτού του στοιχείου. Η πρώτη κατά σειρά εικόνα είναι το πρώτο του παιδί.

Σώζω και βλέπω τις αλλαγές



Για να φέρω τις εικόνες και την παράγραφο να κάτσουν δίπλα-δίπλα θα κάνω χρήση του **flexbox** πάνω στο κουτί **delivered meals**. Αυτό φιλοξενεί στο εσωτερικό του δύο στοιχεία, το κουτί **delivered-imgs** με τις εικόνες και το **κουτί της παραγράφου**.

Αυτά τα δύο στοιχεία αποτελούν άμεσα παιδιά της **delivered-meals div**. Αν εφαρμόσω την ιδιότητα **display: flex** στο γονικό στοιχείο (στην **delivered-meals**) θα καταφέρω να παρατάξω τα άμεσα παιδιά του το ένα δίπλα στο άλλο στον οριζόντιο άξονα.

Στον κανόνα που έχω ήδη για την **delivered-meals** προσθέτω:

```
style.css
.delivered-meals {
    margin-top: 8rem;
    display: flex;
}
```

Σώζω και βλέπω τις αλλαγές



Στοιχίζω τα δύο στοιχεία του **flexbox** στο κέντρο του κατακόρυφου άξονα προσθέτοντας την ιδιότητα **align-items** με τιμή **center**.

```
style.css
.delivered-meals {
  margin-top: 8rem;
  display: flex;
  align-items: center;
}
```

Σώζω και βλέπω τις αλλαγές



Τέλος με την ιδιότητα **gap** του flexbox προσθέτω **απόσταση 16px (1.6rem)** μεταξύ των δύο στοιχείων.

```
style.css
.delivered-meals {
  margin-top: 8rem;
  display: flex;
  align-items: center;
  gap: 1.6rem;
}
```

Σώζω και βλέπω τις αλλαγές



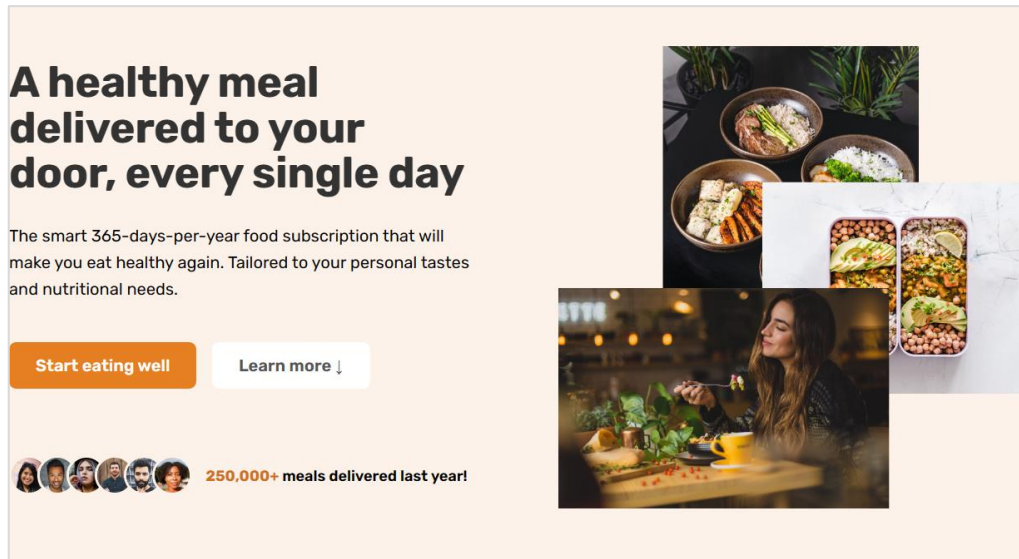
Τελειώνω προσθέτοντας μια **απόσταση** της τάξης των **32px (3.2rem)** μεταξύ του περιεχομένου και της αριστερής και δεξιάς πλευράς του κουτιού της **hero div** μέσα στην οποία περιέχεται.

```
style.css
.hero {
  max-width: 130rem;
  margin: 0 auto;
  display: grid;
  gap: 9.6rem;
  align-items: center;
  grid-template-columns: 1fr 1fr;
  padding: 0 3.2rem;
}
```

