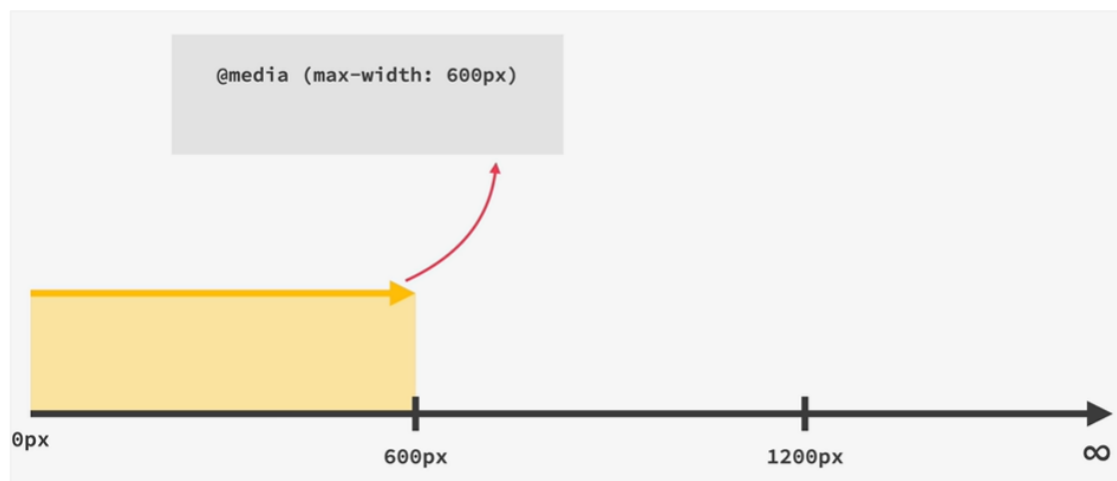


Media Queries

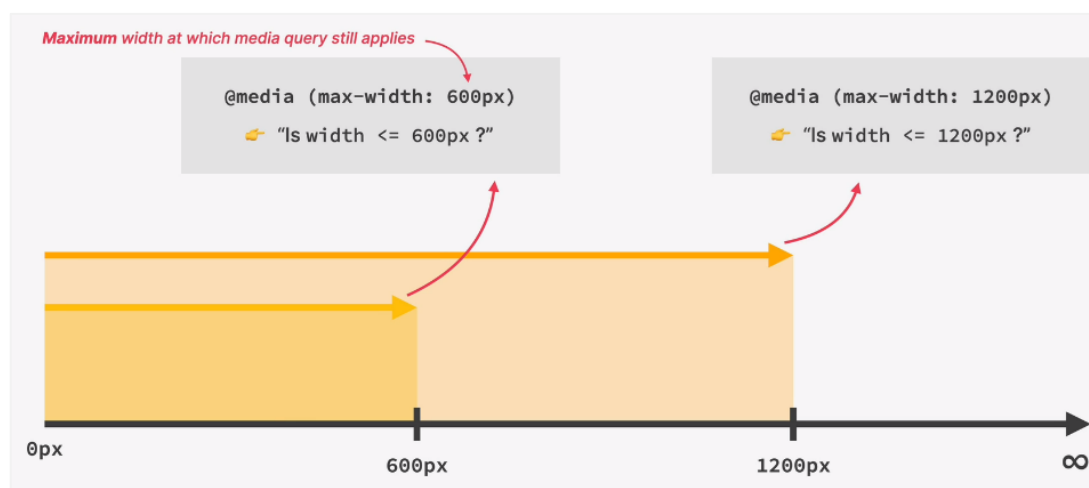
Θα δούμε πως λειτουργούν οι media queries σε συνδυασμό με την **max-width** ιδιότητα, την οποία πρέπει να χρησιμοποιούμε όταν δουλεύουμε με την **desktop-first** στρατηγική. Αντίθετα όταν ακολουθώ την **mobile-first** στρατηγική τότε θα πρέπει να χρησιμοποιώ media queries σε συνδυασμό με την **min-width** ιδιότητα.



