

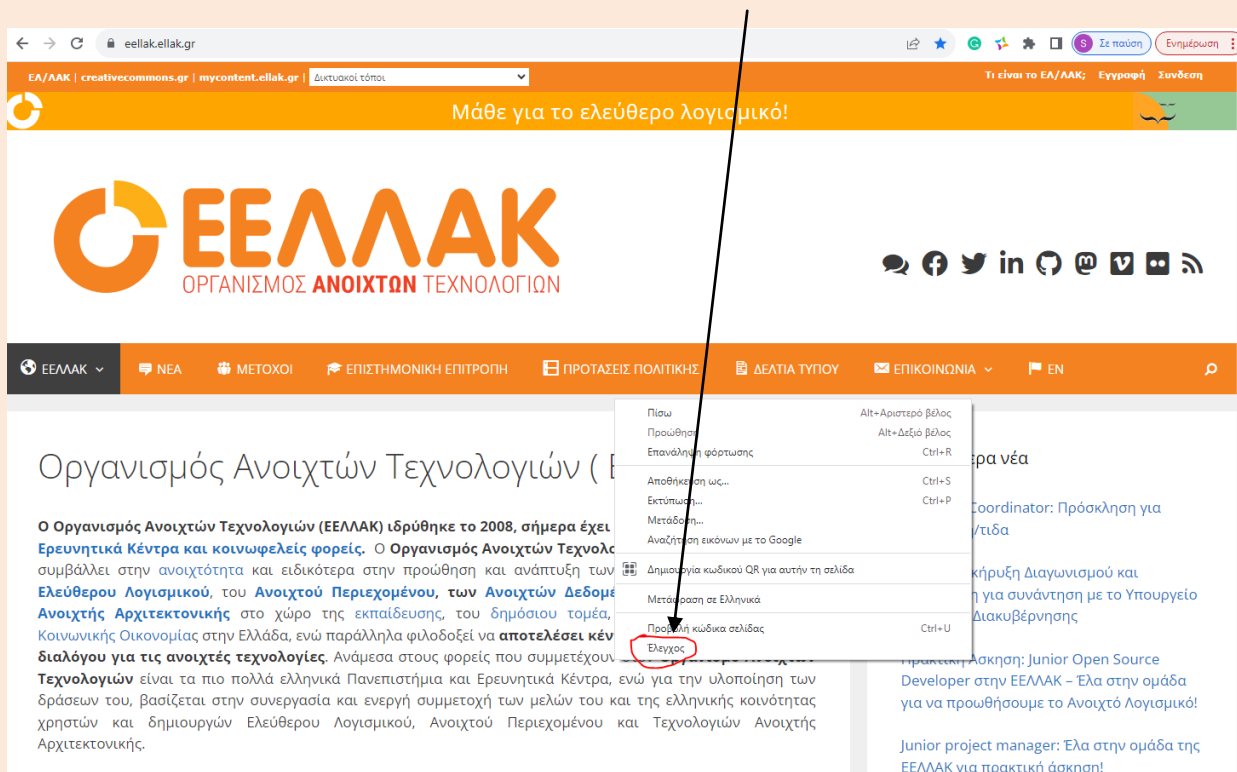
Προσαρμοστική σχεδίαση (responsive design)



