ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Προγράμματα Συμπληρωματικής Εκπαίδευσης

Με τη χρήση καινοτόμων μεθόδων εξ αποστάσεως εκπαίδευσης

1937 - ΔΗΜΙΟΥΡΓΙΑ RESPONSIVE ΣΕΛΙΔΩΝ ΜΕ HTML5 ΚΑΙ CSS

© Copyright 2016, Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών, Π. Ε. Πετράκης.

Η έντυπη, ηλεκτρονική και γενικά κατά οποιοδήποτε τρόπο αναπαραγωγή, δημοσίευση ή χρησιμοποίηση όλου ή μέρους του υλικού έργου αυτού, απαγορεύεται χωρίς την έγγραφη έγκριση του κατόχου των πνευματικών δικαιωμάτων του έργου.

Το παρόν έντυπο αποτελεί τμήμα του εκπαιδευτικού υλικού του μαθήματος **"Δημιουργία Responsive Σελίδων με HTML5 και CSS"**. Αποτελεί απαραίτητο συμπλήρωμα του ηλεκτρονικού υλικού που βρίσκεται στην πλατφόρμα και αναπόσπαστο κομμάτι της εκπαιδευτικής διαδικασίας.

Στην πρώτη διδακτική ενότητα, οι εκπαιδευόμενοι θα προσεγγίσουν και θα κατανοήσουν τον σκοπό που υπηρετεί μία σελίδα Html καθώς και τη δομή της. Στη συνέχεια θα παρουσιαστούν text editors για Windows και Mac OS με τη βοήθεια των οποίων οι εκπαιδευόμενοι θα μπορούν να γράψουν κώδικα για τις Html σελίδες τους.

Με τη δεύτερη διδακτική ενότητα οι εκπαιδευόμενοι θα μάθουν να εισαγάγουν κείμενο με τα κατάλληλα Tags και στη συνέχεια να επεμβαίνουν στην εμφάνιση του κειμένου με τη βοήθεια των CSS ιδιοτήτων.

Στην τρίτη διδακτική ενότητα οι εκπαιδευόμενοι θα μάθουν να εμπλουτίζουν τις σελίδες τους με multimedia στοιχεία όπως εικόνες, videos και ήχους σε διάφορα formats. Επίσης οι εκπαιδευόμενοι θα μάθουν να παραμετροποιούν τα multimedia στοιχεία αλλάζοντας τη θέση και τις διαστάσεις τους. Στη συνέχεια οι εκπαιδευόμενοι θα μάθουν να χρησιμοποιούν το Html Canvas για να σχεδιάζουν εντός μιας Html σελίδας. Τέλος, θα μάθουν να εισάγουν links προς άλλες σελίδες, videos από το You Tube κτλ.

Στην τέταρτη διδακτική ενότητα οι εκπαιδευόμενοι θα μάθουν τις δυνατότητες που προσφέρει το CSS έτσι ώστε να επεμβαίνουν στην εμφάνιση των στοιχείων μιας σελίδας, όπως για παράδειγμα το μέγεθος και το χρώμα μιας γραμματοσειράς, η επιλογή χρώματος background κ.α. Θα μάθουν επίσης, με τη βοήθεια του CSS, να δημιουργούν responsive δομές σελίδων με δύο διαφορετικούς τρόπους. Ο πρώτος τρόπος είναι η δημιουργία σελίδων με το flexbox, ενώ ο δεύτερος τρόπος είναι με τη βοήθεια του καινούργιου CSS Grid Layout. Σε κάθε περίπτωση θα αναλυθεί εκτενώς ο τρόπος δημιουργίας navigation για να μπορεί ο χρήστης να μεταβαίνει από μία σελίδα σε μία άλλη.

Με την πέμπτη διδακτική ενότητα οι εκπαιδευόμενοι θα είναι σε θέση να εφαρμόζουν όλα όσα διδάχθηκαν στο μάθημα της Html. Πιο συγκεκριμένα θα αναλυθούν βήμα προς βήμα η δημιουργία templates βασισμένα στο flexbox και στο CSS Grid Layout.

Τη συγγραφή του συγκεκριμένου εκπαιδευτικού υλικού πραγματοποίησε ο κ. **Βαγγέλης Στελλάτος**, εξωτερικός συνεργάτης του Προγράμματος Συμπληρωματικής εξ Αποστάσεως Εκπαίδευσης του Εθνικού και Καποδιστριακού Πανεπιστημίου Αθηνών.

Π Ε Ρ Ι Ε Χ Ο Μ Ε Ν Α

ΔΙΔΑΚΤΙΚΗ ΕΝΟΤΗΤΑ 1. HTML5 BASICS **5**
ΕΙΣΑΓΩΓΙΚΕΣ ΠΑΡΑΤΗΡΗΣΕΙΣ ***7***
ΥΠΟΕΝΟΤΗΤΑ 1. HOME PAGE-CSS ΚΑΙ HEADER *8*
ΥΠΟΕΝΟΤΗΤΑ 2. TEXT EDITORS *13*
ΥΠΟΕΝΟΤΗΤΑ 3. HELLO HTML - ΔΗΜΙΟΥΡΓΩΝΤΑΣ HTML ΣΕΛΙΔΕΣ *22*
ΣΥΝΟΨΗ ***46***

ΔΙΔΑΚΤΙΚΗ ΕΝΟΤΗΤΑ 2. ΚΕΙΜΕΝΟ ΚΑΙ ΠΙΝΑΚΕΣ ΣΕ HTML ΣΕΛΙΔΕΣ **47**
ΕΙΣΑΓΩΓΙΚΕΣ ΠΑΡΑΤΗΡΗΣΕΙΣ ***49***
ΥΠΟΕΝΟΤΗΤΑ 1. ΓΡΑΦΟΝΤΑΣ ΚΕΙΜΕΝΟ ΣΤΗΝ HTML5 ..... *50*
ΥΠΟΕΝΟΤΗΤΑ 2. ΜΟΡΦΟΠΟΙΗΣΗ ΚΕΙΜΕΝΟΥ ΚΑΙ ΛΙΣΤΕΣ *76*
ΥΠΟΕΝΟΤΗΤΑ 3. ΠΙΝΑΚΕΣ ΚΑΙ FORMS *109*
ΣΥΝΟΨΗ ***128***

ΔΙΔΑΚΤΙΚΗ ΕΝΟΤΗΤΑ 3. ΕΝΣΩΜΑΤΩΣΗ MULTIMEDIA ΣΕ HTML ΣΕΛΙΔΕΣ **129**
ΕΙΣΑΓΩΓΙΚΕΣ ΠΑΡΑΤΗΡΗΣΕΙΣ ***131***
ΥΠΟΕΝΟΤΗΤΑ 1. HTML ΚΑΙ ΕΙΚΟΝΕΣ *132*
ΥΠΟΕΝΟΤΗΤΑ 2. AUDIO, VIDEO ΚΑΙ HTML CANVAS *158*
ΥΠΟΕΝΟΤΗΤΑ 3. ΚΑΜΠΥΛΕΣ ΣΤΟ HTML CANVAS ΚΑΙ LINKS *177*
ΣΥΝΟΨΗ ***216***

ΔΙΔΑΚΤΙΚΗ ΕΝΟΤΗΤΑ 4. CSS3, FLEXBOX ΚΑΙ CSS GRID SYSTEM **219**
ΕΙΣΑΓΩΓΙΚΕΣ ΠΑΡΑΤΗΡΗΣΕΙΣ ***221***
ΥΠΟΕΝΟΤΗΤΑ 1. CSS *222*
ΥΠΟΕΝΟΤΗΤΑ 2. CSS ΚΑΙ FLEXBOX *250*
ΥΠΟΕΝΟΤΗΤΑ 3. FLEXBOX ΚΑΙ GRID SYSTEM *273*
ΣΥΝΟΨΗ ***321***

ΔΙΔΑΚΤΙΚΗ ΕΝΟΤΗΤΑ 5. ΔΗΜΙΟΥΡΓΙΑ TEMPLATES ΜΕ ΤΗ ΒΟΗΘΕΙΑ ΤΗΣ HTML5 ..... **323**
ΕΙΣΑΓΩΓΙΚΕΣ ΠΑΡΑΤΗΡΗΣΕΙΣ ***325***
ΥΠΟΕΝΟΤΗΤΑ 1. MEDIA QUERIES ΚΑΙ CSS ANIMATION *326*
ΥΠΟΕΝΟΤΗΤΑ 2. GRID SYSTEM TEMPLATE *343*
ΥΠΟΕΝΟΤΗΤΑ 3. FLEXBOX TEMPLATE *366*
ΣΥΝΟΨΗ ***377***

ΒΙΒΛΙΟΓΡΑΦΙΑ *378*

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Προγράμματα Συμπληρωματικής Εκπαίδευσης

Με τη χρήση καινοτόμων μεθόδων εξ αποστάσεως εκπαίδευσης

1937 - ΔΗΜΙΟΥΡΓΙΑ RESPONSIVE ΣΕΛΙΔΩΝ ΜΕ HTML5 ΚΑΙ CSS

ΔΙΔΑΚΤΙΚΗ ΕΝΟΤΗΤΑ 1. HTML5 BASICS

Εισαγωγικές Παρατηρήσεις

Στην υποενότητα 1 θα γίνει εισαγωγή στην Html και το CSS. Επίσης θα τονισθεί η σπουδαιότητα των κανόνων web usability που θα πρέπει να πληροί το website. Τέλος θα εξηγηθεί η σπουδαιότητα του Responsive Design.

Στην υποενότητα 2 θα παρουσιαστούν διάφοροι editors για Mac και για Windows καθώς και οι δυνατότητες που προσφέρουν. Επίσης θα αναλυθούν τα στοιχεία p και τα headings h1-h6 της Html.

Στην υποενότητα 3 θα αναλυθεί η δημιουργία buttons για τις ιστοσελίδες μας. Επίσης θα επεξηγηθεί επακριβώς η δομή ενός website. Τέλος θα αναλυθεί βήμα προς βήμα η δημιουργία ενός πολύ απλού website με τρεις σελίδες και στοιχεία p και h.

ΥΠΟΕΝΟΤΗΤΑ 1. HOME PAGE-CSS ΚΑΙ HEADER

Η Html είναι μία γλώσσα προγραμματισμού για να ενσωματώνουμε κείμενο και Multimedia στοιχεία σε μία Web σελίδα. Τα αρχικά Html προέρχονται από τις λέξεις Hypertext Markup Language. Η γλώσσα Html βρίσκεται στην Πέμπτη έκδοσή της (Html5) και είναι μία γλώσσα σήμανσης υπερκειμένου που εμπεριέχεται στις web σελίδες ενός website ή μιας web εφαρμογής.

Η γλώσσα Html εκτελείται από τους Browsers όπως οι Chrome, Mozilla, Microsoft Edge, Safari κα. Ένα αρχείο Html μπορεί να εκτελεστεί ακόμη και αν αυτό είναι απλά αποθηκευμένο σε κάποια διαδρομή του σκληρού μας δίσκου.

Σε κανονικές συνθήκες τα αρχεία Html φιλοξενούνται σε κάποιο απομακρυσμένο server computer. Μόλις ο χρήστης πληκτρολογήσει τη διεύθυνση μιας σελίδας web τότε ο Browser (client) ζητά το αντίστοιχο αρχείο από τον αντίστοιχο server. Με τα δεδομένα που λαμβάνει από το server συγκεντρώνει τις απαραίτητες πληροφορίες (κώδικας και βοηθητικά αρχεία) για να εμφανίσει στην οθόνη του χρήστη το περιεχόμενο της σελίδας.

Όπως αναφέρθηκε προηγουμένως η Html είναι μία γλώσσα σήμανσης. Αυτό σημαίνει ότι με κατάλληλες ετικέτες (tags) η Html δίνει την πληροφορία στο Browser για τον τύπο του περιεχομένου. Κάθε τύπος περιεχομένου αποτελεί και ένα στοιχείο της Html. Συνήθως η σήμανση του περιεχομένου γίνεται με δύο ετικέτες tags της Html. Η ετικέτα έναρξης σηματοδοτεί την αρχή του στοιχείου ενώ η ετικέτα λήξης το πέρας του στοιχείου. Για παράδειγμα έστω ένα στοιχείο p (paragraph):

```
<p> This is my First Web Page  </p>
```

Η ετικέτα έναρξης περιλαμβάνει το Html element p μεταξύ των τελεστών '<' και '>'. Αντιστοίχως η ετικέτα λήξης περιλαμβάνεται μεταξύ των τελεστών '</' και '>'.

Ενώ η Html προσδιορίζει τον τύπο του περιεχομένου η CSS (Cascade style sheets) η οποία συνοδεύει την Html αναλαμβάνει το στυλ των στοιχείων της Html δηλαδή την εμφάνιση των στοιχείων της Html όπως στο ακόλουθο παράδειγμα:

```
<p style="color:blue;"> This is my First Web Page  </p>
```

Στο παραπάνω παράδειγμα οι ετικέτες και προσδιορίζουν ότι το περιεχόμενο είναι τύπου p Html element, ενώ η χαρακτηριστική λέξη style με το περιεχόμενο που ακολουθεί προσδίδει το στυλ εμφάνισης της Html παραγράφου. Πιο συγκεκριμένα με τη λέξη color προσδιορίζεται το μπλε χρώμα ως το χρώμα εμφάνισης της γραμματοσειράς που θα χρησιμοποιηθεί στην εμφάνιση της παραγράφου από τον Browser.
Η τωρινή έκδοση της CSS είναι η έκδοση 3 (CSS3) και μπορεί να αλλάξει το είδος και το χρώμα της γραμματοσειράς, το χρώμα του φόντου καθώς και να δώσει στον Developer πλήθος από δυνατότητες που αφορούν στο στυλ και τη δομή μιας Html σελίδας.

1.1 Στα ενδότερα της Html

Η Html είναι μία γλώσσα εύκολη στην κατανόηση και δεν κάνει διάκριση μεταξύ κεφαλαίων και πεζών χαρακτήρων. Δηλαδή είναι μια not case sensitive language και για παράδειγμα την ετικέτα έναρξης για την παράγραφο μπορεί ο Browser να τη διαβάσει ως όπως επίσης και <P>. Γενικά είναι καλό να χρησιμοποιούμε πεζούς χαρακτήρες για τη σήμανση έτσι ώστε να είναι πιο εύκολα αναγνώσιμος ο κώδικας Html που γράφουμε.
Η γλώσσα Html εκτός από not case sensitivity language είναι μία γλώσσα της οποίας τα λάθη ελάσσονος σημασίας «συγχωρούνται» από τους Browsers. Δηλαδή κατά την εκτέλεση του κώδικα από το Browser αρκετά λάθη παρακάμπτονται έτσι ώστε να μην εμποδιστεί η εμφάνιση του περιεχομένου της σελίδας. Έτσι αν το λάθος δεν είναι κρίσιμο, ο Browser το παραλείπει και τρέχει τις υπόλοιπες γραμμές του κώδικα.

Σημείωση

Ο κώδικας Html είναι ίδιος για Mac και για Windows. Αυτό που αλλάζει είναι τα εργαλεία για τη συγγραφή και την επεξεργασία του κώδικα Html. Κάποια εργαλεία διατίθενται και στις δύο εκδόσεις ενώ κάποια άλλα υπάρχουν αποκλειστικά για Mac ή αποκλειστικά για Windows.

Αρχεία Html και εργαλεία για τη συγγραφή Html κώδικα

Τα αρχεία Html έιναι αρχεία τύπου txt με τη μόνη διαφορά ότι έχουν επέκταση .html και όχι .txt. Εφόσον λοιπόν τα Html αρχεία είναι αρχεία τύπου txt μπορούμε να γράψουμε Html κώδικα σε ένα απλό text editor.

Εκτός από τον απλό text editor υπάρχουν και πιο εξιδεικευμένοι Html editors που παρέχουν παραπάνω λειτουργίες για την Html.

Γενικοί κανόνες που πρέπει να προσέξει κάποιος πριν ξεκινήσει κάποιος να ασχολείται με την Html

1. Στα ονόματα αρχείων γράφουμε αποκλειστικά Λατινικούς χαρακτήρες.

Για παράδειγμα μπορούμε να γράψουμε example1, paradeigma1 αλλά όχι παράδειγμα1.

2. Στα ονόματα των αρχείων δεν αφήνουμε κενά. Αντί για το κενό γράφουμε το χαρακτήρα underscore (_).

Για παράδειγμα δεν γράφουμε example 1 αλλά γράφουμε example_1 ή example1.

3. Δεν χρησιμοποιούμε ποτέ διαδρομές μέσα από το δικό μας Desktop και γενικά διαδρομές μέσα από το φάκελο user.

Παράδειγμα

C:\Users\....\Desktop\hello.Html

4. Δεν αντιγράφουμε ποτέ κώδικα από pdf, Word. Αντιγράφουμε κώδικα μέσα από txt ή Html editors. Αν αντιγράψουμε κώδικα από pdf πολλοί χαρακτήρες θα αντιγραφούν λάθος, οπότε και θα έχουμε πρόβλημα στον κώδικα.
5. Είμαστε προσεκτικοί στην αλλαγή της γλώσσας. Ο Χαρακτήρας Β με Αγγλική Γλώσσα είναι διαφορετικός από τον χαρακτήρα Β της Ελληνικής γλώσσας.

1.1.1 Καλές πρακτικές

Ειδικά στα πρώτα βήματα της Html και γενικά κάθε γλώσσας Προγραμματισμού θα πρέπει να κάνουμε μικρά βήματα. Σε κάθε μικρό βήμα δοκιμάζουμε το αποτέλεσμα. Αν δεν είμαστε ικανοποιημένοι με το αποτέλεσμα κάνουμε undo στον editor και συνεχίζουμε.

Καλό θα είναι να κρατάμε αποθηκευμένες διάφορες εκδόσεις του κώδικα που γράφουμε έτσι ώστε αν κάποια στιγμή ανακαλύψουμε ότι ακολουθήσαμε λάθος δρόμο, να μπορούμε να επιστρέψουμε στην επιθυμητή έκδοση του κώδικα που γράφουμε.

Για να μελετήσουμε καλυτέρα τα παραδείγματα θα πρέπει να κάνουμε open και edit του αρχείου του παραδείγματος και αφού δούμε τις λειτουργίες του κώδικα να κάνουμε μικρές τροποποιήσεις για να πειραματιστούμε.
Για τον Html κώδικα καλό θα είναι να έχουμε έτοιμο κάποιο template με τη βασική δομή ενός Html αρχείου το οποίο θα δείξουμε παρακάτω.
Δεν συνιστάται να χρησιμοποιήσουμε κάποιο Html editor ο οποίος δεν είναι δωρεάν εφόσον υπάρχουν εξαιρετικές επιλογές για free Html editors.
Αφού γράψουμε τον κώδικα θα είναι καλό να τον εκτελέσουμε στους πιο δημοφιλείς Browsers Mozilla Safari, Firefox, Chrome και Microsoft Edge.

1.2 Web Usability και Responsive web pages

Συνεχίζουμε παρουσιάζοντας το Web Usability και το HTML Responsive Design.

1.2.1 Web Usability

Το Web Usability είναι ένα σύνολο κανόνων έτσι ώστε να δημιουργήσουμε user friendly Html σελίδες. Στόχος είναι οι χρήστες να βρίσκουν γρήγορα και εύκολα αυτό που ζητούν. Κάποιοι βασικοί κανόνες είναι οι παρακάτω:
Χρόνος φόρτωσης της σελίδας: Ο χρόνος που απαιτείται για να κατέβει μια σελίδα πρέπει να είναι πολύ μικρός. Αν μια σελίδα χρειάζεται πολύ χρόνο να κατέβει είναι σχεδόν βέβαιο ότι ο χρήστης θα εγκαταλείψει τη σελίδα.
Το logo του website θα πρέπει να βρίσκεται επάνω και αριστερά της σελίδας. Ο τρόπος που ένας χρήστης διαβάζει μια ιστοσελίδα είναι διαφορετικός από τον τρόπου που διαβάζει ένα βιβλίο. Συνήθως στην ιστοσελίδα οι χρήστες ψάχνουν κάτι συγκεκριμένο και θα πρέπει να το βρίσκουν γρήγορα. Έρευνες έχουν δείξει ότι αρχικά οι χρήστες εστιάζουν επάνω και αριστερά της οθόνης, οπότε στο σημείο αυτό θα πρέπει να τοποθετηθούν οι σημαντικές πληροφορίες της σελίδας όπως το logo και ο τίτλος.
Ομοιομορφία: Όλες οι σελίδες θα πρέπει να έχουν κάποια κοινά στοιχεία έτσι ώστε ο χρήστης να έχει την αίσθηση κατά τη διάρκεια της πλοήγησης σε ένα website ότι βρίσκεται στο συγκεκριμένο Website και δεν πλοηγείται σε κάποιο άλλο. Τα κοινά χαρακτηριστικά μπορούν να είναι το logo, κοινοί χρωματικοί συνδυασμοί των σελίδων, χρήση Background κτλ.
Κείμενο: Το κείμενο θα πρέπει να είναι ευανάγνωστο. Για να επιτευχθεί αυτός ο στόχος θα πρέπει το κείμενο να μην είναι πυκνογραμμένο και να έχει τους

κατάλληλους χρωματικούς συνδυασμούς, έτσι ώστε οι χρήστες να μπορούν εύκολα να εστιάσουν σε αυτό και να το διαβάσουν. Γενικά μία πολύ καλή πρακτική για ευανάγνωστο κείμενο είναι η χρησιμοποίηση λευκού Background με μαύρη γραμματοσειρά.

Παράγραφοι: Οι παράγραφοι δεν θα πρέπει να έχουν μεγάλη έκταση και να υπάρχουν σαφείς αποστάσεις μια παραγράφου από την επόμενή της. Σε κάθε παράγραφο αρχικά οι χρήστες εστιάζουν στην πρώτη γραμμή και με κατεύθυνση από αριστερά προς τα δεξιά.

Links: Θα πρέπει όλα τα links είτε προς άλλη σελίδα είτε προς Multimedia στοιχεία της σελίδας να δουλεύουν όλα. Δηλαδή να μην υπάρχουν broken links.

Style: Πρέπει οι σελίδες μας να εμφανίζονται σωστά σε όλες τις διαστάσεις της οθόνης του Browser. Γι' αυτό φροντίζει το Web Responsive Design.

Λειτουργικότητα του Website: Ο χρήστης δε θα πρέπει να καταναλώνει χρόνο για να σκεφθεί τι θα γίνει αν πατήσει ένα button ή ένα link. Θα πρέπει πριν πατήσει ένα button να είναι σχεδόν σίγουρος τι θα προκύψει μετά από αυτή την επιλογή.

Ενημέρωση του Website: Ένα Website Θα πρέπει να είναι πλήρως ενημερωμένο και ως προς τις πληροφορίες που παρέχει στο χρήστη.

1.2.2 HTML Responsive Design

Το Html Responsive Design έχει ως στόχο οι ιστοσελίδες που δημιουργούμε να φαίνονται σωστά σε οποιαδήποτε συσκευή με οποιεσδήποτε διαστάσεις της οθόνης. Οι χρήστες θα πρέπει να απολαμβάνουν την ίδια εμπειρία σε σχέση με την εμφάνιση των Web pages είτε χρησιμοποιούν ένα tablet, είτε Pc, κινητό τηλέφωνο κτλ.

Ακόμα και όταν αυξομειώνονται οι διαστάσεις του παραθύρου του Browser θα πρέπει ο χρήστης να βλέπει καθαρές εικόνες, καθαρό κείμενο όπως επίσης να έχει εύκολη πρόσβαση στις πληροφορίες της ιστοσελίδας.

Οι εικόνες θα πρέπει να έχουν κατάλληλη ανάλυση έτσι ώστε να εμφανίζονται σωστά σε όλα τα πλάτη της οθόνης. Επίσης υπάρχει η δυνατότητα ανάλογα με τη συσκευή που χρησιμοποιούμε να στέλνει ο server στον Browser εικόνες με διαφορετική ανάλυση. Δηλαδή αν ο χρήστης χρησιμοποιεί mobile Browser δε χρειάζεται να εμπεριέχεται στη σελίδα μια εικόνα ανάλυσης 4096X 3500 για

παράδειγμα. Με την ίδια εικόνα σε πολύ χαμηλότερη ανάλυση θα έχει το ίδιο οπτικό αποτέλεσμα ενώ ταυτόχρονα θα μειωθεί και ο χρόνος download σε ένα κινητό τηλέφωνο.

Επίσης οι γραμματοσειρές μπορούν να προσαρμόζονται στην τρέχουσα διάσταση του πλάτους της οθόνης του Browser.

Το Html Responsive Design έχει τεράστια σημασία και η παγκόσμια κοινότητα του internet έχει δώσει πολύ μεγάλη προσοχή.

ΥΠΟΕΝΟΤΗΤΑ 2. TEXT EDITORS

Γενικές οδηγίες για τους Html editors

Για να ξεκινήσουμε με την Html χρειαζόμαστε τις βασικές λειτουργίες ενός editor όπως οι παρακάτω:

Open: Από το File Menu επιλέγουμε open και έπειτα με τη λειτουργία browse επιλέγουμε το αρχείο που επιθυμούμε να ανοίξουμε. Στους περισσότερους editors μπορούμε με ένα drag & drop να ανοίξουμε ένα ή περισσότερα αρχεία στον editor. Κατά τη λειτουργία drag & drop επιλέγουμε ένα ή περισσότερα αρχεία και αφού τα επιλέξουμε κρατώντας πατημένο τα αριστερό mouse button ελευθερώνουμε το mouse button στην περιοχή του παράθυρου του editor και αυτομάτως τα αρχεία ανοίγουν στον editor. Τέλος μπορούμε να κάνουμε drag & drop ένα ολόκληρο folder.

Edit: Από το File Menu επιλέγουμε Edit σε ένα αρχείο που έχουμε ανοίξει και έχουμε τις δυνατότητες για αντιγραφή, αποκοπή κτλ.

New: Από το File Menu επιλέγουμε New για να δημιουργήσουμε ένα νέο αρχείο. Μόλις δημιουργηθεί το αρχείο μπορούμε να κάνουμε Save as από το File Menu και να δώσουμε ένα όνομα στο αρχείο με κατάληξη ".html".

2.1 Text editors για αρχάριους

Όταν χρειαζόμαστε κάποιον Html Editor ο οποίος θα είναι πολύ απλός στη χρήση και η εγκατάσταση του θα είναι πολύ απλή μπορούμε να καταφύγουμε στις παρακάτω λύσεις.

2.1.1 Notepad++ (Windows)

Το Notepad++ είναι ίσως η καλύτερη λύση για κάποιο αρχάριο που χρειάζεται μία γρήγορη εγκατάσταση χωρίς παραμετροποιήσεις και πολύ απλό και φιλικό περιβάλλον με σχετικά αρκετές δυνατότητες.

Η εγκατάσταση είναι πολύ απλή και γρήγορη. Αρκεί να πλοηγηθούμε στη διεύθυνση: https://notepad-plus-plus.org

και να κάνουμε download την τελευταία έκδοση του Notepad++, επιλέγοντας 32 bit ή 64 bit ανάλογα την έκδοση των windows που έχουμε.

Στο Notepad++ υπάρχουν όλες οι επιλογές για επεξεργασία και δημιουργία αρχείων Html.

Έτσι λοιπόν αν επιθυμούμε να επεξεργαστούμε ένα αρχείο Html μπορούμε εύκολα να κάνουμε δεξί κλικ επάνω στο αρχείο και να επιλέξουμε επεξεργασία με Notepad++.

Αν θέλουμε να δημιουργήσουμε ένα αρχείο Html επιλέγουμε File → New και έπειτα επιλέγουμε αποθήκευση ως και δίνουμε όνομα με κατάληξη .html, όπως για παράδειγμα index.html. Όταν το αρχείο αναγνωριστεί ως αρχείο Html αμέσως θα παρατηρήσουμε ότι υπάρχουν διαφορετικοί χρωματισμοί για διάφορες κατηγορίες στοιχείων.

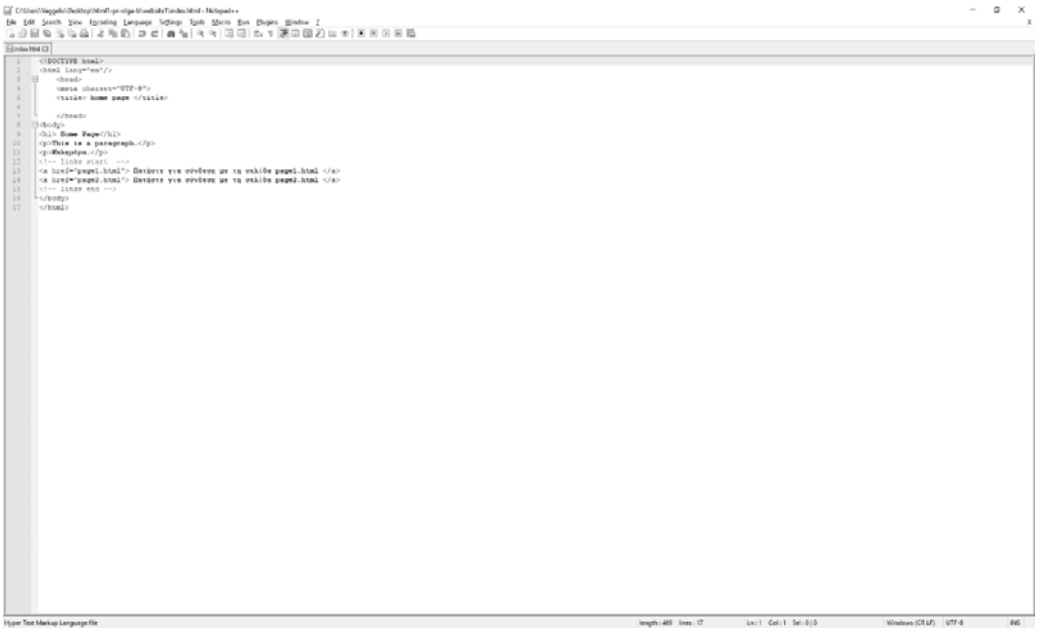
Επίσης κάνοντας κλικ σε ένα άγκιστρο '{' μπορούμε και βλέπουμε σε ποιο άγκιστρο κλεισίματος αντιστοιχεί '}'.

Μια άλλη πολύ καλή λειτουργία του Notepad++ είναι η αναζήτηση.

Για να αναζητήσουμε κάποιο στοιχείο της Html τότε αρκεί να πατήσουμε το control button και έχοντας πατημένο το Control button να πατήσουμε μία φορά το πλήκτρο F.

Τότε θα έχουμε τις επιλογές αναζήτησης στο τρέχον αρχείο ακόμη και σε όλα τα αρχεία που έχουμε ανοίξει στο Notepad++.

Εικόνα 1. Βασική οθόνη Notepad++



2.1.2 Text Edit for Mac

Το Text Edit βρίσκεται στις εφαρμογές του macOs και μπορούμε να το βρούμε ανοίγοντας το Launch Pad ή γράφοντας το όνομα στο Spotlight. Το spotlight μπορούμε να το ανοίξουμε πατώντας το εικονίδιο που απεικονίζει μεγεθυντικό φακό στα δεξιά του Menu Bar του Mac.

Το Text Edit δυστυχώς παρέχει μόνο τις βασικές δυνατότητες, ίδιες με αυτές της απλής επεξεργασίας ενός κειμένου.

Αρχικά ανοίγουμε το Text edit. Στη συνέχεια από το Format επιλέγουμε Make Plain Text.

Έπειτα από το File Menu επιλέγουμε save και θα εμφανιστεί η καρτέλα όπου στο πεδίο Save As θα πληκτρολογήσουμε το όνομα του αρχείου ενώ στο File Format θα επιλέξουμε Web Page (.Html).

Όταν θα χρειαστεί να επεξεργαστούμε το αρχείο μπορούμε να το ανοίξουμε με την επιλογή open από το File Menu και στη συνέχεια θα επιλέξουμε save.

Εικόνα 2. Βασική οθόνη Text Edit

2.1.3 Brackets (Mac και Windows)

Το brackets είναι ένας πολύ δημοφιλής Html editor με πολλές δυνατότητες και η εγκατάσταση του είναι απλή.

Αρκεί να μεταβούμε στη σελίδα: http://brackets.io

και να επιλέξουμε download. Η έκδοση του λειτουργικού που διαθέτουμε θα αναγνωριστεί αυτόματα.

Η εγκατάσταση είναι απλή και σύντομη σε Mac Os καθώς και σε PC. Στο Brackets υπάρχουν πολύ χρήσιμα features όπως το Live Preview. Για να ενεργοποιηθεί το Live Preview θα πρέπει να πατήσουμε το εικονίδιο στη δεξιά πλευρά της οθόνης που μοιάζει με αναπαράσταση κεραυνού.

Μόλις πατηθεί το Button Live Preview θα φορτωθεί στον Browser η σελίδα που έχουμε ανοίξει στο Browser. Το πολύ χρήσιμο είναι ότι κατά τη διάρκεια της πληκτρολόγησης του κώδικα στην οθόνη του Browser γίνεται συγχρόνως ενημέρωση. Για παράδειγμα αν σε ένα στοιχείο p (παράγραφος) πληκτρολογήσω έστω και ένα χαρακτήρα, αυτόματα ο χαρακτήρας θα εμφανιστεί και στην οθόνη του Browser.

Σημείωση

Για να ενεργοποιηθεί το Live Preview θα πρέπει να υπάρχει η σελίδα index.html καθώς και εγκατεστημένος ο Chrome browser.

Στο Brackets υπάρχουν τα κλασσικά menus:

File: Στο File Menu υπάρχουν οι κλασικές επιλογές για δημιουργία και αποθήκευση αρχείων.

Edit: Στο Edit Menu υπάρχουν οι επιλογές για επεξεργασία

Find: Στο Edit Menu υπάρχουν οι επιλογές για αναζήτηση

Navigate: Στο Navigate Menu υπάρχουν επιλογές για μετάβαση από ένα αρχείο σε άλλο

Debug: Υπάρχουν επιλογές για Debug με τα Chrome Developer tools

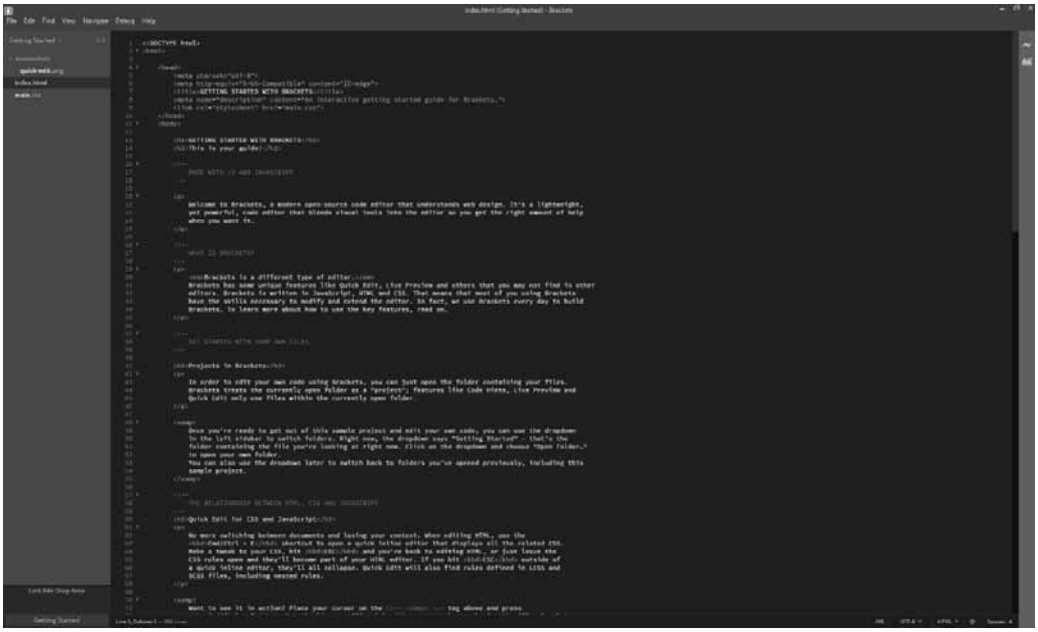
Help: Επιλογές για βοήθεια

Σημείωση

Σε αυτήν τη στιγμή μας είναι απαραίτητες μόνο οι βασικές επιλογές όπως η δημιουργία αρχείου, η αποθήκευση κτλ.

Εικόνα 3. Βασική οθόνη Brackets για Mac OS

Εικόνα 4. Βασική οθόνη Brackets για Windows



2.2 Δομής σελίδας Html

Αναγνωριστικό <!DOCTYPE Html>

Στη πρώτη γραμμή της κάθε σελίδας υπάρχει το αναγνωριστικό

<!DOCTYPE Html>

το οποίο δηλώνει την έκδοση της Html.

Κώδικας Html της σελίδας

Μεταξύ των ετικετών <html> και </html> περικλείεται ο Html κώδικας της σελίδας. Στην ετικέτα <html> μπορεί να συμπεριληφθεί και η δήλωση lang για τη γλώσσα

<html lang="en">

```
<!DOCTYPE html>
<html>
Κώδικας της σελίδας
</html>
```

Στοιχείο Head

Το στοιχείο head περιλαμβάνεται μεταξύ των ετικετών <head> και </head>. Εντός του στοιχείου head μπορούμε να γράψουμε τα δεδομένα metadata τα οποία περιλαμβάνουν πληροφορίες που βοηθούν στην εύρεση της σελίδας από τις μηχανές αναζήτησης, όπως και το στοιχείο title. Στο στοιχείο title γράφουμε τον τίτλο που επιθυμούμε να εμφανίζεται στο tab του Browser στο επάνω μέρος της οθόνης.

```
<!DOCTYPE html>
<html>
<head>
…
…
…
</head>
…
…
Κώδικας της σελίδας
…
…
…
</html>
```

Body

Το στοιχείο body περικλείεται μεταξύ των ετικετών <body> και </body>. Στο στοιχείο body περιλαμβάνεται κείμενο, συνδέσεις προς εικόνες και προς άλλες σελίδες και γενικά ότι μπορεί να γίνει ορατό από το χρήστη.

```
<!DOCTYPE Html>
<html>
<head>
…
</head>
<body>
…
</body>
</html>
```

Εικόνα 5. Δομή ενός Html Αρχείου

```
<!DOCTYPE html>    Αναγνωριστικό Html
<html> Αρχή Html Κώδικα
<head> Αρχή στοιχείου Head
…
</head> Τέλος στοιχείου Head
<body>  Αρχή στοιχείου Body
…
…
…
…
</body>Τέλος  στοιχείου Body
</html> Τέλος Html Κώδικα
```

2.3 Στοιχεία p, h1-h6 και δημιουργία σχολίων

Στη συνέχεια, παρουσιάζουμε την εισαγωγή σχολίων και τα στοιχεία p και h.

2.3.1 Σχόλια και στοιχείο p

Για να δημιουργήσουμε σχόλια εντός του κώδικα των ιστοσελίδων τα περιλαμβάνουμε μεταξύ των ετικετών <!-- και -->. Τα σχόλια δεν εμφανίζονται στη σελίδα που έχουμε δημιουργήσει.