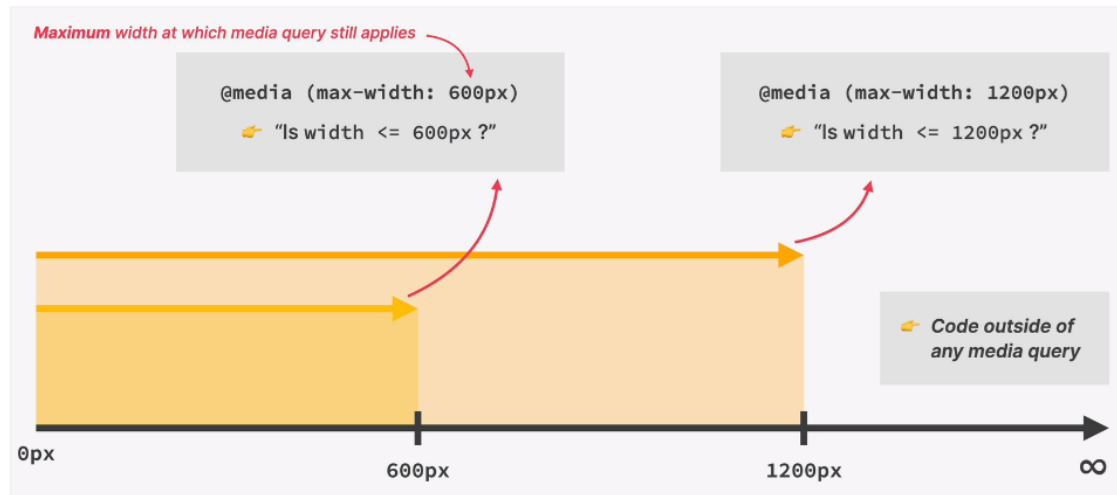
Ας υποθέσουμε ότι θέλουμε να εφαρμόσουμε μια διαφορετική μορφοποίηση για το διάστημα μεταξύ 0 και 600px. Τότε θα χρειαστεί να γράψουμε μια media query όπως η πιο κάτω, η οποία ελέγχει κατά πόσο το εύρος του viewport είναι μικρότερο ή ίσο με 600px. Αν είναι τότε όλος ο css κώδικας που βρίσκεται μέσα στη media query θα επιβληθεί. Διαφορετικά δεν θα επιβληθεί.



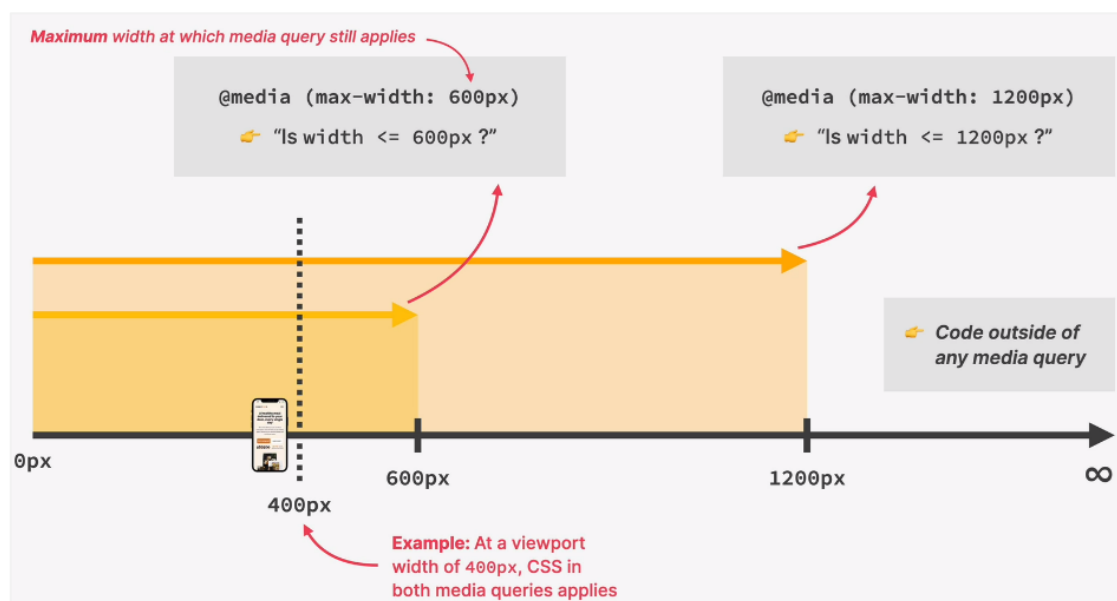
Αν πάλι θέλουμε να καλύψουμε το διάστημα μεταξύ 0 και 1200px θα χρειαστεί να γράψουμε μια media query όπως η πιο κάτω.



Μέσα στις media queries βρίσκονται μορφοποιήσεις οι οποίες αντικαθιστούν αντίστοιχες μορφοποιήσεις που επιβάλλονται από κώδικες CSS που βρίσκονται εκτός των media queries.