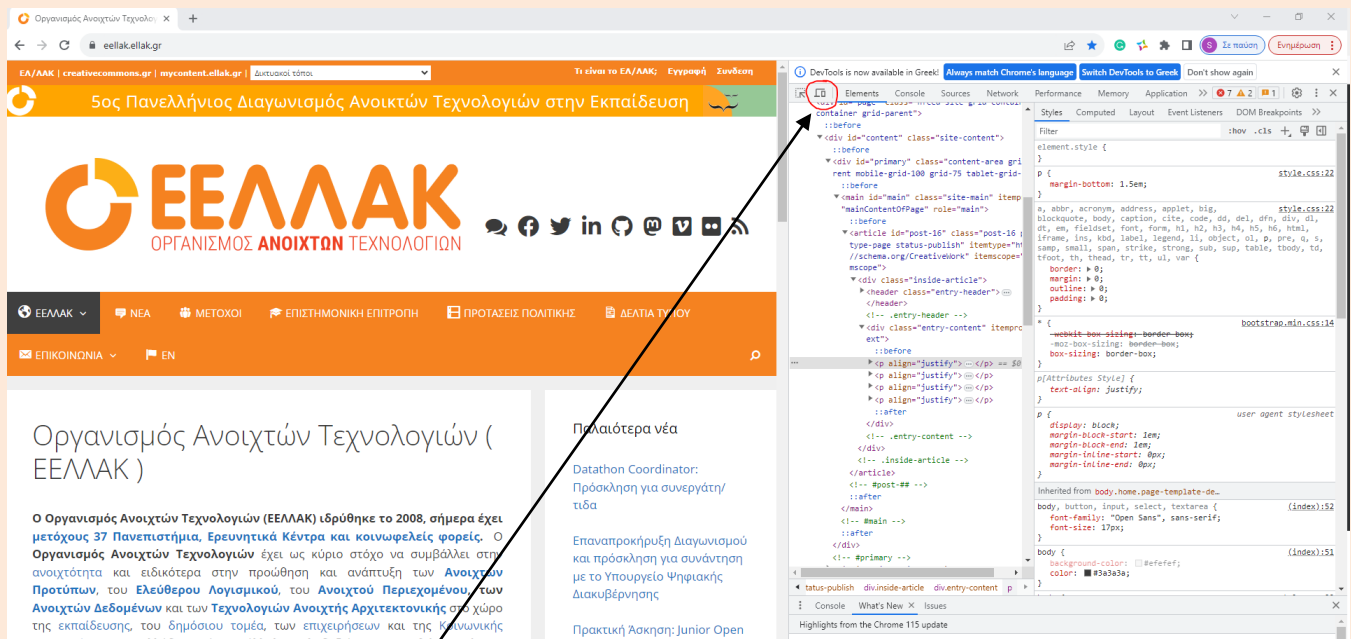
Η **προσαρμοστική σχεδίαση ιστοτόπων (RWD: Responsive Web Design)** είναι η τεχνική σχεδίασης και κατασκευής ιστοτόπων που επιδιώκει να βελτιστοποιήσει τη λειτουργία τους σε ένα μεγάλο εύρος διαφορετικών συσκευών (οθόνες υπολογιστών, τάμπλετ, κινητών, ρολογιών κλπ). Με απλά λόγια, **οι ιστοσελίδες μας πρέπει να προσαρμόζονται σε οιοδήποτε μέγεθος οθόνης**, από την οθόνη ενός ρολογιού μέχρι μιας μεγάλης τηλεόρασης. Ο όρος «**προσαρμοστική σχεδίαση**» (**responsive design**) έχει προέλθει από τον τρόπο που οι φυλλομετρητές ανταποκρίνονται στο περιβάλλον τους. Δεν αποτελεί ξεχωριστή τεχνολογία, αλλά είναι μια ορολογία για να περιγράψουμε ένα σύνολο από πρακτικές που χρησιμοποιούμε για να διαμορφώσουμε μια διάταξη ιστοσελίδας που ανταποκρίνεται στις διάφορες συσκευές που μπορεί να προβληθεί. Η **προσαρμοστική σχεδίαση** ανταποκρίνεται στις ανάγκες των χρηστών και κατά συνέπεια στις συσκευές που αυτοί χρησιμοποιούν. Σ' ένα κινητό π.χ. το περιεχόμενο παρουσιάζεται σε μία στήλη, ενώ σε ένα τάμπλετ μπορεί να προβάλλεται σε δύο στήλες.

Ένα από τα βασικά οφέλη της προσαρμοστικής σχεδίασης είναι η χρησιμοποίηση του ίδιου url και του ίδιου πηγαίου κώδικα για τον ιστότοπο μας, διαχειριζόμενοι ένα μόνο σετ περιεχομένου για όλες τις συσκευές που θα προβάλλεται. Δεν χρειάζεται δηλαδή, για κάθε διαφορετική συσκευή, να φτιάχνουμε διαφορετικό περιεχόμενο με διαφορετικό φορμάτ και διάταξη, που κάθε ένα σετ θα ικανοποιεί και διαφορετικό μέγεθος οθόνης. Με την προσαρμοστική σχεδίαση γίνεται ευκολότερη η υποστήριξη και συντήρηση του υλικού και του κώδικα που απαιτείται για την κατασκευή του ιστότοπου.

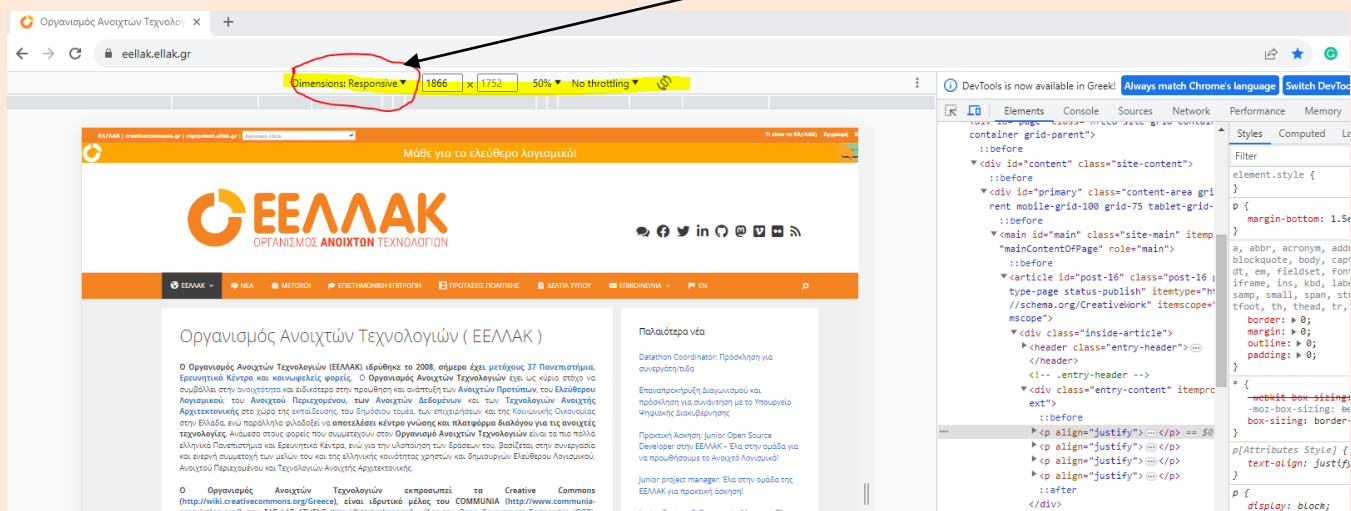
Πριν αρχίσουμε να περιγράφουμε την **προσαρμοστική σχεδίαση**, για την κατανόηση των παραπάνω ας μεταβούμε στην ιστοσελίδα <https://eellak.ellak.gr/> χρησιμοποιώντας ως φυλλομετρητή τον chrome και κάνοντας δεξί κλικ, επιλέγουμε από το πλαίσιο που ανοίγει την επιλογή **Έλεγχος**.