Βασικό στοιχείο της Html είναι το στοιχείο p όπου ανάμεσα στις ετικέτες και μπορούμε να γράψουμε κείμενο με τη μορφή παραγράφου.

Παράδειγμα

```
<p> η πρώτη μου παράγραφος στην Html </p>
```

2.3.2 Στοιχείο h

Το στοιχείο h προσφέρει τη δυνατότητα σήμανσης κειμένου ως επικεφαλίδα.

Η Html5 διαθέτει επικεφαλίδες με διαφορετικά μεγέθη από h1-h6. Το κείμενο της επικεφαλίδας περικλείεται μεταξύ των ετικετών <hi> και </hi> όπου i είναι ένας αριθμός από 1,2…,6. Το μεγαλύτερο μέγεθος στοιχείου h είναι το h1 ενώ το μικρότερο είναι το h6.

Σημείωση

Οι επικεφαλίδες h1-h6 καταλαμβάνουν ολόκληρο το πλάτος της ιστοσελίδας.

Παράδειγμα

```
<h1> This is my first heading</h1>
```

Σημείωση

Όταν χρησιμοποιούμε δύο επικεφαλίδες στη σειρά η δεύτερη θα είναι επεξήγηση της πρώτης. Οπότε η δεύτερη στη σειρά επικεφαλίδα θα πρέπει να έχει μικρότερο μέγεθος.

Παράδειγμα

```
<h1> This is my first heading 1</h1>

<h2> This is my first heading 2</h2>
```

ΥΠΟΕΝΟΤΗΤΑ 3. HELLO HTML - ΔΗΜΙΟΥΡΓΩΝΤΑΣ HTML ΣΕΛΙΔΕΣ

Σε αυτή την υποενότητα θα δούμε πώς γίνεται η η συγγραφή μια τυπικής html σελίδας.

3.1 Η πρώτη μου σελίδα

Τα βήματα που θα πρέπει να ακολουθήσουμε για να δημιουργήσουμε μια Html σελίδα είναι τα παρακάτω:

1. Αρχικά δημιουργούμε ένα νέο αρχείο σε οποιοδήποτε editor

Για το TextEdit(Mac OS):

- Επιλέγουμε File → New και έπειτα Format → Make Plain text

Για το Notepad++ (Windows):

- Επιλέγουμε File → New

Για το Brackets(Mac OS & Windows):

- Επιλέγουμε File → New

2. Γράφουμε στην πρώτη γραμμή το αναγνωριστικό <!DOCTYPE Html>

Κάθε σελίδα Html εκτός από την κατάληξη .html στο όνομα του αρχείου θα πρέπει να ξεκινάει με το αναγνωριστικό <!DOCTYPE Html> το οποίο δηλώνει την έκδοση της Html.

3. Θα πρέπει να γράψουμε στη δεύτερη γραμμή <html> για να δηλώσουμε την αρχή του κώδικα html. Εντός της ετικέτας <html> μπορούμε να ορίσουμε και τη γλώσσα της σελίδας με την ιδιότητα lang. Για παράδειγμα, μπορούμε να γράψουμε <html lang="en"> για την υποστήριξη της Αγγλικής γλώσσας.

4. Θα πρέπει να γράψουμε την ετικέτα <head> για να ορίσουμε την αρχή του στοιχείου head.

5. Εντός του στοιχείου head θα γράψουμε το meta tag με τη δήλωση για την κωδικοποίηση χαρακτήρων UTF-8 (<meta charset="UTF-8">). Η κωδικοποίηση χαρακτήρων περιλαμβάνει ένα σετ χαρακτήρων που θα πρέπει να εμφανίζει ο Browser.

6. Κάτω από την ετικέτα που περιλαμβάνει κωδικοποίηση χαρακτήρων θα γράψουμε τον τίτλο που θα εμφανίζεται στο tab της τρέχουσας σελίδας του Browser (<title> Html Basics </title>).

7. Γράφουμε την ετικέτα </head> για να ορίσουμε το τέλος του στοιχείου head.

8. Γράφουμε <body> για να ορίσουμε την αρχή του στοιχείου body το οποίο θα περιλαμβάνει στοιχεία που θα εμφανίζονται στην οθόνη του browser.

9. Εντός του στοιχείου body γράφουμε τα στοιχεία που επιθυμούμε να περιλαμβάνει η σελίδα μας.

10. Κλείνουμε το στοιχείο body με την ετικέτα </body>

11. Κλείνουμε τον κώδικα της σελίδα με την ετικέτα </html>

12. Αποθηκεύουμε τον κώδικα στο αρχείο σε οποιοδήποτε editor

Για το TextEdit (Mac OS):

Επιλέγουμε File → Save και αυτόματα θα εμφανιστεί ένα μικρό παράθυρο για την αποθήκευση.

Εντός του παραθύρου για την αποθήκευση στο πεδίο Save As πληκτρολογούμε το όνομα του αρχείου και στη συνέχεια από το dropdown File Format, επιλέγουμε Web Page (.Html)

Για το Notepad++ (Windows):

Επιλέγουμε File → Save As και επιλέγουμε διαδρομή και πληκτρολογούμε το όνομα του αρχείου με κατάληξη .html. Παράδειγμα index.Html

Για το Brackets(Mac OS & Windows):

Ακριβώς όπως με το Notepad++

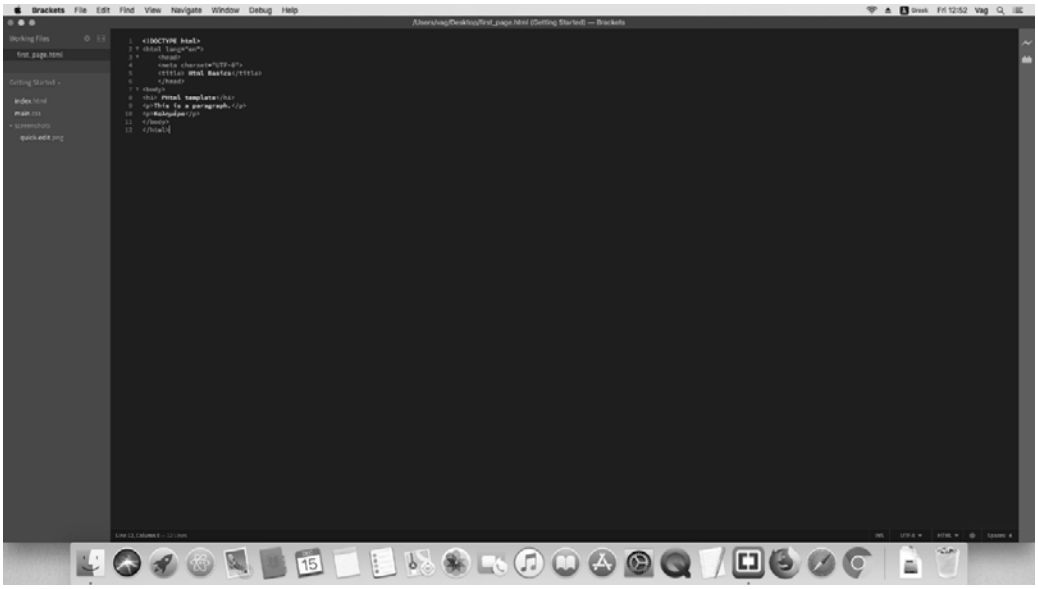
Για να ανοίξουμε ένα αρχείο Html έτσι ώστε να το επεξεργαστούμε:

επιλέγουμε File → Open και στη συνέχεια αφού εντοπίσουμε το αρχείο πατάμε open.

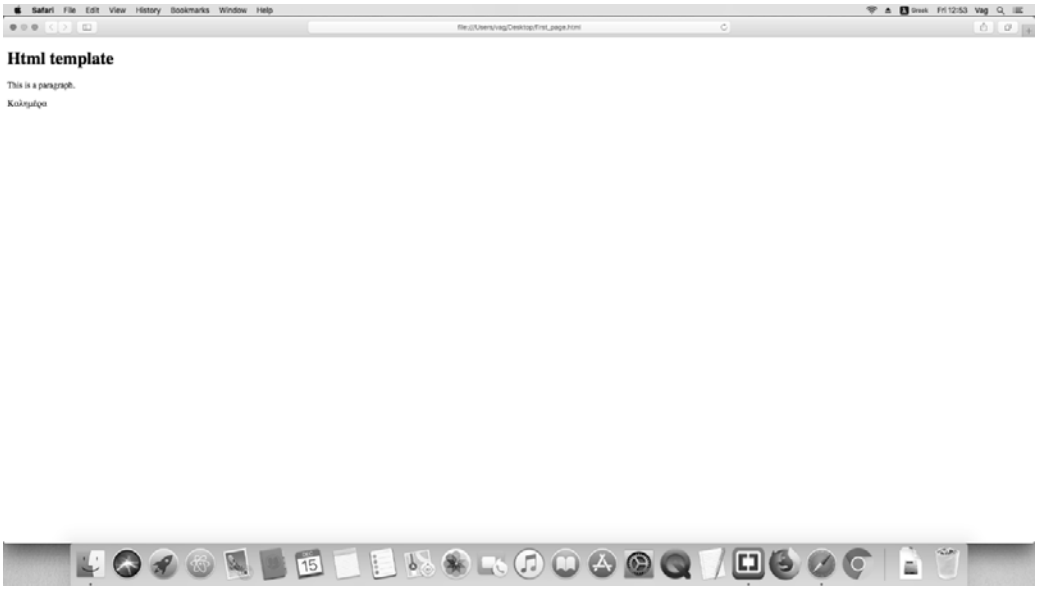
```
<!DOCTYPE Html>
<html lang="en">
   <head>
   <meta charset="UTF-8">
   <title> Html Basics </title>
   </head>
<body>
<h1> Html template  </h1>
<p>This is a paragraph.</p>
<p>Καλημέρα.</p>
</body>
</html>
```

Παράδειγμα 1. Η πρώτη μου σελίδα (Mac)

Κώδικας: first_page.html

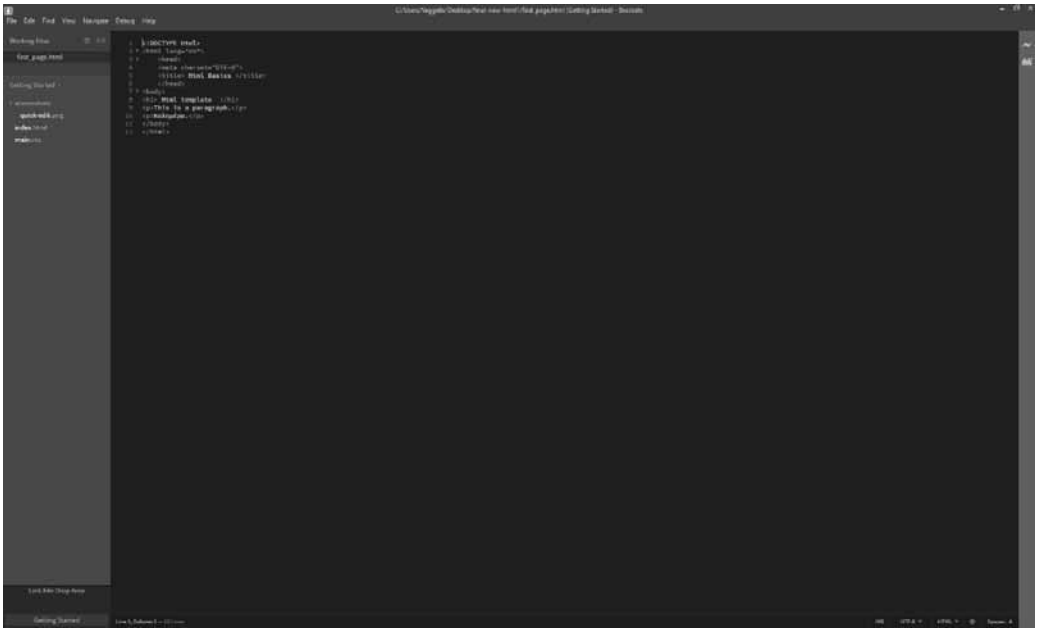


Εικόνα Browser: first_page.html

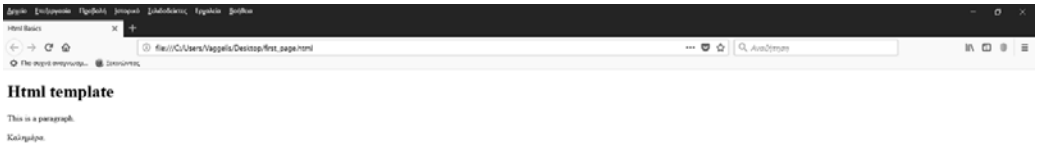


Παράδειγμα 2. Η πρώτη μου σελίδα (Windows)

Κώδικας: first_page.html



Εικόνα Browser: first_page.html



3.2 Το πρώτο μου Button

Τα buttons αποτελούν αναπόσπαστο τμήμα της Html και γενικά της φιλοσοφίας στην οποία βασίζεται η πλοήγηση σε ένα website. Δηλαδή σε ένα website με το πάτημα ενός button μπορούμε να εμφανίσουμε μια εικόνα, να κάνουμε αναπαραγωγή ενός video ή να μεταφερθούμε σε άλλη σελίδα του ίδιου ή ενός διαφορετικού Website.

Στην παρούσα φάση θα εξηγήσουμε πως μπορούμε να δημιουργήσουμε εύκολα και γρήγορα ένα απλό button σε μορφή png. Τα αρχεία png περιέχουν τις πληροφορίες έτσι ώστε να εμφανιστεί μία εικόνα και το συγκεκριμένο format είναι κατάλληλο για ιστοσελίδες. Αν όμως χρειαζόμαστε εικόνες υψηλής ανάλυσης τότε θα πρέπει να επιλέξουνε το format jpg.

Το καλύτερο για ξεκίνημα είναι να χρησιμοποιήσουμε ένα εύχρηστο και απλό Online free tool για τη δημιουργία Buttons όπως το Da button factory το οποίο μπορούμε να χρησιμοποιήσουμε πληκτρολογώντας απλά τη διεύθυνση: https://dabuttonfactory.com

Σημείωση

Το Da button factory εφόσον βρίσκεται στο Web είναι το ίδιο για Mac OS και Windows και δε χρειάζεται εγκατάσταση και παραμετροποίηση. Επίσης αντί για το συγκεκριμένο online tool μπορούμε να χρησιμοποιήσουμε οποιοδήποτε άλλο. Στην παρούσα φάση θα δημιουργήσουμε ένα απλό button χωρίς αλληλεπιδράσεις, όπως για παράδειγμα την αλλαγή των χρωμάτων όταν το mouse cursor αιωρείται πάνω από την περιοχή που καταλαμβάνει το button εντός της σελίδας.

Το Da button factory αποτελείται από τρία τμήματα και ξεκινώντας από τα αριστερά προς τα δεξιά:

Περιοχή 1:

Στην πρώτη περιοχή θα δούμε:

Πεδίο Text → γράφουμε το κείμενο του Button

Font → επιλέγουμε τη γραμματοσειρά

Size και Color → επιλέγουμε μέγεθος και χρώμα γραμματοσειράς

Text shadow → επιλέγουμε το εφέ της σκιάς

Size → το μέγεθος που βρίσκεται σε κατώτερο σημείο σε σχέση με το προηγούμενο μας δίνει δύο επιλογές:

Fit to text → όπου όσο πληκτρολογούμε χαρακτήρες το μέγεθος του button αυξάνει έτσι ώστε να είναι ορατοί οι χαρακτήρες.

Fixed όπου οι διαστάσεις του button δεν αυξομειώνονται ανάλογα με το πλήθος και το μέγεθος των χαρακτήρων εντός του button.

Horizontal Padding → απόσταση του κειμένου από τη δεξιά και την αριστερή πλευρά του Button.

Vertical Padding → απόσταση του κειμένου από την επάνω και κάτω πλευρά του button.

Εικόνα 6. Περιοχή 1 Da button Factory

Περιοχή 2:

Υπάρχει η προεπισκόπηση του Button και κάθε αλλαγή που επιφέρουμε σε αυτό αυτόματα γίνεται ορατή. Επίσης υπάρχουν επιλογές για το format του αρχείου. Όταν ολοκληρώσουμε το σχεδιασμό του button θα επιλέξουμε download έτσι ώστε να αποθηκευτεί το button.

Εικόνα 7. Περιοχή 2 Da button Factory

Περιοχή 3:

Αρχικά στο πεδίο STYLE επιλέγουμε:

Rounded rectangular όπου με την αύξηση του Corner Radius αυξάνουμε την καμπυλότητα στις 4 γωνίες.

Επίσης υπάρχουν και οι επιλογές Background για simple gradient δηλαδή χρωματική κλιμάκωση του Background.

Τέλος υπάρχουν οι επιλογές Border και Shadow για το button καθώς επίσης και επιλογές για προσανατολισμό του κειμένου εντός του Button.

Εικόνα 8. Περιοχή 3 Da button Factory

3.3 Δομή ενός Website και συνδέσεις μεταξύ των σελίδων

Τα βήματα που θα πρέπει να κάνουμε για να δημιουργήσουμε μια απλή μορφή website είναι η παρακάτω:

Αρχικά θα δημιουργήσουμε ένα νέο Folder το οποίο θα φιλοξενήσει τα αρχεία και τους υποφακέλους που θα αποτελούν το Website. Στο σημείο αυτό θα

πρέπει να διευκρινιστεί ότι όταν το website «ανεβεί» στο Web τότε ολόκληρος ο φάκελος θα βρίσκεται στο server που θα φιλοξενεί το Website. Για τις ανάγκες της εκπαιδευτικής διαδικασίας θα δημιουργήσουμε το φάκελο του Website στο desktop του υπολογιστή μας ή σε οποιοδήποτε path του σκληρού μας δίσκου. Στο φάκελο του website θα δώσουμε ένα όνομα π.χ Website1.

Στη συνέχεια θα δημιουργήσουμε εντός του φακέλου τα ακόλουθα folders:

videos → για την αποθήκευση videos

sounds → για την αποθήκευση ήχων

images → για την αποθήκευση αρχείων για εικόνες κτλ.

Έπειτα θα δημιουργήσουμε μια σελίδα η οποία θα είναι η κεντρική σελίδα του Website και θα την αποθηκεύσουμε εντός φακέλου με όνομα index.html.

Σημείωση

Μία πολύ καλή πρακτική είναι να έχουμε πρόχειρο ένα κώδικα-template που έχουμε δημιουργήσει έτσι ώστε τα βασικά μέρη του κώδικα για κάθε σελίδα να είναι έτοιμα και να εξοικονομήσουμε χρόνο για την πληκτρολόγηση του κώδικα. Τα ονόματα των υποφακέλων είναι ενδεικτικά.

Συνδέσεις μεταξύ των σελίδων

Για να κάνουμε μια σύνδεση προς μία άλλη σελίδα θα χρειαστούμε το στοιχείο a.

Το στοιχείο a περιέχεται μεταξύ των ετικετών <a> και </a> και μαζί με τη βοήθεια της ιδιότητας href μας δίνει τη δυνατότητα να δημιουργήσουμε μία σύνδεση προς μία άλλη σελίδα του συγκεκριμένου ή κάποιου άλλου website.

Σύνταξη

```
<a href="διεύθυνση _ιστοσελίδας"> κείμενο της σύνδεσης </a>
```

Παράδειγμα

```
<a href="http://www.uoa.gr/"> Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών
</a>
```

Ως προεπιλογή η εμφάνιση της σύνδεσης εμφανίζεται ως εξής:

- Αν δεν έχει πατηθεί το link: χρώμα μπλε και υπογραμμισμένη
- Αν έχει πατηθεί το link: χρώμα μωβ και υπογραμμισμένη
- Ενεργό link: χρώμα μωβ και υπογραμμισμένο

Παράδειγμα σύνδεσης μεταξύ δύο σελίδων

Έστω ότι έχουμε δημιουργήσει δύο σελίδες με ονόματα index.Html και page1.html και επιθυμούμε να δημιουργήσουμε μία σύνδεση από τη σελίδα

index.html προς τη σελίδα page1.html. Για να υλοποιήσουμε τη σύνδεση εντός του στοιχείου body της index.html προς τη σελίδα page1.html θα γράψουμε

```
<a href="page1.html"> Πατήστε για σύνδεση με τη σελίδα page1.html </a>
```

Για να δημιουργήσουμε μία σύνδεση από τη σελίδα page1.html προς τη σελίδα index.html θα γράψουμε εντός του στοιχείου Body της σελίδας page1.html την παρακάτω γραμμή:

```
<a href="index.html"> Πατήστε για σύνδεση με τη σελίδα page1.html </a>
```

Σημείωση

Έχει γίνει η παραδοχή ότι και οι δύο σελίδες βρίσκονται στο ίδιο folder.

Παρατήρηση

Δεν αντιγράφουμε ποτέ διαδρομές μέσα από το δικό μας desktop, φάκελο documents και γενικά διαδρομές από το φάκελο users κτλ.

3.3.1 Παράδειγμα Δομής Website

Παράδειγμα website με τρεις σελίδες:

Δημιουργία folders

Αρχικά θα δημιουργήσουμε ένα κεντρικό folder το οποίο θα είναι και το folder του Website και θα του δώσουμε ένα όνομα όπως για παράδειγμα Website1.

Στη συνέχεια εντός του κεντρικού folder θα δημιουργήσουμε άλλα τρία:

- images
- sounds
- videos

Δημιουργία σελίδας home page

Η σελίδα home page είναι και η αρχική σελίδα του website και θα πρέπει να έχει το όνομα index.html έτσι ώστε οι browsers να "καταλαβαίνουν" ότι πρόκειται για την αρχική σελίδα (Home Page) του Website.

index.html

Στις δύο πρώτες γραμμές θα γράψουμε το αναγνωριστικό

```
<!DOCTYPE Html>
```

και την ετικέτα έναρξης του κώδικα Html μαζί με τη γλώσσα `<Html lang="en">`

```
<!DOCTYPE html>
<html lang="en">
```

Στη συνέχεια θα γράψουμε το στοιχείο head και εντός του στοιχείου head θα συμπεριλάβουμε το meta tag με την κωδικοποίηση χαρακτήρων UTF-8 καθώς και το στοιχείο title.

```
<head>
  <meta charset="UTF-8">
  <title> home page </title>
</head>
```

Ξεκινάμε το στοιχείο body με την ετικέτα <body>. Ο κώδικας εντός του body εμφανίζει το περιεχόμενο στην οθόνη του Browser.

```
<body>
```

Γράφουμε ένα στοιχείο h1 για την επικεφαλίδα της σελίδας

```
<h1> Home Page</h1>
```

Προσθέτουμε δύο στοιχεία p

```
<p>This is a paragraph.</p>
<p>Καλημέρα</p>
```

Τέλος δημιουργούμε δύο συνδέσεις με τις υπόλοιπες σελίδες page1.html και page2.html

Στην πρώτη σύνδεση θα συνδέσουμε τη σελίδα index.html με τη σελίδα page1.html., οπότε θα γράψουμε ένα στοιχείο a με τιμή της ιδιότητας href page1.html και μεταξύ των ετικετών a θα γράψουμε την περιγραφή της σύνδεσης «Πατήστε για σύνδεση με τη σελίδα page1.html»

```
<a href="page1.html"> Πατήστε για σύνδεση με τη σελίδα page1.html </a>
```

Στη δεύτερη σύνδεση για να συνδέσουμε την index.html με την page2.html θα γράψουμε για την ιδιότητα href την τιμή page2.html και μεταξύ των ετικετών θα γράψουμε το συνοδευτικό κείμενο «Πατήστε για σύνδεση με τη σελίδα page2.html»

```
<a href="page2.Html"> Πατήστε για σύνδεση με τη σελίδα page2.Html </a>
```

Τέλος κλείνουμε το στοιχείο body και γράφουμε </html> για να δηλώσουμε το τέλος του Html κώδικα.

```
</body>
</html>
```

Αφού ολοκληρωθεί ο κώδικας αποθηκεύουμε το αρχείο ως index.html

Δημιουργία σελίδων page1.Html και page2.Html

Για ταχύτερη υλοποίηση μπορούμε να κάνουμε δύο φορές copy+paste τη σελίδα index.html και έπειτα να κάνουμε rename στις δύο σελίδες εκτός από την index.html και να δώσουμε τα ονόματα page1.html και page2.html αλλάζοντας title, h1 και γενικά ότι άλλο διαφέρει.

Αυτό που θα πρέπει να προσέξουμε είναι οι συνδέσεις μεταξύ των σελίδων. Έτσι λοιπόν η σελίδα page1.html θα πρέπει να συνδέεται με τις σελίδες index.html και page2.html, Οπότε θα πρέπει να γράψουμε:

```
<a href="index.html"> Πατήστε για σύνδεση με τη σελίδα index.html </a>
<a href="page2.html"> Πατήστε για σύνδεση με τη σελίδα page2.html </a>
```

Αντίστοιχα η σελίδα page2.html θα πρέπει να συνδέεται με τις σελίδες index.Html και page1.Html οπότε και θα πρέπει να γράψουμε:

```
<a href="index.html"> Πατήστε για σύνδεση με τη σελίδα index.html </a>
<a href="page1.html"> Πατήστε για σύνδεση με τη σελίδα page1.html </a>
```

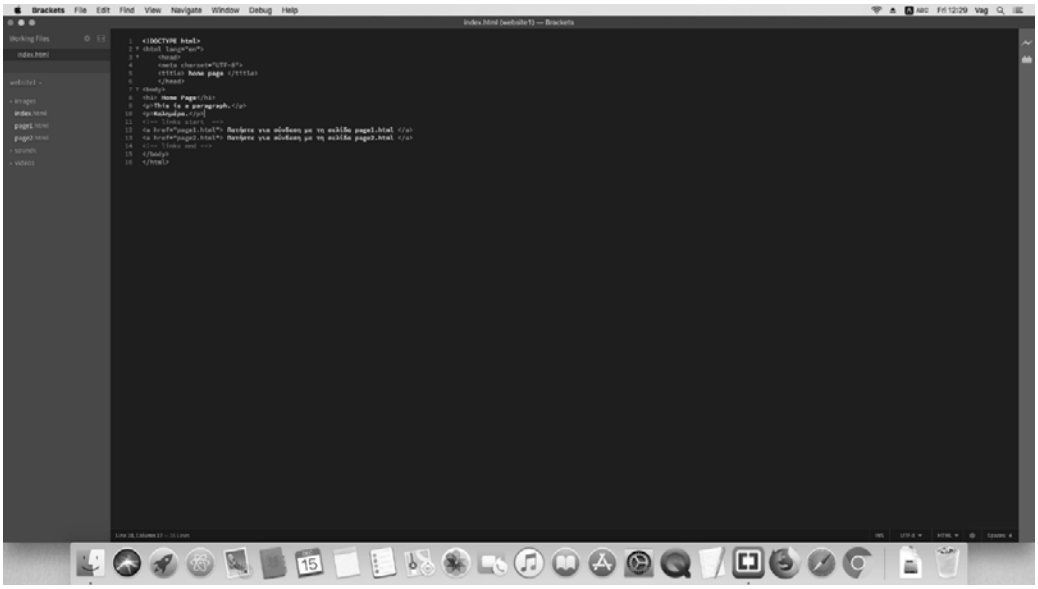
Όταν θα έχουμε τελειώσει τον κώδικα για τις σελίδες μπορούμε να κάνουμε κλικ σε οποιοδήποτε εκ των τριών αρχείων index.html, page1.html και page2.html και να ξεκινήσουμε την πλοήγηση.

Σημείωση

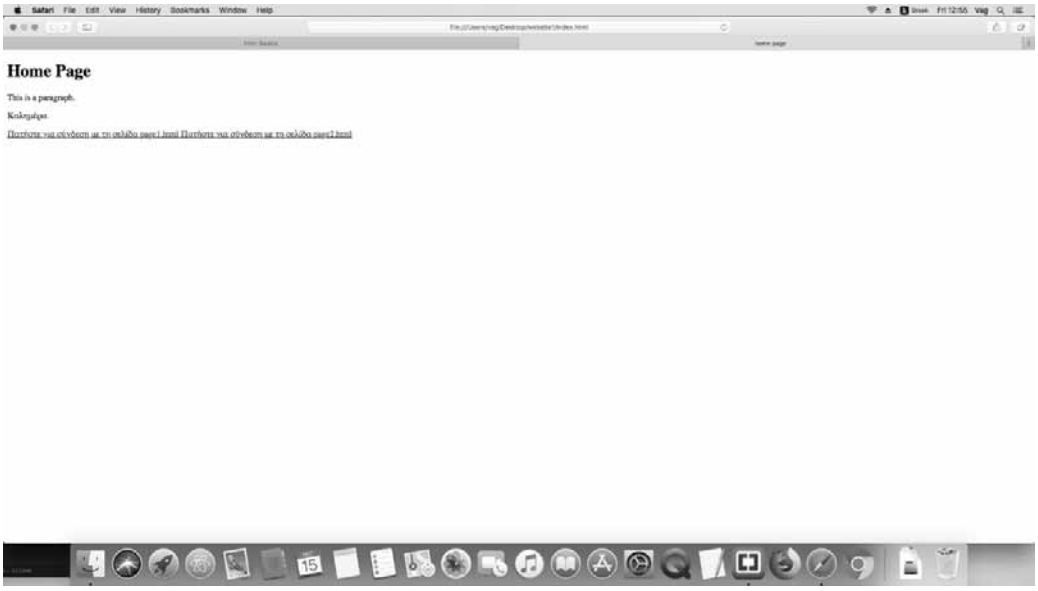
Τα folders για videos και sounds δεν έχουν χρησιμοποιηθεί στο συγκεκριμένο παράδειγμα. Είναι σε αναμονή για την αναπαραγωγή αρχείων ήχου, video και εικόνας.