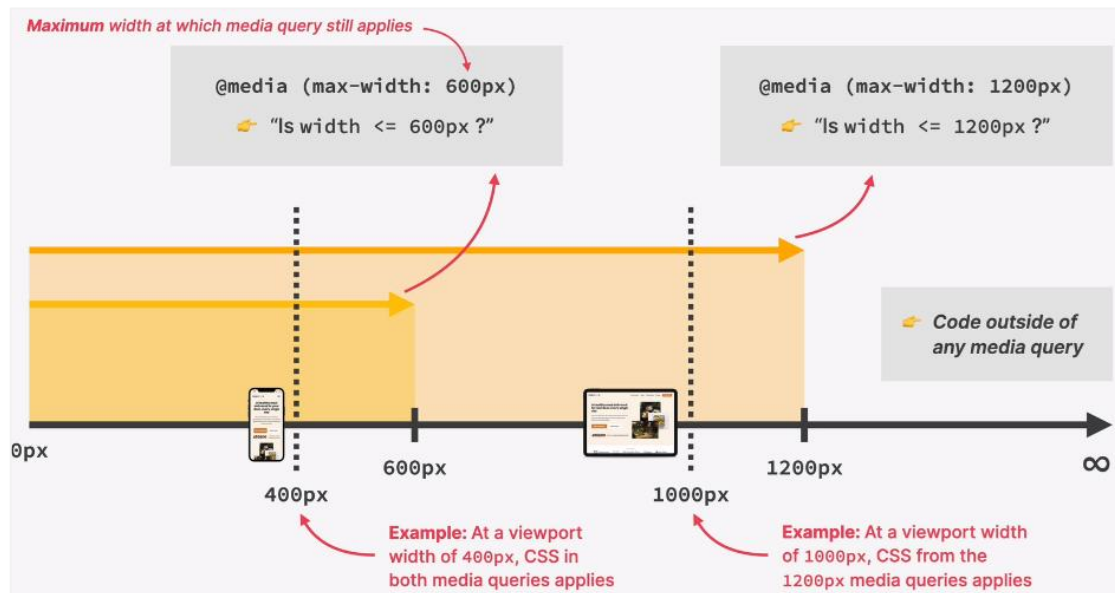


Ας δούμε τώρα τι συμβαίνει αν προσπαθήσουμε να δούμε τη σελίδα από ένα κινητό με πλάτος οθόνης 400px.



Τα 400px είναι λιγότερα από τα 600px αλλά και λιγότερα από τα 1200px με αποτέλεσμα να επιβληθούν οι μορφοποιήσεις και των δύο media queries. Στην περίπτωση που έχουμε συγκρουόμενους κανόνες CSS, τότε επιβάλλεται αυτός που έχει δηλωθεί τελευταίος στον κώδικα.

Αν τώρα πάρουμε την περίπτωση ενός tablet πλάτους 1000px τότε γι' αυτή τη συσκευή μόνο οι κανόνες μορφοποίησης που αφορούν πλάτη οθόνης μικρότερα του 1200 επιβάλλονται.



Επιστρέφουμε στη σελίδα μας και στην Hero section. Αν θέλω το χρώμα υποβάθρου της section να αλλάζει σε κόκκινο όταν το πλάτος της οθόνης πέφτει κάτω από τα 1200px τότε μέσα στο αρχείο style.css θα πρέπει να προσθέσω τον ακόλουθο κώδικα

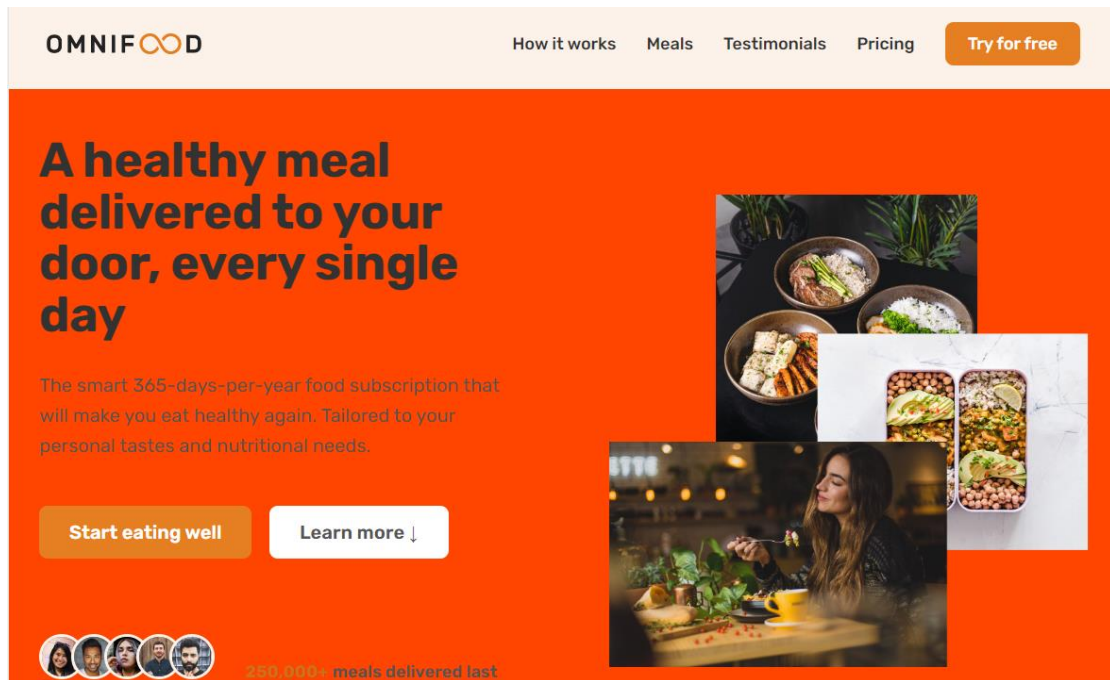
```
/* HERO SECTION */

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

@media (max-width: 1200px) {
  .section-hero {
    background-color: orangered;
  }
}
```