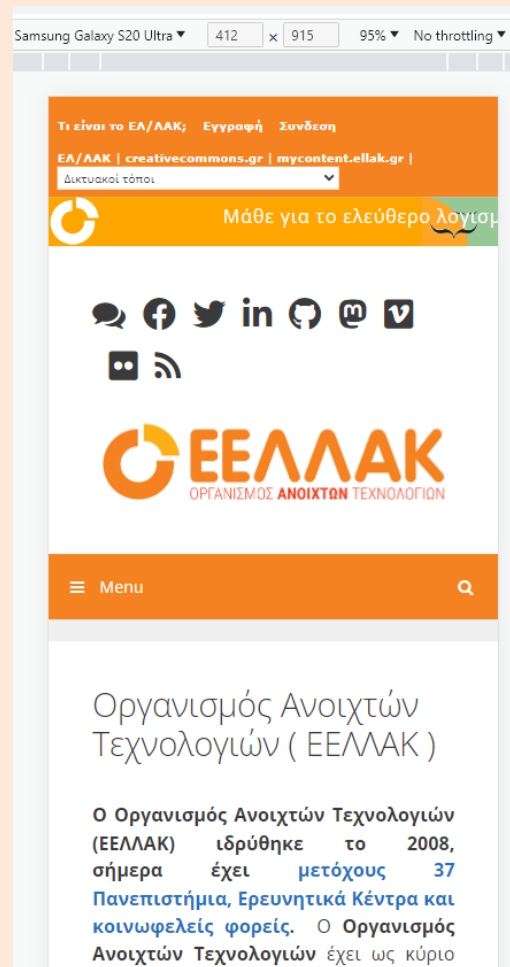
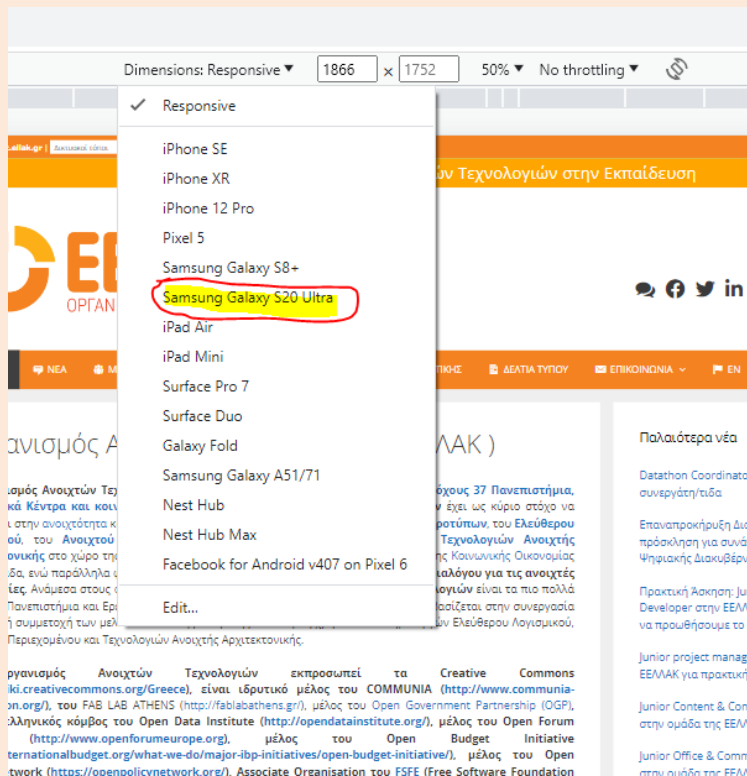
Θα παρατηρήσουμε ότι στο παράθυρο διαιρείται και εμφανίζεται δίπλα στην ιστοσελίδα μας ο πηγαίος κώδικας (κάνοντας κλικ σε κάθε ένα στοιχείο της html, παρουσιάζεται στο διπλανό παράθυρο και το στυλ που έχουμε καθορίσει γι αυτό το στοιχείο).