Παράδειγμα 3. Website template 1 Mac

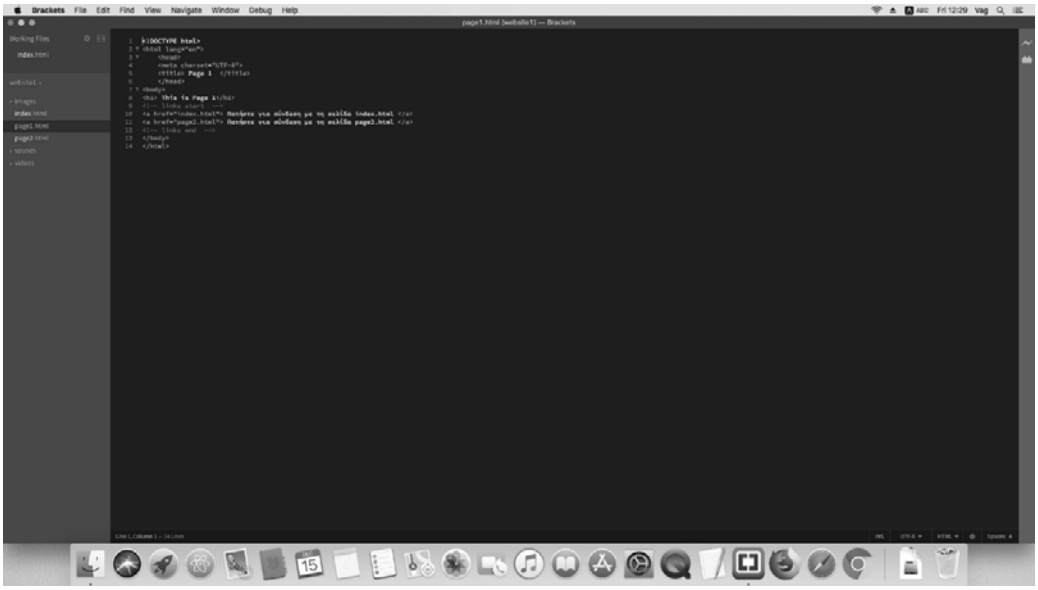
Κώδικας: index.html



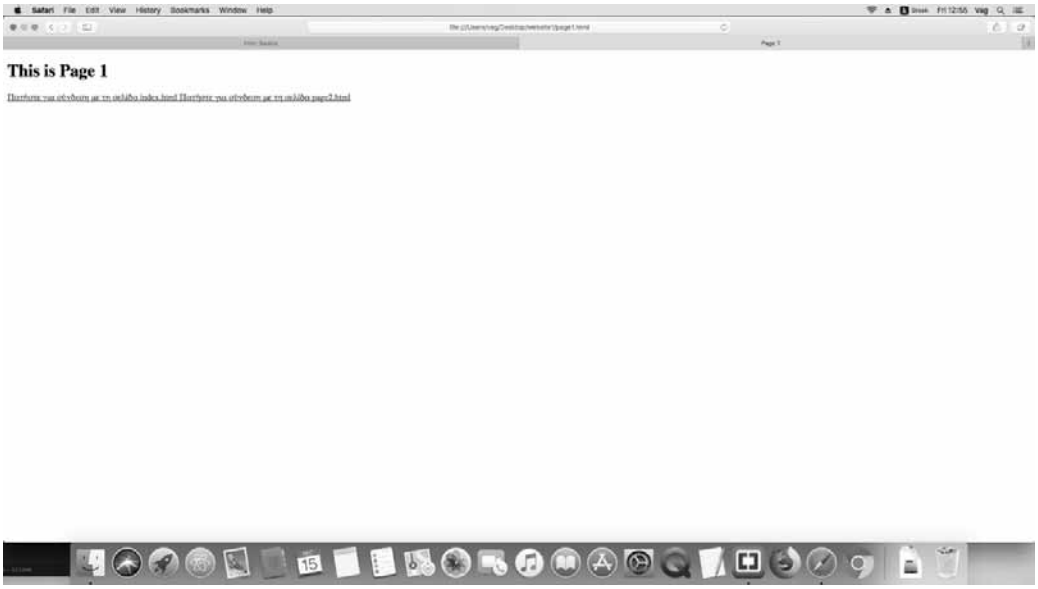
Εικόνα Browser: index.html



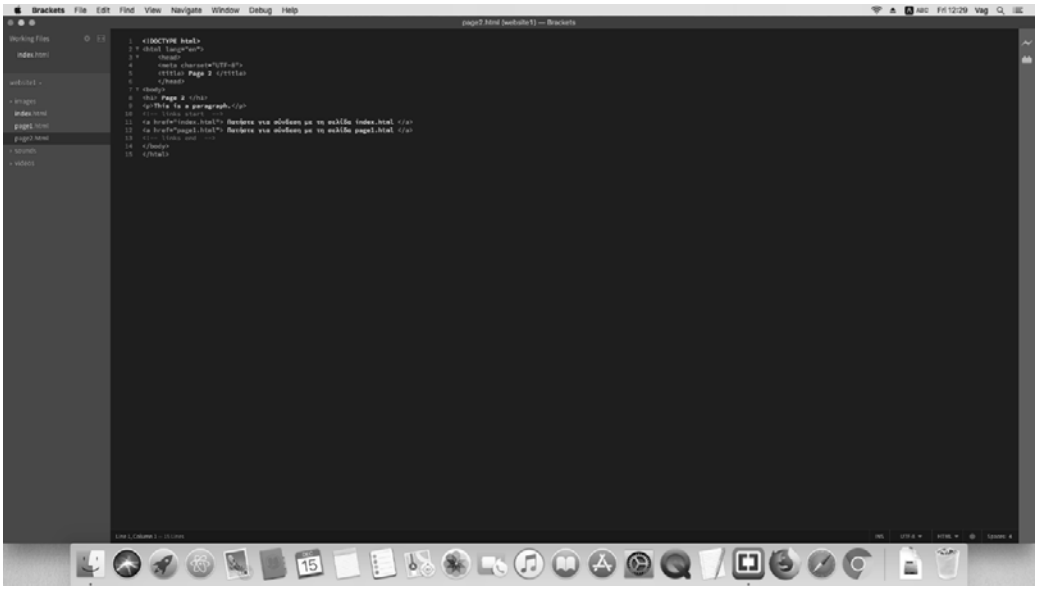
Κώδικας: page1.html



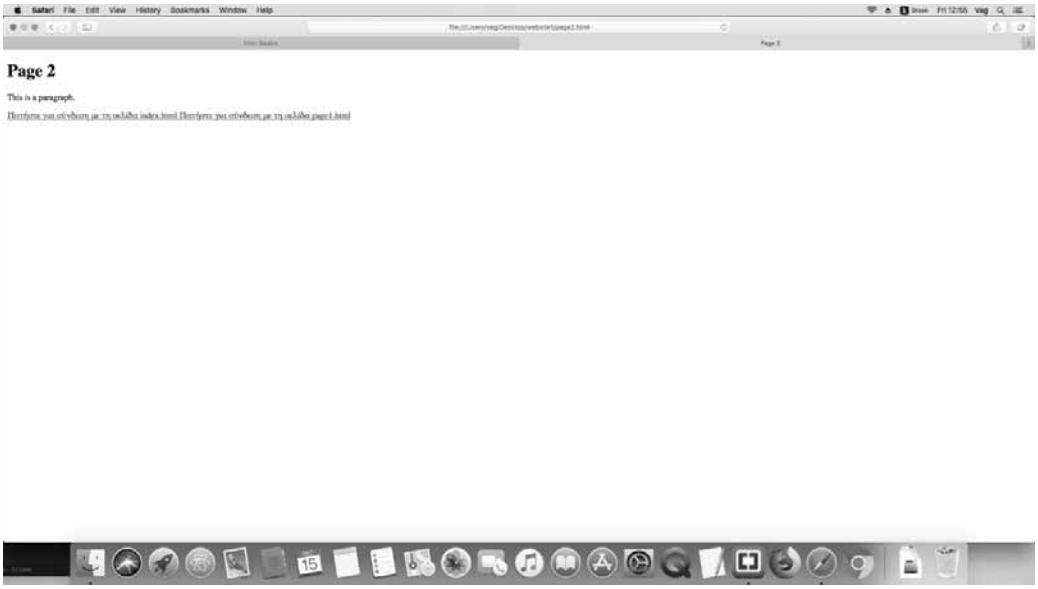
Εικόνα Browser: page1.html



Κώδικας: page2.html

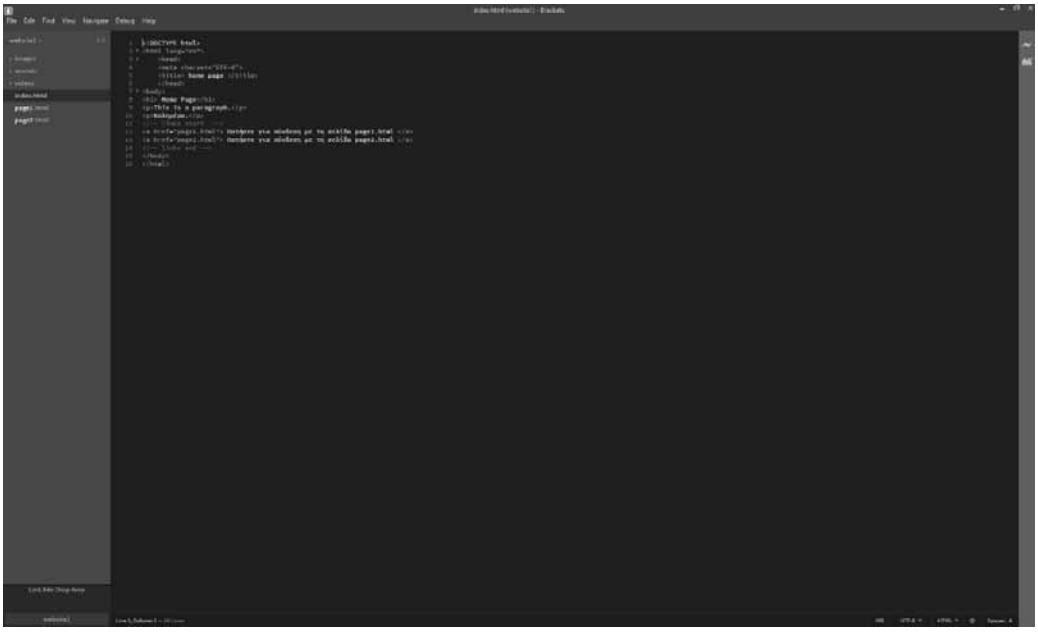


Εικόνα Browser: page2.html

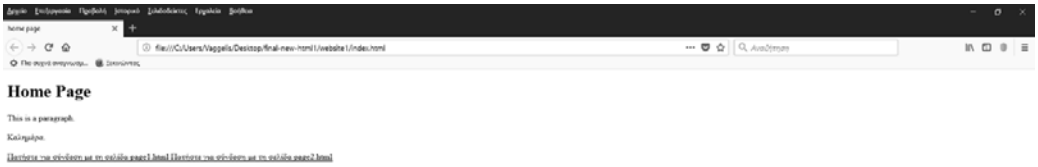


Παράδειγμα 4. Website template 1 Windows

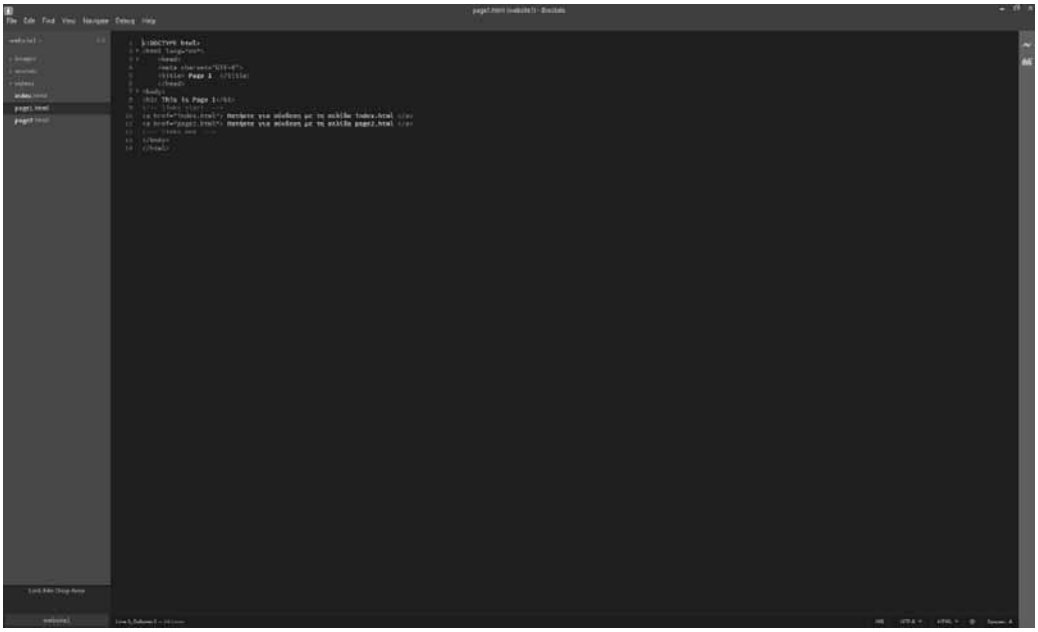
Κώδικας: index.html



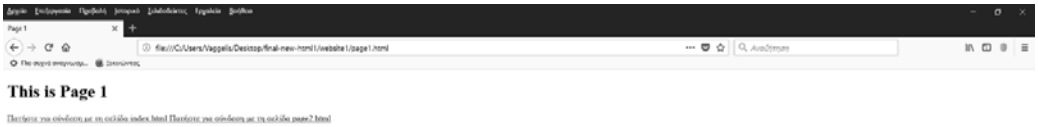
Εικόνα Browser: index.html



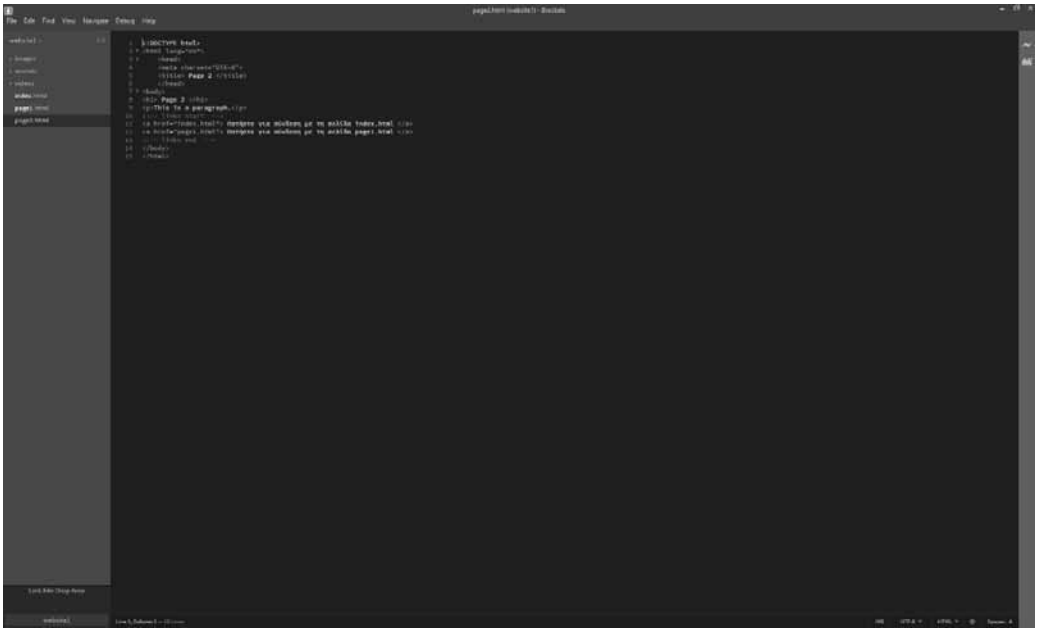
Κώδικας: page1.html



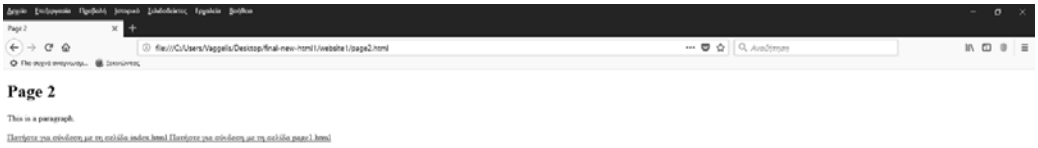
Εικόνα Browser: page1.html



Κώδικας: page2.html



Εικόνα Browser: page2.html



Βίντεο 1. Website template 1 - Mac

Βίντεο 2. Website template 1 - Windows

Μπορείτε να δείτε τα βίντεο στο αντίστοιχο σημείο του ηλεκτρονικού υλικού.

3.3.3 Εισαγωγή Buttons στα Links των σελίδων

Για να κάνουμε εισαγωγή buttons στις σελίδες έτσι ώστε να πλοηγούμαστε στο website θα κάνουμε τις παρακάτω προσθήκες και αλλαγές:

1. Αρχικά θα δημιουργήσουμε με το Da Button factory ή με ένα οποιοδήποτε άλλο εργαλείο τρία buttons για:
   Home Page
   Page 1
   Page 2
2. Θα αποθηκεύσουμε τα buttons στο subfolder images.
3. Θα χρησιμοποιήσουμε και πάλι Links τύπου a href μόνο που στη θέση του κειμένου θα υπάρχει μία σύνδεση προς το αρχείο του button. Πιο αναλυτικά θα χρησιμοποιήσουμε την παρακάτω σύνταξη:

```
<a href="___.html"> <img src="images/όνομα_αρχείου.png" /> </a>
```

Έτσι λοιπόν για τη home page θα γράψουμε τα παρακάτω Links:

```
<a href="page1.Html"> <img src="images/button_page1.png" /> </a>
```

```
<a href="page2.Html"> <img src="images/button_page2.png" /> </a>
```

Στο πρώτο link θα δημιουργήσουμε σύνδεση από τη home page προς τη σελίδα page1.html., οπότε ανάμεσα στις ετικέτες του στοιχείου a θα παρεμβάλουμε μια εικόνα με πηγή (img src) τη διαδρομή images/button_page1.png.

Αυτό σημαίνει ότι ο Browser θα ψάξει και θα βρει το αρχείο button_page1.png εντός του sub folder images και θα το εμφανίσει στη θέση του κειμένου του Link όπως είχαμε δει στο προηγούμενο παράδειγμα.

Προσοχή

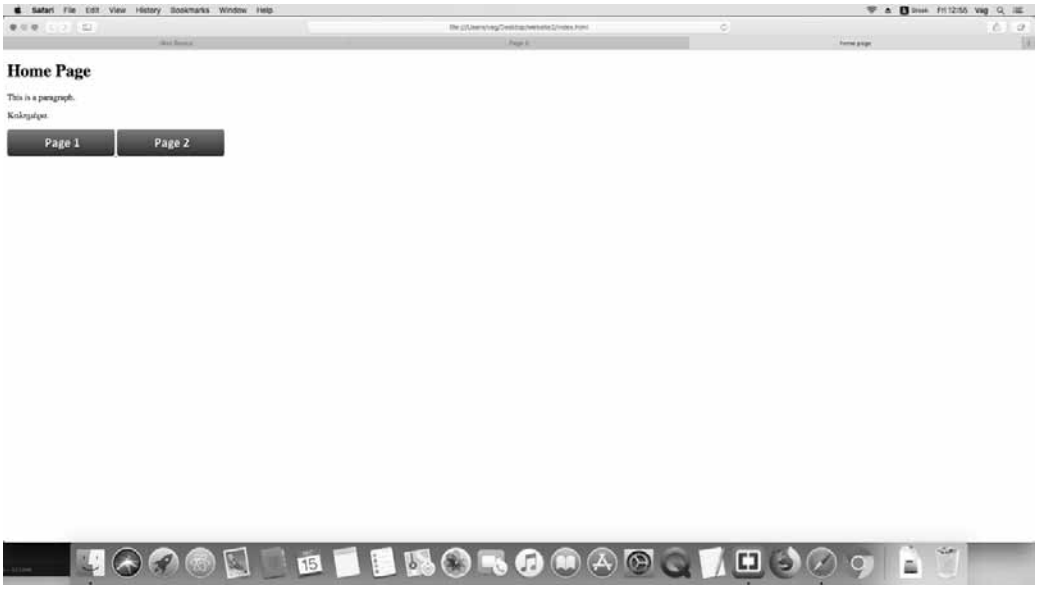
Αν το όνομα του αρχείου δεν είναι σωστό ή το αρχείο δε βρίσκεται στο sub folder images δεν θα μπορεί να εμφανιστεί.

Αντίστοιχα στο δεύτερο Link θα δημιουργήσουμε μια σύνδεση από τη home page προς τη σελίδα page2.Html. Το όνομα του αρχείου για το button θα είναι button_page2.png και θα βρίσκεται εντός του sub folder images.

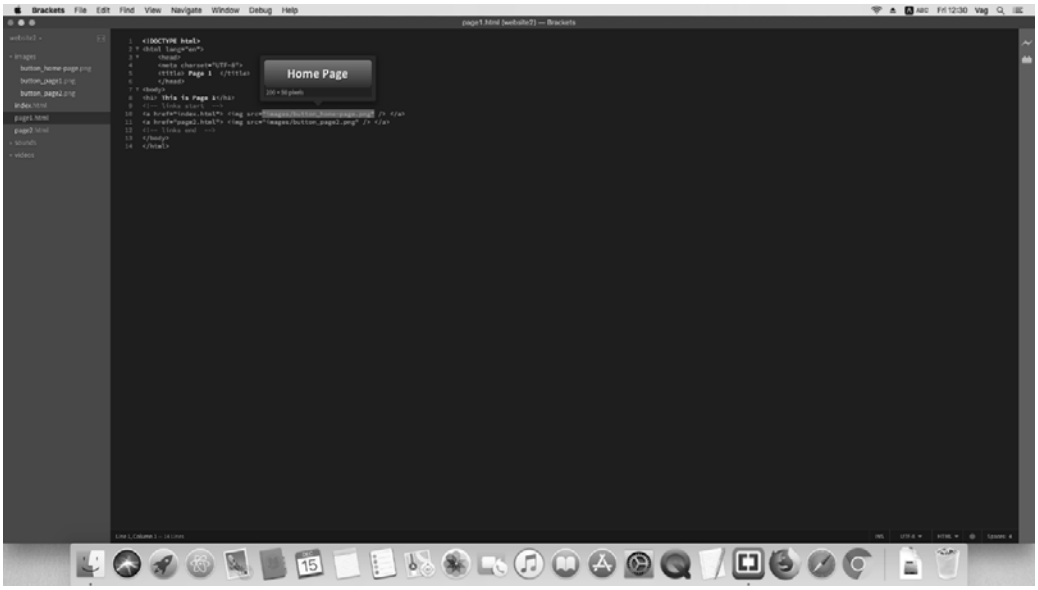
Παράδειγμα 5. Website template 2 Mac

Κώδικας: index.html

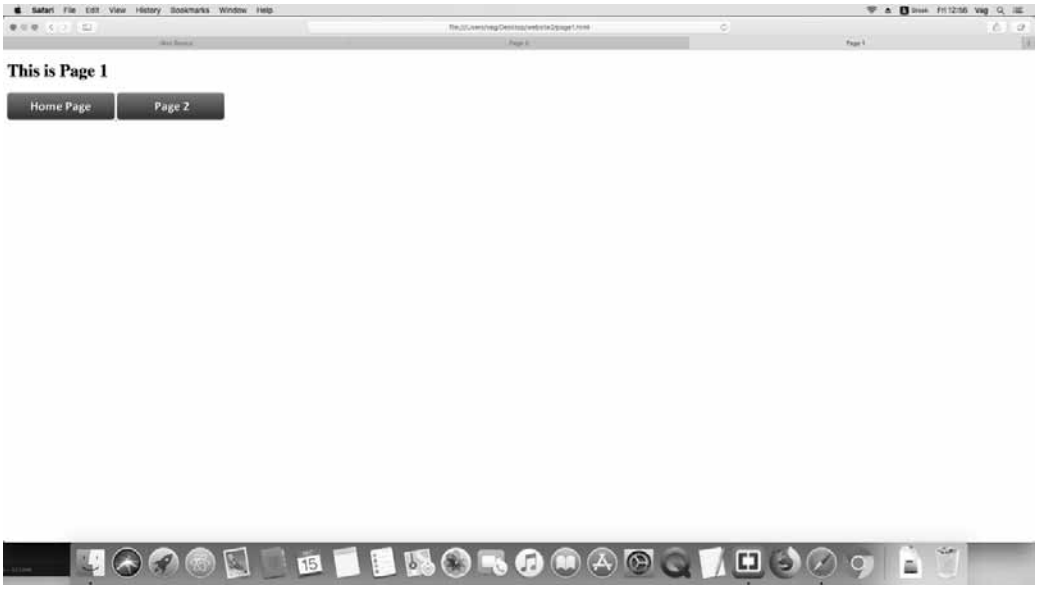
Εικόνα Browser: index.html



Κώδικας: page1.html



Εικόνα Browser: page1.html

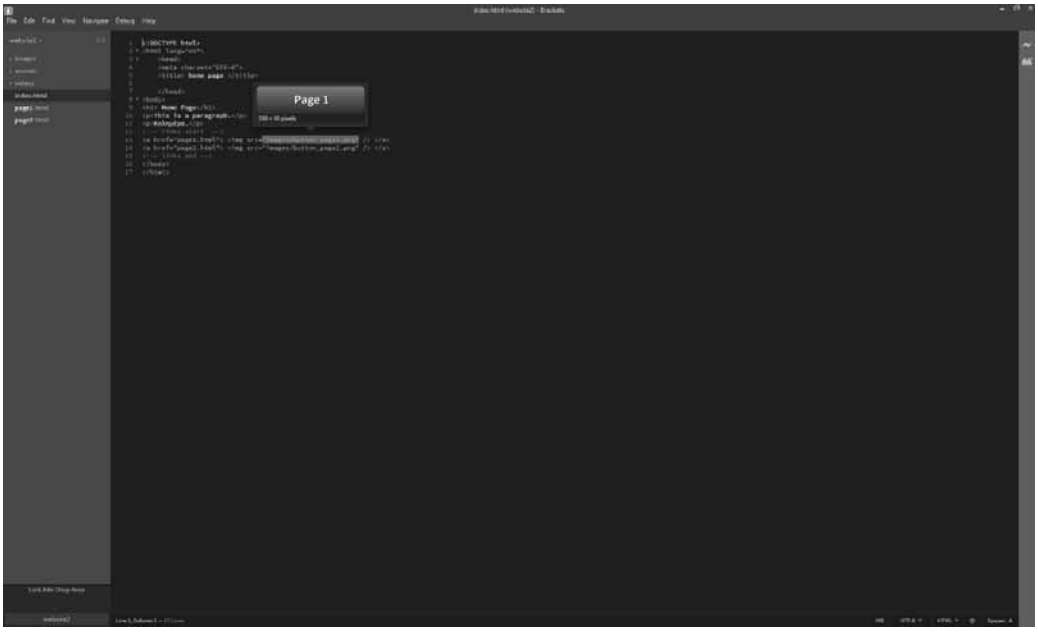


Κώδικας: page2.html

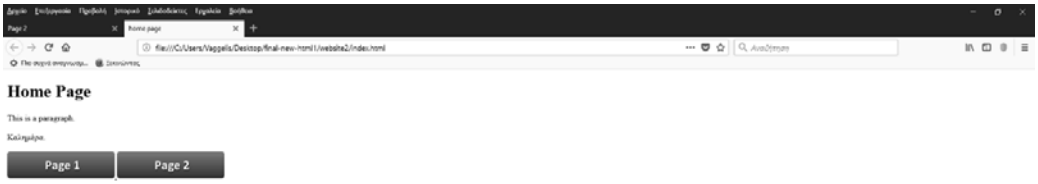
Εικόνα Browser: page2.html

Παράδειγμα 6. Website template 2 Windows

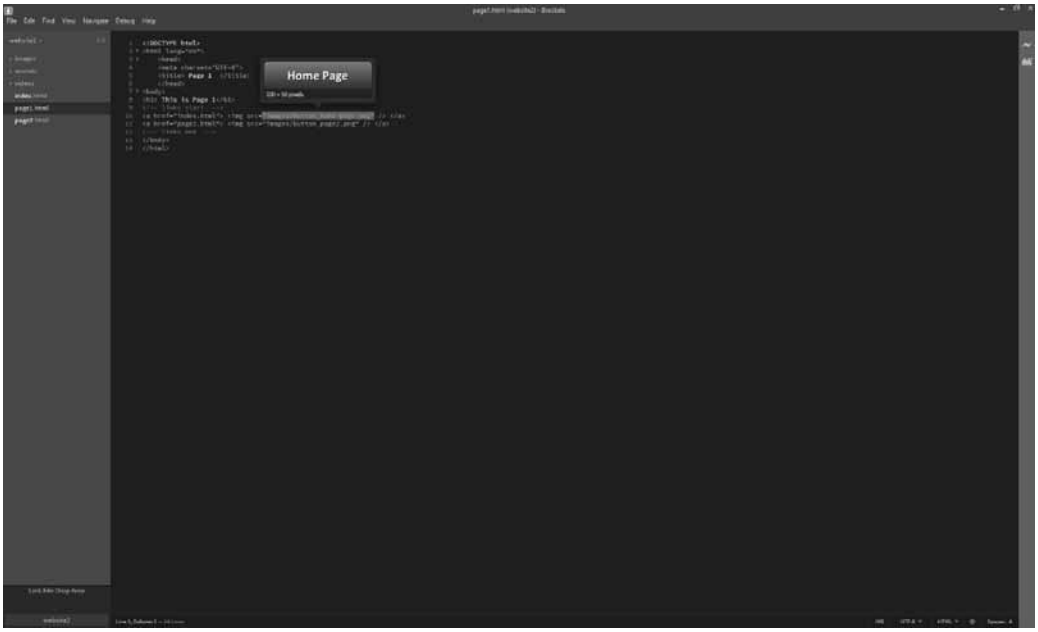
Κώδικας: index.html



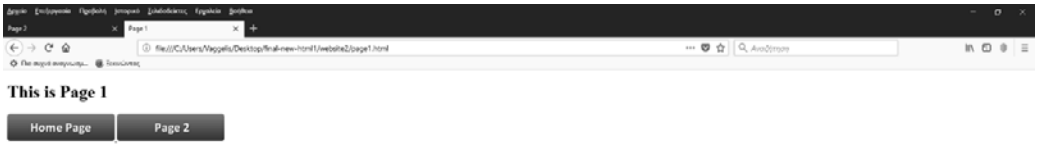
Εικόνα Browser: index.html



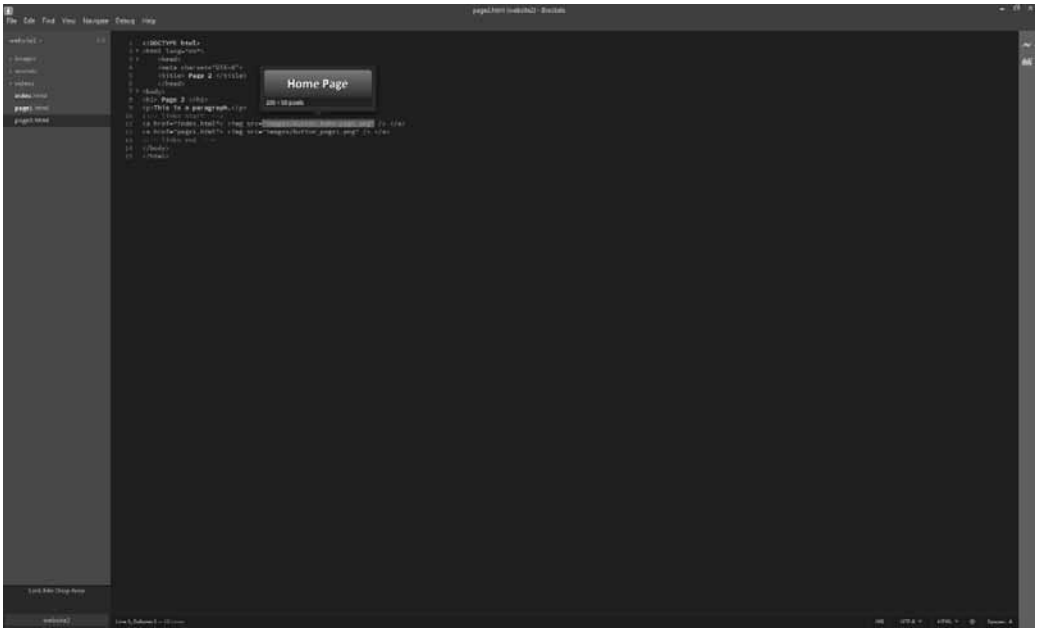
Κώδικας: page1.html



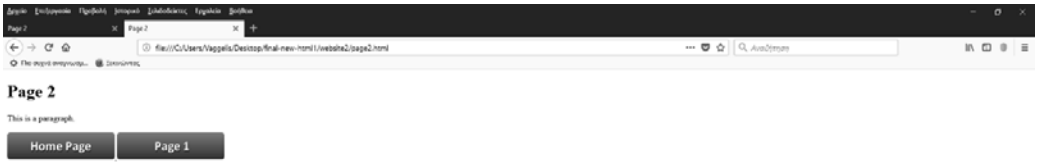
Εικόνα Browser: page1.html



Κώδικας: page2.html



Εικόνα Browser: page2.html



Βίντεο 3. Website template 2 - Mac

Βίντεο 4. Website template 2 - Windows

Μπορείτε να δείτε τα βίντεο στο αντίστοιχο σημείο του ηλεκτρονικού υλικού.

Σημείωση

Τα παραδείγματα της ενότητας τρέχουν σε οποιοδήποτε λειτουργικό σύστημα. Ακόμη και σε mobile OS όπως το Android.

Σύνοψη

Μετά την ολοκλήρωση της συγκεκριμένης διδακτικής ενότητας γνωρίζετε ότι:

- Μπορούμε να δημιουργούμε και να επεξεργαζόμαστε Html αρχεία με ένα text editor
- Μπορούμε να επιλέξουμε ένα Html editor για τη δημιουργία και επεξεργασία αρχείων Html ο οποίος μας προσφέρει επιπλέον δυνατότητες από έναν απλό text editor.
- Μπορούμε να δημιουργήσουμε σελίδες Html λαμβάνοντας υπόψη Web Usability κανόνες καθώς και τις προδιαγραφές που θέτει το Web Responsive Design
- Μπορούμε στο στοιχείο body να προσθέσουμε όλα τα στοιχεία τα οποία θα μπορεί να βλέπει ο επισκέπτης της ιστοσελίδας που θα δημιουργήσουμε,
- Μπορούμε να προσθέσουμε σχόλια με την ακόλουθη σύνταξη: <!-- και -->
- Μπορούμε να προσθέσουμε headers στις σελίδες μας με τα στοιχεία h1-h6 λαμβάνοντας υπόψη ότι το μεγαλύτερο μέγεθος αντιστοιχεί στο h1 header.
- Μπορούμε εύκολα να δημιουργήσουμε απλά buttons με τη βοήθεια ενός free online εργαλείου όπως το Da Button Factory
- Μπορούμε να δημιουργήσουμε Links μεταξύ των σελίδων με τη βοήθεια του στοιχείου a και της ιδιότητας href
- Μπορούμε στα Links με τα a στοιχεία να ενσωματώσουμε buttons με τη βοήθεια του img στοιχείου

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Προγράμματα Συμπληρωματικής Εκπαίδευσης

Με τη χρήση καινοτόμων μεθόδων εξ αποστάσεως εκπαίδευσης

1937 - ΔΗΜΙΟΥΡΓΙΑ RESPONSIVE ΣΕΛΙΔΩΝ ΜΕ HTML5 ΚΑΙ CSS

ΔΙΔΑΚΤΙΚΗ ΕΝΟΤΗΤΑ 2. ΚΕΙΜΕΝΟ ΚΑΙ ΠΙΝΑΚΕΣ ΣΕ HTML ΣΕΛΙΔΕΣ

Εισαγωγικές Παρατηρήσεις

Στην υποενότητα 1 θα γίνει ανάλυση της εισαγωγής κειμένου σε μία ιστοσελίδα με το στοιχείο p καθώς και παρουσίαση των επιλογών σε σχέση με τη μορφή και τη διάταξη του κειμένου. Επίσης θα παρουσιαστούν οι επιλογές σε σχέση με τις γραμματοσειρές και το χρώμα των γραμματοσειρών. Τέλος θα παρουσιαστούν οι χρωματικοί κώδικες που είναι συμβατοί με την Html.

Στην υποενότητα 2 θα παρουσιαστούν οι επιλογές για τη στοίχιση του κειμένου καθώς και τη μορφή του σε σχέση με το χώρο που καταλαμβάνει στη ιστοσελίδα. Επίσης θα αναλυθούν διατεταγμένες λίστες, μη διατεταγμένες λίστες και λίστες ορισμών.

Στην υποενότητα 3 θα παρουσιαστούν και θα αναλυθούν οι δυνατότητες που προσφέρει η Html σε σχέση με τη δημιουργία πινάκων. Τέλος θα παρουσιαστούν οι διαφόρων τύπων φόρμες της Html καθώς και τα dropdown lists.

ΥΠΟΕΝΟΤΗΤΑ 1. ΓΡΑΦΟΝΤΑΣ ΚΕΙΜΕΝΟ ΣΤΗΝ HTML5

1.1 Εισαγωγή κειμένου στην Html

Πριν ξεκινήσουμε να γράφουμε κείμενο στην Html5 θα πρέπει να γνωρίζουμε ότι παρόλο που η Html5 προφέρει πλήθος δυνατοτήτων και επιλογών, ο τρόπος που γράφουμε και εμφανίζουμε το κείμενο στην οθόνη του Browser δεν έχει καμία σχέση με το χειρόγραφο κείμενο ή με το κείμενο που γράφουμε σε ένα κειμενογράφο όπως το Microsoft Word.

Στοιχείο p

Το στοιχείο p αντιπροσωπεύει μία παράγραφο σε μια ιστοσελίδα. Ανάμεσα στις ετικέτες(tags) και γράφουμε το κείμενο της παραγράφου. Γενικά στο στοιχείο p μπορούμε να γράψουμε κείμενο όπως το παρακάτω:

```
<p>εισαγωγή κειμένου με το στοιχείο p </p>
```

Μια γενική αρχή για το στοιχείο p είναι ότι η διάρθρωση του κειμένου εντός της ιστοσελίδας που εμφανίζεται στην οθόνη του browser δεν ταυτίζεται με τη διάρθρωση του κειμένου όπως το έχουμε γράψει στον Html editor. Αυτό οφείλεται στον τρόπο που ο Browser «διαβάζει» και εμφανίζει το κείμενο που περικλείεται ανάμεσα στις ετικέτες και .

Γονικά στοιχεία και απόγονοι

Το Γονικό στοιχείο(parent element) περιλαμβάνει άλλα στοιχεία τα οποία ονομάζουμε απογόνους.

Παράδειγμα

```
<body>
<h1> ... </h1>
<p>……. </p>
</body>
```

Στο παραπάνω παράδειγμα το στοιχείο body είναι το γονικό στοιχείο (parent element) ενώ τα στοιχεία h1 και p είναι οι απόγονοι του στοιχείου body.

Html entities

Για τις ανάγκες της συγγραφής Html κώδικα κάποιοι χαρακτήρες είναι δεσμευμένοι, όπως για παράδειγμα οι μαθηματικοί τελεστές μεγαλύτερο (>) ή μικρότερο (<). Γράφοντας με τη μορφή κειμένου τους τελεστές μεγαλύτερο ή μικρότερο ο browser πιθανότατα θα καταλάβει ότι πρόκειται για tags με αποτέλεσμα να δημιουργούν λάθη στον κώδικα. Αυτούς τους ειδικούς χαρακτήρες θα πρέπει να τους γράφουμε με διαφορετική μορφή για να μην προκαλούνται λάθη κατά την εκτέλεση του Html κώδικα.

Για τον κάθε ειδικό χαρακτήρα έχουμε δύο συμβατές μορφές:

- &entity_name
- &entity_number

Έτσι λοιπόν αν σε μια παράγραφο επιθυμούμε να γράψουμε με τη μορφή κειμένου τους χαρακτήρες για μεγαλύτερο ή μικρότερο μπορούμε να γράψουμε

 χαρακτήρες < > Σε μορφή entity name

 χαρακτήρες < > Σε μορφή entity number

Πίνακας 1. Html entities

Result	Description	Entity Name	Entity Number
	non-breaking space		
<	less than	<	<
>	greater than	>	>
&	ampersand	&	&
"	double quotation mark	"	"
'	single quotation mark (apostrophe)	'	'
¢	cent	¢	¢
£	pound	£	£
¥	yen	¥	¥
€	euro	€	€
©	copyright	©	©
®	registered trademark	®	®

Πηγή: *https://www.w3schools.com*

**Ετικέτα
**

Η ετικέτα
 μόλις εκτελεστεί προσθέτει αλλαγή γραμμής στην εμφάνιση των στοιχείων της σελίδας. Η ετικέτα
 δεν γράφεται ως ζεύγος.

Παράδειγμα

```
<p> line break with br  <br> </p>
```

Ετικέτα <hr>

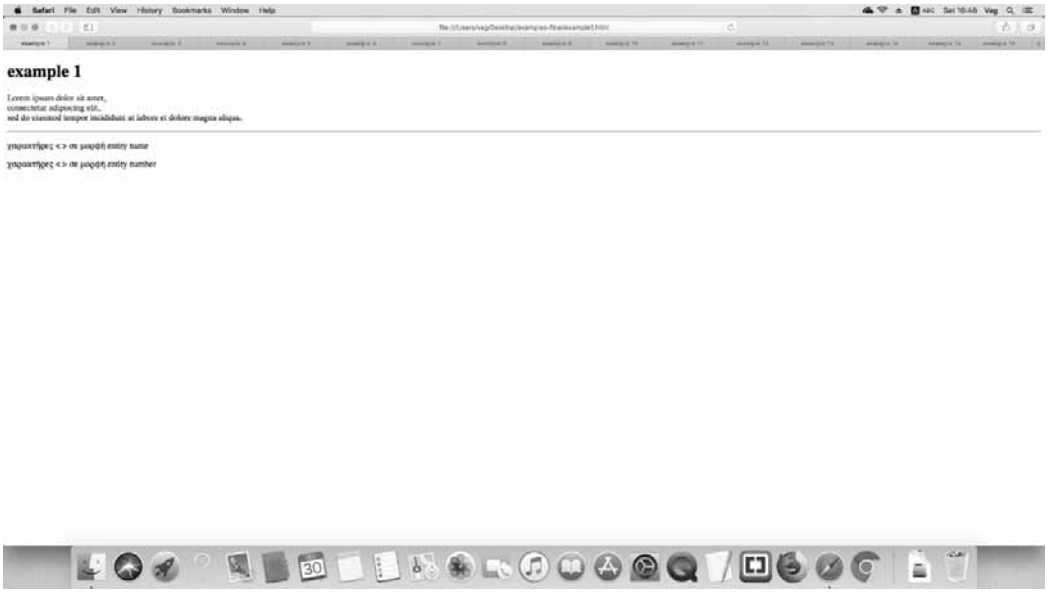
Προσθέτει μια οριζόντια γραμμή στην ιστοσελίδα μας και δεν υφίσταται ως ζεύγος ετικετών. Ορίζει θεματικές περιοχές στη σελίδα μας.

```
<p>τέλος Θεματικής ενότητας 1 </p>

<hr>

<p>Αρχή Θεματικής ενότητας 2 </p>
```

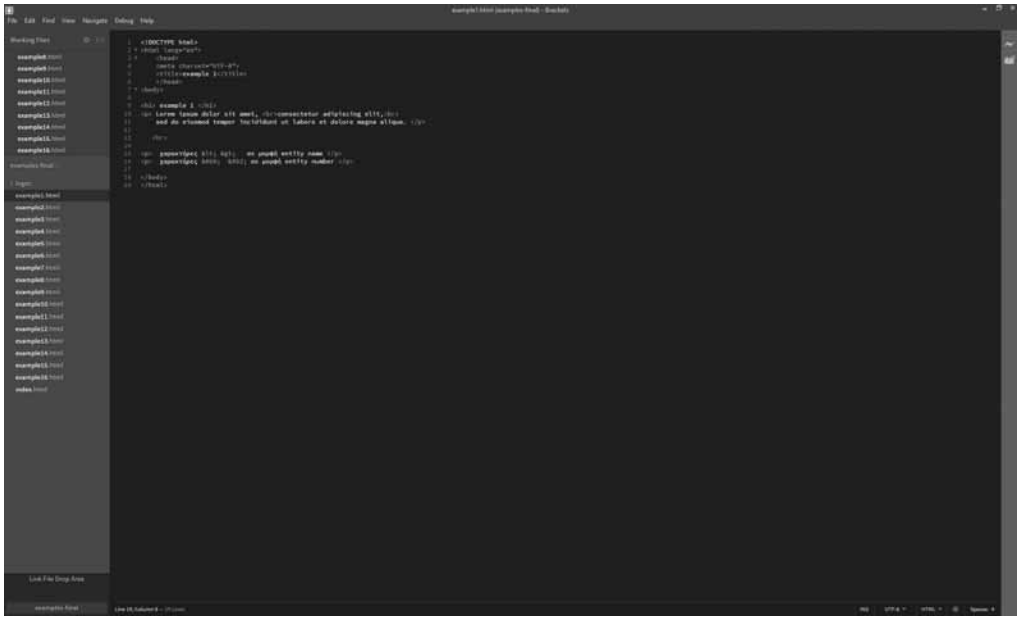
*Παράδειγμα 1. <p>, hr>,
 και html entities*

Κώδικας(Mac): <p>, hr>,
 και html entities

Εικόνα Browser(Mac): <p>, hr>,
 και html entities



Κώδικας(Win): <p>, hr>,
και html entities



Εικόνα Browser(Win):

Στοιχείο footer

Μπορούμε να εισάγουμε υποσέλιδο γράφοντας κείμενο μεταξύ των ετικετών <footer> και </footer>

Παράδειγμα

```
<footer>
<p>στοιχεία επικοινωνίας: ...</p>
</footer>
```

Βασικές μορφοποιήσεις κειμένου σε Html σελίδες

Με τη βοήθεια του CSS θα εφαρμόσουμε βασικές μορφοποιήσεις για το κείμενο μιας ιστοσελίδας. Το CSS θα αναλυθεί περισσότερο στις επόμενες ενότητες. Πριν ξεκινήσουμε να χρησιμοποιούμε το CSS καλό θα είναι να γνωρίζουμε ένα βασικό κανόνα, όπου κατά την εμφάνιση της σελίδας αν για κάποιο στοιχείο υπάρχουν περισσότερες από μια τιμές για μια CSS ιδιότητα, μετράει η τιμή που έχει γραφεί τελευταία. Δηλαδή, η επόμενη τιμή μιας ιδιότητας διαγράφει την προηγούμενη.

Style στο στοιχείο head εφαρμοζόμενο σε όλα τα Html elements ίδιου τύπου

Ένα τρόπος να εφαρμόσουμε CSS style είναι να γράψουμε κώδικα εντός του στοιχείου head, δηλαδή ανάμεσα στις ετικέτες:

```
<head>
```

και

```
</head>
```

Τον CSS κώδικα σε αυτή την περίπτωση θα πρέπει να τον γράψουμε ανάμεσα στις ετικέτες:

```
<style>
```

και

```
</style>
```

Για να δηλώσουμε σε ποια στοιχεία θα εφαρμοστεί το CSS style θα χρειαστούμε τους selectors (επιλογείς). Αρχικά θα γράφουμε τον επιλογέα ο οποίος θα είναι και το όνομα του στοιχείου στο οποίο αναφερόμαστε και στη συνέχεια εντός αγκίστρων θα γράφουμε τις ιδιότητες και τις τιμές αυτών όπως ακολούθως:

```
όνομα_στοιχείου {ιδιότητα:τιμή; ιδιότητα:τιμή; ιδιότητα:τιμή;}
```

Στο τέλος της κάθε τιμής θα προσθέτουμε και τον χαρακτήρα ';' (Ελληνικό ερωτηματικό).