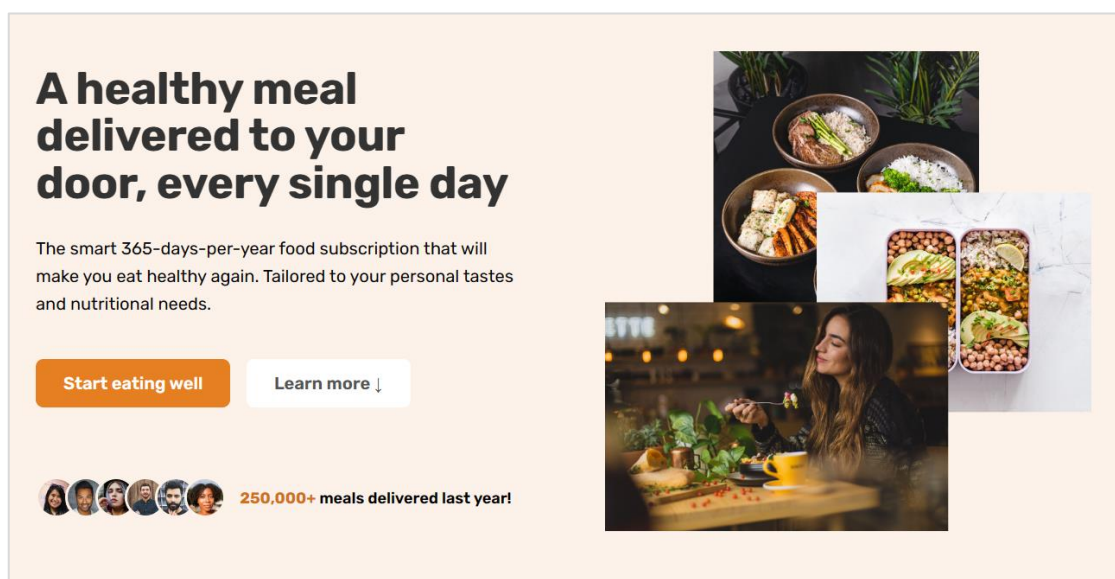
Η προσθήκη της απόστασης αυτής έχει νόημα αν προσπαθήσουμε να δούμε τη σελίδα μέσω μικρότερων συσκευών - όπου το πλάτος της οθόνης πέφτει κάτω από τα **1300px (130rem)**. Σ' αυτές τις συσκευές το κουτί **hero** που φιλοξενεί το περιεχόμενο θα αρχίσει να συρρικνώνει το πλάτος του προσαρμόζοντάς το στο πλάτος της συσκευής.

Αυτό είναι αποτέλεσμα της χρήσης της ιδιότητας **max-width** που αναγκάζει το πλάτος της **hero div** να **προσαρμοστεί** στο πλάτος της συσκευής, όταν το τελευταίο πέσει κάτω από την τιμή που έχω ορίσει (εδώ τα **1300px**). Σε αντίθεση με την ιδιότητα **width** η οποία κρατά το πλάτος του στοιχείου **σταθερό** ανεξάρτητα από το πλάτος της συσκευής.

Αν δεν έχω ορίσει κάποια απόσταση μεταξύ περιεχομένου και των πλαϊνών του κουτιού της **hero** div τότε στις μικρότερες συσκευές το περιεχόμενο θα φαίνεται να κολλά στα τοιχώματα της συσκευής όπως πιο κάτω.



Πιο κάτω φαίνεται η μορφή που παίρνει η σελίδα με την προσθήκη αποστάσεων **32px (3.2rem)** μεταξύ περιεχομένου και αριστερής και δεξιάς πλευράς του κουτιού μέσω της ιδιότητας **padding**.



