

ΠΕΡΙΕΧΟΜΕΝΑ:

1. Media Queries
2. Που γράφουμε τα @media
3. RWD με τον Google Chrome
4. Οδηγία meta
5. Στρατηγικές για RWD

- **Responsive Design:** Η σελίδα μας να εμφανίζεται διαφορετικά, ανάλογα με τα χαρακτηριστικά της οθόνης.

At-rule @media:

- Ορίζει σύνολο κανόνων αν το μέσο (π.χ. οθόνη) έχει ορισμένα χαρακτηριστικά:

```
@media constraint {
  ...
}
```

- όπου ο περιορισμός (constraint) μπορεί να είναι:

- τύπος του μέσου, με τιμές:

all	screen	print	speech
-----	---------------	-------	--------

π.χ.

```
@media screen {
  ...
}
```

- περιορισμοί χαρακτηριστικών του μέσου ως ζεύγη χαρακτηριστικού-τιμής. Ζεύγη (μεταξύ άλλων):

width	πλάτος viewport (μήκος)
{min,max}-width	ελάχιστο, μέγιστο πλάτος viewport (μήκος)
height	ύψος viewport (μήκος)
{min,max}-height	ελάχιστο, μέγιστο ύψος viewport (μήκος)
orientation	landscape, portrait
resolution	Ανάλυση οθόνης

π.χ.

```
@media (min-width: 640px) {
  ...
}
```

- σύνθετοι περιορισμοί (λογικές πράξεις):

- ένωση με λογικό και, π.χ.:

```
@media (min-width: 480px) and (max-width: 640px) { ... }
```

- ένωση με λογικό ή (συμβ. με κόμμα), π.χ.:

```
@media screen, speech and (max-width: 640px) { ... }
```

- αντιστροφή με λογικό not

- Σχεσιακοί τελεστές (ακόμη όχι υλοποιημένο καθολικά) π.χ.:

```
@media 480px <= width <= 640px { ... }
```

Παρατήρηση:

- Θα εστιάσουμε, στην πιο συνηθισμένη χρήση για την κατασκευή σελιδών:
 - Να αλλάζει η εμφάνιση ανάλογα με το πλάτος της οθόνης

Παράδειγμα 1: responsive

```
body {
  background-color: blue;
}
@media screen and (min-width: 500px) {
  body {
    background-color: red;
  }
}
@media screen and (min-width: 1000px) {
  body {
    background-color: green;
  }
}
```

```
<body>
<h1>Responsive</h1>
</body>
```



Εναλλακτικές θέσεις ορισμού των @media:

- Ο πιο συνηθισμένος τρόπος είναι σε ένα κοινό αρχείο CSS (όπως είδαμε στην προηγούμενη διαφάνεια)
- Ωστόσο μπορούμε να ενσωματώσουμε το πεδίο εφαρμογής κανόνων αρχείων CSS κατά την ενσωμάτωση τους:

- είτε με το στοιχείο **link** στο **head** (βλ. και project: linked), π.χ.:

```
<head>
...
<style>
  body {
    background-color: blue;
  }
</style>
<link rel="stylesheet" href="styles/style1.css" media="screen and (min-width: 700px)">
<link rel="stylesheet" href="styles/style2.css" media="screen and (min-width: 1000px)">
</head>
```

- είτε **imported** στο **style** (βλ. και project: imported), π.χ.:

```
<head>
...
<style>
  @import url("styles/style.css");
  @import url("styles/style1.css") screen and (min-width: 700px);
  @import url("styles/style2.css") screen and (min-width: 1000px);
</style>
</head>
```

Διαφορά <link> με @import:

- @import: ενσωματώνει τα περιεχόμενα ενός αρχείου CSS σε περιοχή σύνταξης κανόνων CSS
- link: ενσωματώνει τους κανόνες CSS στο head της HTML
- Το import μπλοκάρει το rendering της σελίδας μέχρι να κατέβει, γι' αυτό συνίσταται να χρησιμοποιούμε πάντα το link

Επισημάνση:

- Ανεξάρτητα από τον τρόπο ορισμού των @media:
 - Αν η συνθήκη του @media είναι αληθής, τότε οι κανόνες παρατίθενται σαν να υπάρχει ένα κοινό αρχείο με τον κώδικα CSS.
 - Δεν υπάρχει δηλαδή ειδική εξειδίκευση κανόνων, το μόνο που γίνεται είναι επανεγγραφή κανόνων