Σημείωση

Με αυτή τη μέθοδο θα εφαρμοστεί CSS σε όλα τα στοιχεία του ίδιου τύπου. Στο παρακάτω παράδειγμα θα εφαρμοστεί η συγκεκριμένη τιμή χρώματος σε όλα τα στοιχεία p(αν δεν υπάρχει άλλη τιμή CSS για την ίδια ιδιότητα γραμμένη στον κώδικα μετά από αυτή).

Παράδειγμα

```
<style>
   p {color:rgb(255, 0, 191);}
      </style>
```

Σημείωση

Μπορούμε να γράψουμε και παραπάνω επιλογείς για ένα σετ ιδιοτήτων:

```
όνομα_στοιχείου1, όνομα_στοιχείου2  {ιδιότητα:τιμή; ιδιότητα:τιμή; ιδιότητα:τιμή;}
```

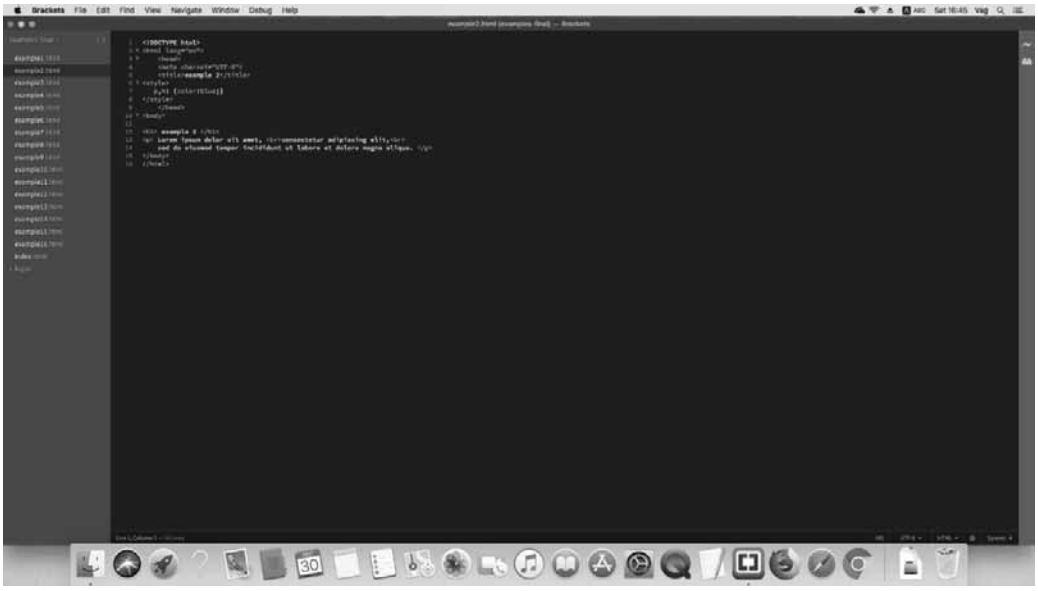
Στο παρακάτω παράδειγμα θα εφαρμοστεί η τιμή του χρώματος για όλα τα στοιχεία p καθώς και για όλα τα στοιχεία h1.

Παράδειγμα

```
<style>
  p, h1 {color:blue;}
</style>
```

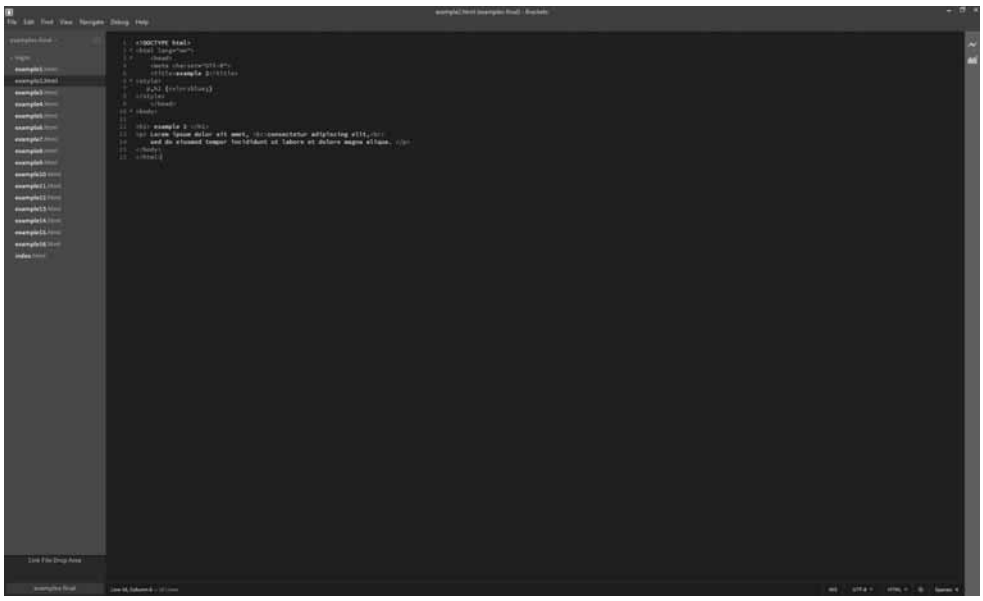
Παράδειγμα 2. style

Κώδικας(Mac): <style>



Εικόνα Browser(Mac): <style>

Κώδικας(Win): <style>



Εικόνα Browser(Win): <style>

Εφαρμόζοντας ιδιότητες CSS για συγκεκριμένα στοιχεία της σελίδας μας

Έστω ότι θέλουμε να εφαρμόσουμε CSS σε ένα ή περισσότερα στοιχεία της σελίδας μας αλλά όχι σε όλα του ίδιου τύπου, όπως για παράδειγμα όχι σε όλα τα στοιχεία p. Οι τρόποι που μπορούμε να το πετύχουμε είναι οι ακόλουθοι:

1. Χρήση style εντός της ετικέτας έναρξης τους στοιχείου:

Εντός της ετικέτας έναρξης γράφουμε τη δήλωση

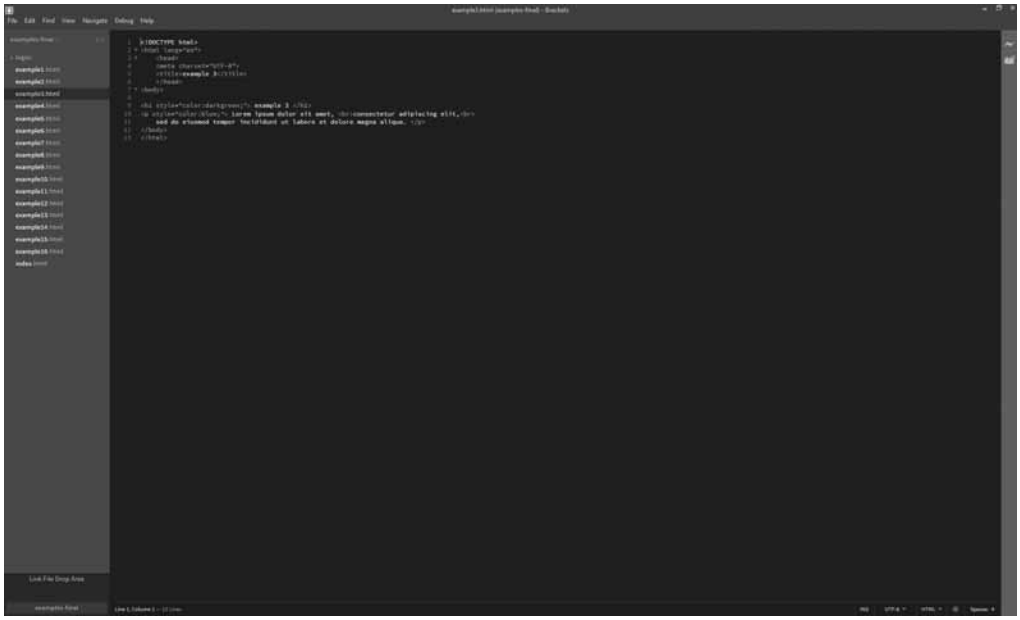
style="ιδιότητα:τιμή; ιδιότητα:τιμή; ιδιότητα:τιμή;"

Παράδειγμα 3. Style εντός της ετικέτας

Κώδικας(Mac): style εντός της ετικέτας

Εικόνα Browser(Mac): style εντός της ετικέτας

Κώδικας(Win): style εντός της ετικέτας



Εικόνα Browser(Win): style εντός της ετικέτας

Παράδειγμα

```
<p style="color:orangered">This is a paragraph </p>
```

2. Δημιουργώντας κλάσεις. Με τη δήλωση class μπορούμε να γράψουμε ένα σύνολο ιδιοτήτων CSS και στη συνέχεια να τις εφαρμόζουμε σε όποιο στοιχείο Html περιέχει στην ετικέτα έναρξης το όνομα της συγκεκριμένης κλάσης.

Τα βήματα που θα πρέπει να ακολουθήσουμε είναι τα παρακάτω:

I. Εντός του στοιχείου head και μεταξύ των ετικετών <style> και </style> ορίζουμε μια κλάση γράφοντας '.' και στη συνέχεια το όνομα της κλάσης(χωρίς να μεσολαβεί κενός χαρακτήρας). Μετά το όνομα της κλάσης εντός των αγκίστρων γράφουμε τις CSS ιδιότητες.

Σύνταξη ορισμού κλάσης

```
<style>
    .όνομα_κλάσης ={ιδιότητα:τιμή; ιδιότητα:τιμή; ιδιότητα:τιμή;}
    </style>
```

Παράδειγμα ορισμού κλάσης

```
<style>
    .color1 {color: mediumvioletred;background-color: dodgerblue;}
</style>
```

Σημείωση

Η ιδιότητα background-color αναφέρεται στο χρώμα του φόντου.

II. Κλήση της κλάσης σε στοιχείο Html

Για να καλέσουμε την κλάση σε ένα ή περισσότερα στοιχεία Html αρκεί στην ετικέτα έναρξης του στοιχείου να γράψουμε:

```
class="όνομα_κλάσης"
```

Παράδειγμα

```
<p class="color1">    Lorem ipsum dolor sit amet, consectetur adipiscing elit,
sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. </p>
```

Σημείωση

Μπορούμε σε ένα στοιχείο να καλέσουμε και παραπάνω από μία κλάσεις αφήνοντας ένα κενό χαρακτήρα ανάμεσα στα ονόματα των κλάσεων:

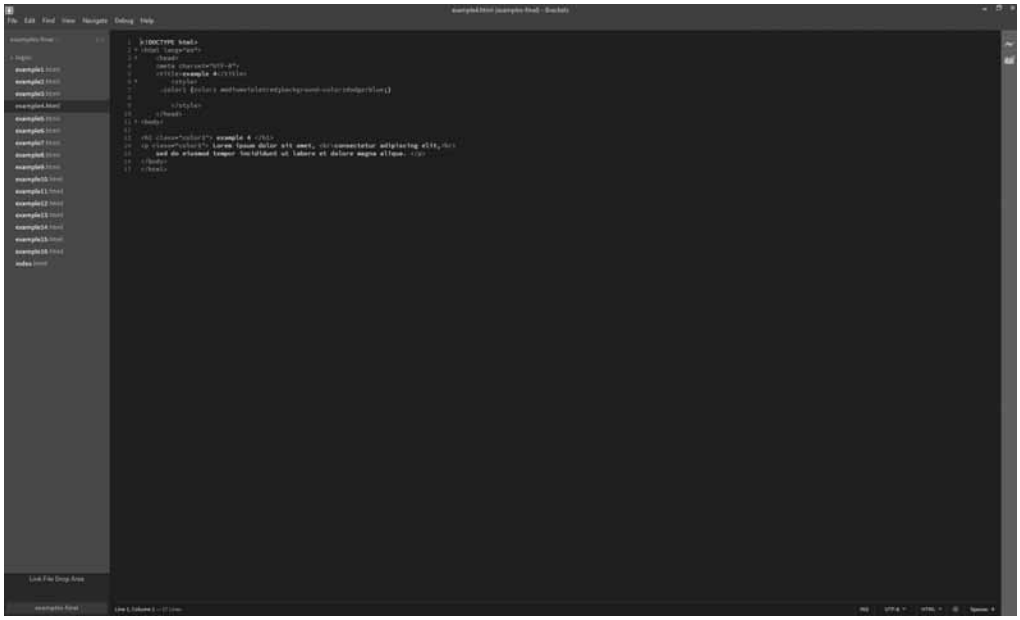
```
class="όνομα_κλάσης όνομα_κλάσης2…"
```

Παράδειγμα 4. class

Κώδικας(Mac): class

Εικόνα Browser(Mac): class

Κώδικας(Win): class



Εικόνα Browser(Win): class

1.2 Γραμματοσειρές

Font-family

Με την ιδιότητα font-family μπορούμε να ορίσουμε μια ομάδα γραμματοσειρών για την εμφάνιση του κειμένου στην ιστοσελίδα μας. Στην ιδιότητα font family γράφουμε τα ονόματα των γραμματοσειρών με σειρά

προτεραιότητας ως προς την υποστήριξη των συγκεκριμενών γραμματοσειρών από τον Browser με την ακόλουθη σύνταξη:

```
p {
   font-family: όνομα_γραμματοσειράς1, όνομα_γραμματοσειράς2,
όνομα_γραμματοσειράς3;
}
```

Σημείωση

Αν το όνομα της γραμματοσειράς αποτελείται από δύο ή παραπάνω λέξεις τότε το όνομα της γραμματοσειράς θα περιλαμβάνεται εντός διπλών εισαγωγικών όπως για παράδειγμα "Times New Roman".

Λειτουργία της ιδιότητας font-family

Ο Browser κατά την εκτέλεση της ιδιότητας font-family διαβάζει την πρώτη γραμματοσειρά. Στην περίπτωση που την υποστηρίζει εφαρμόζει τη συγκεκριμένη γραμματοσειρά. Αν δεν την υποστηρίζει διαβάζει τη δεύτερη και αν την υποστηρίζει την εμφανίζει κοκ. Οπότε οι γραμματοσειρές από τη δεύτερη θέση και μετά λειτουργούν ως εφεδρείες. Αν ο Browser υποστηρίζει την πρώτη δε θα χρησιμοποιηθούν οι υπόλοιπες.

Παράδειγμα

```
p {
   font-family: "Times New Roman", Times, Georgia, serif;
}
```

Πίνακας 2. Επιλογές για Font-Family

Font-family	Sample
'sans-serif': normal fonts *without* serifs	
Arial, sans-serif	The Quick Brown Fox Jumps Over The Lazy Dog
Helvetica, sans-serif	The Quick Brown Fox Jumps Over The Lazy Dog
Verdana, sans-serif	The Quick Brown Fox Jumps Over The Lazy Dog
Trebuchet MS, sans-serif	The Quick Brown Fox Jumps Over The Lazy Dog
Gill Sans, sans-serif	The Quick Brown Fox Jumps Over The Lazy Dog
Noto Sans, sans-serif	The Quick Brown Fox Jumps Over The Lazy Dog
Avantgarde, TeX Gyre Adventor, URW Gothic L, sans-serif	The Quick Brown Fox Jumps Over The Lazy Dog
Optima, sans-serif	The Quick Brown Fox Jumps Over The Lazy Dog
Arial Narrow, sans-serif	The Quick Brown Fox Jumps Over The Lazy Dog
sans-serif	The Quick Brown Fox Jumps Over The Lazy Dog

'serif': normal fonts *with* serifs	
Times, Times New Roman, serif	The Quick Brown Fox Jumps Over The Lazy Dog
Didot, serif	The Quick Brown Fox Jumps Over The Lazy Dog
Georgia, serif	The Quick Brown Fox Jumps Over The Lazy Dog
Palatino, URW Palladio L, serif	The Quick Brown Fox Jumps Over The Lazy Dog
Bookman, URW Bookman L, serif	The Quick Brown Fox Jumps Over The Lazy Dog
New Century Schoolbook, TeX Gyre Schola, serif	The Quick Brown Fox Jumps Over The Lazy Dog
American Typewriter, serif	The Quick Brown Fox Jumps Over The Lazy Dog
serif	The Quick Brown Fox Jumps Over The Lazy Dog

'monospace': fixed-width fonts	
Andale Mono, monospace	The Quick Brown Fox Jumps Over The Lazy Dog
Courier New, monospace	The Quick Brown Fox Jumps Over The Lazy Dog
Courier, monospace	The Quick Brown Fox Jumps Over The Lazy Dog
FreeMono, monospace	The Quick Brown Fox Jumps Over The Lazy Dog
OCR A Std, monospace	The Quick Brown Fox Jumps Over The Lazy Dog
DejaVu Sans Mono, monospace	The Quick Brown Fox Jumps Over The Lazy Dog
monospace	The Quick Brown Fox Jumps Over The Lazy Dog

'cursive': fonts that emulate handwriting	
Comic Sans MS, Comic Sans, cursive	The Quick Brown Fox Jumps Over The Lazy Dog
Apple Chancery, cursive	The Quick Brown Fox Jumps Over The Lazy Dog
Bradley Hand, cursive	The Quick Brown Fox Jumps Over The Lazy Dog
Brush Script MT, Brush Script Std, cursive	*The Quick Brown Fox Jumps Over The Lazy Dog*
Snell Roundhand, cursive	The Quick Brown Fox Jumps Over The Lazy Dog
URW Chancery L, cursive	The Quick Brown Fox Jumps Over The Lazy Dog
cursive	The Quick Brown Fox Jumps Over The Lazy Dog

'fantasy': decorative fonts, for titles, etc.	
Impact, fantasy	The Quick Brown Fox Jumps Over The Lazy Dog
Luminari, fantasy	THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG
Chalkduster, fantasy	THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG
Jazz LET, fantasy	THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG
Blippo, fantasy	THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG
Stencil Std, fantasy	THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG
Marker Felt, fantasy	THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG
Trattatello, fantasy	THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG
fantasy	THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG

Πηγή: *https://www.w3.org*

Μέγεθος της γραμματοσειράς (font-size)

Για το μέγεθος της γραμματοσειράς θα χρησιμοποιήσουμε την ιδιότητα font-size με την παρακάτω σύνταξη:

font-size:τιμή;

Στη θέση της τιμής θα συμπληρώσουμε μία από τις επιλογές του σχετικού πίνακα 3.

Πίνακας 3. Font size

Τιμή font-size	Μέγεθος
Medium **xx-small** **x-small** **small** **large** **x-large** **xx-large**	Medium xx-small x-small small large x-large xx-large
smaller	Μικρότερο μέγεθος από το πατρικό στοιχείο
larger	Μεγαλύτερο μέγεθος από το πατρικό στοιχείο
length	Απόλυτο μέγεθος σε px, cm, em
%	Μέγεθος ίσο με την επί % αναλογία του πατρικού στοιχείου
initial	Θέτει την προεπιλεγμένη τιμή
inherit	Κληρονομεί το μέγεθος γραμματοσειράς του γονικού στοιχείου

Η μονάδα 1px αντιστοιχεί σε ένα pixel της οθόνης μας. Οι μονάδες των απόλυτων μεγεθών θα πρέπει να γράφονται αμέσως μετά τα μεγέθη χωρίς να υπάρχει κενός χαρακτήρας:

- Σωστό 1px
- Λάθος 1 px

Παράδειγμα:

```
p{font-size:150%;}
p{font-size:12px;}
p{font-size:0.5em;}
```

Η μονάδα em ορίζεται σε σχέση με το πατρικό στοιχείο και προσδιορίζει το μέγεθος της γραμματοσειράς σε σχέση με το πατρικό στοιχείο. Έτσι λοιπόν αν το πατρικό στοιχείο έχει γραμματοσειρά μεγέθους 22px και έχουμε γράψει

```
p{font-size:0.5em;}
```

τότε η συγκεκριμένη γραμματοσειρά θα έχει μέγεθος 0.5 X 22=11px.

Χρώμα Γραμματοσειράς

Μπορούμε να προσθέσουμε χρώμα γραμματοσειράς με την ιδιότητα color σύμφωνα με την παρακάτω σύνταξη:

p {color:τιμή}

Για την τιμή στην ιδιότητα color υπάρχουν οι παρακάτω επιλογές:

1. Επιλογή κατά όνομα (color name)

Η Html5 προσφέρει τη δυνατότητα να γράψουμε το όνομα ενός εκ των προεπιλεγμένων χρωμάτων όπως red, blue, aqua, yellow κ.α. Τα συγκεκριμένα χρώματα υποστηρίζονται από όλους τους Browsers.

Πίνακας 4. Χρωματικός πίνακας color name

Color Name	HEX	Color
AliceBlue	#F0F8FF	
AntiqueWhite	#FAEBD7	
Aqua	#00FFFF	
Aquamarine	#7FFFD4	
Azure	#F0FFFF	
Beige	#F5F5DC	
Bisque	#FFE4C4	
Black	#000000	
BlanchedAlmond	#FFEBCD	
Blue	#0000FF	
BlueViolet	#8A2BE2	
Brown	#A52A2A	
BurlyWood	#DEB887	
CadetBlue	#5F9EA0	
Chartreuse	#7FFF00	
Chocolate	#D2691E	
Coral	#FF7F50	
CornflowerBlue	#6495ED	
Cornsilk	#FFF8DC	
Crimson	#DC143C	
Cyan	#00FFFF	
DarkBlue	#00008B	
DarkCyan	#008B8B	
DarkGoldenRod	#B8860B	
DarkGray	#A9A9A9	
DarkGrey	#A9A9A9	
DarkGreen	#006400	
DarkKhaki	#BDB76B	
DarkMagenta	#8B008B	
DarkOliveGreen	#556B2F	
DarkOrange	#FF8C00	
DarkOrchid	#9932CC	
DarkRed	#8B0000	
DarkSalmon	#E9967A	
DarkSeaGreen	#8FBC8F	
DarkSlateBlue	#483D8B	
DarkSlateGray	#2F4F4F	

DarkSlateGrey	#2F4F4F	
DarkTurquoise	#00CED1	
DarkViolet	#9400D3	
DeepPink	#FF1493	
DeepSkyBlue	#00BFFF	
DimGray	#696969	
DimGrey	#696969	
DodgerBlue	#1E90FF	
FireBrick	#B22222	
FloralWhite	#FFFAF0	
ForestGreen	#228B22	
Fuchsia	#FF00FF	
Gainsboro	#DCDCDC	
GhostWhite	#F8F8FF	
Gold	#FFD700	
GoldenRod	#DAA520	
Gray	#808080	
Grey	#808080	
Green	#008000	
GreenYellow	#ADFF2F	
HoneyDew	#F0FFF0	
HotPink	#FF69B4	
IndianRed	#CD5C5C	
Indigo	#4B0082	
Ivory	#FFFFF0	
Khaki	#F0E68C	
Lavender	#E6E6FA	
LavenderBlush	#FFF0F5	
LawnGreen	#7CFC00	
LemonChiffon	#FFFACD	
LightBlue	#ADD8E6	
LightCoral	#F08080	
LightCyan	#E0FFFF	
LightGoldenRodYellow	#FAFAD2	
LightGray	#D3D3D3	
LightGrey	#D3D3D3	
LightGreen	#90EE90	
LightPink	#FFB6C1	

LightSalmon	#FFA07A	
LightSeaGreen	#20B2AA	
LightSkyBlue	#87CEFA	
LightSlateGray	#778899	
LightSlateGrey	#778899	
LightSteelBlue	#B0C4DE	
LightYellow	#FFFFE0	
Lime	#00FF00	
LimeGreen	#32CD32	
Linen	#FAF0E6	
Magenta	#FF00FF	
Maroon	#800000	
MediumAquaMarine	#66CDAA	
MediumBlue	#0000CD	
MediumOrchid	#BA55D3	
MediumPurple	#9370DB	
MediumSeaGreen	#3CB371	
MediumSlateBlue	#7B68EE	
MediumSpringGreen	#00FA9A	
MediumTurquoise	#48D1CC	
MediumVioletRed	#C71585	
MidnightBlue	#191970	
MintCream	#F5FFFA	
MistyRose	#FFE4E1	
Moccasin	#FFE4B5	
NavajoWhite	#FFDEAD	
Navy	#000080	
OldLace	#FDF5E6	
Olive	#808000	
OliveDrab	#6B8E23	
Orange	#FFA500	
OrangeRed	#FF4500	
Orchid	#DA70D6	
PaleGoldenRod	#EEE8AA	
PaleGreen	#98FB98	
PaleTurquoise	#AFEEEE	
PaleVioletRed	#DB7093	
PapayaWhip	#FFEFD5	

PeachPuff	#FFDAB9	
Peru	#CD853F	
Pink	#FFC0CB	
Plum	#DDA0DD	
PowderBlue	#B0E0E6	
Purple	#800080	
RebeccaPurple	#663399	
Red	#FF0000	
RosyBrown	#BC8F8F	
RoyalBlue	#4169E1	
SaddleBrown	#8B4513	
Salmon	#FA8072	
SandyBrown	#F4A460	
SeaGreen	#2E8B57	
SeaShell	#FFF5EE	
Sienna	#A0522D	
Silver	#C0C0C0	
SkyBlue	#87CEEB	
SlateBlue	#6A5ACD	
SlateGray	#708090	
SlateGrey	#708090	
Snow	#FFFAFA	
SpringGreen	#00FF7F	
SteelBlue	#4682B4	
Tan	#D2B48C	
Teal	#008080	
Thistle	#D8BFD8	
Tomato	#FF6347	
Turquoise	#40E0D0	
Violet	#EE82EE	
Wheat	#F5DEB3	
White	#FFFFFF	
WhiteSmoke	#F5F5F5	
Yellow	#FFFF00	
YellowGreen	#9ACD32	

Πηγή: *https://www.w3schools.com*

Παράδειγμα

```
<style>
   p {color:steelblue;}
</style>
```

2. Επιλογή με τη μορφή δεκαεξαδικού χρωματικού κώδικα

Σε αυτή την περίπτωση γράφουμε τον χαρακτήρα # και στη συνέχεια ανά δυάδες δίνουμε δεκαεξαδικές τιμές για τα χρώματα red, green και blue σύμφωνα με την παρακάτω σύνταξη:

#rrggbb

Παράδειγμα

```
<style>
    p {color: #F0F8FF}
    </style>
```

Για την επιλογή χρωμάτων με δεκαεξαδική μορφή μπορούμε να δημιουργήσουμε το δικό μας custom color επιλέγοντας τιμές για τα βασικά χρώματα. Για να επιλέξουμε και να συνθέσουμε χρώματα μπορούμε να μεταβούμε στη σελίδα:

https://www.w3schools.com/colors/colors_hexadecimal.asp

Επιλογή με τη μορφή rgb και rgba χρωματικού κώδικα

Μπορούμε όπως και στην αναζήτηση χρώματος σε δεκαεξαδικό χρωματικό κώδικα να δημιουργήσουμε το δικό μας χρώμα με τη βοήθεια της σελίδας Html color Picker:

https://www.w3schools.com/colors/colors_picker.asp

Ο κώδικας rgb περιλαμβάνει τρεις τιμές διαχωριζόμενες από κόμμα σύμφωνα με την παρακάτω σύνταξη:

rgb(r,g,b)

Οι τιμές r,g,b είναι φυσικοί αριθμοί από το 0 έως το 255 και αντιπροσωπεύουν τιμές για τα τρία χρώματα κόκκινο, πράσινο και μπλε.

Παράδειγμα

```
    <style>
p {color:rgb(255, 0, 191);}
    </style>
```

Ο χρωματικός κώδικας rgba σε σχέση με τον rgb διαθέτει ακόμη μία τιμή-παράμετρο α για την ιδιότητα opacity(διαφάνεια). Η παράμετρος α παίρνει τιμές από 0.0 (πλήρως διαφανής) μέχρι την τιμή 1.0 (πλήρως αδιαφανής)

Παράδειγμα

```
<style>
   p {color:rgba(255, 0, 191,0.6);}
      </style>
```

Χρωματικός κώδικας hsl και hsla

Στο χρωματικό κώδικα hsl θα πρέπει να γράψουμε τιμές για τις τρεις παραμέτρους:

- h (hue) → χροιά με τιμές από 0-360
- s (saturation) → κορεσμός με τιμές από 0%-100%
- l (lightness) → φωτεινότητα με τιμές από 0%-100%

Παράδειγμα

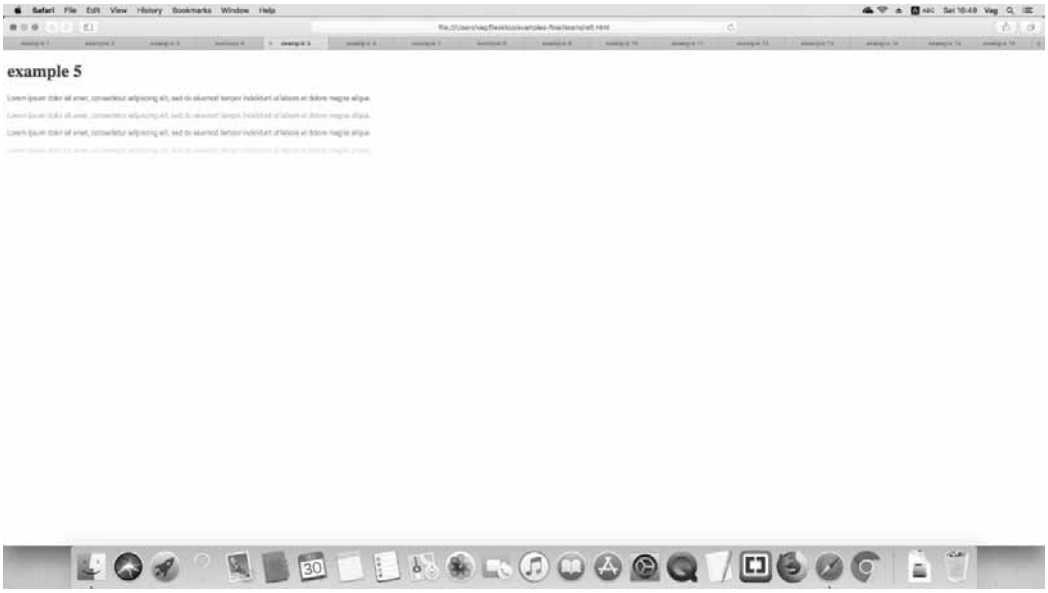
```
<style>
      p {color:  hsl(120, 100%, 75%);}
      </style>
```

Ο χρωματικός κώδικας hsla σε αντιστοιχία με τον rgba περιλαμβάνει ακόμη μία παράμετρο την παράμετρο της διαφάνειας. Η παράμετρος διαφάνειας παίρνει τιμές από 0.0 (πλήρως διαφανής) έως 1.0 (πλήρως αδιαφανής).

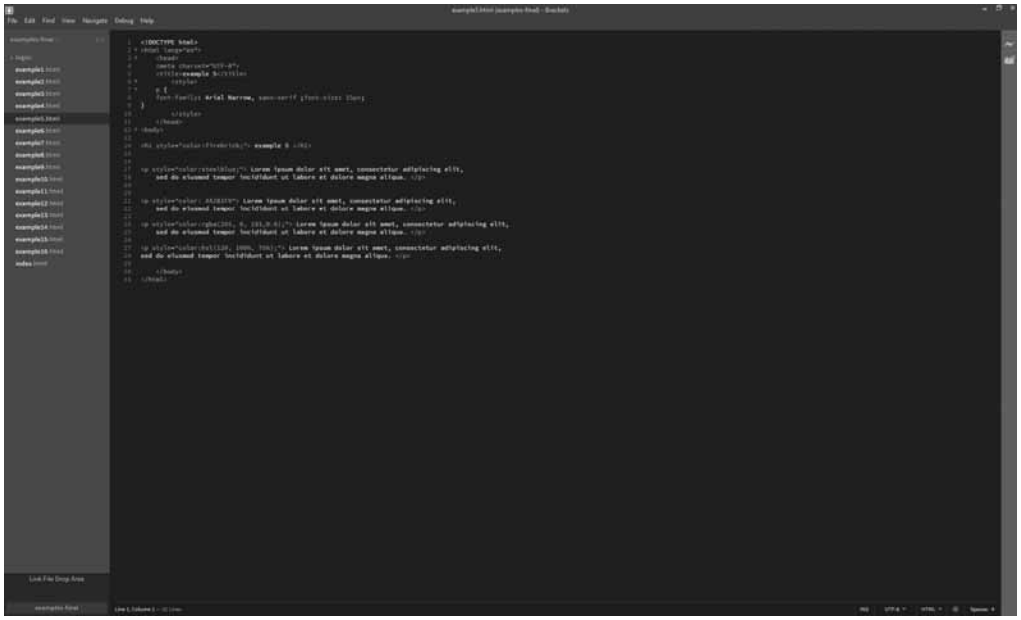
Παράδειγμα 5. Χρωματικοί κώδικες, font-family και font-size

Κώδικας(Mac): Χρωματικοί κώδικες, font-family και font-size

Εικόνα Browser(Mac): Χρωματικοί κώδικες, font-family και font-size



Κώδικας(Win): Χρωματικοί κώδικες, font-family και font-size



Εικόνα Browser(Win): Χρωματικοί κώδικες, font-family και font-size

ΥΠΟΕΝΟΤΗΤΑ 2. ΜΟΡΦΟΠΟΙΗΣΗ ΚΕΙΜΕΝΟΥ ΚΑΙ ΛΙΣΤΕΣ

2.1 Στοίχιση και μορφή κειμένου στην html

Για να στοιχίσουμε το κείμενο κατά την οριζόντια διεύθυνση στην Html θα χρειαστούμε την ιδιότητα text align με την παρακάτω σύνταξη:

text-align: τιμή;

Ως τιμές για την ιδιότητα text-align υπάρχουν οι επιλογές του πίνακα 5.

Πίνακας 5. Text Align

Τιμή text-align	Θέση οριζόντιας στοίχισης
left	Στοίχιση στα αριστερά
right	Στοίχιση στα δεξιά
center	Στοίχιση στο κέντρο
justify	Επιμήκυνση της γραμμής έτσι ώστε όλες οι γραμμές να έχουν το ίδιο πλάτος
initial	Θέτει την προεπιλεγμένη τιμή
inherit	Αντιγράφει τη στοίχιση του γονικού στοιχείου

Παράδειγμα

```
p {text-align: left;}
```

text-align-last

Η ιδιότητα text-align-last μας δίνει τη δυνατότητα να στοιχίσουμε στην οριζόντια διεύθυνση την τελευταία γραμμή της παραγράφου. Οι τιμές για την text-align-last είναι ίδιες με αυτές της ιδιότητας text-align.

Σημείωση

Ειδικά για το Mozilla στην ιδιότητα text-align-last θα γράψουμε και το πρόθεμα "-moz-":

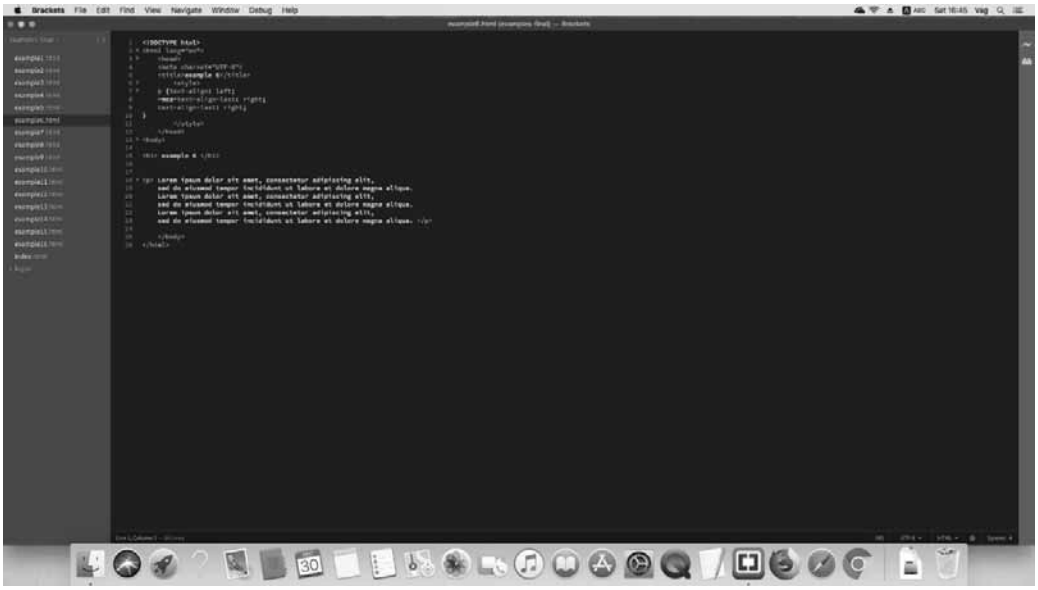
```
-moz-text-align: τιμή;
```

Παράδειγμα

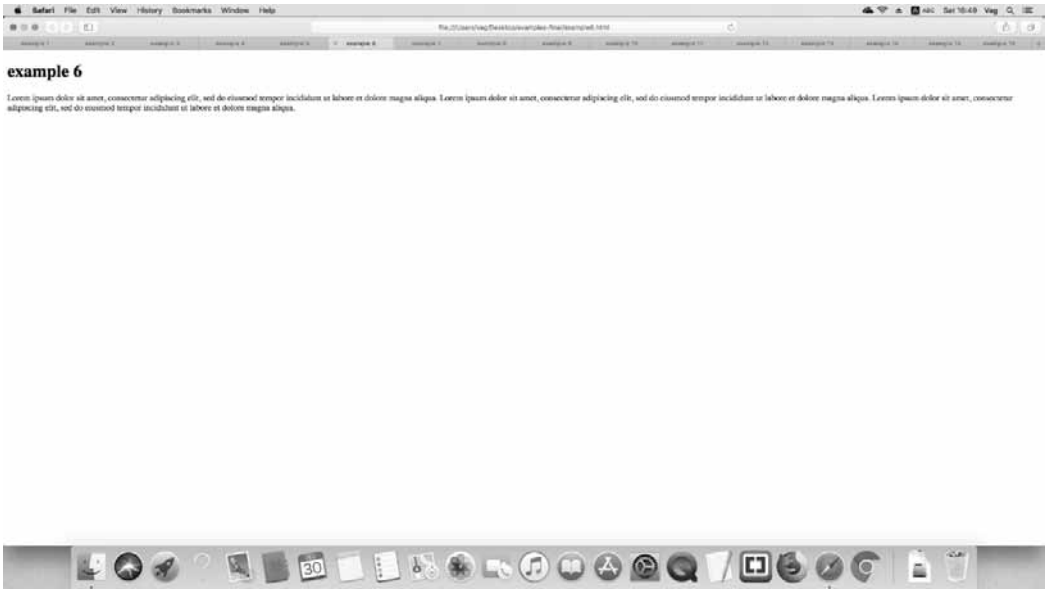
```
-moz-text-align-last: right;
```

Παράδειγμα 6. text-align και text-align-last

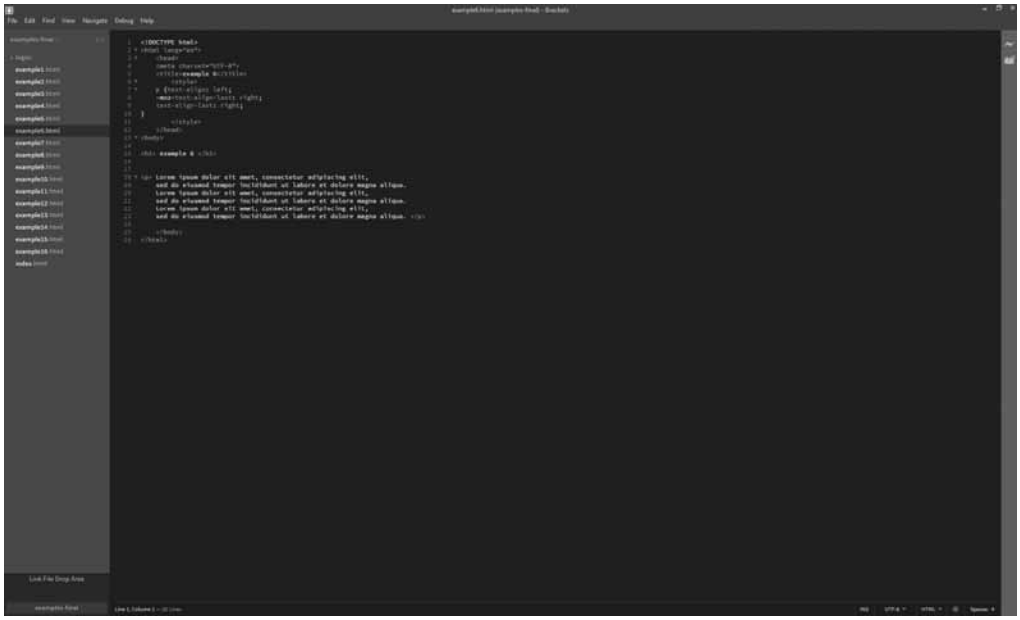
Κώδικας(Mac): text-align και text-align-last



Εικόνα Browser(Mac): text-align και text-align-last



Κώδικας(Win): text-align και text-align-last



Εικόνα Browser(Win): text-align και text-align-last

Σημείωση

Σε παρόμοιες περιπτώσεις που διαφέρει ο κώδικας σε κάποιο Browser γράφουμε κανονικά την ιδιότητα και προσθέσουμε και την ειδική σύνταξη για το συγκεκριμένο browser.