Κάνοντας κλικ στο εικονίδιο  στο οριζόντιο μενού παρατηρούμε την εμφάνιση μιας οριζόντιας μπάρας, πάνω από την ιστοσελίδα μας, που μας δίνει τη δυνατότητα να **επιλέξουμε** συσκευή και να δούμε πως θα παρουσιάζεται η ιστοσελίδα μας σε αυτή τη συσκευή,



Κάνοντας κλικ στο βέλος και επιλέγοντας π.χ. τη συσκευή ενός κινητού (Samsung Galaxy S20 Ultra) θα δούμε πως προσαρμόζεται η εμφάνιση της ιστοσελίδας μας στην οθόνη της συγκεκριμένης συσκευής.



Μπορείτε να δοκιμάσετε και τις υπόλοιπες επιλογές, με μια οποιαδήποτε ιστοσελίδα και να παρατηρήσετε πως εμφανίζεται η ιστοσελίδα σε διαφορετικές συσκευές και σε διάφορα μεγέθη οθονών.

Τι χρειάζεται για μια προσαρμοστική σχεδίαση

Στη συνέχεια θα γνωρίσουμε πως μπορούμε να σχεδιάσουμε μια προσαρμοστική ιστοσελίδα, αξιοποιώντας ορισμένες γνώσεις που έχουμε ήδη αποκτήσει και θα τις εμπλουτίσουμε με ορισμένες καινούργιες. Το περιεχόμενο της ιστοσελίδας μας πρέπει να είναι προσαρμοστικό, τόσο τα κείμενα, όσο και οι εικόνες που χρησιμοποιούμε. Απαραίτητο είναι να χρησιμοποιούμε **ευέλικτες μονάδες μέτρησης** για τον καθορισμό των μεγεθών των στοιχείων της ιστοσελίδας μας και η **διάταξη** των στοιχείων πρέπει να είναι **ευέλικτη**. Μερικές άλλες έννοιες που θα γνωρίσουμε στη συνέχεια, με τις οποίες μπορούμε να κάνουμε μια προσαρμοστική σχεδίαση, είναι το **viewport**, τα **Media queries** και τα **breakpoints**. Ας αναλύσουμε λοιπόν περισσότερο τον τρόπο που πρέπει να σχεδιάσουμε την ιστοσελίδα μας, ώστε να είναι προσαρμοστική.

Ευέλικτες μονάδες μέτρησης

Στην προσαρμοστική σχεδίαση δεν πρέπει να χρησιμοποιούμε σταθερές μονάδες μέτρησης μεγεθών της CSS, όπως px, cm αλλά να χρησιμοποιούμε τις σχετικές μονάδες μέτρησης όπως % και em. Αν για παράδειγμα ορίσουμε ως το εύρος μιας παραγράφου σε 70px, η εμφάνιση της σε οποιαδήποτε οθόνη θα είναι σταθερή. Σε μια οθόνη κινητού τηλεφώνου δεν θα χωράει να εμφανισθεί πλήρης στο εύρος της οθόνης με αποτέλεσμα να δημιουργείται οριζόντια κύλιση στο πλαίσιο εμφάνισης του κειμένου. Αυτό είναι άβολο για τους χρήστες, καθώς χρειάζεται να σκρολάρουν συνέχεια για να δουν το κείμενο, με

αποτέλεσμα να τους προκαλεί δυσарέσκεια. Γι' αυτό το λόγο είναι κάτι που αποφεύγεται προσαρμοστική σχεδίαση. Έχουμε ήδη γνωρίσει τη μονάδα μέτρησης μεγέθους **em**, η οποία δεν είναι σταθερή αλλά είναι σε εξάρτηση από το αντίστοιχο μέγεθος του στοιχείου γονέα (αν π.χ. οι χαρακτήρες του γονέα έχουν μέγεθος 16px και θέσουμε στο τρέχον στοιχείο `font-size: 2em`, οι χαρακτήρες του στοιχείου αυτού θα γίνουν $2 \times 16\text{px} = 32\text{px}$). Μια άλλη ευέλικτη μονάδα μέτρησης μεγέθους είναι η **rem**, η οποία είναι σε εξάρτηση από το αντίστοιχο μέγεθος του αρχικού στοιχείου (**root element**), δηλαδή του `html`. Αν έχουμε ορίσει οι χαρακτήρες του αρχικού στοιχείου να είναι 12px (`html {font-size: 12px;}`) και οι χαρακτήρες του γονέα να είναι 16px και θέσουμε στο τρέχον στοιχείο `font-size: 2rem`, οι χαρακτήρες του στοιχείου αυτού θα γίνουν $2 \times 12\text{px} = 24\text{px}$ (δεν εξαρτώνται από τον γονέα, αλλά από το αρχικό στοιχείο).

Δύο ακόμα ευέλικτες μονάδες μέτρησης μεγέθους, που είναι εξαιρετικά χρήσιμες στην προσαρμοστική σχεδίαση, είναι οι **vw** (viewport width) και **vh** (viewport height), οι οποίες λειτουργούν όπως τα ποσοστά και αναφέρονται στην τρέχουσα ορατή οθόνη. Αν για παράδειγμα ορίσουμε για ένα στοιχείο `width: 10vw`; θα καταλάβει το 10% του εύρους της διαθέσιμης ορατής οθόνης. Παρομοίως, αν ορίσουμε για ένα στοιχείο `height: 30vh`; θα καταλάβει το 30% του ύψους της διαθέσιμης ορατής οθόνης.