Ο πρώτος κανόνας επιβάλλει στην section μαζί με τα paddings χρώμα υποβάθρου μπεζ. Μόλις όμως το πλάτος της οθόνης πέσει κάτω από τα 1200px τότε ενεργοποιείται ο κανόνας της media query ο οποίος αλλάζει το χρώμα που επιβάλλει ο προηγούμενος κανόνας σε κόκκινο.

Αν ανοίξω τα Dev Tools και αρχίσω να μαζεύω το παράθυρο θα δω ότι μόλις το πλάτος πέσει κάτω από τα 1200 τότε το χρώμα αλλάζει σε κόκκινο.



Ας προσθέσουμε τώρα τον επόμενο κανόνα ο οποίος προσθέτει ένα μπλε διακεκομμένο περίγραμμα πάχους 20px γύρω από το κουτί της section μόλις το πλάτος της οθόνης πέσει κάτω από τα 600px.

```
/* HERO SECTION */
```

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

```
@media (max-width: 1200px) {  
  .section-hero {  
    background-color: orangered;  
  }  
}
```

```
@media (max-width: 600px) {  
  .section-hero {  
    border: 20px dashed blue;  
  }  
}
```

Παρατηρείστε ότι μόλις το πλάτος πέσει κάτω από τα 600px το περίγραμμα προστίθεται. Προσέξτε ότι συνεχίζει να επιβάλλεται ο κόκκινος φόντος από τον προηγούμενο κανόνα ενώ τώρα προστίθεται και το περίγραμμα.



Αν τώρα προσθέσουμε και την πιο κάτω αλλαγή στο χρώμα του φόντου αν το πλάτος οθόνης πέσει κάτω από τα 600

```
/* HERO SECTION */

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

@media (max-width: 1200px) {
  .section-hero {
    background-color: orangered;
  }
}

@media (max-width: 600px) {
  .section-hero {
    border: 20px dashed blue;
    background-color: blue;
  }
}
```

Θα δω ότι μόλις το πλάτος πέσει κάτω από τα 600 το χρώμα που ορίζει ο τελευταίος κανόνας επιβάλλεται αντικαθιστώντας το κόκκινο που επέβαλλε ο προηγούμενος κανόνας.



KANONAS: Στην περίπτωση αλληλοσυγκρουόμενων κανόνων ο τελευταίος επιβάλλεται πάντα.

Έτσι αν αλλάξουμε τη σειρά των κανόνων στο αρχείο css θα δούμε ότι ο κανόνας για τις οθόνες κάτω των 600px δεν θα επιβληθεί ποτέ γιατί με το που επιβάλλεται στη συνέχεια αναιρείται από τον κανόνα που ακολουθεί.

```
/* HERO SECTION */

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

@media (max-width: 600px) {
  .section-hero {
    border: 20px dashed blue;
    background-color: blue;
  }
}

@media (max-width: 1200px) {
  .section-hero {
    background-color: orangered;
  }
}
```