Παρατηρείστε ότι στις μικρότερες συσκευές το περιεχόμενο δεν κολλά πλέον στις πλευρές της οθόνης όσο κι αν μικρύνει το πλάτος της οθόνης.

Εδώ η **HERO SECTION** ολοκληρώθηκε. Ακολουθεί το αρχείο **index.html** & οι μορφοποιήσεις της ενότητας HERO μέσα στο αρχείο **style.css**.

```
index.html

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Omnifood</title>
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link href="https://fonts.googleapis.com/css2?family=Rubik:wght@400;500;600;700&display=swap" rel="stylesheet">
  <link href="style.css" rel="stylesheet">
</head>
<body>
  <header>
    
    <nav>
      <ul class="nav-list">
        <li><a class="btn-menu" href="#">How it works</a></li>
        <li><a class="btn-menu" href="#">Meals</a></li>
        <li><a class="btn-menu" href="#">Testimonials</a></li>
        <li><a class="btn-menu" href="#">Pricing</a></li>
        <li><a class="btn-menu btn-try" href="#">Try for free</a></li>
      </ul>
    </nav>
  </header>
  <main>
    <section class="section-hero">
      <div class="hero">
        <div class="hero-text-box">
          <h1 class="heading-primary">
            A healthy meal delivered to your door, every single day
          </h1>
          <p class="hero-description">
            The smart 365-days-per-year food subscription that will make you
            eat healthy again. Tailored to your personal tastes and
            nutritional needs.
          </p>
          <a href="#" class="btn btn-left">Start eating well</a>
        </div>
      </div>
    </section>
  </main>
</body>
</html>
```

```

<a href="#" class="btn btn-right">Learn more &darr;</a>
<div class="delivered-meals">
  <div class="delivered-imgs">
    
    
    
    
    
    
  </div>
  <p class="delivered-text">
    <span>250,000+ </span>meals delivered last year!
  </p>
</div>
</div>
<div class="hero-img-box">
  
</div>
</div>
</section>
</main>
</body>
</html>

```

style.css

```

/* HERO SECTION */
.section-hero {
  background-color: #fdf2e9;
  padding: 4.8rem 0 9.6rem 0;
}

.hero {
  max-width: 130rem;
  margin: 0 auto;
  display: grid;
  gap: 9.6rem;
  align-items: center;
  grid-template-columns: 1fr 1fr;
  padding: 0 3.2rem;
}

```

```
.hero-img {
    width: 100%;
}

.heading-primary {
    font-size: 5.2rem;
    font-weight: 700;
    line-height: 1.05;
    color: #333;
    letter-spacing: -0.5px;
    margin-bottom: 3.2rem;
}

.hero-description {
    font-size: 2rem;
    line-height: 1.6;
    margin-bottom: 4.8rem;
}

.btn:link, .btn:visited {
    font-size: 2rem;
    text-decoration: none;
    display: inline-block;
    padding: 1.6rem 3.2rem;
    border-radius: 9px;
    font-weight: 600;
    transition: all 0.3s;
}

.btn-left:link, .btn-left:visited {
    background-color: #e67e22;
    color: #fff;
    margin-right: 1.6rem;
}

.btn-right:link, .btn-right:visited {
    background-color: #fff;
    color: #555;
}
```

```

.btn-left:hover, .btn-left:active {
    background-color: #cf711f;
}

.btn-right:hover, .btn-right:active {
    background-color: #fdf2e9;
    box-shadow: inset 0 0 0 3px #fff;
}

.delivered-meals {
    margin-top: 8rem;
    display: flex;
    align-items: center;
    gap: 1.6rem;
}

.delivered-imgs {
    display: flex;
}

.delivered-imgs img {
    height: 4.8rem;
    width: 4.8rem;
    border-radius: 50%;
    margin-left: -1.6rem;
    border: 3px solid #fdf2e9;
}

.delivered-imgs img:first-child {
    margin-left: 0;
}

.delivered-text {
    font-size: 1.8rem;
    font-weight: 600;
}

.delivered-text span {
    color: #cf711f;
    font-weight: 700;
}

```