Προσαρμοστική σχεδίαση με ευέλικτη διάταξη στοιχείων

Μια ευέλικτη διάταξη στοιχείων, όπως το μοντέλο **Flexbox** ή το μοντέλο **Grid**, μας επιτρέπει να επιτύχουμε μια προσαρμοστική σχεδίαση της ιστοσελίδας μας.

Το μοντέλο **Flexbox** είναι ιδανικό όταν τα στοιχεία της ιστοσελίδας μας έχουν διαφορετικά μεγέθη και επιθυμούμε να εμφανίζονται βολικά σε μία ή περισσότερες σειρές με τα στοιχεία να παίρνουν αναλογικά τον χώρο που τους αναλογεί (τα μικρότερου μεγέθους λιγότερο χώρο και τα μεγαλύτερου μεγέθους περισσότερο). Τα στοιχεία θα εμφανίζονται σε μια σειρά (ή στήλη, αν καθορίσουμε κατακόρυφη ροή) και μπορούν να αναδιπλώνονται σε περισσότερες, όσο ο διαθέσιμος χώρος μειώνεται. Επίσης, μπορούν με τον κατάλληλο σχεδιασμό να μεγεθύνονται ή να συρρικνώνονται, ώστε να καταλαμβάνουν ομοιόμορφα τον χώρο που τους διατίθεται για την εμφάνισή τους.

Η διάταξη **Grid** υπερτερεί όταν έχουμε **δισδιάστατη τοποθέτηση** των στοιχείων, καθώς δημιουργεί εύκολα ένα πλέγμα. Με χρησιμοποίηση του `fr` ως μονάδας μέτρησης του μεγέθους των στηλών, γίνεται με αναλογικό τρόπο η κατανομή του διαθέσιμου χώρου σε αυτές. Αυξομειώνοντας τον διαθέσιμο χώρο, η αναλογία αυτή διατηρείται.

Καθορισμός του viewport

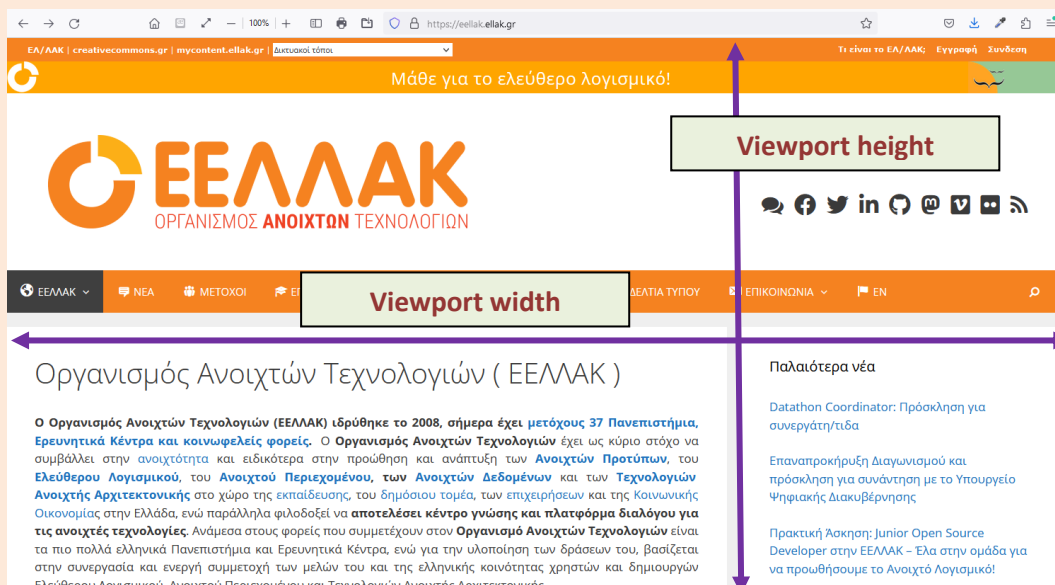
Το **viewport**, είναι το μέρος του φυλλομετρητή στο οποίο είναι ορατή η ιστοσελίδα μας. Όταν το εύρος του περιεχομένου της ιστοσελίδας είναι μικρότερο ή μεγαλύτερο από το εύρος του viewport ενδέχεται να μην αποδίδεται σωστά σε μικρές οθόνες, όπως αυτές των έξυπνων κινητών. Αν, για παράδειγμα είναι πολύ μεγάλο μπορεί να προξενεί μεγάλη σμίκρυνση του κειμένου και να διαβάζεται με δυσκολία. Ο καθορισμός του εύρους (width) και της κλίμακας (scaling) του viewport γίνεται με τη χρήση του **viewport meta tag** content, το οποίο περιέχει μια λίστα από χαρακτηριστικά και τις τιμές τους, χωριζόμενα με κόμμα. (τα κυριότερα είναι το width και το initial-scale). Τοποθετείται στο τμήμα head του αρχείου `html` Μερικά παραδείγματα:

```
<meta name="viewport" content="width=1024, initial-scale=1">
```

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


Στο πρώτο παράδειγμα, για μια μη προσαρμοστική ιστοσελίδα με εύρος 980px, έχουμε ορίσει το εύρος σε 1024 ώστε η οθόνη να έχει ένα μικρό αέρα. Στο δεύτερο παράδειγμα, αντιστοιχεί σε ιστοσελίδα με προσαρμοστική σχεδίαση, όπου θέλουμε το εύρος του viewport να είναι ίσο με το εύρος της οθόνης της συσκευής (**device-width**) που χρησιμοποιούμε. Επίσης, θέλουμε η αρχική εμφάνιση της ιστοσελίδας μας να είναι με το αρχικά προκαθορισμένο ζούμ του φυλλομετρητή, γι αυτό θέτουμε, **initial-scale=1**.