Παράδειγμα 2: multiple-files

```
<body>
<div class="container">
  <div class="item1"> Item 1</div>
  <div class="item2"> Item 2</div>
  <div class="item3"> Item 3</div>
</div>
</body>
```

```
<link rel="stylesheet" href="styles/style1.css"
media="screen and (min-width: 700px)">
<link rel="stylesheet" href="styles/style2.css"
media="screen and (min-width: 1000px)">
```

```
<style>
  body {
    background-color: lightblue;
    font-size: 30px;
  }
  div.container {
    display: grid;
    grid-template-rows: repeat(3, 100px);
    grid-template-columns:
      "item1"
      "item2"
      "item3";
  }
  div[class^=item] {
    border: 2px black solid;
    padding: 30px;
    text-align: center;
  }
  div.item1 { grid-area: item1; }
  div.item2 { grid-area: item2; }
  div.item3 { grid-area: item3; }
</style>
```

```
div.container {
  display: grid;
  grid-template-rows:
    repeat(2, 200px);
  grid-template-columns:
    repeat(2, 1fr);
  grid-template-areas:
    "item1 item1"
    "item2 item3";
}

div.container {
  display: grid;
  grid-template-rows:
    repeat(2, 200px);
  grid-template-columns:
    repeat(2, 1fr);
  grid-template-areas:
    "item1 item1"
    "item2 item3";
}
```

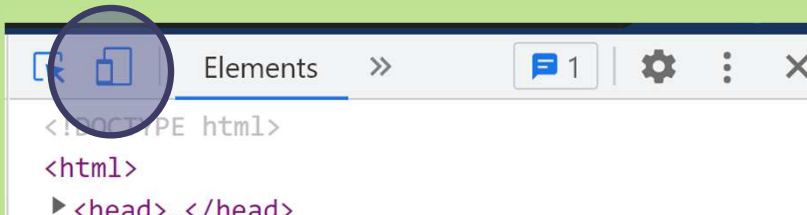
Item 1	
Item 2	Item 3

Item 1	Item 2	Item 3
--------	--------	--------

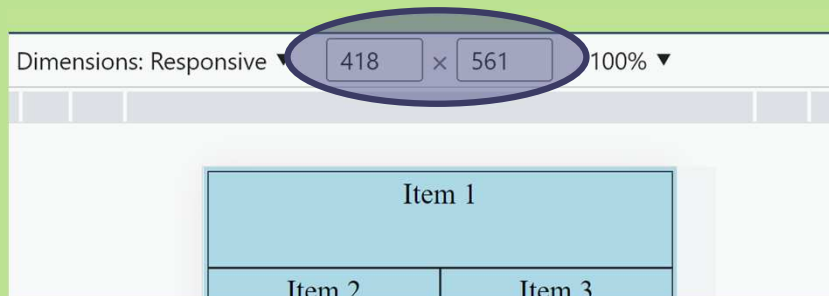
Item 1
Item 2
Item 3

RWD Inspection με τον Google Chrome:

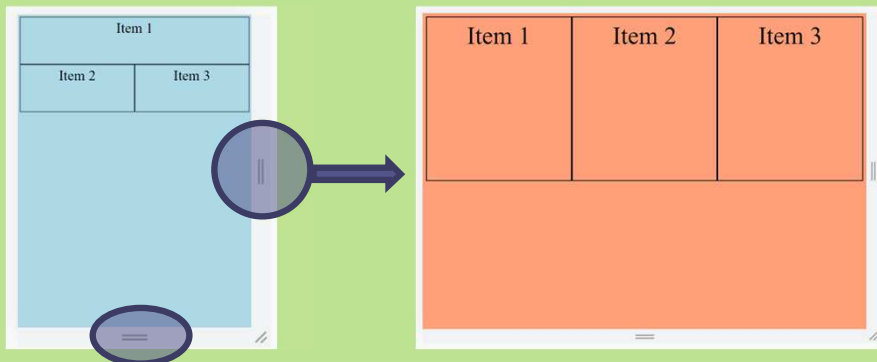
- Δεξί κλικ και Inspect.
- Κλικ πάνω στο εικονίδιο:



- Επιλογή πλάτους οθόνης:
 - είτε ρητά γράφοντας το επιθυμητό ύψος/πλάτος του viewport:

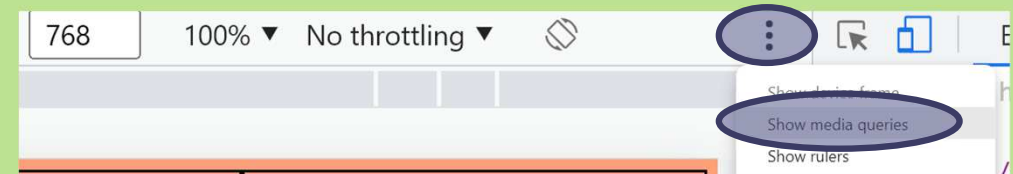


- είτε συροντας τα εικονίδια δεξιά και κάτω:

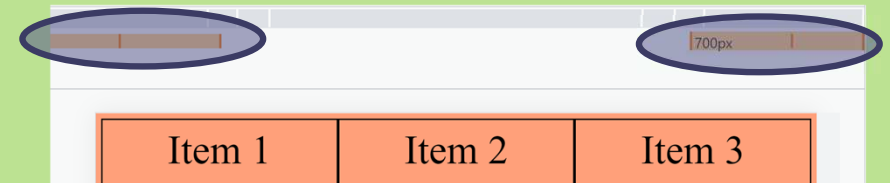


Άλλες Ενδιαφέρουσες Επιλογές:

- Εμφάνιση των media queries:



- Εμφανίζονται έξτρα μπάρες που δείχνουν τα σημεία breakpoint του κώδικά μας:

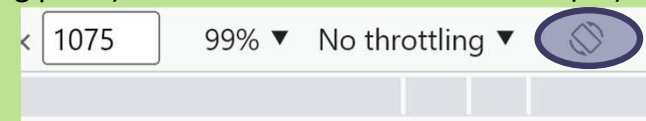


- Επιλογή διαστάσεων συγκεκριμένης συσκευής:



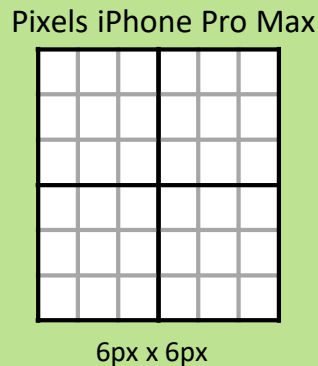
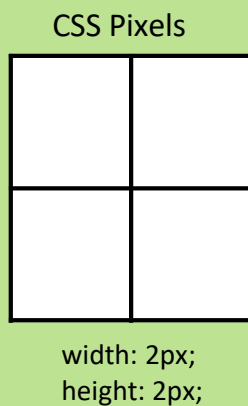
- Εμφανίζονται έξτρα μπάρες που δείχνουν τα σημεία breakpoint του κώδικά μας:

- Switching μεταξύ των δύο κατευθύνσεων σε κινητές συσκευές



Παρερμηνεία:

- Τα pixels της οθόνης μιας συσκευής ΔΕΝ είναι το ίδιο πράγμα με τα pixels που γράφουμε στον κώδικά μας.
 - Οι σημερινές συσκευές ενδέχεται να έχουν τεράστιες αναλύσεις. Π.χ το iPhone 13 Pro Max έχει ανάλυση 1284px x 2778px
 - Ωστόσο κάθε συσκευή επιλέγει σε πόσα πραγματικά pixel θα απεικονίσει 1 pixel
 - Για παράδειγμα **μπορεί 1 CSS Pixel να αντιστοιχεί σε 3x3 πραγματικά pixels** (αυτό π.χ. συμβαίνει στο iPhone Pro Max)
 - (Η απόφαση αυτή, λαμβάνεται από το σχεδιαστή της συσκευής, το λειτουργικό σύστημα, τον browser κ.ο.κ.)
 - Από αυτήν τη σχέση (που ονομάζεται CSS Pixel Ratio, εδώ ίσο με 3) καθορίζονται οι διαστάσεις σε CSS Pixels της συσκευής.
 - έτσι π.χ. το iPhone Pro Max έχει διαστάσεις σε **CSS Pixels: 428 px x 926px (το viewport)**

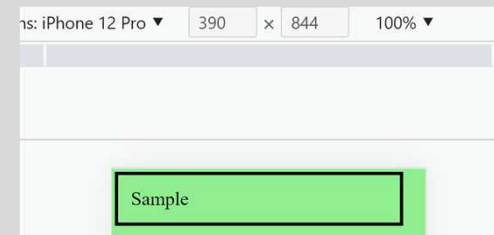


Το ViewPort:

- Default ViewPort:
 - Μία σελίδα προσαρμόζεται αυτόματα στην οθόνη του κινητού ή του tablet => Γίνεται Zoom Out ώστε να χωρέσει αυτόματα σε πλάτος περίπου 1000px.