Παράδειγμα:

```
p {text-align: left;
   -moz-text-align-last: right;
   text-align-last: right;
}
```

Εμφάνιση κειμένου στον Browser

Ο Browser κατά την εκτέλεση του κώδικα σε σχέση με το κείμενο εντός του στοιχείου p:

- Δεν εμφανίζει πολλαπλά κενά στην ιστοσελίδα. Δηλαδή κατά τη συγγραφή κειμένου στον Html editor σε στοιχείο p δεν μπορούμε να αλλάξουμε τη δομή του κειμένου προσθέτοντας κενά, διότι ο browser τα παραλείπει και συνενώνει όλα τα συνεχόμενα κενά σε ένα.
- Ο Browser δε διαβάζει αλλαγές γραμμών όπως αυτές εμφανίζονται στο κείμενο που έχουμε γράψει στον Html editor.

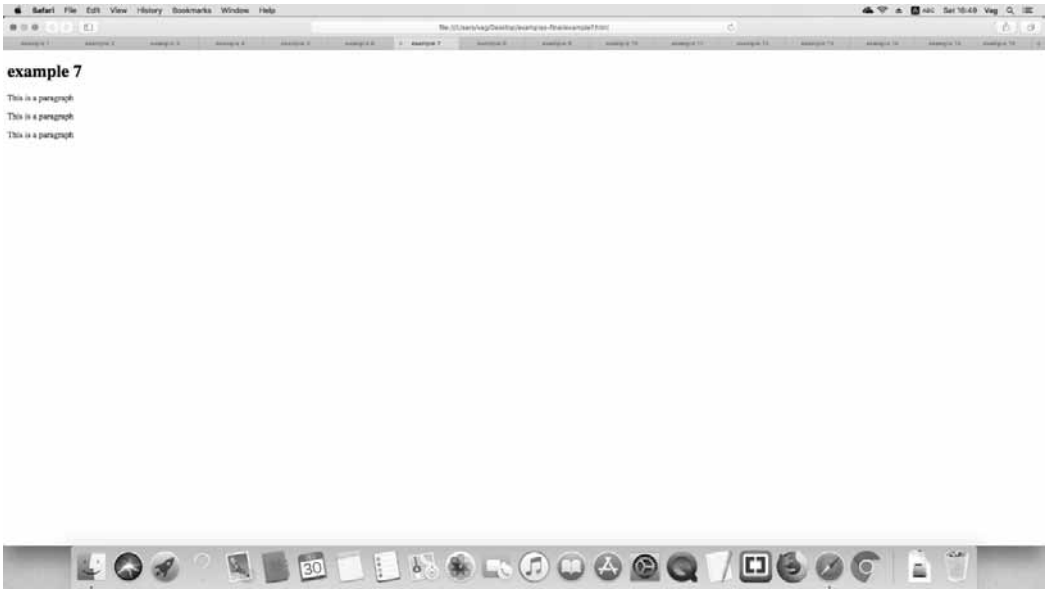
Για παράδειγμα στις επόμενες παραγράφους η εμφάνιση του κειμένου θα είναι η ίδια.

```
<p>This is a paragraph</p>
<!-- Χρησιμοποιώντας συνεχόμενα κενά -->
<p>This     is     a     paragraph     </p>
  <!-- Χρησιμοποιώντας αλλαγές γραμμών -->
<p>This
  is
  a
  paragraph</p>
```

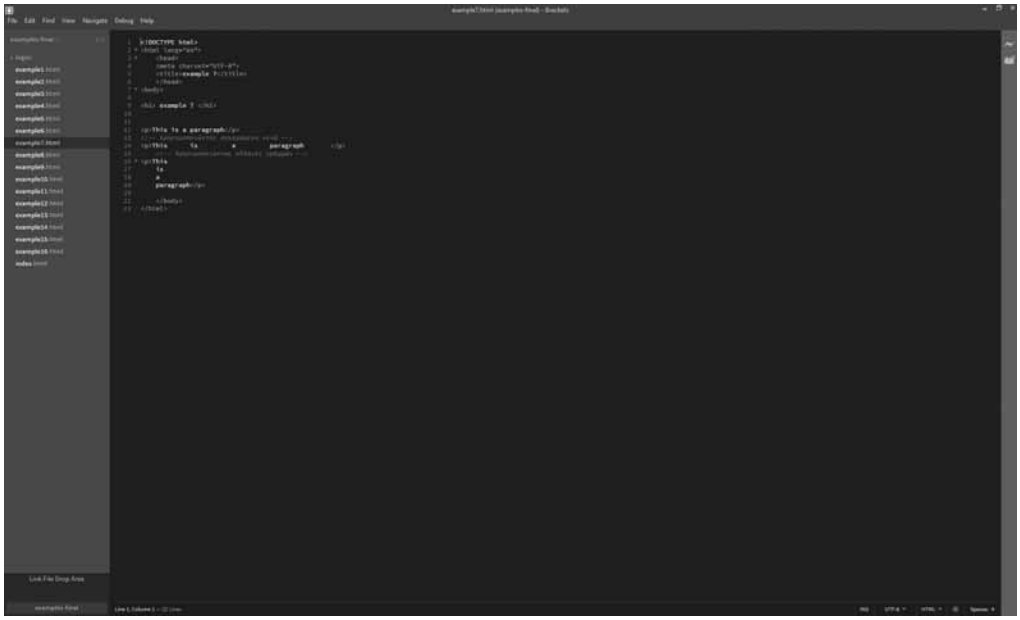
Παράδειγμα 7. Εμφάνιση κειμένου στον Browser

Κώδικας(Mac): Εμφάνιση κειμένου στον Browser

Εικόνα Browser(Mac): Εμφάνιση κειμένου στον Browser



Κώδικας(Win): Εμφάνιση κειμένου στον Browser



Εικόνα Browser(Win): Εμφάνιση κειμένου στον Browser

Προμορφοποιημένο κείμενο -ετικέτα <pre>

Με την ετικέτα pre έχουμε τη δυνατότητα να εμφανίσουμε στην ιστοσελίδα μας κείμενο όπως ακριβώς το έχουμε γράψει και στον κώδικα. Δηλαδή το κείμενο στην οθόνη του browser θα έχει τις ακολουθίες κενών χαρακτήρων και τις αλλαγές γραμμών που υπάρχουν και στον κώδικα.

Σύνταξη

```
<pre> ...  </pre>
```

Παράδειγμα

```
<pre> σ υ ν ε χ     όμενα κε       να   αλλαγή γραμμής
            Κ   ε     ν   ά     αλλαγή γραμμής
</pre>
```

White space

Με τη ιδιότητα white-space μπορούμε και χειριζόμαστε τον κενό χώρο ενός στοιχείου Html. Οι τιμές που συνοδεύουν την ιδιότητα white-space είναι οι παρακάτω:

normal: Ακολουθίες κενών χαρακτήρων εμφανίζονται ως ένας μόνο κενός χαρακτήρας. Υπάρχει αναδίπλωση κειμένου όταν αυτή είναι απαραίτητη.

nowrap: Ακολουθίες κενών χαρακτήρων εμφανίζονται ως ένας μόνο κενός χαρακτήρας. Δεν υποστηρίζεται αναδίπλωση κειμένου.

pre: Διατηρούνται οι ακολουθίες κενών χαρακτήρων. Γίνεται αναδίπλωση κειμένου μόνο στις αλλαγές γραμμών. Έχει τα ίδια αποτελέσματα με την ετικέτα <pre>.

pre-wrap: Διατηρούνται οι ακολουθίες κενών χαρακτήρων. Υποστηρίζεται η αναδίπλωση κειμένου όταν αυτή είναι απαραίτητη.

initial: Θέτει την ιδιότητα στην προεπιλεγμένη τιμή (default value).

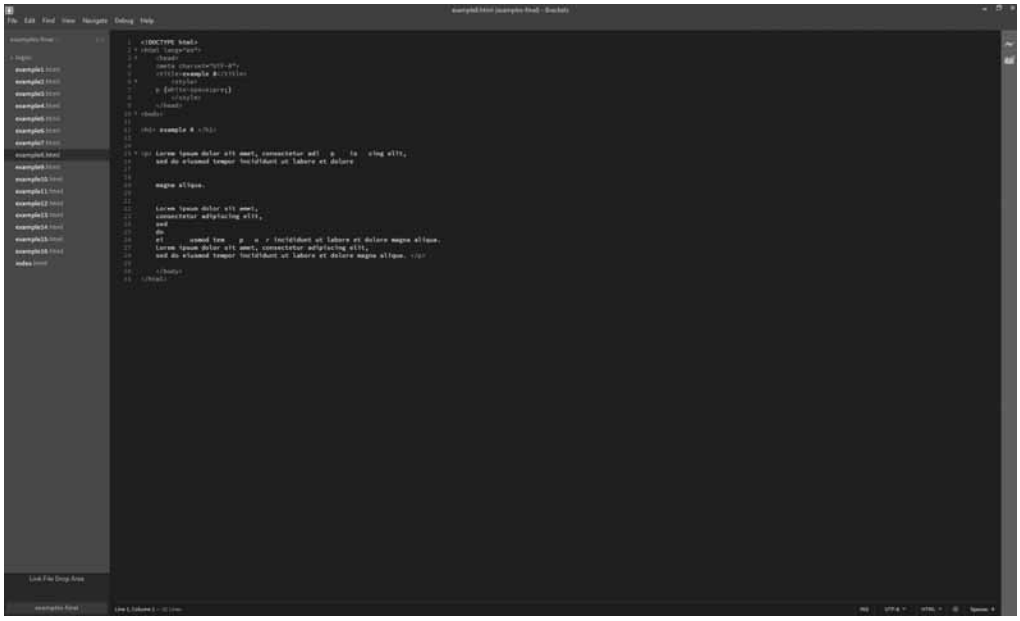
Inherit: Κληρονομεί την ιδιότητα από το πατρικό στοιχείο.

Παράδειγμα 8. white-space

Κώδικας(Mac): white-space

Εικόνα Browser(Mac): white-space

Κώδικας(Win): white-space



Εικόνα Browser(Win): white-space

Απόσταση μεταξύ των χαρακτήρων letter-spacing

Μπορούμε με την ιδιότητα CSS letter-spacing να ορίσουμε την οριζόντια απόσταση μεταξύ των χαρακτήρων με την παρακάτω σύνταξη:

letter-spacing: τιμή;

Στη θέση της τιμής μπορούμε να γράψουμε κάποια από τις επιλογές του πίνακα 6.

Πίνακας 6. Letter Spacing

Τιμή letter-spacing	Περιγραφή
normal	Είναι η προεπιλογή. Δεν αφήνει παραπάνω οριζόντιο διάστημα μεταξύ των χαρακτήρων.
length	Ορίζει επιπλέον οριζόντια απόσταση από την κανονική. Οι τιμές μπορεί να είναι σε μονάδες px, cm. Επιτρέπονται αρνητικές τιμές οι οποίες μειώνουν την οριζόντια απόσταση μεταξύ των χαρακτήρων. Σημείο αναφοράς των τιμών είναι η προεπιλεγμένη απόσταση.
initial	Θέτει την ιδιότητα στην προεπιλεγμένη τιμή
Inherit	Κληρονομεί την ιδιότητα από το πατρικό στοιχείο

Παράδειγμα

```
<p style="letter-spacing: 5px">
    Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
    </p>
```

Απόσταση μεταξύ των λέξεων (word-spacing)

Με την ιδιότητα word-spacing μπορούμε να ορίσουμε την απόσταση μεταξύ των λέξεων σύμφωνα με την παρακάτω σύνταξη:

```
word-spacing: τιμή;
```

Όπου τιμή μπορούμε να γράψουμε κάποια από τις επιλογές του πίνακα 7.

Πίνακας 7. Word Spacing

Τιμή word-spacing	Περιγραφή
normal	Είναι η προεπιλογή. Δεν αφήνει παραπάνω οριζόντιο διάστημα μεταξύ των λέξεων
length	Ορίζει επιπλέον οριζόντια απόσταση από την κανονική. Οι τιμές μπορεί να είναι σε μονάδες px, cm. Επιτρέπονται αρνητικές τιμές οι οποίες μειώνουν την οριζόντια απόσταση μεταξύ των λέξεων. Σημείο αναφοράς των τιμών είναι η προεπιλεγμένη απόσταση.
initial	Θέτει την ιδιότητα στην προεπιλεγμένη τιμή.
inherit	Κληρονομεί την ιδιότητα από το πατρικό στοιχείο.

Παράδειγμα

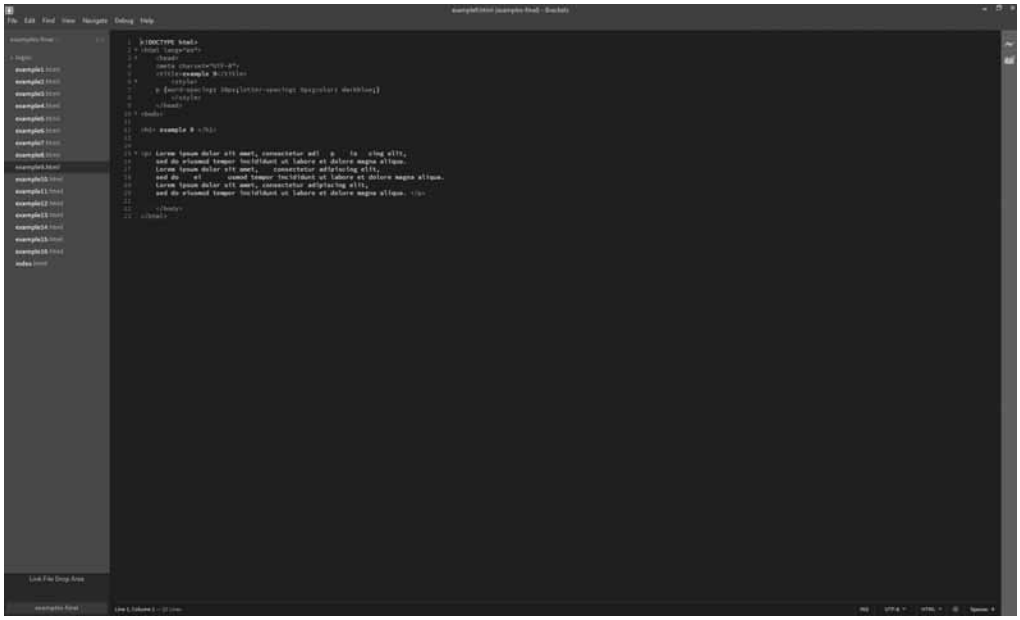
```
<p style="word-spacing: 30px">
    Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
    </p>
```

Παράδειγμα 9. Word και letter spacing

Κώδικας(Mac): word και letter spacing

Εικόνα Browser(Mac): word και letter spacing

Κώδικας(Win): word και letter spacing



Εικόνα Browser(Win): word και letter spacing

Δημιουργώντας εσοχές στις πρώτες γραμμές των παραγράφων (text-indent)

Με την ιδιότητα text-indent μπορούμε να δημιουργήσουμε εσοχή στην πρώτη γραμμή της παραγράφου με την ακόλουθη σύνταξη:

text-indent:τιμή; όπου τιμή θα επιλέξουμε μία από τις τιμές του πίνακα 8.

Πίνακας 8. Text indent

Τιμή text-indent	Περιγραφή
length	Απόλυτη τιμή σε px,cm, em
%	Ορίζει ποσοστιαία αναλογία σε σχέση με το πλάτος του πατρικού στοιχείου
initial	Θέτει την προεπιλεγμένη τιμή(default value)
inherit	Κληρονομεί την ιδιότητα από το parent element

Παράδειγμα

```
<style>
  p {text-indent: 25px;}
</style>
```

2.2 Text Formatting

Η Html5 διαθέτει τις γνωστές μορφοποιήσεις κειμένου όπως και οι κειμενογράφοι τύπου Microsoft Word. Στη συνέχεια θα παρουσιαστούν οι δυνατότητες μορφοποιήσεις κειμένου.

Έντονη γραφή-Ιδιότητα font-weight

Με την ιδιότητα font-weight μπορούμε να καθορίσουμε το πάχος των χαρακτήρων και κατά συνέπεια το πόσο έντονη θα εμφανίζεται η γραμματοσειρά στο browser.

Η σύνταξη της ιδιότητας font-weight είναι η παρακάτω:

```
font-weight:τιμή;
```

Πίνακας 9. Font Weight

Τιμή font-weight	Περιγραφή
normal	Είναι η προεπιλογή.
bold	Ορίζει έντονους χαρακτήρες
bolder	Ορίζει πολύ έντονους χαρακτήρες
lighter	Ορίζει λεπτούς χαρακτήρες
Διακριτές τιμές: 100,200,300,400,500,600,700,800,900	Ορίζει διαβαθμίσεις έντονης γραφής. Η τιμή 400 αντιστοιχεί στο normal και η τιμή 700 αντιστοιχεί στο bold. Οι επιτρεπτές τιμές είναι μόνο αυτές. Δεν μπορούμε να γράψουμε για παράδειγμα τιμή 150.
initial	Θέτει την προεπιλεγμένη τιμή(default value)
inherit	Κληρονομεί την ιδιότητα από το parent element

Παράδειγμα

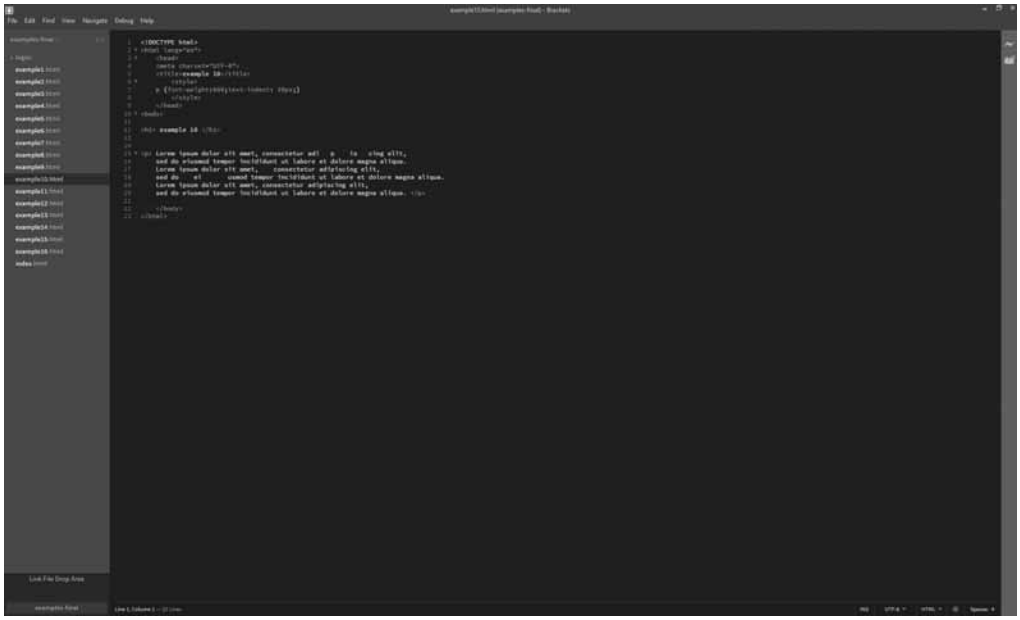
```
<style>
  p {font-weight:600;}
      </style>
```

Παράδειγμα 10. font-weight και text-indent

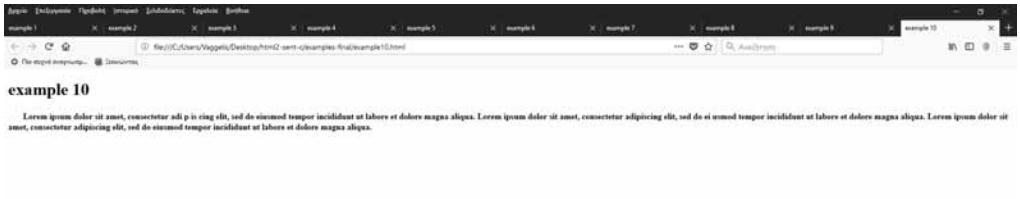
Κώδικας(Mac): font-weight και text-indent

Εικόνα Browser(Mac): font-weight και text-indent

Κώδικας(Win): font-weight και text-indent



Εικόνα Browser(Win): font-weight και text-indent



Bold Text

Για να εμφανίζουμε μέρος του κειμένου με έντονους χαρακτήρες, περικλείουμε το κείμενο μεταξύ των ετικετών και

Παράδειγμα

```
<b> bold text</b>
```

Σημαντικό Κείμενο

Για να ορίσουμε ένα σημαντικό κείμενο το περικλείουμε μεταξύ των ετικετών και .

Παράδειγμα

```
<strong> strong text </strong>
```

Κείμενο με πλαγιαστή γραμματοσειρά

Για να γράψουμε κείμενο με πλαγιαστή γραμματοσειρά το περικλείουμε μεταξύ των ετικετών και

Παράδειγμα

```
<i>  italic text</i>
```

Κείμενο με έμφαση

Για να γράψουμε κείμενο με έμφαση, χρησιμοποιούμε τις ετικέτες και .

Παράδειγμα

```
<em>em text </em>
```

Σημείωση

Οι browsers εμφανίζουν στις ιστοσελίδες τις ετικέτες ως <b> και ως , ωστόσο υπάρχει διαφορά στη σημασία των ετικετών. Ενώ οι ετικέτες <b> και εμφανίζουν bold και italic κείμενο, οι ετικέτες και δίνουν τη σημασία του σημαντικού κειμένου.

Κείμενο με μικρότερη Γραμματοσειρά

Για να γράψουμε μέρος του κειμένου με μικρότερη γραμματοσειρά χρησιμοποιούμε τις ετικέτες <small> και </small>

Παράδειγμα

```
<small> small text </small>
```

Highlighted text

Αν επιθυμούμε να εμφανίσουμε highlighted text όπως κάνουμε με ένα highlight μαρκαδόρο σε βιβλία και σημειώσεις τότε περικλείουμε λέξεις ή και φράσεις μεταξύ των ετικετών <mark> και </mark>. Οι χαρακτήρες μεταξύ των

ετικετών <mark> και </mark> θα εμφανιστούν με κίτρινο background χρώμα το οποίο είναι και η προεπιλογή για τις ετικέτες <mark> και </mark>. Αν επιθυμούμε μπορούμε να αλλάξουμε το χρώμα του μαρκαδόρου με την ιδιότητα background-color καθώς επίσης και χρώμα της γραμματοσειράς του highlighted text με την ιδιότητα color.

```
<style>
mark {color:blue;background-color: fuchsia;}
</style>
```

Παράδειγμα

```
<mark> this is highlighted </mark>
```

Διαγραμμένο κείμενο

Για να εμφανίσουμε κείμενο που έχει διαγραφεί περικλείουμε το κείμενο μεταξύ των ετικετών <del> και </del>. Το κείμενο θα εμφανιστεί με μια οριζόντια γραμμή στο μισό του ύψους της γραμματοσειράς.

Παράδειγμα

```
<del> this is deleted text </del>
```

Κείμενο που έχει προστεθεί

Για να ορίσουμε κείμενο που έχει προστεθεί το περικλείουμε μεταξύ των ετικετών <ins> και </ins>. Στη σελίδα το κείμενο θα εμφανιστεί ως υπογραμμισμένο δηλαδή με μια γραμμή στο κάτω μέρος της γραμματοσειράς.

Παράδειγμα

```
<ins> this is inserted text </ins>
```

Αναφορά πηγής

Μπορούμε να γράψουμε την πηγή κάποιου αρχείου, εγγράφου κτλ. ανάμεσα στις ετικέτες <cite> και </cite>

Παράδειγμα

```
<cite>πηγή εικόνας:... </cite>
```

Δείκτες και εκθέτες

Για να εμφανίσουμε κείμενο στη θέση εκθέτη, δηλαδή στην επάνω δεξιά θέση του χαρακτήρα το περικλείουμε μεταξύ των ετικετών ^{και}

Παράδειγμα

X ⁵

Για να εμφανίσουμε κείμενο στη θέση δείκτη περικλείουμε το κείμενο μεταξύ των ετικετών _{και}.

Παράδειγμα

X₃

Δείγμα κώδικα (<code>)

Ανάμεσα στις ετικέτες <code> και </code> περικλείουμε δείγμα κώδικα Γλώσσας Προγραμματισμού. Το κείμενο του κώδικα θα εμφανιστεί με γραμματοσειρά σταθερού πλάτους.

Παράδειγμα

<code>var x = 0; </code>

Δείγμα κειμένου (<samp>)

Για να γράψουμε δείγμα κειμένου χρησιμοποιούμε τις ετικέτες <samp> και </samp>. Η εμφάνιση του δείγματος κειμένου στον Browser είναι ίδια με αυτή της ετικέτας <code>.

Παράδειγμα

<samp> Δείγμα κειμένου</samp>

Keyboard input (<kbd>)

Το κείμενο ανάμεσα στις ετικέτες <kbd> και </kbd> υποδηλώνει κείμενο που θα πληκτρολογηθεί από το χρήστη

Παράδειγμα

<kbd> keyboard input</kbd>

Μεταβλητή (<var>)

Για να δηλώσουμε το όνομα μιας μεταβλητής χρησιμοποιούμε τις ετικέτες <var> και </var>.

Παράδειγμα

Μεταβλητή <var>y</var>

Όρος ορισμού (<dfn>)

Για να δηλώσουμε τον όρο ενός ορισμού χρησιμοποιούμε τις ετικέτες <dfn> και </dfn>.

Παράδειγμα

Η γλώσσα <dfn> Html5</dfn> εκτελείται από τους browsers

Συντομογραφία

Για να συμπεριλάβουμε στο κείμενό μας μια συντομογραφία, τότε περικλείουμε τη συντομογραφία ανάμεσα στις ετικέτες <abbr> και </abbr>.

Παράδειγμα

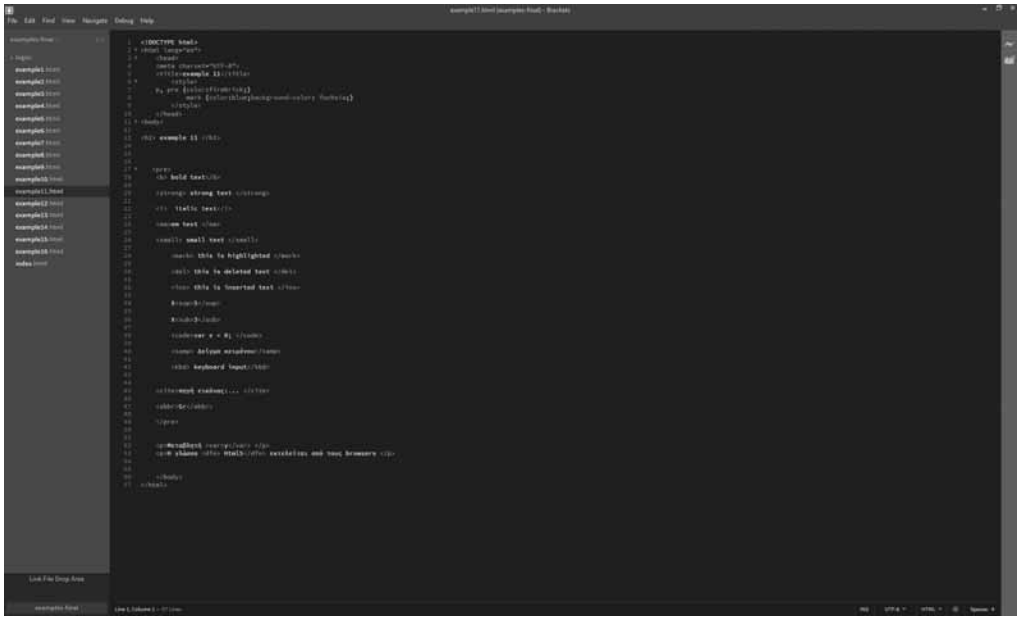
Συντομογραφία για Ελλάδα <abbr>Gr</abbr>

Παράδειγμα 11. Ειδικές ετικέτες κειμένου

Κώδικας(Mac): ειδικές ετικέτες κειμένου

Εικόνα Browser(Mac): ειδικές ετικέτες κειμένου

Κώδικας(Win): ειδικές ετικέτες κειμένου



Εικόνα Browser(Win): ειδικές ετικέτες κειμένου

Σημείωση

Όταν γενικά επιθυμούμε να γράψουμε κείμενο γράφουμε το κείμενο ανάμεσα στις ετικέτες και . Όταν υπάρχει ετικέτα που προσδιορίζει με πιο εξειδικευμένο τρόπο το κείμενο τότε χρησιμοποιούμε τη συγκεκριμένη ετικέτα και όχι την ετικέτα .

Για παράδειγμα την πηγή μιας πληροφορίας εικόνας κτλ. δεν τη γράφουμε μεταξύ των ετικετών και αλλά μεταξύ των ετικετών <cite> και </cite>.

Επίσης ένα στοιχείο p μπορεί να περιλαμβάνει και στοιχεία κειμένου όπως dfn, var κτλ.

Λάθος → πηγή: …

Σωστό → <cite> πηγή: … </cite>

Παράθεση φράσεων και αυτούσιου αποσπάσματος κειμένου με q και blockquote και text-area

Στοιχείο q

Στην περίπτωση που επιθυμούμε να παραθέσουμε μια φράση από ένα κείμενο ή ένα απόφθεγμα τότε χρησιμοποιούμε το στοιχείο q. Συνήθως οι Browsers Προσθέτουν εισαγωγικά στη φράση που περιέχεται στο στοιχείο q.

<q> φράση… </q>

Παράδειγμα

```
<p> Νίκος Καζαντζάκης: <q> Έχεις τα πινέλα, έχεις τα χρώματα, ζωγράφισε τον
Παράδεισο και μπες μέσα </q> </p>
```

Στοιχείο blockquote

Όταν επιθυμούμε να παραθέσουμε αυτούσιο απόσπασμα κειμένου χρησιμοποιούμε το στοιχείο blockquote.

Στην ετικέτα έναρξης του blockquote μπορούμε(προαιρετικά) να συμπεριλάβουμε και την πηγή με το στοιχείο cite. Πιο συγκεκριμένα στην ετικέτα έναρξης του Blockquote γράφουμε:

```
cite="…"
```

και μεταξύ των διπλών εισαγωγικών γράφουμε την πηγή.

```
<blockquote cite="… ">

…

</blockquote>
```

Παράδειγμα

```
<blockquote cite="https://www.forbes.com/quotes/185/">

"The most beautiful thing we can experience is the mysterious. It is the source of
all true art and all science.

 He to whom this emotion is a stranger, who can no longer pause to wonder and
stand rapt in awe,  is as good as dead: his eyes are closed."

   Albert Einstein

</blockquote>
```

Text area

Μπορούμε να δημιουργήσουμε μια περιοχή κειμένου με το στοιχείο text-area

Μεταξύ των ετικετών <textarea> και </textarea> θα γράψουμε το περιεχόμενο της περιοχής κειμένου.

Στην ετικέτα έναρξης <textarea> μπορούμε να ορίσουμε και τις διαστάσεις της περιοχής με τις ιδιότητες rows και cols προσδιορίζοντας τον αριθμό των γραμμών και των στηλών που θα καταλαμβάνει το στοιχείο textarea.

```
<textarea rows="…" cols="…">

…

</textarea>
```

Παράδειγμα

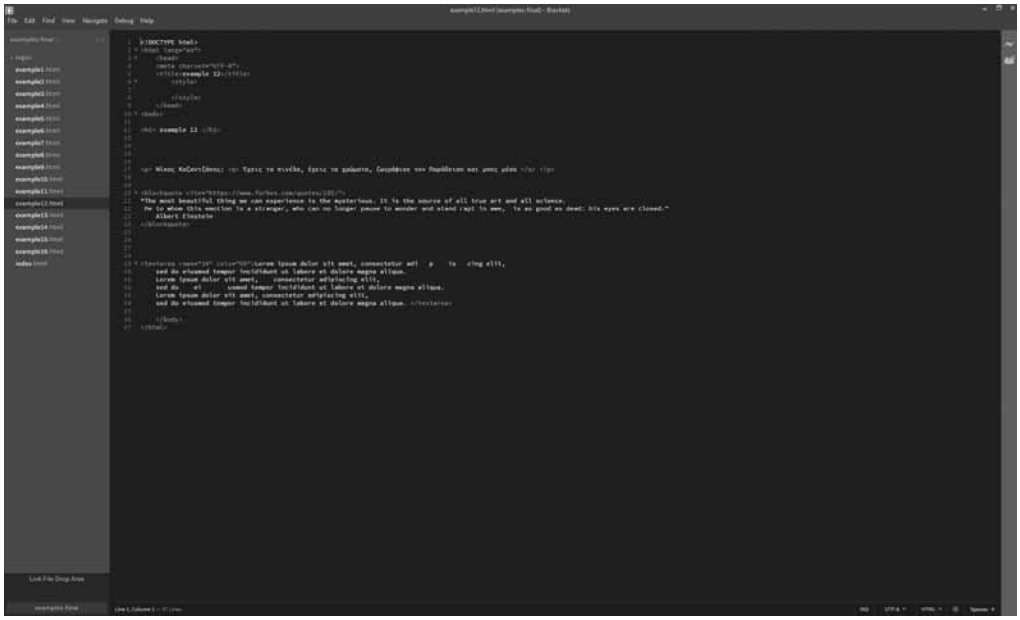
```
<textarea rows="10" cols="50">Lorem ipsum dolor sit amet, consectetur adi    p
is   cing elit,

   sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

   Lorem ipsum dolor sit amet,    consectetur adipiscing elit,

   sed do    ei       usmod tempor incididunt ut labore et dolore magna aliqua.

   Lorem ipsum dolor sit amet, consectetur adipiscing elit,

   sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

</textarea>
```

Παράδειγμα 12. Στοιχεία q, blockquote και Text Area

Κώδικας(Mac): στοιχεία q, blockquote και Text Area

Εικόνα Browser(Mac): στοιχεία q, blockquote και Text Area

Κώδικας(Win): στοιχεία q, blockquote και Text Area



Εικόνα Browser(Win): στοιχεία q, blockquote και Text Area

2.3 Διατεταγμένες λίστες, μη διατεταγμένες λίστες και λίστες ορισμών

Η οργάνωση κειμένου σε διατεταγμένες και μη διατεταγμένες λίστες σε μία ιστοσελίδα είναι πολύ χρήσιμη και διευκολύνει τον αναγνώστη-χρήστη να κατανοήσει το περιεχόμενο της ιστοσελίδας.

Στη διατεταγμένη λίστα η σειρά με την οποία θα γράφουμε τους σημειωτές-Bullets δεν παίζει κάποιο ρόλο, ενώ στις διατεταγμένες λίστες η σειρά είναι απαραίτητη.

Έστω για παράδειγμα ότι διαβάζουμε μια συνταγή για να παρασκευάσουμε ένα κέικ. Η σειρά με την οποία θα είναι γραμμένα τα υλικά δεν έχει σημασία (χρήση μη διατεταγμένης λίστας).

Υλικά για κέικ

- Γάλα 250ml
- Βούτυρο 100 g
- Αλεύρι 400 g

Όταν όμως έχουμε συγκεντρώσει τα υλικά και είμαστε έτοιμοι για την εκτέλεση της συνταγής τότε η σειρά έχει μεγάλη σημασία (χρήση διατεταγμένης λίστας).

Εκτέλεση συνταγής

1. Ρίχνουμε το γάλα στο μπολ

2. Ρίχνουμε το αλεύρι
3. Ρίχνουμε το βούτυρο

Μη διατεταγμένη λίστα unordered list (<ul>)

Για να γράψουμε μία διατεταγμένη λίστα τότε περικλείουμε το περιεχόμενό της μεταξύ των ετικετών <ul> και </ul>.

```
<ul>
…
…
</ul>
```

Κάθε στοιχείο της λίστας (list item) περικλείεται μεταξύ των ετικετών <li> και </li>, όποτε η δομή της μη διατεταγμένης λίστας είναι η παρακάτω:

```
<ul>
<li>…</li>
<li>…</li>
<li>…</li>
…
</ul>
```

Σημείωση

Το κάθε li στοιχείο της λίστας μπορεί να περιέχει άλλη λίστα, εικόνες Links κα.