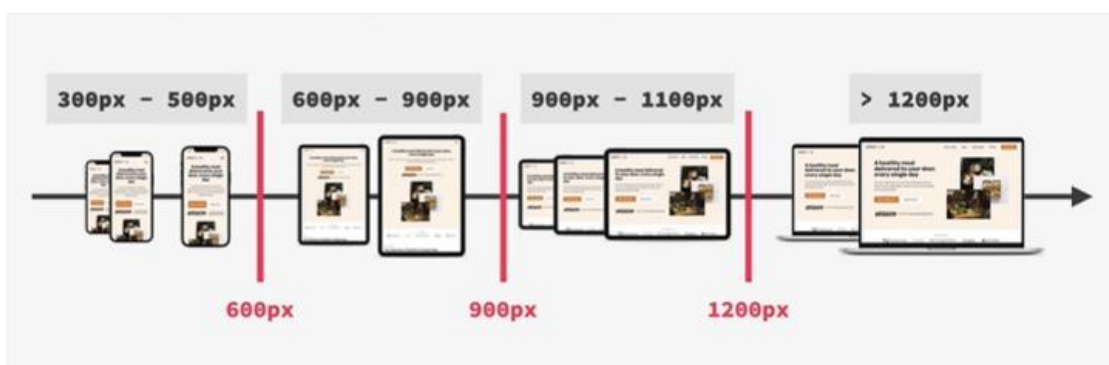
Breakpoints

Είναι εκείνα τα πλάτη οθόνης (viewport) για τα οποία η σχεδίασή μας επιβάλλεται να αλλάξει, ή με άλλα λόγια είναι εκείνες οι τιμές σε pixels για τις οποίες θα χρειαστεί να ορίσουμε media queries.

Παλαιότερα ως breakpoints εκλαμβάνονταν τα πλάτη των δημοφιλών συσκευών. Αυτή η τεχνική είχε ένα μειονέκτημα την βελτιστοποίηση της σελίδας για συγκεκριμένες μόνο συσκευές ενώ παρέκαμπτε όλες τις υπόλοιπες.

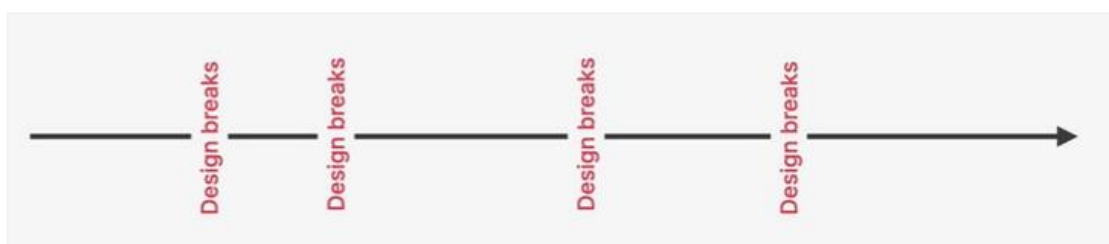


Η επόμενη στρατηγική έχει να κάνει με το διάστημα στο οποίο μπορεί να ανήκει το πλάτος της οθόνης του χρήστη. Αυτή είναι καλύτερη τεχνική σε σχέση με την προηγούμενη γιατί οι συσκευές ανάλογα με το πλάτος τους έχουν ομαδοποιηθεί σε μεγάλες κατηγορίες.



Έτσι μπορούμε να υποθέσουμε ότι οι περισσότερες οθόνες κινητών είναι μεταξύ 300px & 500px ενώ τα πιο πολλά tablet είναι μεταξύ 600px & 900px. Έτσι μπορούμε να θέσουμε ένα breakpoint στα 600px. Επίσης τα περισσότερα landscape tablets είναι μεταξύ 900px & 1200px, ενώ τα desktops είναι πάνω από 1200px. Έτσι μπορούμε να έχουμε δύο Breakpoints ένα στα 900px και άλλο ένα στα 1200px.

Η καλύτερη τεχνική όμως είναι να τοποθετούμε Breakpoints στα σημεία που η σχεδίαση της σελίδας αρχίζει να καταστρέφεται.



Σ' αυτή την τεχνική αγνοούμε εντελώς τις συσκευές ή τις κατηγορίες συσκευών και καθοδηγούμαστε μόνο από το περιεχόμενο και τη σχεδίαση.

ΠΡΟΣΟΧΗ! Σ' αυτό το σημείο θα πρέπει να διαγράψουμε μέσα από το αρχείο style.css τα media queries που είχαμε προσθέσει.

```
/* HERO SECTION */

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

@media (max-width: 600px) {
  .section-hero {
    border: 20px dashed blue;
    background-color: blue;
  }
}

@media (max-width: 1200px) {
  .section-hero {
    background-color: orangered;
  }
}
```

Σχεδίαση για Μικρά Laptops

Προτού ξεκινήσουμε την προσαρμοστική σχεδίαση θα πρέπει να προσθέσουμε την πιο κάτω μεταετικέτα στον HTML κώδικα της σελίδας μας μέσα στην περιοχή <head></head>.

```
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <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:ital,wght@0,300..900;1,300..900&
display=swap" rel="stylesheet" />
  <link href="style.css" rel="stylesheet" />
  <title>Omnifood</title>
</head>
```