Με την τιμή **device-width**, η ιστοσελίδα μας θα ματσάρει το εύρος της οθόνης σε ανεξάρτητα pixels (τα οποία σε μια οθόνη υψηλής ανάλυσης μπορεί να αποτελούνται από πολλά φυσικά pixels). Αυτό επιτρέπει στην ιστοσελίδα να αναδιατάξει το περιεχόμενο της, ώστε να προσαρμόζεται σε διαφορετικά μεγέθη οθονών (είτε στην οθόνη ενός κινητού τηλεφώνου, είτε σε μια μεγάλη οθόνη υπολογιστή). Μερικοί φυλλομετρητές όταν περιστρέφουμε την οθόνη (ειδικά στα κινητά τηλέφωνα) κρατάνε σταθερό το εύρος της σελίδας και δεν αναδιατάσσουν το περιεχόμενο να γεμίσει την οθόνη. Με την τιμή **initial-scale=1**, καθοδηγούμε τους φυλλομετρητές να έχουν μια σχέση 1:1 μεταξύ των CSS pixels και των ανεξάρτητων pixels της συσκευής, ανεξάρτητα από τον προσανατολισμό της οθόνης. Με τον τρόπο αυτό επιτυγχάνουμε να εκμεταλλευόμαστε πάντοτε ολόκληρο το εύρος της οθόνης (οπότε γεμίζει ολόκληρη η οθόνη, όταν περιστρέφουμε το κινητό ή το τάμπλετ μας).



Προσαρμογή του περιεχομένου στο viewport

Όταν το περιεχόμενο μας περιέχει στοιχεία με σταθερά μεγέθη (το πιο συχνό είναι να έχει μια εικόνα με σταθερό πλάτος) είναι πιθανό να μην ταιριάζει με αυτά που έχουμε καθορίσει στο **viewport meta tag**. Αποτέλεσμα να χρειάζεται οριζόντια κύλιση του περιεχομένου για να μπορεί να αναγνωσθεί από τους χρήστες. Η αποφυγή τους επιτυγχάνεται με τον καθορισμό του **max-width** αντί του width στα στοιχεία μας και μάλιστα με ευέλικτη μονάδα μέτρησης μεγέθους, όπως σε 100%. Π.χ.

```
img { max-width: 100%; display: block; }
```

Όταν χρησιμοποιούμε το max-width, επικαλύπτεται η διάσταση width που πιθανόν έχουμε καθορίσει στο tag , γιατί οι φυλλομετρητές χρησιμοποιούν την πληροφορία αυτή για να δεσμεύσουν τον χώρο για την εικόνα προτού την φορτώσουν.

Προσαρμοστική σχεδίαση με χρήση Media queries και breakpoints

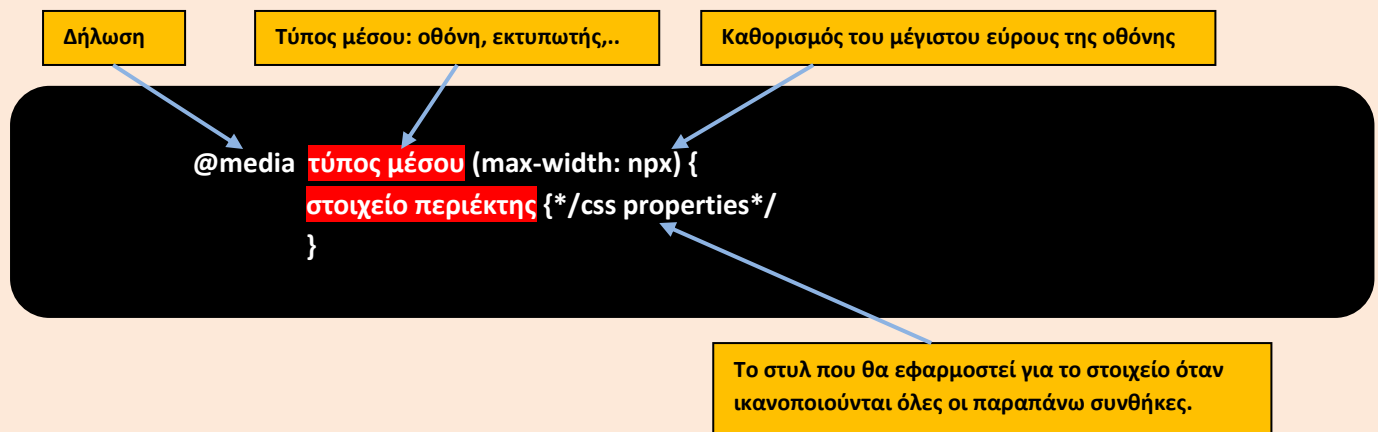
Όταν χρειαζόμαστε να εφαρμόσουμε περισσότερες αλλαγές στη διάταξη μας ώστε να υποστηρίξουμε ένα μέγεθος οθόνης, χρησιμοποιούμε τα **Media queries**, τα οποία είναι απλά φίλτρα που μπορούμε να εφαρμόσουμε στα στυλ της CSS.