Για την επιλογή των σημειωτών μπορούμε εντός της ετικέτας της λίστας να γράψουμε:

```
<ul style="list-style-type:τιμή;">
```

Όπου τιμή μία από τις παρακάτω επιλογές:

- disk → κυκλικός δίσκος (προεπιλογή)
- circle → περιφέρεια κύκλου (χωρίς γέμισμα)
- square → τετράγωνα
- none → χωρίς σημειωτές

Παράδειγμα

```
<ul style="list-style-type:circle;">
<li>Γάλα 250ml</li>
<li>Βούτυρο 100 g</li>
<li>Αλεύρι 400 g</li>
</ul>
```

Διατεταγμένη λίστα (ordered list)

Για να δημιουργήσουμε μια διατεταγμένη λίστα περικλείουμε το περιεχόμενό της μεταξύ των ετικετών <ol> και </ol>

```
<ol>
…
…
</ol>
```

Το κάθε στοιχείο της λίστας το γράφουμε μεταξύ των ετικετών <li> και </li> όπως και στις μη διατεταγμένες λίστες.

```
<ol>
<li>…</li>
<li>…</li>
<li>…</li>
…
</ol>
```

Για να προσδιορίσουμε τον τρόπο της αρίθμησης των διατεταγμένων λιστών μπορούμε να γράψουμε στην ετικέτα έναρξης της διατεταγμένης λίστας:

```
<ol  type="τιμή">
```

όπου τιμή για την ιδιότητα type συμπληρώνουμε μία από τις παρακάτω:

1 → σύστημα αρίθμησης με αριθμούς από το δεκαδικό σύστημα 1,2,3,…

A → σύστημα αρίθμησης με κεφαλαία γράμματα

a → σύστημα αρίθμησης με πεζά γράμματα

I → σύστημα αρίθμησης με κεφαλαίους Λατινικούς αριθμούς

i → σύστημα αρίθμησης με πεζούς Λατινικούς αριθμούς

Παράδειγμα

```
  <ol type="I">
 <li>Ρίχνουμε το γάλα στο μπολ </li>
 <li>Ρίχνουμε το αλεύρι </li>
 <li>Ρίχνουμε το βούτυρο</li>
    </ol>
```

Εμφωλευμένες λίστες

Μπορούμε να δημιουργήσουμε εύκολα εμφωλευμένες λίστες αν προσθέσουμε απλά μία καινούργια λίστα σε ένα li στοιχείο. Δηλαδή το στοιχείο li της αρχικής λίστας θα παραστάνει μία ένθετη λίστα.

Για να δημιουργήσουμε μία ένθετη λίστα σε διατεταγμένες και μη διατεταγμένες λίστες εντός του στοιχείου li που επιθυμούμε γράφουμε τον κώδικα για την ένθετη λίστα.

Παράδειγμα

```
<h5>Εμφωλευμένη  Λίστα ul</h5>
```

Αρχικά θα δημιουργήσουμε τη ul list και στη συνέχεια θα προσθέσουμε τα στοιχεία της.

```
  <ul>
<li>Γάλα 250ml</li>
<li>Βούτυρο 100 g</li>
<li>Αλεύρι 400 g</li>
```

Στη συνέχεια έστω ότι επιθυμούμε να προσθέσουμε μια εμφωλευμένη λίστα. Για να το πετύχουμε αυτό θα γράψουμε αρχικά την ετικέτα έναρξης **<li>**, θα προσθέσουμε προαιρετικά κείμενο και στη συνέχεια θα προσθέσουμε μια νέα λίστα.

```
<li> Γλυκαντικές ουσίες
        <ul>
         <li>ζάχαρη 150 g</li>
         <li> ζάχαρη άχνη 50 g</li>
         <li> ζάχαρη μαύρη 30 g</li>
        </ul>
```

Αφού ολοκληρώσουμε την εμφωλευμένη λίστα θα γράψουμε την ετικέτα λήξης του στοιχείου li **<li>** και θα συνεχίσουμε γράφοντας το υπόλοιπο μέρος της αρχικής λίστας.

```
         </li>
  <li>Μαγιά 10 g</li>
 </ul>
```

Σημείωση

Την ίδια μεθοδολογία χρησιμοποιούμε και για εμφωλευμένες διατεταγμένες λίστες.

Λίστες ορισμών (<dl>)

Εκτός από τις διατεταγμένες και μη διατεταγμένες λίστες μπορούμε να δημιουργήσουμε και λίστες ορισμών. Οι λίστες ορισμών περιλαμβάνουν στοιχεία τα οποία είναι ορισμοί. Ο κάθε ορισμός περιλαμβάνει δύο τμήματα:

1. Τον όρο του ορισμού
2. Την επεξήγηση του ορισμού.

```
Όρος1
Επεξήγηση1
Όρος2
Επεξήγηση2
…
```

Για να δημιουργήσουμε μία λίστα ορισμών τότε θα πρέπει το περιεχόμενο της συγκεκριμένης λίστας να περιλαμβάνεται μεταξύ των ετικετών <dl> και </dl> όπως ακολούθως:

```
<dl>
…
…
…
</dl>
```

Για κάθε στοιχείο-ορισμό της λίστας θα γράφουμε:

1. Τον όρο του ορισμού ανάμεσα στις ετικέτες <dt> και </dt>
2. Την επεξήγηση του ορισμού ανάμεσα στις ετικέτες <dd> και </dd>

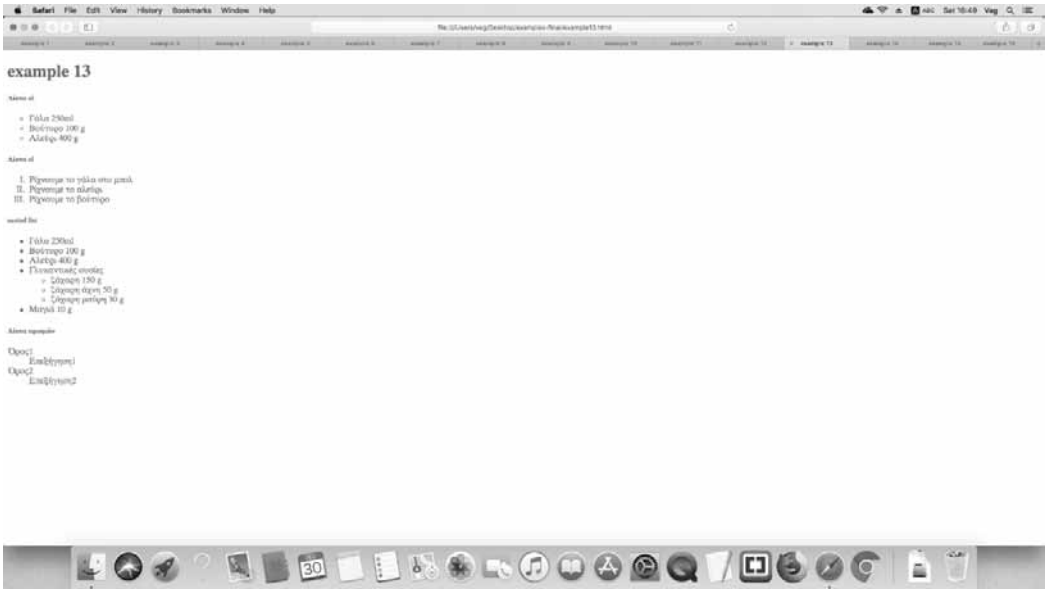
Παράδειγμα

```
<dl>
<dt> Όρος1</dt>
<dd> Επεξήγηση1</dd>
<dt> Όρος2</dt>
<dd> Επεξήγηση2</dd>
</dl>
```

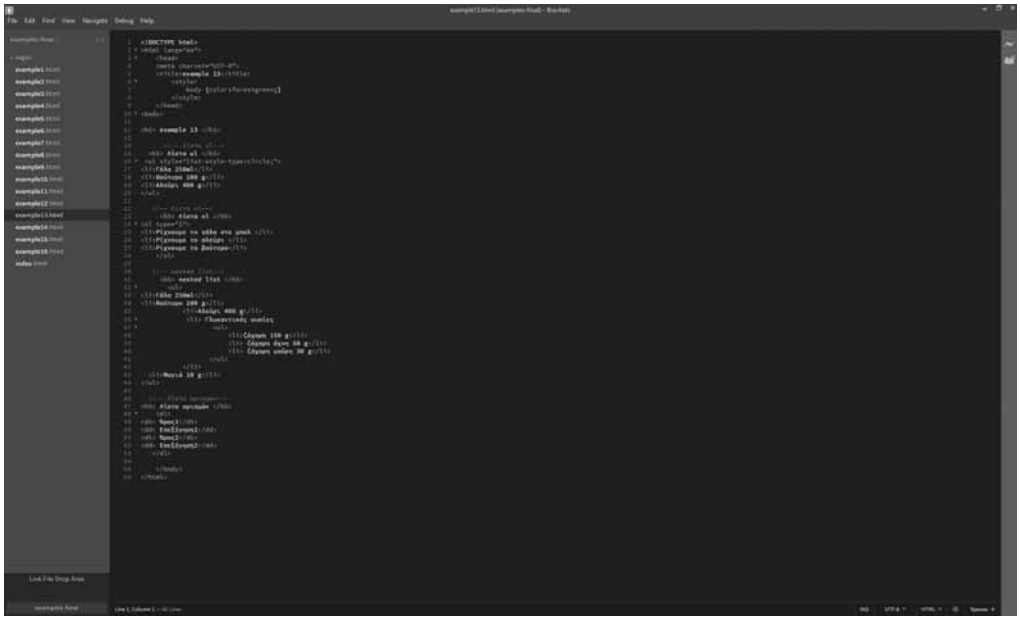
Παράδειγμα 13. Λίστες

Κώδικας(Mac): λίστες

Εικόνα Browser(Mac): λίστες



Κώδικας(Win): λίστες



Εικόνα Browser(Win): λίστες

ΥΠΟΕΝΟΤΗΤΑ 3. ΠΙΝΑΚΕΣ ΚΑΙ FORMS

3.1 Πίνακες στην Html

Με τη βοήθεια του στοιχείου table μπορούμε να δημιουργήσουμε πίνακες στην ιστοσελίδα μας. Ολόκληρο το περιεχόμενο του πίνακα περιλαμβάνεται μεταξύ των ετικετών <table> και </table>. Σε κάθε πίνακα μπορούμε προαιρετικά να γράψουμε λεζάντα. Για να κατανοήσουμε καλύτερα τη συγγραφή του κώδικα για ένα πίνακα είναι σημαντικό να ξέρουμε ότι γράφουμε τον πίνακα γραμμή προς γραμμή και με κατεύθυνση από επάνω προς τα κάτω. Δηλαδή πρώτα θα γράψουμε προαιρετικά τη λεζάντα που αντιστοιχεί στην πρώτη γραμμή. Στη συνέχεια θα γράψουμε προαιρετικά τη γραμμή που θα περιέχει τα πεδία δηλαδή τα ονόματα των στηλών. Για τις υπόλοιπες γραμμές που περιέχουν τις τιμές του πίνακα πάλι θα γράφουμε κώδικα ξεχωριστά για κάθε γραμμή.

```
<table>

...

...

...

</table>
```

Τα βήματα που πρέπει να ακολουθήσουμε για τη δημιουργία ενός πίνακα είναι τα παρακάτω:

1. Γράφουμε την ετικέτα έναρξης του πίνακα <table>
2. Προαιρετικά γράφουμε μία λεζάντα για τον πίνακα ανάμεσα στις ετικέτες <caption> και </caption>
3. Γράφουμε τον κώδικα για την γραμμή επικεφαλίδα του πίνακα που θα περιέχει τα ονόματα των στηλών του πίνακα. Για τη γραμμή επικεφαλίδα καθώς και για τις γραμμές με τις τιμές του πίνακα θα ξεκινάμε τον κώδικα με την ετικέτα <tr> και θα τελειώνουμε με την ετικέτα </tr>.

Οπότε και για τη γραμμή επικεφαλίδα θα ξεκινήσουμε με την ετικέτα <tr>. Στη συνέχεια το κάθε όνομα μιας στήλης θα το γράψουμε μεταξύ των ετικετών <th> και </th>. Αφού γράψουμε τα ονόματα των στηλών θα κλείσουμε με την ετικέτα </tr>.

Μέχρι εδώ η σύνταξη του πίνακα είναι η παρακάτω:

```
<table>
<caption> … </caption>
<tr>
<th>όνομα_στήλης_1   </th>
<th>όνομα_στήλης_2   </th>
<th>όνομα_στήλης_3   </th>
</tr>
…
```

4. Γράφουμε τις επόμενες γραμμές με τις τιμές του πίνακα. Η κάθε γραμμή θα αρχίζει και θα τελειώνει με τις ετικέτες <tr> και </tr>. Κάθε τιμή ενός κελιού του πίνακα θα περιλαμβάνεται μεταξύ των τιμών <td> και </td>.

```
<tr>
<td>τιμή_γραμμής1_στήλης_1   </td>
<td> τιμή_γραμμής1_στήλης_2   </td>
<td> τιμή_γραμμής1_στήλης_3   </td>
</tr>

<tr>
<td>τιμή_γραμμής2_στήλης_1   </td>
<td>όν τιμή_γραμμής2_στήλης_2   </td>
<td> τιμή_γραμμής2_στήλης_3   </td>
</tr>
```

5. Αφού γράψουμε και την τελευταία γραμμή του πίνακα οριοθετούμε το τέλος του πίνακα με την ετικέτα </table>

Σημείωση

Εντός ενός στοιχείου td μπορούμε να συμπεριλάβουμε ονόματα, τιμές, εικόνες, logos, και γενικά στοιχεία τα οποία είναι απαραίτητα για την εμφάνιση του πίνακα.

Παράδειγμα

Έστω ότι χρειαζόμαστε να εμφανίσουμε στην ιστοσελίδα μας τον παρακάτω πίνακα με στατιστικά από το NBA:

logo	Team Name	W	L	WIN%
	Toronto	23	8	.742
	Boston	27	10	.730
	Cleveland	24	10	.706
	Indiana	19	14	.576
	Detroit	18	14	.563
	Washington	19	15	.559
	Milwaukee	17	14	.548
	Miami	17	16	.515

Πηγή Στατιστικών: stats.nba.com

Πηγή logos: sportslogos.net

Υλοποίηση

Για να υλοποιήσουμε τον πίνακα με τις στατιστικές ομάδων του NBA θα πρέπει να ακολουθήσουμε τα παρακάτω βήματα:

1. Αρχικά εντός του στοιχείου head θα γράψουμε CSS style για τον πίνακα html.

```
<style>
```

Για τα στοιχεία table, th, td και caption θα θέσουμε περίγραμμα πλάτους 1 pixel και χρώματος συμπαγούς πράσινου.

```
  table, th, td, caption {
   border: 1px solid green;
}
```

Ειδικά για τα στοιχεία th(στοιχεία γραμμής επικεφαλίδας) και τα στοιχεία td που θα περιλαμβάνουν τις τιμές των στατιστικών στοιχείων έχουμε επιλέξει η στοίχιση του περιεχομένου να γίνεται στο κέντρο.

```
    th, td {text-align: center;}
  </style>
```

Στην ετικέτα έναρξης έχουμε γράψει width (πλάτος) του πίνακα 400 pixels και align center για να στοιχηθεί ο πίνακας στο κέντρο της σελίδας. Επίσης έχουμε γράψει λεζάντα για τον πίνακα με την ετικέτα <caption>

```
<table width="400" align="center">
   <caption> NBA statistics</caption>
```

2. Στην συνέχεια θα γράψουμε τη γραμμή επικεφαλίδα ξεκινώντας με την ετικέτα <tr> για να δηλώσουμε τη δημιουργία μιας καινούργια γραμμής. Το κάθε όνομα της στήλης Logo, W κτλ. θα γραφεί ανάμεσα στις ετικέτες <th> και </th>.

```
<tr>
<th> Logo  </th>
<th> W  </th>
<th> L   </th>
<th> WIN %   </th>
</tr>
```

3. Στη δεύτερη καθώς και στις υπόλοιπες γραμμές θα γράψουμε τις τιμές από τις επιδόσεις των ομάδων καθώς και τις εικόνες από τα logos. Έτσι κάθε γραμμή θα περικλείεται μεταξύ των ετικετών <tr> και </tr>, ενώ κάθε τιμή από τις επιδόσεις των ομάδων καθώς και οι εικόνες από τα logos θα περιλαμβάνονται μεταξύ των ετικετών <td> και </td>.

Σε κάθε γραμμή το πρώτο στοιχείο td αντιπροσωπεύει το logo της ομάδας της συγκεκριμένης γραμμής. Όλα τα logos των ομάδων είναι αποθηκευμένα στο subfolder logos. Έτσι λοιπόν για να εμφανιστεί το logo ανάμεσα στις ετικέτες <td> και </td> θα γράψουμε ένα στοιχείο img και θα δηλώσουμε τη διεύθυνση του subfolder logos μαζί με το όνομα του αρχείου για να μπορέσει ο browser να βρει το logo.

```
img src="logos/toronto.gif"
```

Επίσης θα γράψουμε:

```
alt="…"
```

ένα κείμενο που θα περιγράφει το περιεχόμενο της εικόνας στην περίπτωση που για οποιοδήποτε λόγο η εικόνα δε θα μπορεί να εμφανιστεί στη σελίδα μας.

```
alt="toronto logo"
```

Τέλος έχουμε δηλώσει για το κάθε αρχείο διατάσεις ύψος Χ πλάτος 35Χ40 pixels.

```
height="35" width="40"
```

Αναλυτικά ο κώδικας για τις γραμμές των τιμών είναι ο παρακάτω:

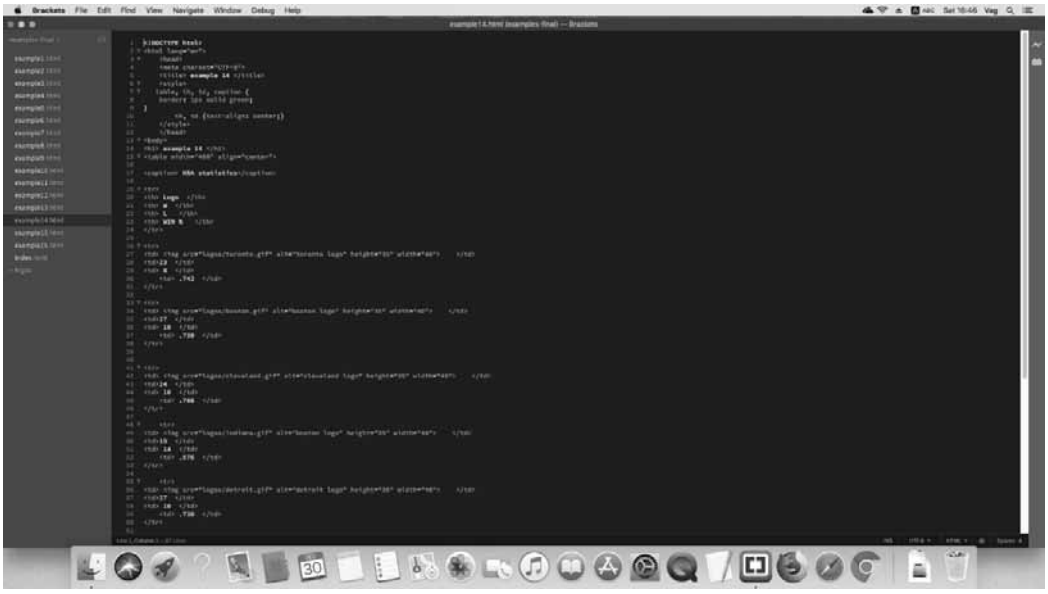
```
<tr>

<td> <img src="logos/toronto.gif" alt="toronto logo" height="35" width="40">
</td>

<td>23  </td>

<td> 8  </td>

   <td> .742  </td>

</tr>


<tr>

<td> <img src="logos/boston.gif" alt="boston logo" height="35" width="40">
</td>

<td>27  </td>

<td> 10  </td>

   <td> .730  </td>

</tr>



<tr>
```

```
<td> <img src="logos/cleveland.gif" alt="cleveland logo" height="35" width="40">
</td>

<td>24  </td>

<td> 10  </td>

   <td> .706  </td>

</tr>


   <tr>

<td> <img src="logos/indiana.gif" alt="boston logo" height="35" width="40">
</td>

<td>19  </td>

<td> 14  </td>

   <td> .576  </td>

</tr>


   <tr>

<td> <img src="logos/detroit.gif" alt="detroit logo" height="35" width="40">   </td>

<td>27  </td>

<td> 10  </td>

   <td> .730  </td>
```

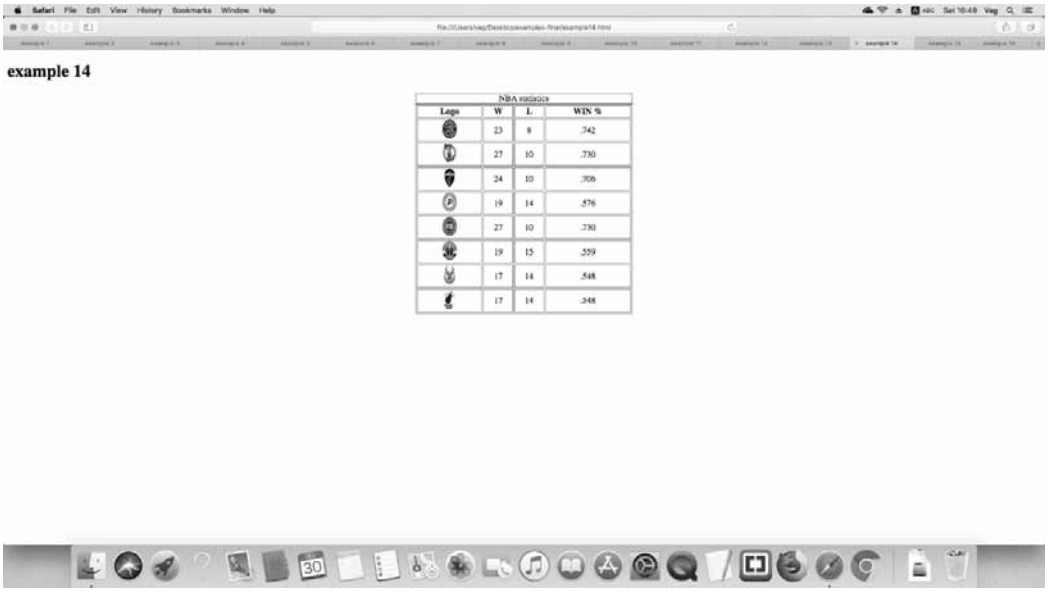
```
</tr>

  <tr>
<td> <img src="logos/washington.gif" alt="washington logo" height="35"
width="40">   </td>
<td>19 </td>
<td> 15  </td>
   <td> .559  </td>
</tr>

  <tr>
<td> <img src="logos/Milwaukee.gif" alt="washington logo" height="35"
width="40">   </td>
<td>17 </td>
<td> 14  </td>
   <td> .548  </td>
</tr>

     <tr>
<td> <img src="logos/miami.gif" alt="washington logo" height="35" width="40">
</td>
<td>17 </td>
<td> 14  </td>
   <td> .548  </td>
</tr>

   </table>
```

Παράδειγμα 14. HTML Πίνακας

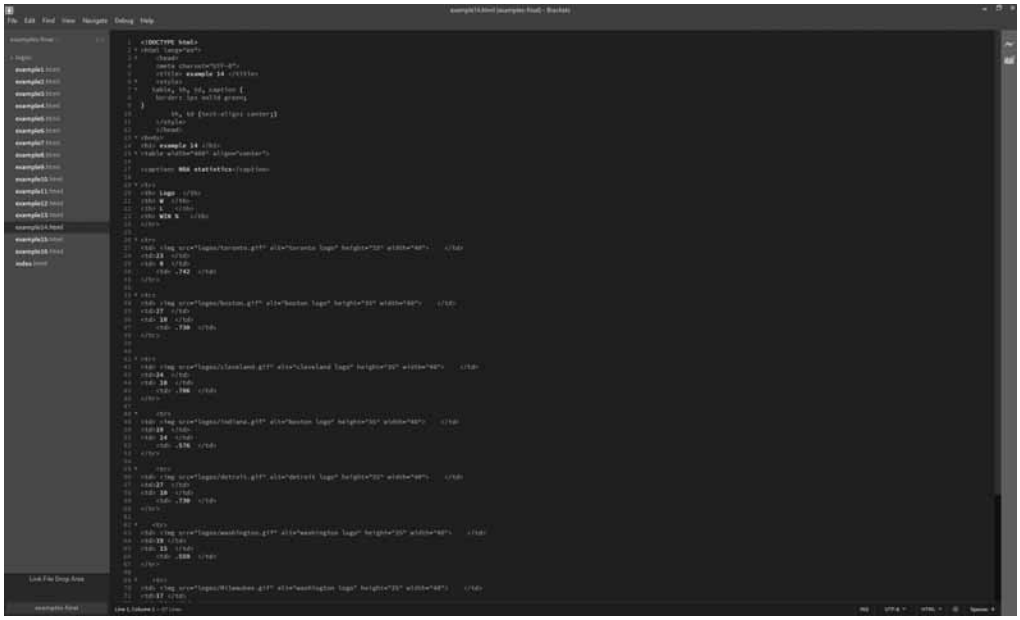
Κώδικας(Mac): Html Πίνακας



Εικόνα Browser(Mac): Html Πίνακας



Κώδικας(Win): Html Πίνακας



Εικόνα Browser(Win): Html Πίνακας

example 14

NBA statistics

Logo	W	L	WIN %
	23	8	.742
	27	10	.730
	24	10	.706
	19	14	.576
	27	10	.730
	19	15	.559
	17	14	.548
	17	14	.548

3.2 Html Forms

Το στοιχείο form της Html έχει ως σκοπό τη συλλογή δεδομένων από το χρήστη και περιλαμβάνεται μεταξύ των ετικετών <form> και </form>.

```
<form>
…
…
</form>
```

Για το κάθε input που θα περιλαμβάνεται σε ένα στοιχείο form μπορούμε να γράψουμε κείμενο περιγραφής και στη συνέχεια να προσθέσουμε μια ετικέτα <input> με ιδιότητες όπως type, name και value ως ακολούθως:

```
<form>
…
<input type="…" name="…" value="…" > <br>
…
</form>
```

Η τιμή name αναφέρεται στο όνομα του Input και η τιμή value θα εμφανιστεί εντός το πεδίου του Input ως ενδεικτική τιμή.

Για την ιδιότητα type υπάρχουν οι παρακάτω επιλογές:

1. **text** για εισαγωγή κειμένου
2. **email** για εισαγωγή email

Παράδειγμα

```
<h5> Text and email  type</h5>
 <form>
  First name:<br>
  <input type="text" name="first_name" value="John"><br>
  Last name:<br>
  <input type="text" name="last_name" value="Kalipolitis"> <br>
 email:<br>
  <input type="email" name="email" value="jkalipolitis@fastemail.com">
</form>
```

3. **password** για εισαγωγή κωδικού. Στο συγκεκριμένο Input δεν είναι ορατοί οι χαρακτήρες που πληκτρολογεί ο χρήστης. Στη θέση των χαρακτήρων εμφανίζονται κυκλικοί δίσκοι ή αστερίσκοι.

Παράδειγμα

```
<h5> Text and password type</h5>
 <form>
  First name:<br>
  <input type="text" name="user_name" value="John"><br>
 Password:<br>
  <input type="password" name="password1" >
</form>
```

4. **button** για να ορίσουμε ένα απλό button. Στην τιμή για την ιδιότητα value θα γράψουμε το κείμενο που θα εμφανίζεται στο button.

Παράδειγμα

```
<h5> button</h5>
 <form>
  <input type="button"  value="click here"><br>
</form>
```

5. **submit** για να ορίσουμε button υποβολής δεδομένων
6. **reset** για να κάνει επαναφορά των τιμών στις τιμές που έχουν οριστεί ως προεπιλογές

Παράδειγμα

```
<h5> Text type Submit and reset</h5>
 <form>
  First name:<br>
  <input type="text" name="first_name" value="John"><br>
  Last name:<br>
  <input type="text" name="last_name" value="Kalipolitis"> <br>
<input type="submit">
 <input type="reset" >
</form>
```

7. **radio** για να ορίσουμε radio button. Στη φόρμα με radio buttons ο χρήστης επιτρέπεται να κάνει κλικ σε μια μόνο επιλογή. Σε ένα Input μπορούμε να γράψουμε την ιδιότητα checked έτσι ώστε να είναι προεπιλεγμένο το συγκεκριμένο radio button.

Παράδειγμα

```
<h5> Radio type</h5>
 <form>
 <form>
   Επιλέξτε είδος χρεωστικής κάρτας: <br>
  <input type="radio" name="card" value="master" checked> MasterCard<br>
  <input type="radio" name="card" value="visa"> Visa<br>
</form>
```

8. **color** για να ορίσουμε επιλογή χρώματος. Σε αυτή την περίπτωση θα εμφανιστεί κατάλληλο μενού για να επιλέξει ο χρήστης ένα χρώμα.

Παράδειγμα

```
<h5> Color type</h5>
    <form>
  Please Select color:
  <input type="color" name="color1" >
</form>
```

9. **date** για την υποβολή ημερομηνίας. Αν ο χρήστης κάνει κλικ επάνω σε αυτό το πεδίο τότε θα εμφανιστεί ημερολόγιο για επιλογή ημερομηνίας (date picker). Στην επιλογή date μπορούμε να προσθέσουμε και περιορισμούς με τις ιδιότητες max και min.

10. **datetime-local** για εισαγωγή ημερομηνίας και ώρας χωρίς time zone

11. **month** για εισαγωγή μήνα και χρόνου. (Δεν υποστηρίζεται από Firefox και internet Explore 11 ή προγενέστερη έκδοση).

Παράδειγμα

```
<h5> date datetime</h5>
<form>
 Select Date:<br>
  <input type="date" name="date1">
    Select  date and time:<br>
  <input type="datetime-local" name="dtime1">
     Select month and year:<br>
  <input type="month" name="month1">
</form>
```

12. **number** για εισαγωγή αριθμού. Και σε αυτό το πεδίο μπορούμε να θέσουμε περιορισμούς min και max ορίζοντας με αυτό τον τρόπο τον ελάχιστο και το μέγιστο αριθμό.

Παράδειγμα

```
<h5> number</h5>
 <form>
Please select number (between 6 and 15):
 <input type="number" name="number1" min="6" max="15">
</form>
```

13. **search** για πεδίο αναζήτησης

Παράδειγμα

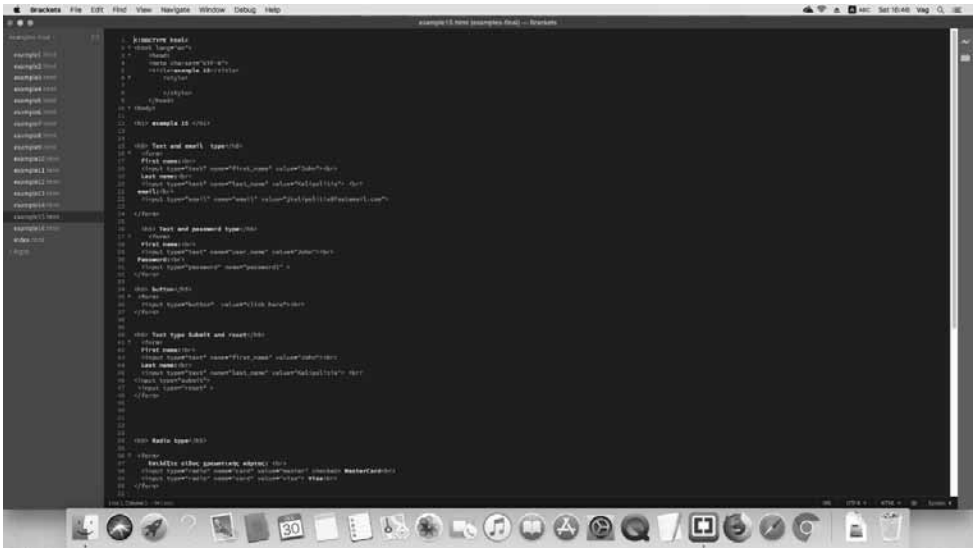
```
 <form>
 Search Google:
 <input type="search" name="search1">
</form>
```

Σημείωση

Για να ολοκληρωθεί η λειτουργία της φόρμας σε σχέση με την εισαγωγή τιμών από το χρήστη απαιτεί γνώση και άλλων γλωσσών Προγραμματισμού.

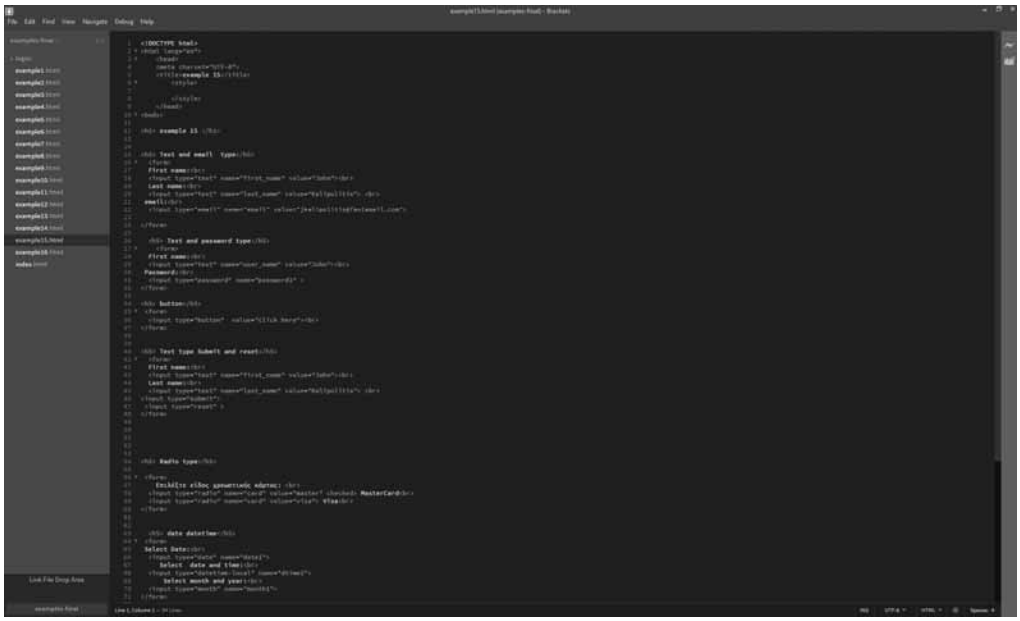
Παράδειγμα 15. HTML Forms

Κώδικας(Mac): Html Forms



Εικόνα Browser(Mac): Html Forms

Κώδικας(Win): Html Forms



Εικόνα Browser(Win): Html Forms

Dropdown list

Το dropdown list είναι στην ουσία ένα μενού επιλογών με προκαθορισμένες τιμές. Στην οθόνη του browser εμφανίζεται ως ένα πεδίο με ένα βελάκι. Όταν ο χρήστης πατήσει το βελάκι τότε θα εμφανιστεί και η λίστα με τις επιλογές. Ο χρήστης επιλέγει πατώντας το αριστερό κλικ του mouse.

Για να δημιουργήσουμε ένα dropdown list γράφουμε το περιεχόμενο μεταξύ των ετικετών <select> και </select>.

Κάθε επιλογή του dropdown list γράφεται μεταξύ των ετικετών <option> και </option>.

Συνολικά η σύνταξη του dropdown list έχει ως ακολούθως:

```
<select>
  <option value="…">…</option>
  <option value="…">…</option>
  <option value="…">…</option>
  <option value="…">…</option>
</select>
```

Παράδειγμα

```
<select>
  <option value="1">Επιλογή 1</option>
  <option value="2">Επιλογή 2</option>
  <option value="3">Επιλογή 3</option>
  <option value="4">Επιλογή 4</option>
</select>
```

Σημείωση

Εντός της ετικέτα έναρξης <option> γράφουμε τιμή για την ιδιότητα value, ενώ ανάμεσα στις ετικέτες <option> και </option> γράφουμε την περιγραφή της επιλογής που θα δει ο χρήστης στο dropdown list.

Η τιμή της ιδιότητας value θα είναι αυτή που θα σταλεί στο server που φιλοξενεί το website μόλις ο χρήστης επιλέξει τιμή από το dropdown list.

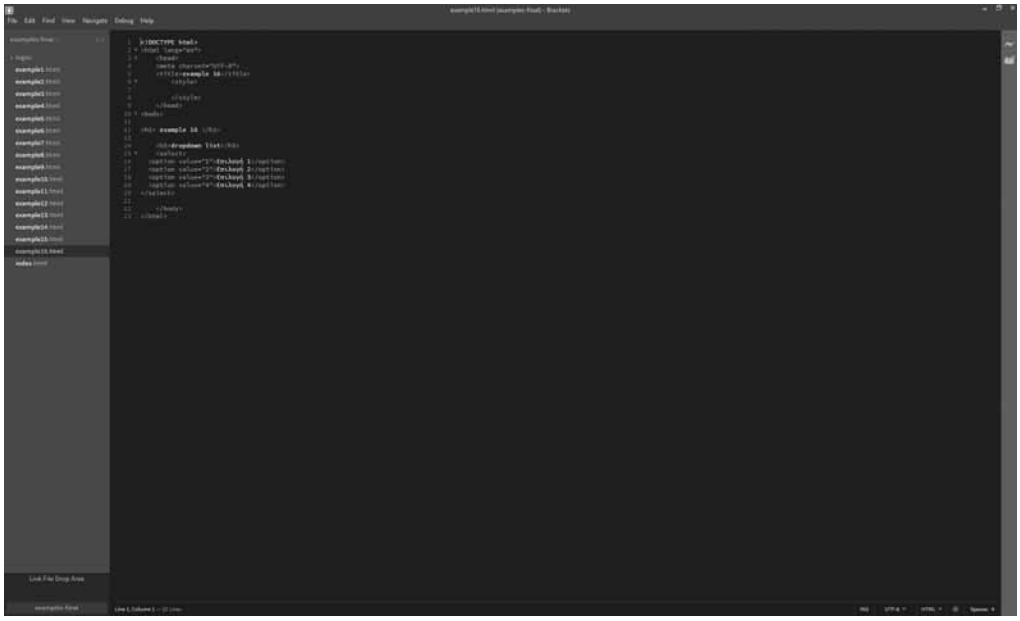
Στο προηγούμενο παράδειγμα ο χρήστης θα δει ως πρώτη επιλογή στη σελίδα «Επιλογή 1» όταν κάνει τη συγκεκριμένη επιλογή στο server θα σταλεί η τιμή 1.

Παράδειγμα 16. Dropdown lists

Κώδικας(Mac): dropdown lists

Εικόνα Browser(Mac): dropdown lists

Κώδικας(Win): dropdown lists



Εικόνα Browser(Win): dropdown lists

3.3 Βίντεο

Η ενότητα κλείνει με ορισμένα βίντεο που παρουσιάζουν όσα αναλύθηκαν στην παρούσα ενότητα, τα οποία μπορείτε να δείτε στο αντίστοιχο σημείο του ηλεκτρονικού υλικού.
Τα βίντεο 1-16 αφορούν mac και τα βίντεο 17-32 αφορούν pc.