Θα φυλάξουμε τους νέους κανόνες που θα επιβάλλονται μέσω των media queries σε νέο αρχείο με το όνομα **queries.css**, το οποίο θα πρέπει να δημιουργήσω και να συνδέσω με τη σελίδα μου ως εξής:

```
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <link rel="preconnect" href="https://fonts.googleapis.com" />
  <link href="https://fonts.googleapis.com/css2?family=Rubik:ital,wght@0,300..900;1,300..900&
display=swap" rel="stylesheet" />
  <link href="style.css" rel="stylesheet" />
  <link href="queries.css" rel="stylesheet" />
  <title>Omnifood</title>
</head>
```

ΠΡΟΣΟΧΗ!

- Η χρήση των pixels στα media queries είναι απαγορευτική.
- Μέσα στα media queries τα rem & em δεν εξαρτώνται από το μέγεθος γραμμάτων του ριζικού html στοιχείου. Αντίθετα 1rem=1em=16px.
- Το rem ορίζεται σε σχέση με το μέγεθος του ριζικού html στοιχείου της σελίδας
- Επειδή τα rem έχουν κάποια προβλήματα σε ορισμένους browsers καλό είναι στα media queries να χρησιμοποιώ την μονάδα em. Είναι ασφαλέστερο και έχει το ίδιο αποτέλεσμα.

Ανοίγω τη σελίδα στον browser και μεταβαίνω στα Dev Tools. Αρχίζω να μαζεύω το παράθυρο. Παρατηρώ ότι καθώς το πλάτος οθόνης πέφτει κάτω από τα 1350px και έως τα 1200px - μια περιοχή που αφορά τα μικρά desktop - αρχίζουν να εμφανίζονται κάποια προβλήματα.

Πρώτα-πρώτα η **hero div** – η div που συγκρατεί τα περιεχόμενα της **Hero section** στο κέντρο - είναι πιο ωραίο για τη συγκεκριμένη περιοχή να έχει το ίδιο πλάτος 1200px με αυτό όλων των υπόλοιπων div που εξυπηρετούν τον ίδιο σκοπό και βρίσκονται σε κάθε μια από τις υπόλοιπες ενότητες.

Θυμηθείτε ότι μέσω του κανόνα που ακολουθεί και μέσω της ιδιότητας **max-width** έχω ορίσει το **μέγιστο πλάτος** της hero div να είναι **1300px**. Αυτό είναι το μέγιστο πλάτος που μπορεί να πάρει το κουτί της div. Έτσι σε οθόνες μικρότερες από 1300px το κουτί θα προσαρμόζεται στο πλάτος της οθόνης. Σε οθόνες από 1300px και πάνω το κουτί παύει να είναι προσαρμοστικό και θα παραμένει σταθερό στα 1300px.

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

Έτσι στο διάστημα από 1200px έως 1300px ενώ οι υπόλοιπες div παραμένουν σταθερές στα 1200px πλάτος η hero div έχει λίγο μεγαλύτερο πλάτος αφού προσαρμόζεται στο παράθυρο, κάτι που δημιουργεί ένα αντιαισθητικό αποτέλεσμα γι' αυτές τις οθόνες.

Θα το διορθώσω προσθέτοντας ένα **breakpoint** στα **1350px** περίπου (λίγο παραπάνω από τα 1300). Στο αρχείο **queries.css** προσθέτω το επόμενο **media** query μαζί με τον σχετικό κανόνα ο οποίος αλλάζει γι' αυτό το διάστημα την τιμή της ιδιότητας **max-width** της hero div σε 1200px.

```
/* Κάτω από τα 1350px (Μικρά Desktops) */  
  
@media (max-width: 84em) {                               /* 1350px/16px=84em */  
    .hero {  
        max-width: 120rem;  
    }  
}
```

Σώζω και βλέπω τις αλλαγές.

Παρατηρώ ότι όλα τα περιεχόμενα είναι τώρα εναρμονισμένα.

A healthy meal delivered to your door, every single day

The smart 365-days-per-year food subscription that will make you eat healthy again. Tailored to your personal tastes and nutritional needs.

[Start eating well](#)[Learn more ↓](#)

250,000+ meals delivered last year!