Παράδειγμα 2: default-viewport

```
body {
  background-color: lightgreen;
  width: 800px;
  ...
}
```



Οδηγία meta για μέγεθος οθόνης:

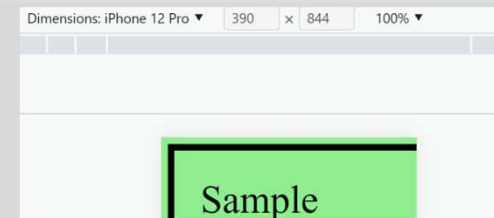
- Η οδηγία:


```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

 - Δίνει την εντολή: Το πλάτος του viewport να οριστεί ίσο με το πλάτος τα συσκευής σε CSS Pixels
 - και την εντολή: Αρχικά να μη γίνει κανένα αυτόματο zoom στην οθόνη (απαραίτητο για κάποιους τύπους συσκευής)
 - (θα την προσθέτουμε πάντα, στο head της html)**

Παράδειγμα 2: meta

```
body {
  width: 800px;
  ...
}
```








Στρατηγικές για Responsive Design:

- Χρήση background images: Έτσι μπορούμε μέσω CSS να αλλάζουμε την εικόνα που προβάλλεται ανάλογα με το μέγεθος της οθόνης.
- Διαφορετικό περιεχόμενο: Είναι συχνό κάποια μέρη της σελίδας που προβάλλονται σε μεγάλες οθόνες, να μην προβάλλονται σε μικρές οθόνες.
- Αλλαγή των font sizes: Άλλα μεγέθη γραμματοσειράς (κυρίως για τα γραφικά μέρη)
- Αλλαγή των μενού πλοήγησης: Π.χ. οριζόντιο μενού για μεγάλη οθόνη και ένα απλό κουμπί για μικρές οθόνες.

Μεγέθη οθόνης:

- Συνηθισμένα Μεγέθη Οθόνης (ενδεικτικό πλάτος σε CSS Pixels):

Κινητό (h)	Tablet (l)	Laptop	Desktop	TV
				
320-480px	481-768px	769px-1024px	1025-1200px	>1201px

Breakpoints:

- Τα σημεία που αλλάζουμε το σχεδιασμό της σελίδας

Ορισμός breakpoints για RWD:

- Προσέγγιση 1: Breakpoints για κάθε συσκευή.
 - Οι συσκευές είναι άπειρες. Απορρίπτεται.
- Προσέγγιση 2: Breakpoints για κάθε τύπο συσκευής:
 - Ένα αρχικό βασικό στυλ
 - Έπειτα διαφορετικό στυλ για κάθε κατηγορία από τις 5 παραπάνω (π.χ. ένα στυλ που θα εφαρμόζεται μόνο για 481-768px κ.ο.κ.)
- Προσέγγιση 3:
 - Ξεκινάμε από έναν βασικό σχεδιασμό για κινητά, και έπειτα προσθέτουμε breakpoints όταν το περιεχόμενο είναι αντιαισθητικό

- Προσέγγιση 4 (που θα ακολουθήσουμε, ~ mobile first):

- Ξεκινάμε με κοινές ιδιότητες στυλ + σχεδιασμό για κινητά
- Έπειτα κάνουμε τροποποιήσεις αν η συσκευή έχει ελάχιστο πλάτος που είναι Tablet (481px)
- Έπειτα κάνουμε τροποποιήσεις αν η συσκευή έχει ελάχιστο πλάτος που είναι Laptop (769px)
- Κ.Ο.Κ.

Παράδειγμα 2: responsive-skeleton

```

<meta name="viewport" content="width=device-width,
initial-scale=1">
<style>
body {
  background-color: #A52422;
  color: white;
}

h1::after {
  content: "mobile";
}

@media screen and (min-width: 481px) { /* tablet */
  h1::after {
    content: "tablet";
  }
}

@media screen and (min-width: 769px) { /* laptop */
  h1::after {
    content: "laptop";
  }
}

@media screen and (min-width: 1025px) { /* desktop */
  h1::after {
    content: "desktop";
  }
}

@media screen and (min-width: 1201px) { /* tv */
  h1::after {
    content: "tv";
  }
}

```