Σύνοψη

Μετά την ολοκλήρωση της συγκεκριμένης διδακτικής ενότητας γνωρίζετε ότι:

- μπορούμε να εισάγουμε κείμενο ανάμεσα στις ετικέτες και ,
- μπορούμε να εισάγουμε αλλαγή γραμμής με την ετικέτα <br>,
- μπορούμε να δημιουργήσουμε οριζόντια γραμμή με την ετικέτα <hr>,
- μπορούμε να ορίσουμε ομάδα γραμματοσειρών με την ιδιότητα font-family,
- μπορούμε να στοιχίσουμε το κείμενο μας με την ιδιότητα text-align,
- μπορούμε να εισάγουμε μεμονωμένες φράσεις ή ολόκληρα αποσπάσματα κειμένου με τις ετικέτες <q> και <blockquote> αντίστοιχα,
- μπορούμε να δημιουργήσουμε διατεταγμένες λίστες, μη διατεταγμένες λίστες και λίστες ορισμών με τις ετικέτες <ol>, <ul> και <dl> αντίστοιχα,
- μπορούμε να δημιουργούμε html πίνακες με τη βοήθεια του στοιχείου table,
- μπορούμε να εισάγουμε εικόνες σε κελιά πινάκων με το στοιχείο img,
- μπορούμε να δημιουργούμε Html forms με την ετικέτα <form>,
- μπορούμε να προσδιορίζουμε τον τύπο της φόρμας με την ιδιότητα type,
- μπορούμε να εισάγουμε τα στοιχεία της φόρμας με την ετικέτα <input>.

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Προγράμματα Συμπληρωματικής Εκπαίδευσης

Με τη χρήση καινοτόμων μεθόδων εξ αποστάσεως εκπαίδευσης

1937 - ΔΗΜΙΟΥΡΓΙΑ RESPONSIVE ΣΕΛΙΔΩΝ ΜΕ HTML5 ΚΑΙ CSS

ΔΙΔΑΚΤΙΚΗ ΕΝΟΤΗΤΑ 3. ΕΝΣΩΜΑΤΩΣΗ MULTIMEDIA ΣΕ HTML ΣΕΛΙΔΕΣ

Εισαγωγικές Παρατηρήσεις

Στην υποενότητα 1 θα γίνει αναφορά στους τύπους των αρχείων που υποστηρίζει η Html 5. Στη συνέχεια θα αναλυθεί η εισαγωγή εικόνας με τη βοήθεια του στοιχείου img και τέλος θα αναλυθούν οι τρόποι τοποθέτησης της εικόνας καθώς και η επικάλυψη στοιχείων.

Στην υποενότητα 2 θα παρουσιαστούν οι τρόποι εισαγωγής audio και video σε μια html σελίδα. Επίσης θα γίνει εισαγωγή στο Html Canvas και θα αναλυθούν οι τρόποι σχεδίασης γραμμών και ορθογωνίων.

Στην υποενότητα 3 θα παρουσιαστούν και θα αναλυθούν ο τρόποι εισαγωγής τόξων, κύκλων και καμπυλών στο html Canvas. Επίσης θα αναλυθεί ο τρόπος ορισμού Gradient color καθώς και των ιδιοτήτων μετασχηματισμού στο Html Canvas. Τέλος θα αναλυθούν οι τρόποι εισαγωγής Links προς άλλες σελίδες καθώς και τρόποι εισαγωγής bookmarks.

ΥΠΟΕΝΟΤΗΤΑ 1. HTML ΚΑΙ ΕΙΚΟΝΕΣ

Όταν δημιουργούμε ένα website δεν θα πρέπει να ξεχνάμε ότι εκτός από την προσέλκυση επισκεπτών λόγω του περιεχομένου, το website θα πρέπει να είναι χρηστικό και εντυπωσιακό έτσι ώστε η πλοήγηση του χρήστη να είναι και χρήσιμη αλλά και ευχάριστη. Για αυτό το σκοπό φροντίζουμε να εμπλουτίζουμε τις σελίδες μας με εικόνες, φωτογραφίες, videos, ήχους και γενικά με στοιχεία multimedia.

1.1 Τύποι αρχείων

Γενικά όταν πληκτρολογήσουμε τη διεύθυνση μιας σελίδας μέσω της επικοινωνίας του υπολογιστή (ή γενικά της συσκευής που χρησιμοποιεί ο χρήστης) με το server που φιλοξενεί το website,θα πρέπει να έχουμε κατανοήσει ότι για να εμφανιστεί η σελίδα θα πρέπει ο browser να κάνει download όλες τις απαραίτητες πληροφορίες. Για παράδειγμα, αν μια σελίδα περιλαμβάνει 3 εικόνες θα πρέπει ο browser να κάνει download εκτός από τον απαραίτητο κώδικα και τις 3 εικόνες. Αυτό έχει ως επακόλουθο να πρέπει να λαμβάνονται υπόψη οι περιορισμοί της ταχύτητας του download του χρήστη.
Για παράδειγμα μια εικόνα 10MB είναι αρκετά δύσκολο να κατέβει με μια αργή σύνδεση σε 2-3 δευτερόλεπτα. Επίσης δε θα πρέπει να ξεχνάμε ότι πολλές φορές στις φορητές συσκευές όταν δεν είναι συνδεδεμένες σε κάποιο Wi-Fi spot υπάρχει αρκετά μεγάλη χρέωση ανά MB downloading. Γι' αυτόν το λόγο υπάρχουν τεχνικές που επιτρέπουν τη μείωση της χωρητικότητας των αρχείων των εικόνων.

Τύποι αρχείων εικόνας

Jpeg

Το jpeg πρότυπο εικόνας (Join Photograph Experts) υποστηρίζει χρώμα 24 Bit και είναι κατάλληλο format για web σελίδες. Η συμπίεση που υποστηρίζει είναι αρκετά καλή καθώς και ρυθμιζόμενη. Στο jpeg format μπορούμε να ρυθμίσουμε το λόγο συμπίεσης ανάλογα με τις ανάγκες που εξυπηρετεί η εικόνα. Αν η εικόνα για παράδειγμα θα εμφανίζεται σε maximum διαστάσεις 40 Pixels πλάτος X 40 Pixels ύψος τότε μπορούμε να επιλέξουμε μεγάλη συμπίεση.

Καθώς αυξάνουμε τη συμπίεση η χωρητικότητα του αρχείου μικραίνει και κερδίζουμε σε χρόνο και κόστος(αν υπάρχει χρέωση ανά GB) downloading της εικόνας και χάνουμε αντίστοιχα σε ποιότητα της εικόνας εφόσον κάποιες πληροφορίες για την εικόνα χάνονται.

Στο προηγούμενο παράδειγμα όμως με εικόνα που έχει διαστάσεις 40X40 ο χρήστης δεν μπορεί να αντιληφθεί εύκολα τη διαφορά στην ποιότητα της εικόνας αν έχει μεγάλο λόγο συμπίεσης. Οπότε στην περίπτωση αυτή δεν υπάρχουν οι αρνητικές επιπτώσεις στην ποιότητα της εικόνας. Το jpeg format δεν υποστηρίζει animation.

Σημείωση

Μια καλή πρακτική είναι να έχουμε το αρχείο της εικόνας σε ασυμπίεστη μορφή έτσι ώστε να μπορούμε να την επεξεργαστούμε ξανά από την αρχή χωρίς να έχουμε χάσει μέρος των δεδομένων του αρχείου της εικόνας.

Jpg

Το Jpg format επιτρέπει την εμφάνιση της εικόνας σταδιακά στη σελίδα. Αρχικά η εικόνα φαίνεται κάπως θολή και στη συνέχεια σταδιακά αυξάνεται η λεπτομέρειά της μέχρι να φτάσει στην τελική μορφή. Στο jpg format αποθηκεύεται η εικόνα σε διάφορα στάδια και δεν υποστηρίζει διαφάνεια. Επίσης το jpg format δεν υποστηρίζει animation.

Png

Το Png format υπάρχει σε δύο εκδόσεις. Η μία έκδοση υποστηρίζει χρώμα 8 bit που συνεπάγεται σε χρωματική παλέτα 2^8=256 χρωμάτων ενώ η δεύτερη έκδοση 24 bit υποστηρίζει παλέτα 2^{24} χρωμάτων. Η πρώτη έκδοση (8 bit) είναι κατάλληλη για μικρές εικόνες και logos ενώ η δεύτερη έκδοση (24) bit είναι κατάλληλη και για φωτογραφίες. Το png format υποστηρίζει καλύτερη συμπίεση εικόνας σε σχέση με το jpg, υποστηρίζει διαφάνεια αλλά όχι animation.

gif

Το gif format υποστηρίζει animation και διαφάνεια αλλά δεν μπορεί να έχει ποιότητα εικόνας εφόσον υποστηρίζει χρωματική παλέτα 8bit. Το gif υποστηρίζει την τεχνική dithering κατά την οποία γίνεται μίξη χρωμάτων σε γειτονικά pixels για να παραχθούν καινούργια χρώματα. Η συγκεκριμένη τεχνική στην ουσία δημιουργεί την ψευδαίσθηση των περισσότερων χρωμάτων

όταν οι εικόνες βρίσκονται σε μικρές διαστάσεις. Όταν ο χρήστης προκαλέσει μεγέθυνση της εικόνας βρίσκεται στη δυσάρεστη έκπληξη ενός ψηφιακού θορύβου ως αποτέλεσμα της τεχνικής της μίξης των χρωμάτων.

1.2 Εισαγωγή εικόνας

Εισαγωγή εικόνας <img>

Κάθε φορά που χρειάζεται να εμφανιστεί κάποια εικόνα στη σελίδα μας ο browser χρειάζεται:

1. το αρχείο της εικόνας
2. το CSS που ορίζει τον τρόπο εμφάνισης της εικόνας

Σημείωση

Για να γράψουμε σχόλια στο CSS κώδικα γράφουμε τα σχόλια με την παρακάτω σύνταξη:

```
/*  Σχόλια   */
```

Για να εμφανιστεί σωστά η εικόνα θα πρέπει να έχουμε φροντίσει έτσι ώστε ο browser να μπορεί να εκτελέσει τις παραπάνω λειτουργίες. Στην πρώτη λειτουργία θα πρέπει να έχουμε δώσει την ακριβή θέση όπου είναι αποθηκευμένη η εικόνα στο φάκελο του server που φιλοξενεί το website. Αν η πληροφορία της θέσης είναι λάθος ή το αρχείο είναι κατεστραμμένο, ο Browser δε θα λάβει το αρχείο της εικόνας από το server και τελικά η εικόνα δεν θα εμφανιστεί καθόλου.

Στη δεύτερη λειτουργία θα πρέπει να γράψουμε CSS ιδιότητες σε σχέση με τις διαστάσεις της εικόνας, τη θέση της εικόνας κτλ.

Στοιχείο img

Με τη βοήθεια του στοιχείου img μπορούμε να κάνουμε εισαγωγή εικόνας σε μία σελίδα. Εντός της ετικέτας img θα γράψουμε και τιμές για τις ιδιότητες src και alt σύμφωνα με την παρακάτω σύνταξη:

```
<img src="….path/όνομα_αρχείου" alt="κείμενο στην περίπτωση αδυναμίας
εμφάνισης της εικόνας" >
```

Το στοιχείο img δεν περιλαμβάνει ζεύγος ετικετών, δηλαδή ετικέτα αρχής και ετικέτα τέλους.

Ιδιότητα src

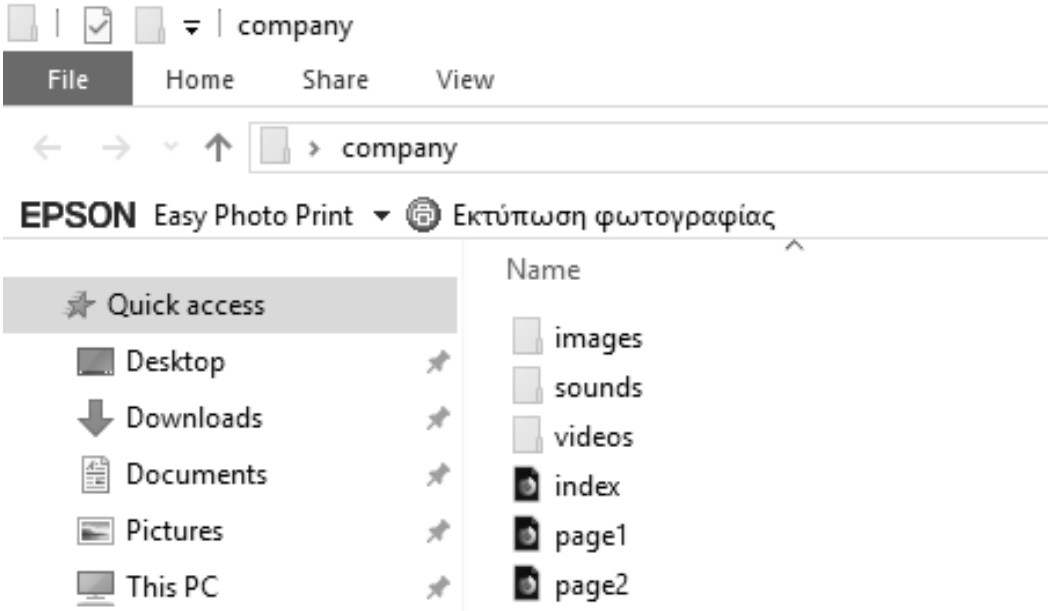
Στην ιδιότητα src θα πρέπει αρχικά να γράψουμε τη διεύθυνση στην οποία ο browser θα βρει το αρχείο της εικόνας και έπειτα θα γράψουμε τον χαρακτήρα slash '/' και τέλος το όνομα του αρχείου.

Θα πρέπει να διευκρινιστεί ότι στη θέση του path γράφουμε σχετικές διαδρομές έχοντας ως σημείο αναφοράς το folder στο οποίο υπάρχουν τα αρχεία του website. Σε καμία περίπτωση δεν γράφουμε ένα path που να περιγράφει θέσεις στο δικό μας desktop ή γενικά σε ένα συγκεκριμένο υπολογιστή.

Έστω για παράδειγμα ότι έχουμε δημιουργήσει ένα folder με όνομα company όπου θα περιέχει όλα τα αρχεία ενός website μιας εταιρείας.

Εντός του folder company έχουμε αποθηκευμένες τις σελίδες του website. Στη συνέχεια μέσα στο folder company δημιουργούμε άλλο folder(subfolder) με το όνομα images.

Εικόνα 1. Δομή φακέλων σε ένα website



Έστω ότι έχουμε αποθηκευμένο το αρχείο εικόνας με το όνομα pets-top-img.jpg.

Πηγή εικόνας: *goodrx.com/pets*

Περιπτώσεις

1. Αν το αρχείο είναι αποθηκευμένο απευθείας στο φάκελο company θα γράψουμε για το στοιχείο img:

```
<img src=" pets-top-img.jpg" alt=" cat and dog image" " width="456" height="268>
```

2. Αν το αρχείο είναι αποθηκευμένο στο subfolder images τότε θα γράψουμε:

```
<img src="images/pets-top-img.jpg" alt="cat and dog image" height="268" width="456">
```

3. Αν έχουμε αποθηκεύσει το αρχείο στο subfolder videos θα γράψουμε

```
<img src="videos/pets-top-img.jpg" alt="cat and dog image" height="268" width="456">
```

Εννοείται ότι το σωστό είναι να αποθηκεύσουμε την εικόνα στο subfolder images. Οι περιπτώσεις που αναλύθηκαν εξυπηρετούν περισσότερο το στόχο της κατανόησης. Επίσης στις ετικέτες img γράψαμε διαστάσεις 456X268 (πλάτος X ύψος) σε μονάδες pixels.

Σημείωση

Το path για την εύρεση κάποιου αρχείου δεν αλλάζει ανάλογα με τον τύπο του αρχείου. Τα paths είναι ίδια είτε πρόκειται για σελίδες, είτε για αρχεία videos κτλ.

Σημείωση

Τεχνικά δε γίνεται εισαγωγή εικόνας σε μια ιστοσελίδας αλλά σύνδεση της εικόνας με μία συγκεκριμένη html σελίδα. Το στοιχείο img φροντίζει για να αφήνει τον κατάλληλο κενό χώρο έτσι ώστε μόλις μεταφερθεί το αρχείο της εικόνας από το server τότε να εμφανιστεί η εικόνα στο συγκεκριμένο κενό χώρο. Ο χώρος αυτός θα έχει συγκεκριμένη θέση και διαστάσεις.

Εικόνες και κείμενο

Έστω ότι σε ένα στοιχείο p έχουμε εισάγει ένα στοιχείο img και επιθυμούμε αναδίπλωση του κειμένου σε σχέση με την εικόνα. Με την ιδιότητα float μπορούμε να τοποθετήσουμε την εικόνα αριστερά ή δεξιά του κειμένου. Για να πετύχουμε την αναδίπλωση του κειμένου γράφουμε για την εικόνα την ιδιότητα float όπως παρακάτω:

float:τιμή;

Όπου τιμή γράφουμε:

float:τιμή;
• **left για να επιπλέει το στοιχείο img ή άλλο στοιχείο στα αριστερά**
• **right για να επιπλέει το στοιχείο img ή άλλο στοιχείο στα δεξιά**
• **none είναι η προεπιλογή χωρίς float**
• **inherit κληρονομεί τιμή της ιδιότητας float από το πατρικό στοιχείο.**

Οπότε για να εμφανίζεται η εικόνα στα δεξιά του κειμένου γράφουμε

```
float:right;
```

ενώ για να εμφανίζεται στα αριστερά του κειμένου γράφουμε:

```
float:left;
```

Παράδειγμα

Γράφουμε δύο κλάσεις για να τις εφαρμόσουμε σε δύο διαφορετικά img στοιχεία. Στην πρώτη κλάση θα γράψουμε:

```
float:left;
```

ενώ στη δεύτερη:

```
float:right;
```

```
.border1 {
    border:3px solid red;
    float:left;
    }
    .border2 {
    border:3px solid red;
    float:right;
    }
```

Στη συνέχεια γράφουμε δύο στοιχεία p και μέσα σε κάθε στοιχείο p ενσωματώνουμε ένα στοιχείο img:

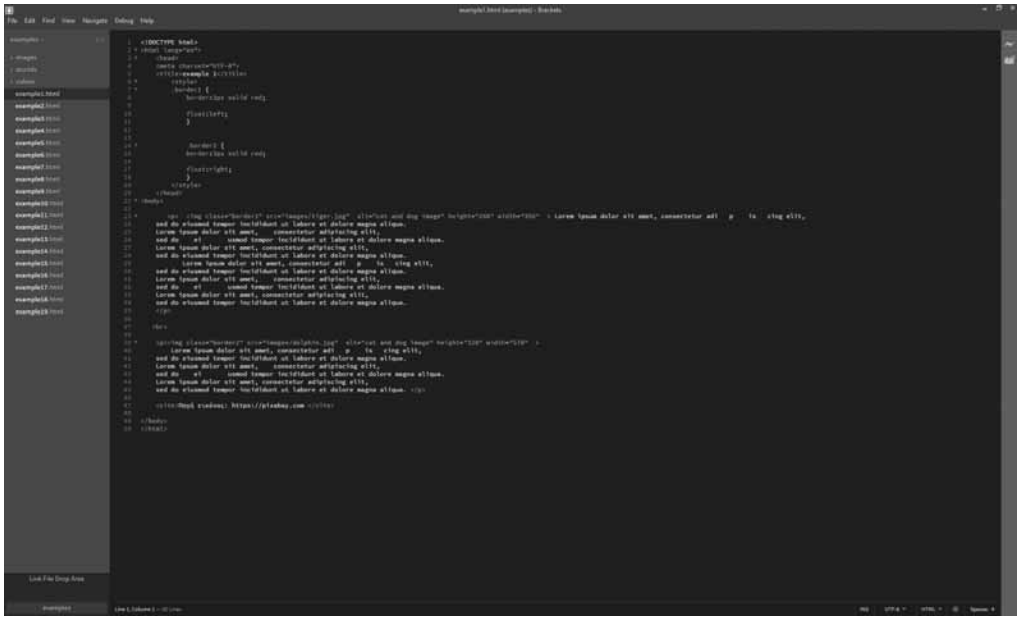
```
<p>   <img class="border1" src="images/tiger.jpg"   alt="cat and dog image"
height="268" width="356"  > Lorem…

   </p>



   <p><img class="border2" src="images/dolphin.jpg"   alt="cat and dog image"
height="320" width="570"  >

       Lorem …

</p>
```

Παράδειγμα 1. Αναδίπλωση Κειμένου

Κώδικας(Mac): αναδίπλωση κειμένου

Εικόνα Browser(Mac): αναδίπλωση κειμένου

Κώδικας(Win): αναδίπλωση κειμένου



Εικόνα Browser(Win): αναδίπλωση κειμένου

1.3 Τοποθέτηση εικόνας εντός της σελίδας (Positioning)

Σε αρκετές περιπτώσεις θα χρειαστεί να ορίζουμε μια ομάδα στοιχείων ως div:

div

Μπορούμε να ορίσουμε μια ξεχωριστή ενότητα εντός της ιστοσελίδας γράφοντας στοιχεία html μεταξύ των ετικετών και . Στην ετικέτα έναρξης του div μπορούμε να γράψουμε την ιδιότητα style ή να καλέσουμε μια ή περισσότερες κλάσεις με CSS ιδιότητες για να εφαρμόσουμε μορφοποιήσεις σε ολόκληρο το div.

Ιδιότητα Position

Με την ιδιότητα Position του CSS μπορούμε να προσδιορίσουμε τον τύπο positioning ενός στοιχείου όπως ένα στοιχείο img εντός της ιστοσελίδας μας σύμφωνα με την παρακάτω σύνταξη:

```
position:τιμή;
```

όπου τιμή θα γράψουμε μία από τις παρακάτω πέντε επιλογές:

- static
- relative
- fixed
- absolute

static

Η τιμή static είναι και η προεπιλογή. Στον τύπο static το στοιχείο τοποθετείται σύμφωνα με την κανονική ροή της σελίδας και δεν επηρεάζεται από τις ιδιότητες top, bottom left και right.

Παράδειγμα

Αρχικά θα γράψουμε δύο κλάσεις, μία για το div και μία για την εικόνα:

Στην κλάση **.back** θα γράψουμε ιδιότητες για το χρώμα του φόντου(background-color), τις διαστάσεις(width:800px ; height:800px;) και το περίγραμμα του div(border: 3px solid blue;).

```
.back{
        background-color: darkgray;
        width:800px;
        height:800px;
        border: 3px solid blue;
    }
```

Στη συνέχεια θα ορίσουμε την κλάση **.position1** με τιμή static για την ιδιότητα position. Επίσης θα ορίσουμε χρώμα περιγράμματος καθώς και χρώμα φόντου(background-color: yellowgreen;).

```
. position1 {
position:static ;
border:  3px solid red;
background-color: yellowgreen;
}
```

Στο στοιχείο body θα γράψουμε ένα div με κλάση **.back** και εντός του div θα γράψουμε ένα img στοιχείο. Εντός της ετικέτας img θα γράψουμε την κλάση static1.

```
<div class="back">
  <img class="position1" src="images/pets-top-img.jpg"  alt="cat and dog image"
height="268" width="456"  >
  </div>
```

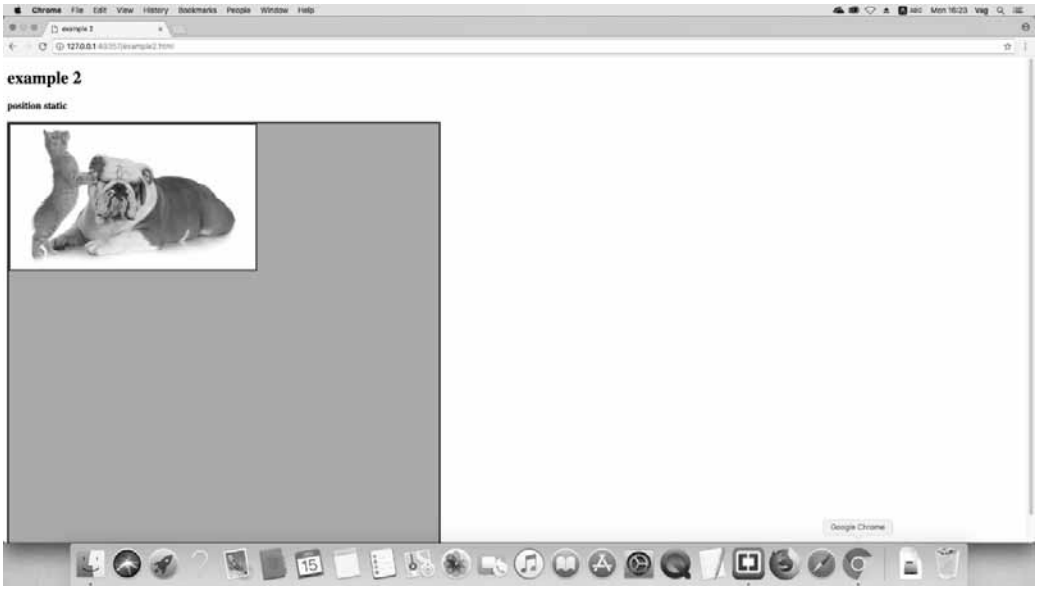
Σχόλια

Με αυτές τις ιδιότητες η normal position της εικόνας θα βρίσκεται με την επάνω αριστερή γωνία της να ταυτίζεται με την επάνω αριστερή γωνία του div εφόσον το στοιχείο img έχει γραφεί στην πρώτη γραμμή του συγκεκριμένου div.

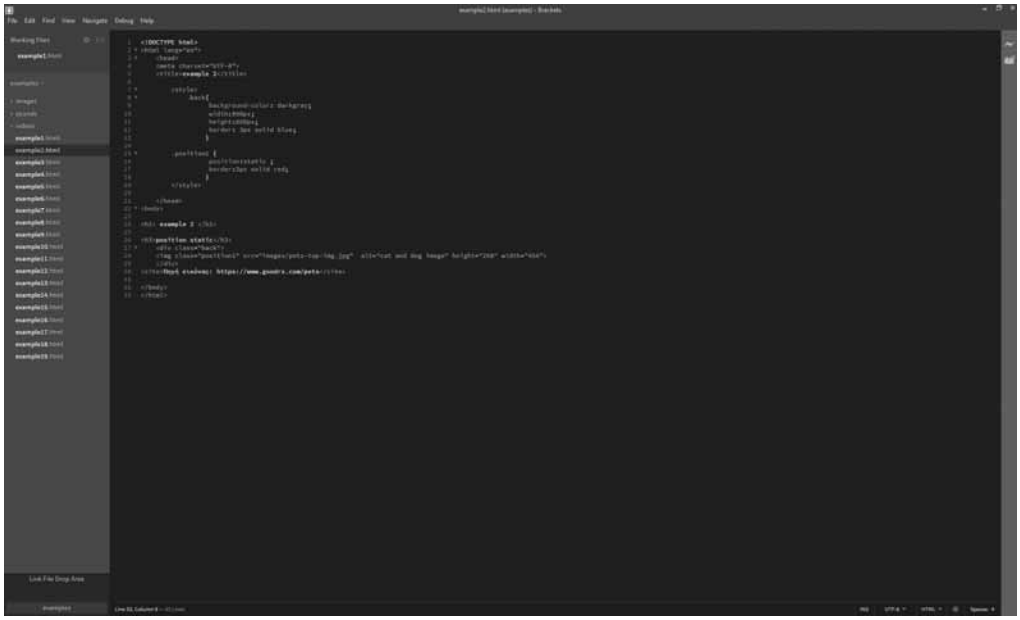
Παράδειγμα 2. Static Positioning

Κώδικας(Mac): static positioning

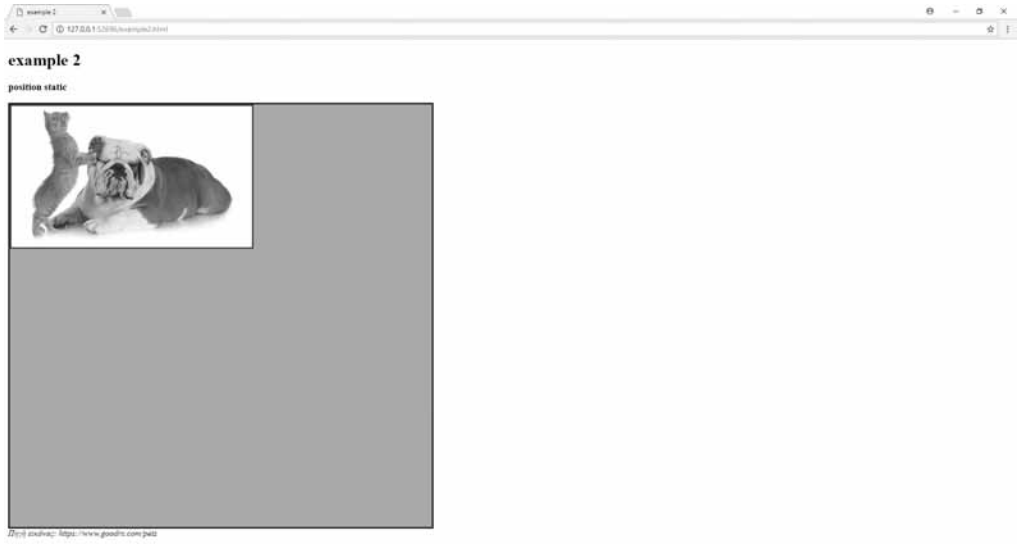
Εικόνα Browser(Mac): static positioning



Κώδικας(Win): static positioning



Εικόνα Browser(Win): static positioning



Relative

Η τιμή relative υποδηλώνει μία σχετική θέση ως προς τη normal position. Αρχικά το στοιχείο τοποθετείται στην κανονική του θέση σαν να ήταν τύπου static. Από την αρχική θέση που παίρνει μπορούμε να μετακινήσουμε τη θέση του στοιχείου με τις ιδιότητες top, right, bottom και left.

Πιο αναλυτικά γράφουμε μία από τις τέσσερις ιδιότητες και μετά μια τιμή σε px:

Μετακίνηση Θέσης
top →μετακίνηση σε px από επάνω προς τα κάτω
bottom→ μετακίνηση σε px από κάτω προς τα επάνω
left →μετακίνηση σε px από αριστερά προς τα δεξιά
right →μετακίνηση σε px από δεξιά προς τα αριστερά

Παράδειγμα

left: 130px; (μετακίνηση 130px της επάνω αριστερής γωνίας με κατεύθυνση από αριστερά προς τα δεξιά).

top:140px ; (μετακίνηση 140px της επάνω αριστερής γωνίας με κατεύθυνση από επάνω προς τα κάτω).

Παράδειγμα

Στο προηγούμενο παράδειγμα θα αλλάξουμε τις ιδιότητες της εικόνας στην κλάση .position1. Στην κλάση **.position1** θα γράψουμε position:relative έτσι ώστε να μπορούμε να μετακινήσουμε την εικόνα. Πιο συγκεκριμένα αρχικά η εικόνα θα τοποθετηθεί στην επάνω αριστερή γωνία όπως στο παράδειγμα με την static position. Στη συνέχεια για να μετακινήσουμε την επάνω αριστερή γωνία 130px από αριστερά προς τα δεξιά θα γράψουμε

left: 130px;

ενώ για να μετακινήσουμε την επάνω αριστερή γωνία της εικόνας κατά 140px και με κατεύθυνση από επάνω προς τα κάτω θα γράψουμε

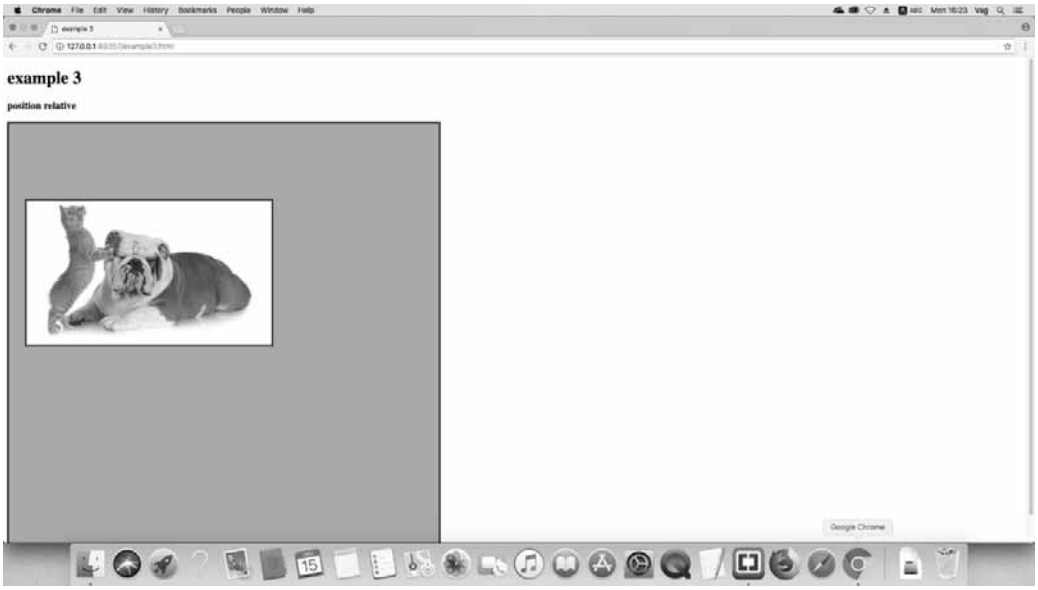
top:140px ;

```
.border1 {  position:relative ;
   border:3px solid red;
   left: 130px;
   top:140px ;
   }
```

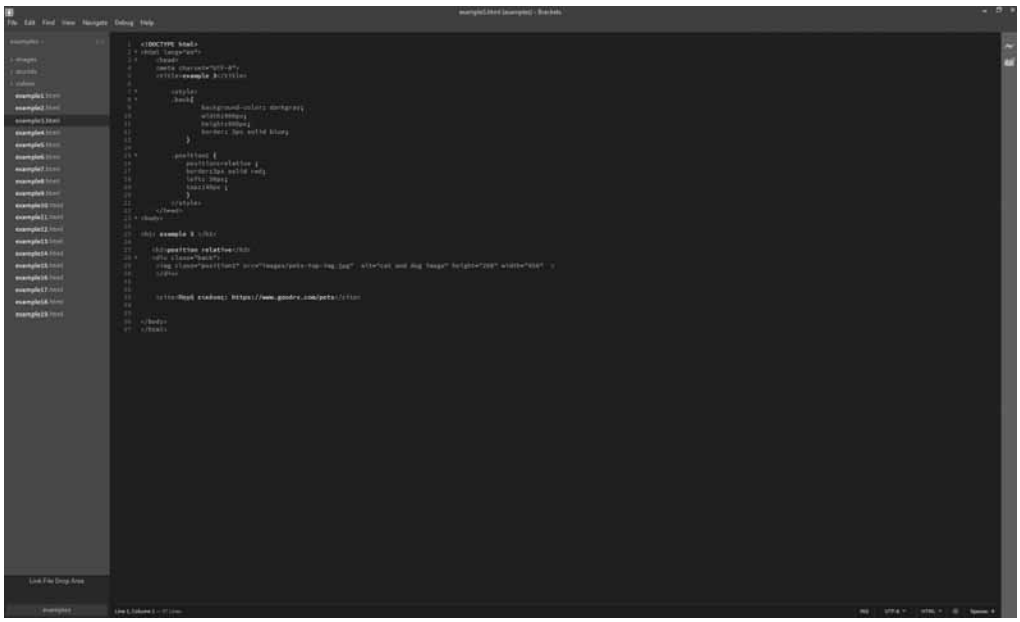
Παράδειγμα 3. Relative Positioning

Κώδικας(Mac): relative positioning

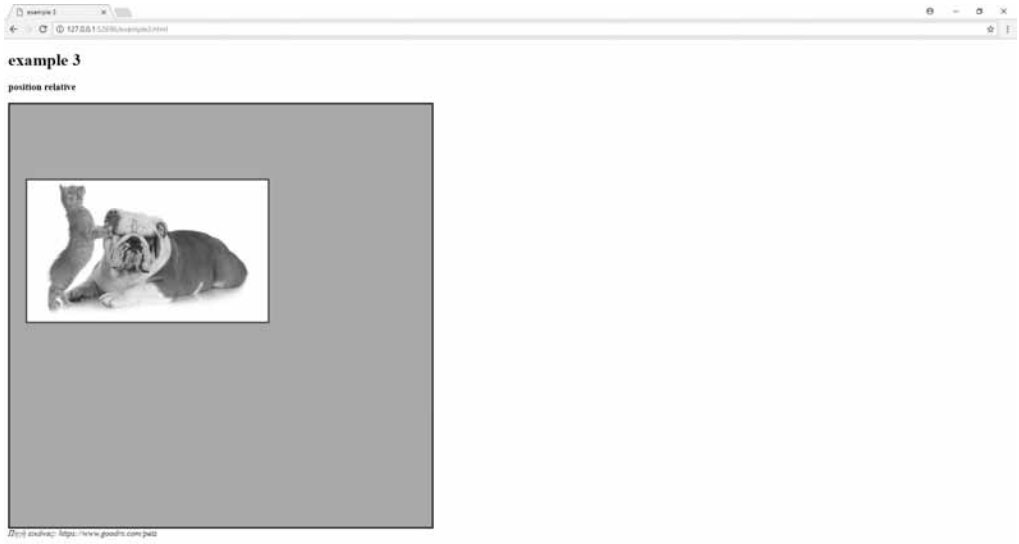
Εικόνα Browser(Mac): relative positioning



Κώδικας(Win): relative positioning



Εικόνα Browser(Win): relative positioning



fixed

Ο τύπος fixed δηλώνει μία συγκεκριμένη θέση για το στοιχείο στο παράθυρο του browser. Η θέση του στοιχείου δεν αλλάζει αν ο χρήστης επιχειρήσει scroll στη σελίδα. Η συμπεριφορά του fixed positioning είναι σαν να έχουμε κολλήσει ένα sticky note επάνω στην οθόνη του Browser. Καθώς γίνεται το scroll δεν

υπάρχει κενός χώρος στη σελίδα εξαιτίας του στοιχείου που έχει τοποθετηθεί με fixed positioning. Οι ιδιότητες top, bottom, left και right χρησιμεύουν στο να τοποθετήσουμε το στοιχείο στη σταθερή θέση σε σχέση με το παράθυρο του browser.

Ιδιότητες top, bottom, left και right

ιδιότητα top

Αντιστοιχεί στην κάθετη απόσταση της επάνω πλευράς της εικόνας από την επάνω πλευρά του παράθυρου του Browser.

ιδιότητα bottom

Αντιστοιχεί στην κάθετη απόσταση της κάτω πλευράς της εικόνας από την κάτω πλευρά του παράθυρου του Browser.