AS FEATURED IN



HOW IT WORKS

Your daily dose of health in 3 simple steps

01

Tell us what you like (and what not)

Never again waste time thinking about what to eat! Omnifood AI will create a 100% personalized weekly meal plan just for you. It makes sure you get all the nutrients and vitamins you need, no



Λόγω όμως του μικρότερου πλάτους η επικεφαλίδα **h1** της hero div δεν χωράει πλέον σε 3 γραμμές και απλώνει σε 4. Θα πρέπει αρχικά να **μειώσω το μέγεθος των γραμμμάτων της h1** το οποίο έχει επιβληθεί μέσω του επόμενου κανόνα. Θυμηθείτε ότι η επικεφαλίδα h1 έχει συνδυαστεί με την κλάση **heading-primary**.

```
.heading-primary {  
  font-size: 5.2rem;  
  font-weight: 700;  
  line-height: 1.05;  
  color: #333;  
}
```

```
letter-spacing: -0.5px;
margin-bottom: 3.2rem;
}
```

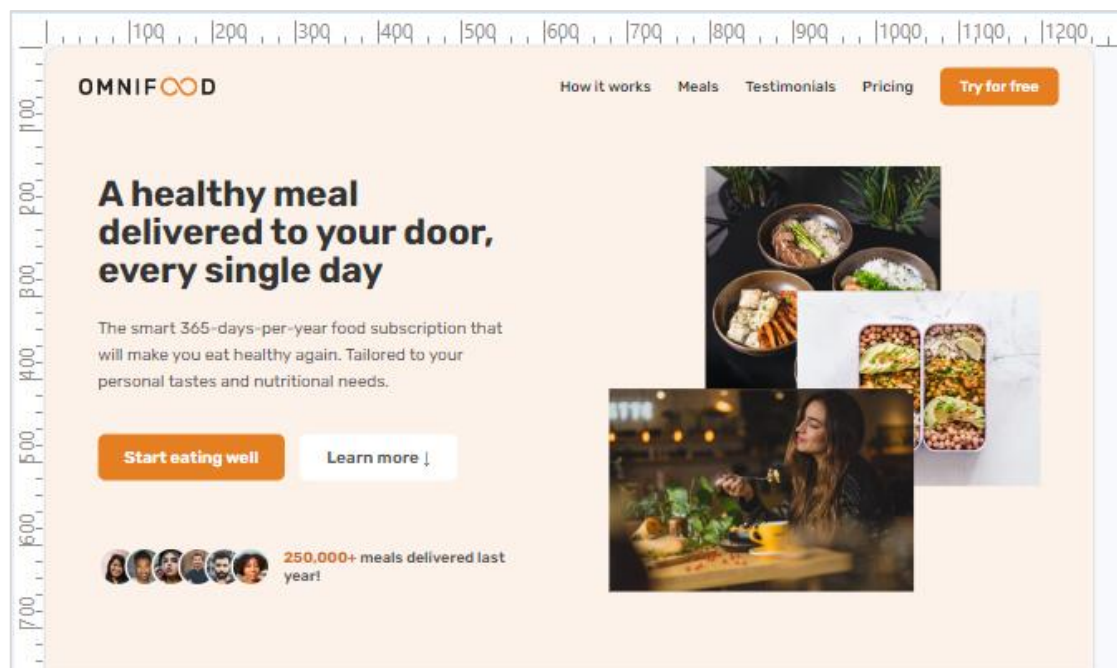
Μέσα στο αρχείο **queries.css** προσθέτω τον επόμενο κανόνα ο οποίος αλλάζει την τιμή **5.2rem** (52px) που επιβάλλει ο πρώτος κανόνας σε **4.4rem** (44px).

```
@media (max-width: 84em) {
  /* 1350px/16px=84em */

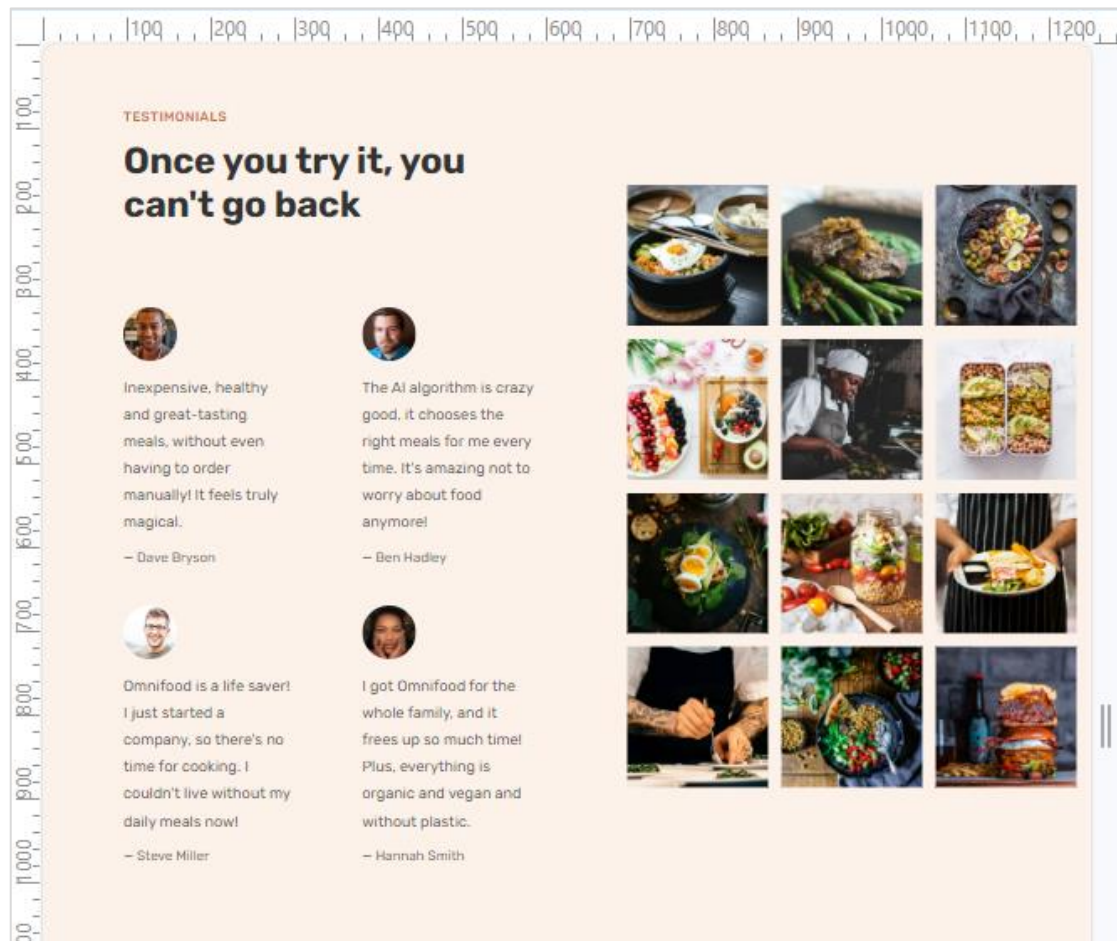
  .hero {
    max-width: 120rem;
  }

  .heading-primary {
    font-size: 4.4rem;
  }
}
```