Ένα απλό παράδειγμα είναι όταν θέλουμε να καθορίσουμε ένα διαφορετικό στυλ για εκτύπωση της ιστοσελίδας μας. Χρησιμοποιούμε ένα media query για να συμπεριλάβουμε το στυλ για εκτύπωση στο κύριο stylesheet.

```
@media print {  
  /* print style */  
}
```

Στην προσαρμοστική σχεδίαση χρησιμοποιούμε τα **Media queries**, στα χαρακτηριστικά της συσκευής, ώστε να καθορίσουμε ένα διαφορετικό στυλ για διαφορετικά μεγέθη οθονών. Αυτό που συνήθως ανιχνεύουμε είναι το μέγεθος της οθόνης της συσκευής και χρησιμοποιούμε **Media queries**, βασισμένα στο εύρος του **viewport**. Χρησιμοποιούμε το max-width σε ένα media query και καθορίζουμε, για την τιμή που αντιστοιχεί στην οθόνη της συσκευής, το στυλ που επιθυμούμε να έχει το στοιχείο όταν εμφανίζεται σε αυτή τη συσκευή.

Τα **Media queries** είναι συναρτήσεις που χρησιμοποιούμε για να διαμορφώνουμε κάποιο στυλ ανάλογα με το αν ισχύει η συνθήκη που περιέχουν. Αποτελούν το βασικό εργαλείο στην προσαρμοστική σχεδίαση ιστοσελίδων, γιατί επιτρέπουν στον συγγραφέα να καθορίζει το στυλ της ιστοσελίδας βασισμένο στις διαστάσεις της οθόνης της συσκευής. Το περισσότερο χρησιμοποιούμενο είναι το εύρος **max-width**. Συντάσσεται ως εξής:



Το παραπάνω σημαίνει ότι θα εφαρμοσθούν οι ρυθμίσεις που υπάρχουν ανάμεσα στις αγκύλες, αν το μέγιστο πλάτος είναι n pixels, το οποίο είναι ένα **breakpoint** (δηλαδή θα εφαρμοσθούν αν η ιστοσελίδα μας εμφανισθεί σε οθόνη κινητού ή τάμπλετ). Π.χ.

```
@media (max-width: 600px) {  
  body {background-color: red;  
  Color: white;  
}
```

```
@media (min-width: 601px) and (max-width: 900px) {  
  body {background-color: green;  
  Color: blue;  
}
```

Αν το εύρος του παραθύρου του φυλλομετρητή μας γίνει μικρότερο από 600px, θα γίνει το υπόβαθρο της ιστοσελίδας μας κόκκινο και οι χαρακτήρες άσπροι. Αν το εύρος του παραθύρου του φυλλομετρητή μας γίνει μικρότερο από 900px και μεγαλύτερο από 600 px (δηλαδή μεταξύ 601 και 900 pixels), θα γίνει το υπόβαθρο της ιστοσελίδας μας πράσινο και οι χαρακτήρες μπλε. Αν δεν αναφέρουμε τύπο μέσου, **η προκαθορισμένη τιμή είναι screen**.

Τα **breakpoints** είναι τιμές που βάζουμε στον κώδικα μας, ώστε η ιστοσελίδα μας να προσαρμόζεται σε αυτά σημεία ανάλογα με το μέγεθος της οθόνης της συσκευής. Παρακάτω παραθέτουμε ορισμένες τιμές, που αφορούν σε breakpoints, διαφόρων συσκευών:

Οθόνη	breakpoints
Smart phones	320px – 480px
Tablets	481px – 768px
Laptops / Small Screens	769px – 1024px
Desktops / Large Screens	1025px – 1200px
TV / Extra Large Screens	1201px - ...

Χρησιμοποιώντας media queries για διαφορετικά breakpoints, μπορούμε να αλλάζουμε το στυλ για τα αρχικά στοιχεία (root elements). Χρησιμοποιώντας και μονάδες rem, μπορούμε να προσαρμόζουμε το περιεχόμενο μας σε διάφορα μεγέθη οθονών και να κάνουμε προσαρμοστική την ιστοσελίδα μας.

Μπορούμε να χρησιμοποιούμε πολλαπλά media queries για να διαμορφώνουμε κατάλληλα στυλ ώστε να προσαρμόζεται η ιστοσελίδα μας σε διάφορα μεγέθη οθονών. Κάθε media query ενεργοποιείται σε κάποιο συγκεκριμένο σημείο που είναι γνωστό ως **breakpoint**.

Πρέπει να αποφεύγουμε να ορίζουμε στυλ για συγκεκριμένες συσκευές ή προϊόντα, γιατί θα κάνει τη συντήρηση τους εφιαλτική. Μια καλή πρακτική είναι να σχεδιάζουμε μια διάταξη ιστοσελίδας με μία στήλη για συσκευές με μικρές οθόνες όπως τα έξυπνα κινητά και στη συνέχεια να σχεδιάζουμε για ευρύτερες οθόνες (στα βασικά breakpoints) με διάταξη που περιλαμβάνει πολλαπλές στήλες. Μια τέτοια προσέγγιση είναι γνωστή ως **mobile first** σχεδίαση.