Σώζω και βλέπω τις αλλαγές.



Αν κατέβω προς τα κάτω θα δω ότι σ' αυτή την περιοχή πλάτους ξεκινούν προβλήματα με την ενότητα TESTIMONIALS και πιο συγκεκριμένα με την δεξιά στήλη που περιέχει τις εικόνες. Το πρόβλημα μεγεθύνεται για ακόμη μικρότερα πλάτη.



Το πρόβλημα ξεκινά σχετικά νωρίς γύρω στα 1450px δεν είναι όμως καλή πρακτική να προσθέτω πολλά breakpoints. Θα προχωρήσω λοιπόν στις αλλαγές που χρειάζονται για τα 1350px και κάτω (για το breakpoint που έχω ήδη θέσει).

Θυμηθείτε ότι οι εικόνες βρίσκονται συγκεντρωμένες μέσα σε μια div με το όνομα **gallery** η οποία με τη βοήθεια του grid έχει χωριστεί σε **3 στήλες**. Οι εικόνες φιλοξενούνται στις στήλες του πλέγματος. Ο σχετικός κανόνας φαίνεται πιο κάτω.

```
.gallery {  
  display: grid;  
  grid-template-columns: 1fr 1fr 1fr;  
  gap: 1.6rem;  
  padding: 1.6rem;  
}
```

Για πλάτη κάτω των 1350px θα ήταν καλύτερο το πλέγμα να αποτελείται από **2 στήλες** αντί για 3. Για να το πετύχω αυτό στο αρχείο **queries.css** προσθέτω τον επόμενο κανόνα ο οποίος αλλάζει την τιμή της ιδιότητας **grid-template-columns** από **1fr 1fr 1fr** που επέβαλε ο προηγούμενος κανόνας σε **1fr 1fr**. Ακολουθεί ο κανόνας και το αποτέλεσμα.

```
@media (max-width: 84em) {  
  /* 1350px/16px=84em */  
  
  .hero {  
    max-width: 120rem;  
  }  
  
  .heading-primary {  
    font-size: 4.4rem;  
  }  
  
  .gallery {  
    grid-template-columns: 1fr 1fr;  
  }  
}
```