Μια απλή ιστοσελίδα html είναι από τη φύση της προσαρμοστική στις διάφορες οθόνες που προβάλλεται. Η διαμόρφωση του στυλ της μπορεί να διαταράσσει την προσαρμοστικότητα της, οπότε είναι χρήσιμο να τηρούμε την εφαρμογή όσων αναφέραμε παραπάνω. Όταν χρησιμοποιούμε στην ιστοσελίδα μας πολυμέσα (εικόνες, βίντεο) πρέπει να διασφαλίζουμε ότι δεν ξεπερνούν τις διαστάσεις του περιέκτη τους. Χρησιμοποιούμε για τον σκοπό αυτό την ιδιότητα **max-width** με τιμές ευέλικτες (**img, picture, video {max-width: 100%;}**).

Μερικές μέθοδοι διάταξης, όπως Grid και Flexbox, που έχουμε γνωρίσει, είναι από τη φύση τους προσαρμοστικές. Γενικά, το Flexbox είναι εξαιρετικό όταν έχουμε στοίχιση των στοιχείων προς μια κατεύθυνση, ενώ το Grid υπερτερεί όταν έχουμε δισδιάστατη τοποθέτηση των στοιχείων.

Προσαρμογή του περιεχομένου

Όταν ακολουθούμε μια προσαρμοστική σχεδίαση πρέπει να φροντίζουμε να προσαρμόζουμε και το περιεχόμενο μας ανάλογα με το μέγεθος της οθόνης προβολής. Μερικές χρήσιμες συμβουλές είναι οι ακόλουθες:

Όταν έχουμε στην ιστοσελίδα μας μια μπάρα πλοήγησης (navigation bar), η εμφάνιση της σε μεγάλες οθόνες είναι καλό να είναι σε οριζόντια διάταξη, ενώ όταν προβάλλεται σε μικρές οθόνες (όπως των κινητών τηλεφώνων) την προσαρμόζουμε σε κατακόρυφη διάταξη.

Αν έχουμε στην ιστοσελίδα μας κάποιο μενού, σε πολύ μικρές οθόνες το αποκρύπτουμε και στη θέση του συνήθως βάζουμε τη λέξη: Μενού ή κάποιο σχετικό σύμβολο, ώστε αν κάνει ο χρήστης κλικ πάνω του να εμφανίζεται αναλυτικά το μενού με τις επιλογές του.

Όταν η οθόνη μας είναι μεγάλη, συνήθως το περιεχόμενο μας διατάσσεται σε περισσότερες από μία στήλες, ενώ σε μικρές οθόνες παρουσιάζεται σε μία στήλη.

Σε ότι αφορά σε εικόνες, μπορούμε να χρησιμοποιούμε διαφορετικές εκδόσεις της ίδιας εικόνας σε διαφορετικά μεγέθη οθονών. Χρησιμοποιούμε το στοιχείο `picture` για να έχουμε καλύτερη διαχείριση τους. Το στοιχείο `picture` μας επιτρέπει να εμφανίζουμε διαφορετικές εικόνες για διαφορετικές συσκευές ή για διαφορετικά μεγέθη οθονών. Το στοιχείο `picture` περιέχει ένα ή περισσότερα στοιχεία `source` τα οποία στο χαρακτηριστικό τους `srcset` καθορίζεται διαφορετικό αρχείο εικόνας και στο χαρακτηριστικό τους `media` καθορίζεται τότε το συγκεκριμένο αρχείο είναι το πλέον κατάλληλο. Π.χ.

```
<picture>
  <source media="(min-width: 600px)" srcset="img1.jpg">
  <source media="(min-width: 460px)" srcset="img2.jpg">
</picture>
```

Όταν έχουμε κείμενα, αυτό που συνιστάται σε κάθε γραμμή είναι να υπάρχουν γύρω στους 80 χαρακτήρες (περίπου 10 λέξεις). Όταν έχουμε υπέρβαση, προτείνεται να βάλουμε ένα `breakpoint` και να καθορίσουμε κατάλληλα το εύρος του περιέκτη του κειμένου. Π.χ.

```
@media (min-width: 575px) {
  article {
    width: 550px;
    margin-left: auto;
    margin-right: auto;
  }
}
```

Στο παραπάνω παράδειγμα χρησιμοποιούμε ένα `media query` για να ανιχνεύσουμε αν η οθόνη που θα προβληθεί η ιστοσελίδα μας είναι πάνω από 575px. Τότε καθορίζουμε στον περιέκτη του κειμένου μας (που είναι ένα άρθρο) να έχει σταθερό πλάτος 550px (`width: 550px;`) και τα περιθώρια, δεξιά και αριστερά να καθορίζονται αυτόματα. Με τον τρόπο αυτό εξασφαλίζουμε ότι το κείμενο μας δεν θα ξεχειλώσει υπερβολικά και θα γίνει δυσανάγνωστο.