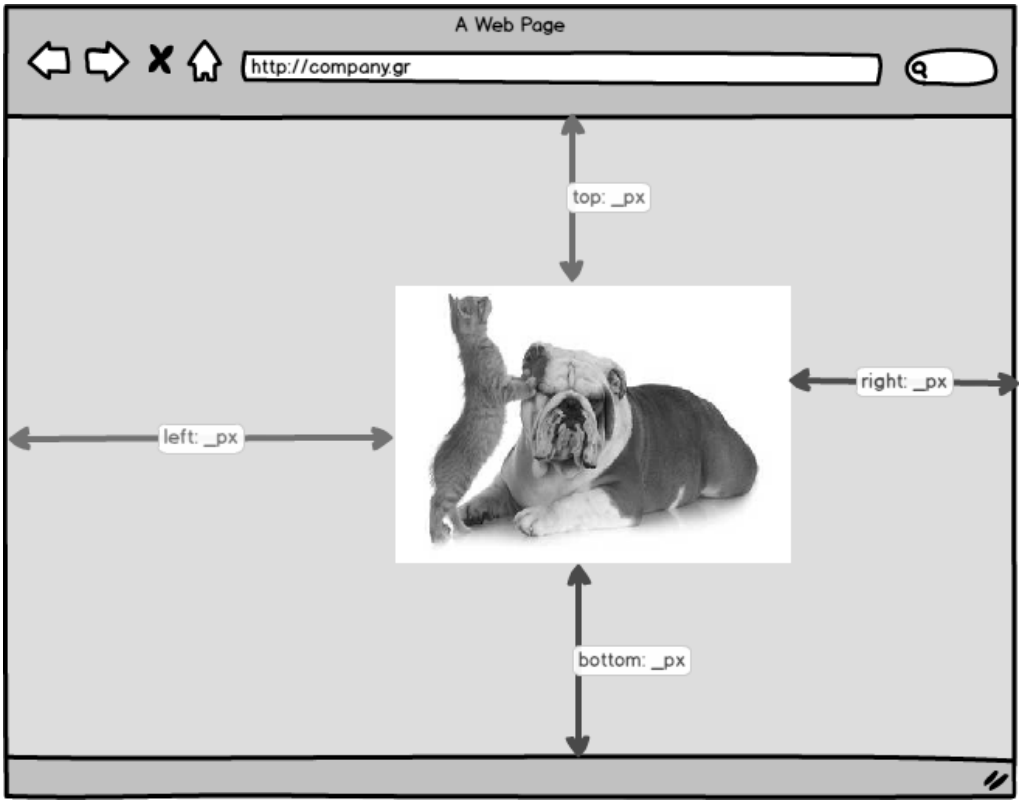
ιδιότητα left

Αντιστοιχεί στην οριζόντια απόσταση της αριστερής πλευράς της εικόνας από την αριστερή πλευρά του παράθυρου του Browser.

ιδιότητα right

Αντιστοιχεί στην οριζόντια απόσταση της δεξιάς πλευράς της εικόνας από την δεξιά πλευρά του παράθυρου του Browser.

Εικόνα 2. Αποστάσεις εικόνας



Για να προσδιορίσουμε τη θέση της εικόνας στην περιοχή του παραθύρου του browser χρειάζεται να προσδιορίσουμε μία από τις ιδιότητες top και bottom και μία από τις ιδιότητες left και right.

Έτσι λοιπόν αν θέλουμε να τοποθετήσουμε την εικόνα:

Στην επάνω αριστερή γωνία:

```
top:0px; left:0px;
```

Στην κάτω αριστερή γωνία:

```
bottom:0px; left:0px;
```

Στην επάνω δεξιά γωνία:

```
top:0px; right:0px;
```

Στην κάτω δεξιά γωνία:

```
bottom:0px; right:0px;
```

Παράδειγμα

```
.position1 {
        position:relative ;
        border:3px solid red;
        left: 30px;
        top:140px ;
        }
```

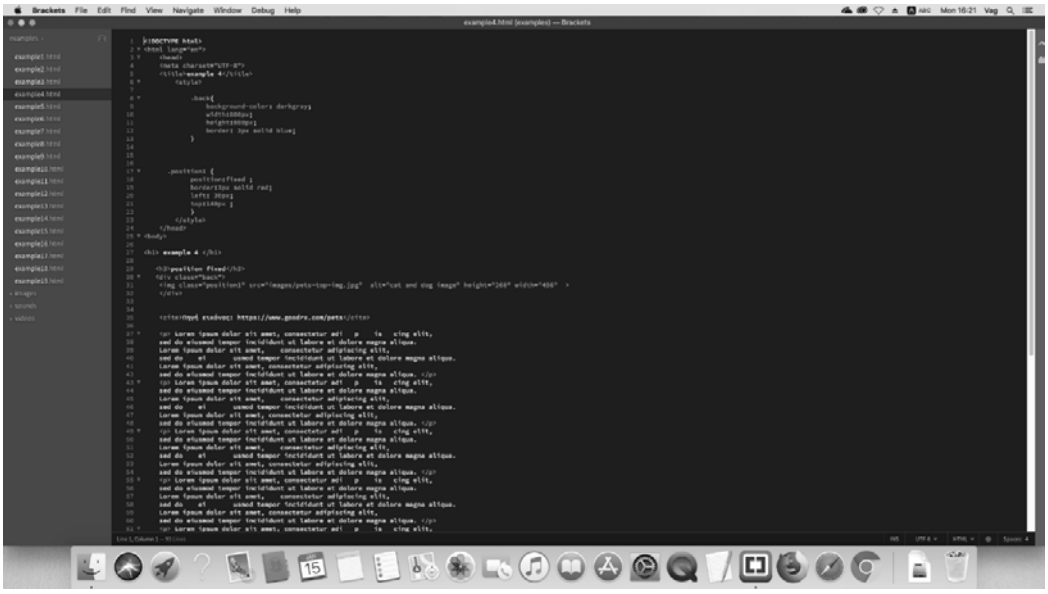
Σχόλια

Έχουμε τοποθετήσει την εικόνα στο παράθυρο του browser έτσι ώστε:

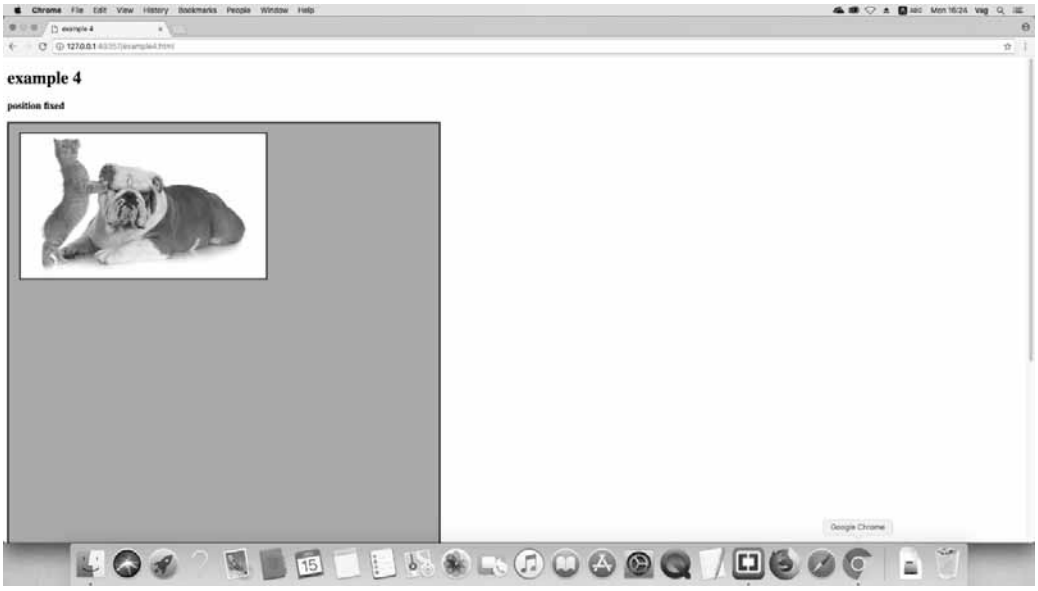
- Η οριζόντια απόσταση αριστερής πλευράς εικόνας-αριστερής πλευράς παραθύρου Browser να είναι ίση με 30 pixels.
- Η κάθετη απόσταση επάνω πλευράς εικόνας-επάνω πλευράς παραθύρου Browser να είναι ίση με 140 pixels

Παράδειγμα 4. Fixed Positioning

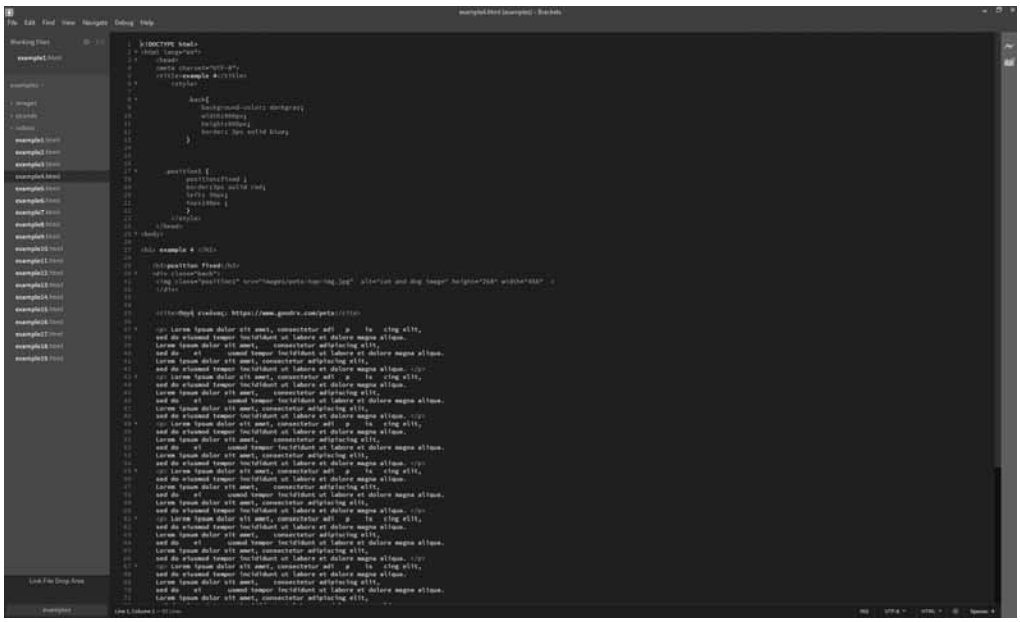
Κώδικας(Mac): fixed positioning



Εικόνα Browser(Mac): fixed positioning



Κώδικας(Win): fixed positioning



Εικόνα Browser(Win): fixed positioning

absolute

Στο absolute positioning γίνεται τοποθέτηση σε σχέση με τον πιο κοντινό πρόγονο του στοιχείου. Δηλαδή εάν για παράδειγμα δημιουργήσουμε ένα div και εντός του div κάνουμε εισαγωγή εικόνας:

```
<div>
<img ……..>
</div>
```

Ο πιο κοντινός πρόγονος του img στοιχείου είναι το συγκεκριμένο div. Στην ουσία το absolute μοιάζει με το fixed positioning με τη διαφορά ότι στο fixed τοποθετούμε το στοιχείο εντός του παραθύρου του browser ενώ στο absolute τοποθετούμε το στοιχείο σε σχέση με το χώρο που καταλαμβάνει ο πιο κοντινός πρόγονος.

Τον πιο κοντινό πρόγονο θα τον τοποθετήσουμε με **relative positioning**.

Παράδειγμα

Θα χρησιμοποιήσουμε τη δομή των παραδειγμάτων για το positioning, θα δημιουργήσουμε ένα div και εντός του div θα γράψουμε ένα στοιχείο img. Έτσι ο πιο κοντινός πρόγονος του img θα είναι το συγκεκριμένο στοιχείο div.

Εφόσον το div είναι ο πιο κοντινός πρόγονος θα τοποθετήσουμε το div με relative positioning με τη βοήθεια της παρακάτω κλάσης:

```
.back{ position: relative;
        left: 30px;
        top:40px ;
         background-color: darkgray;
         width:800px;
         height:800px;
        border: 3px solid blue;
        }
```

Στην κλάση έχουμε γράψει position:relative και έχουμε δώσει τιμές για τις ιδιότητες width, height, border και background-color. Επίσης έχουμε μετακινήσει το div 30px από αριστερά προς τα δεξιά και 40px από επάνω προς τα κάτω.

Στη συνέχεια θα γράψουμε την κλάση για το στοιχείο img με position:absolute όπως παρακάτω:

```
.position1 { position:absolute;
        border:3px solid red;
        right: 0px;
         top:0px ;
         }
```

Με τις παραπάνω ιδιότητες έχουμε τοποθετήσει το στοιχείο img στην επάνω δεξιά γωνία του div. Δηλαδή όπου και να εμφανιστεί το div η εικόνα θα βρίσκεται στην ίδια θέση σε σχέση με το div.

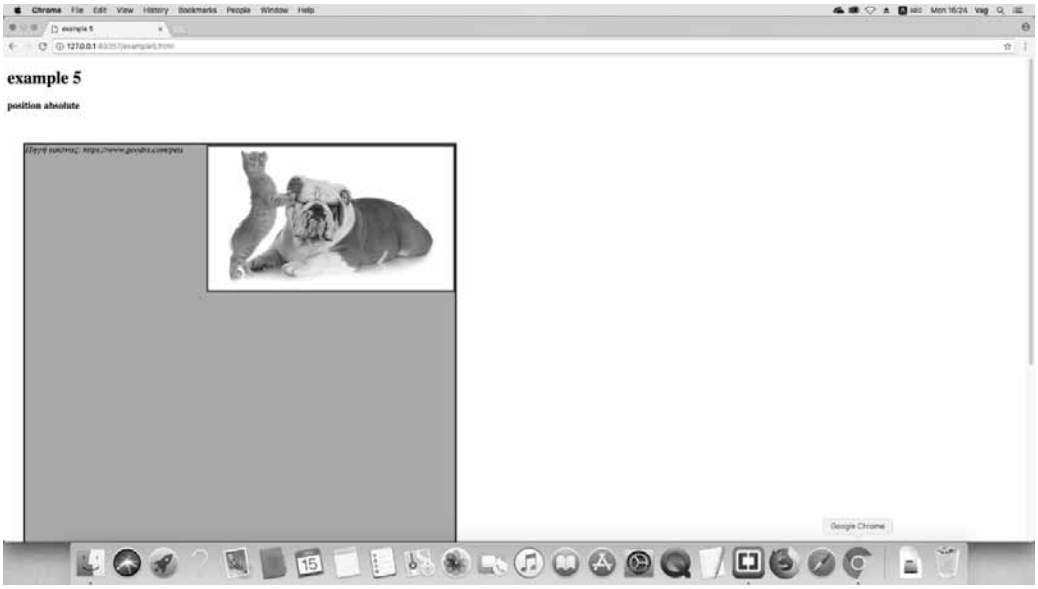
Τέλος εντός του στοιχείου body έχουμε γράψει

```
<div class="back">
   <img class="border1" src="images/pets-top-img.jpg"  alt="cat and dog image"
height="268" width="456" >
   </div>
```

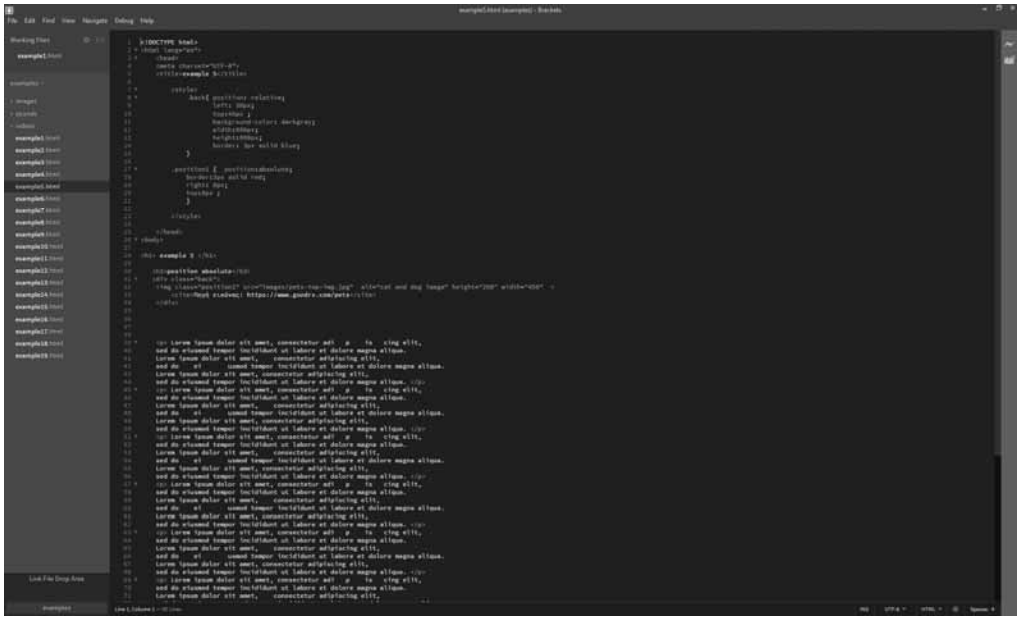
Παράδειγμα 5. Absolute Positioning

Κώδικας(Mac): absolute positioning

Εικόνα Browser(Mac): absolute positioning



Κώδικας(Win): absolute positioning



Εικόνα Browser(Win): absolute positioning

1.4 Επικάλυψη στοιχείων

Όταν τοποθετούνται στοιχεία στη σειρά και το ένα βρίσκεται πίσω από κάποιο άλλο τότε έχουμε επικάλυψη στοιχείων. Με την ιδιότητα z-index μπορούμε να ορίσουμε τη σειρά με την οποία θα εμφανίζονται τα στοιχεία. Η ιδιότητα z-index μπορεί να πάρει θετικές και αρνητικές τιμές.

Ένα στοιχείο με μεγαλύτερη τιμή θα εμφανιστεί μπροστά από ένα στοιχείο με μικρότερη τιμή. Αν δύο στοιχεία έχουν την ίδια τιμή στην ιδιότητα z-index τότε αυτό που γράφτηκε μετά στον κώδικα html θα εμφανιστεί μπροστά.

Σημείωση

Μπορούμε με την ιδιότητα z-index σε μια σελίδα με κείμενο και εικόνα να ορίσουμε στην εικόνα τιμή -1 για την ιδιότητα z-index και η εικόνα να εμφανίζεται πίσω από το κείμενο.

Παράδειγμα

Στο προηγούμενο παράδειγμα θα προσθέσουμε τρία στοιχεία img με το ίδιο positioning fixed και διαφορετικές διαστάσεις με τις κλάσεις border1, border2 και border3.

Σε κάθε κλάση θα προσθέσουμε την ιδιότητα z-index με αριθμούς από 1-3. Έτσι λοιπόν πρώτη στη σειρά θα εμφανιστεί η εικόνα με z-index:3, δεύτερη η εικόνα με z-index:2 και τελευταία η εικόνα με z-index:1.

Οι τρεις κλάσεις είναι οι παρακάτω:

```
.border1 {  position:fixed ;
    border:3px solid red;
    left: 30px;
    top:140px ;
    z-index: 1;
    }
    .border2 {  position:fixed ;
    border:3px solid red;
    left: 30px;
    top:140px ;
    z-index: 2;
    }
    .border3 {  position:fixed ;
    border:3px solid red;
    left: 30px;
    top:140px ;
    z-index: 3;
    }
```

Μέσα στο στοιχείο Body θα γράψουμε:

```
<div class="back">
    <img class="border1" src="images/tiger.jpg" alt="cat and dog image" height="268" width="656" >
    <img class="border2" src="images/dolphin.jpg" alt="cat and dog image" height="320" width="570" >
    <img class="border3" src="images/eagle.jpg" alt="cat and dog image" height="290" width="500" >
...
...
...
</div>
```

Οπότε πρώτη θα εμφανίζεται η εικόνα με τον αετό, δεύτερη η εικόνα με το δελφίνι και τρίτη η εικόνα με την τίγρη.

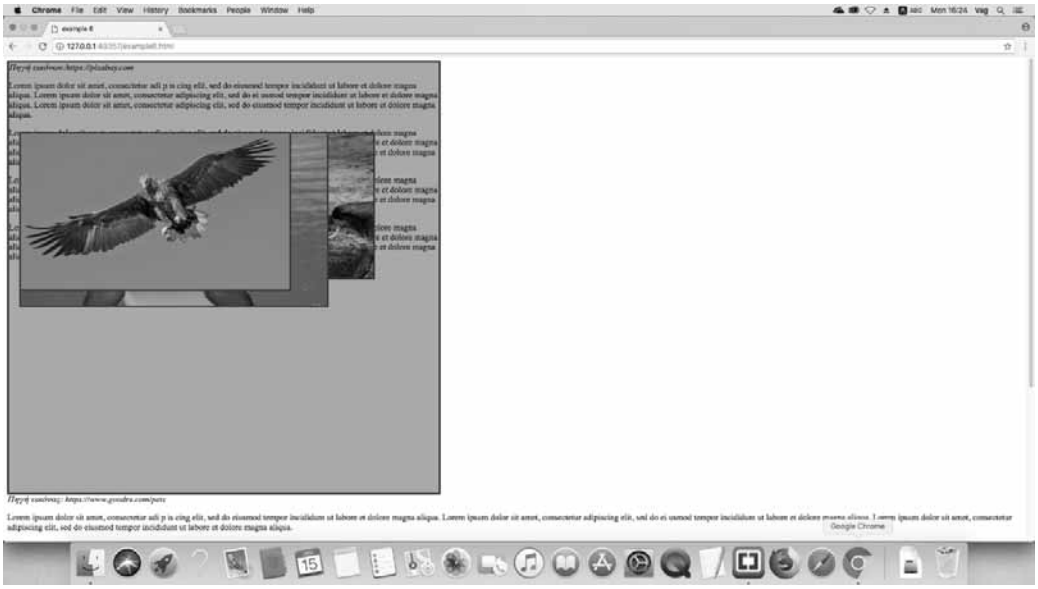
Σημείωση

Το CSS positioning ισχύει γενικά για τα στοιχεία της html και όχι μόνο για τις εικόνες.

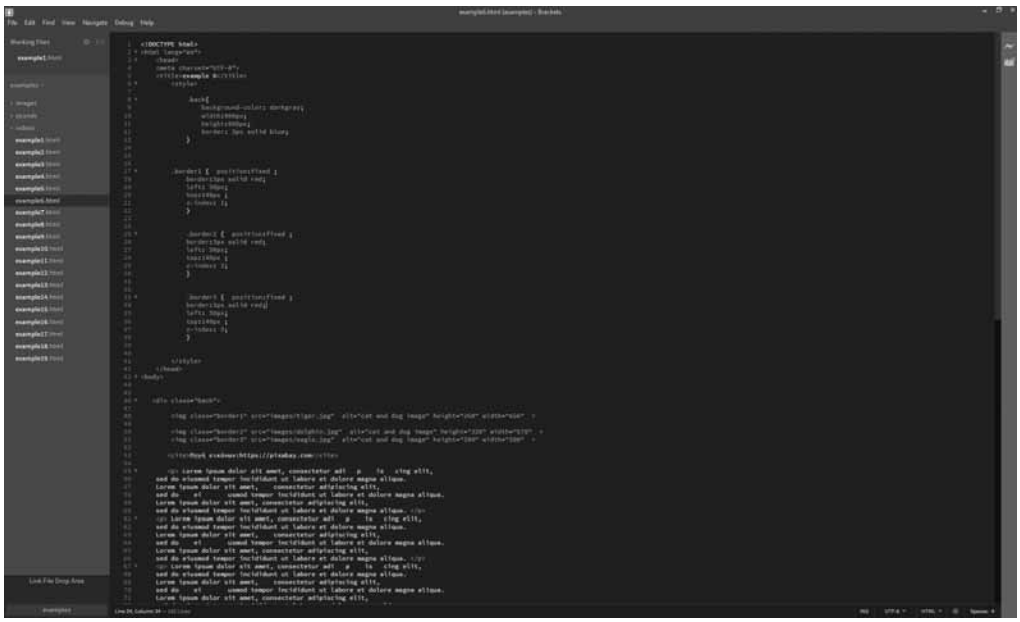
Παράδειγμα 6. z-index

Κώδικας(Mac): z-index

Εικόνα Browser(Mac): z-index



Κώδικας(Win): z-index



Εικόνα Browser(Win): z-index

ΥΠΟΕΝΟΤΗΤΑ 2. AUDIO, VIDEO ΚΑΙ HTML CANVAS

2.1 Εισαγωγή video και audio

Η ετικέτα video υποστηρίζει τις παρακάτω μορφές αρχείων

- Mp4
- WebM
- Ogg

Για να κάνουμε εισαγωγή video σε μία σελίδα θα χρησιμοποιήσουμε την ετικέτα <video> σύμφωνα με την παρακάτω σύνταξη:

```
<video>
<source src="videos/όνομα_αρχείου.mp4" type="video/mp4">
  <source src="videos/ όνομα_αρχείου.ogg" type="video/ogg">
  Your browser does not support the video tag.
</video>
```

Στην ετικέτα έναρξης έχουμε τις ακόλουθες επιλογές για την αναπαραγωγή του video:

- width="…" height="…" για τις διαστάσεις του video.
- muted για να έχουμε ενεργοποιημένη τη σίγαση ήχου.
- controls για να εμφανιστούν τα controls για την αναπαραγωγή video.
- autoplay για αυτόματη αναπαραγωγή video με τη φόρτωση της σελίδας. Η συγκεκριμένη επιλογή χρειάζεται ιδιαίτερη προσοχή καθώς μπορεί να προκαλέσει προβλήματα σε κάποιο χρήστη όταν ανοίξει μια σελίδα και ξεκινήσει ο ήχος του video. Οπότε η επιλογή θα πρέπει να αποφεύγεται ή τουλάχιστον να συνδυάζεται με την ιδιότητα muted έτσι ώστε να ξεκινάει αυτόματα το Video αλλά με σίγαση ήχου.
- scr για τη διαδρομή που θα βρει ο browser το αρχείο του video.

Στην ετικέτα video επιτρέπεται η χρήση δύο ή περισσότερων πηγών που περιλαμβάνουν διαφορετικά formats για το ίδιο video, έτσι ώστε στην περίπτωση που δεν μπορεί να γίνει η αναπαραγωγή μιας πηγής με συγκεκριμένο format να μπορεί να γίνει αναπαραγωγή σε άλλο format.

Το μήνυμα "Your browser does not support the audio tag" θα εμφανιστεί αν ο Browser δεν υποστηρίζει το στοιχείο video.

Παράδειγμα

```
<video width="640" height="480" controls muted autoplay>
  <source src="videos/Squirrel.mp4" type="video/mp4">
  <source src="videos/Squirrel.ogg" type="video/ogg">
  Your browser does not support the video tag.
</video>
```

Εισαγωγή videos από Youtube

Πριν ξεκινήσουμε τη διαδικασία ενσωμάτωσης Video από το YouTube θα πρέπει να γνωρίζουμε ότι λόγω πνευματικών δικαιωμάτων το πιθανότερο είναι να μην είναι δυνατή η αναπαραγωγή από το δικό μας website, αλλά με link που εμφανίζεται να μπορεί ο χρήστης να παρακολουθήσει το video μόνο από τη σελίδα του YouTube.

Τα βήματα που θα πρέπει να ακολουθήσουμε είναι:

1. Επιλογή video από τη σελίδα https://www.youtube.com/
2. Επιλογή ενσωμάτωση
3. Επιλογή αντιγραφή
4. Επικόλληση στον κώδικα Html εντός του στοιχείου Body

Ο κώδικας που θα αντιγράψουμε από το YouTube περικλείεται μεταξύ των ετικετών <iframe> και </iframe>.

Παράδειγμα

```
<iframe width="560" height="315"
src="https://www.youtube.com/embed/MtrOYsNCPmg" frameborder="0"
gesture="media" allow="encrypted-media" allowfullscreen></iframe>
```

Εισαγωγή audio

Η ετικέτα audio υποστηρίζει τις παρακάτω μορφές αρχείων:

- Mp3
- Ogg
- Wav

Το στοιχείο audio έχει την ακόλουθη σύνταξη:

```
<audio>
<audio controls>
 <source src=" sounds/όνομα_αρχείου.mp3" type="audio/mpeg">
  <source src=" sounds/όνομα_αρχείου.ogg" type="audio/ogg">
  Your browser does not support the audio tag
  </audio>
```

Και στο στοιχείο audio στην ετικέτα έναρξης μπορούμε να γράψουμε πηγές src με διαφορετικές μορφές για το ίδιο αρχείο audio έτσι ώστε αν ο browser δεν υποστηρίζει τη κάποια μορφή να υπάρχει εναλλακτική μορφή για το ίδιο audio. Το μήνυμα Your browser does not support the audio tag. Θα εμφανιστεί αν ο Browser δεν υποστηρίζει το στοιχείο audio.

Στη ετικέτα έναρξης του στοιχείου audio υπάρχουν οι αντίστοιχες επιλογές που συναντήσαμε στο στοιχείο video:

- muted για να έχουμε ενεργοποιημένη τη σίγαση ήχου.
- controls για να εμφανιστούν τα controls για την αναπαραγωγή video
- autoplay για αυτόματη αναπαραγωγή μόλις φορτώσει η σελίδα.

Παράδειγμα

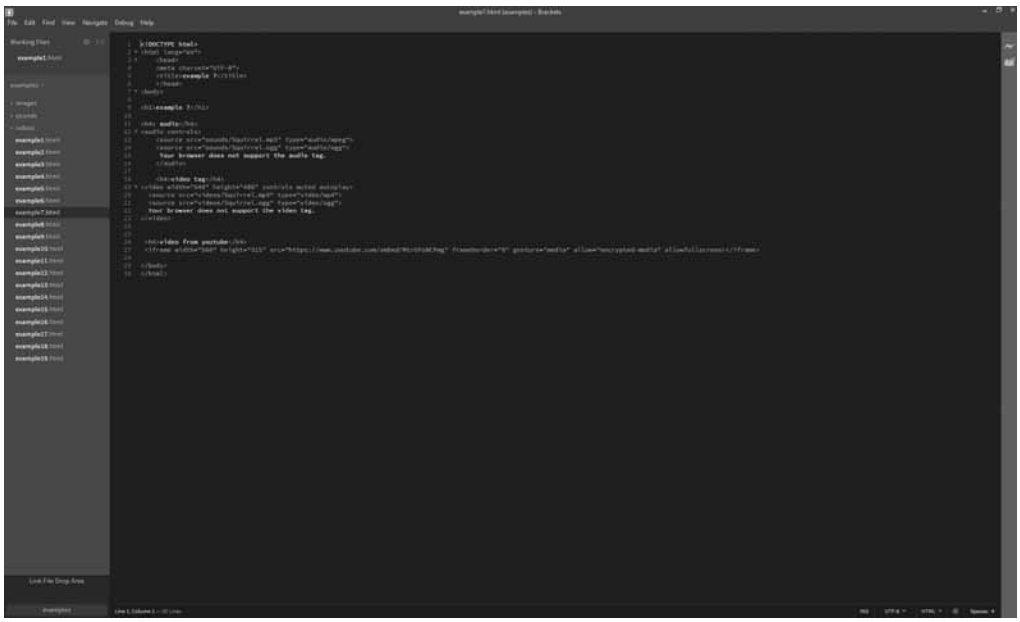
```
 <audio controls>
    <source src="sounds/Squirrel.mp3" type="audio/mpeg">
     <source src="sounds/Squirrel.ogg" type="audio/ogg">
  Your browser does not support the audio tag.
   </audio>
```

Παράδειγμα 7. Video και Audio

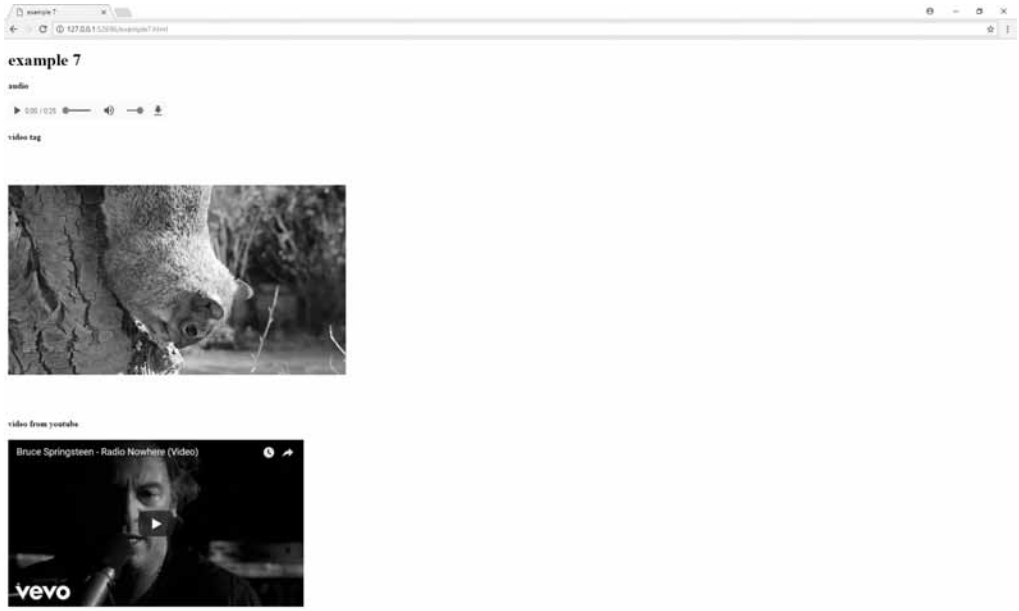
Κώδικας(Mac): video & audio

Εικόνα Browser(Mac): video & audio

Κώδικας(Win): video & audio



Εικόνα Browser(Win): video & audio



2.2 Εισαγωγή στο Html canvas

Με το στοιχείο Html Canvas μπορούμε να σχεδιάσουμε γραφικά γράφοντας κώδικα. Για την υποστήριξη του html canvas είναι απαραίτητη η JavaScript. Θα πρέπει να γίνει κατανοητό ότι το html canvas είναι ένας στοιχείο που περιέχει γραφικά.

Με το html Canvas μπορούμε σε μία σελίδα να σχεδιάζουμε γραμμές, τόξα, κύκλους σχήματα, κείμενο κτλ

Γενικά το html Canvas μπορεί να κάνει τα παρακάτω:

1. Σχεδιασμός γραφικών
2. Σχεδιασμός κειμένου
3. Ενσωμάτωση animation
4. Αλληλεπίδραση και σύνδεση με JavaScript events όπως mouse click, button click κα
5. Σεισμός Canvas σε games.

Πριν ξεκινήσουμε να γράφουμε κώδικα για το html Canvas θα πρέπει να ξέρουμε:

- Εφόσον το canvas υποστηρίζεται από την JavaScript υπάρχει διάκριση μεταξύ πεζών και κεφαλαίων γραμμάτων.
- Ένα μικρό λάθος στον κώδικα για το canvas μπορεί να στοιχίσει ακόμη και τη μη εμφάνιση του στη σελίδα.
- Κατά τη διάρκεια της συγγραφής του κώδικα θα πρέπει να κάνουμε δοκιμές για να είμαστε σίγουροι ότι δεν έχουμε κάνει λάθος. Δεν έχει νόημα να γράψουμε κώδικα για 11 σχήματα για παράδειγμα και λόγω λάθους να μην έχει ενεργοποιηθεί το canvas.
- Δεν αντιγράφουμε κώδικα από pdf, word διότι υπάρχει σοβαρή πιθανότητα να αντιγράφει λάθος κάποιος χαρακτήρας και να μην τρέχει ο κώδικας.

Ενεργοποίηση του canvas

Για να ενεργοποιήσουμε το στοιχείο canvas θα γράψουμε τις ιδιότητες του Canvas ανάμεσα στις ετικέτες <canvas> και </canvas>.

Στην ετικέτα έναρξης του στοιχείου canvas θα γράψουμε id διαστάσεις με τις ιδιότητες width και height και προαιρετικά style.

```
<canvas id="όνομα_canvas" width="…" height="…" style="…">

</canvas>
```

Σημείωση

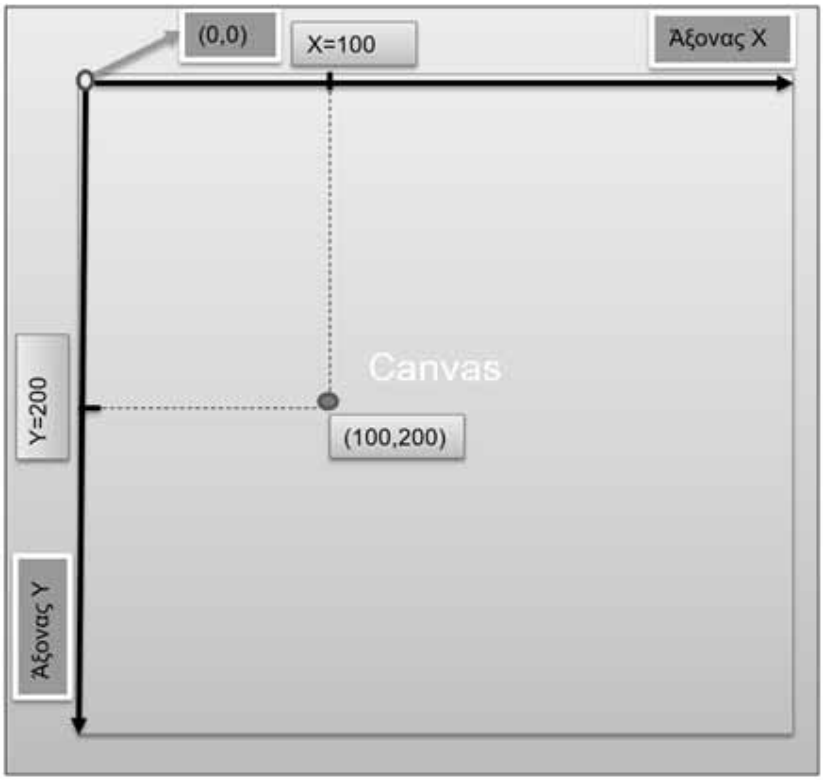
Το id στην ουσία χρησιμεύει στο να αναφερόμαστε στο συγκεκριμένο canvas. Αν κατά το γράψουμε λάθος δε θα μπορεί να σχεδιαστεί οτιδήποτε στο canvas.

Συντεταγμένες στο canvas

Το canvas είναι ένα ορθογώνιο παραλληλόγραμμο με διαστάσεις που ορίζουμε εμείς. Στο παραλληλόγραμμο του canvas ορίζονται δύο άξονες x,y. Ο οριζόντιος άξονας x συμπίπτει με τη επάνω οριζόντια πλευρά του παραλληλογράμμου, ενώ ο κάθετος άξονας y συμπίπτει με την κάθετη αριστερή πλευρά του παραλληλογράμμου.

Το σημείο (0,0) των συντεταγμένων του Canvas συμπίπτει με την επάνω αριστερή γωνία του Canvas. Για να ορίσουμε ένα σημείο x,y στο canvas μετακινούμαστε προς τα δεξιά και οριζόντια τόσα pixels όσο μας δείχνει η τιμή του x. Στη συνέχεια μετακινούμαστε κάθετα και προς τα κάτω όσα pixels μας δείχνει η τιμή y.

Εικόνα 3. Συντεταγμένες Canvas



Δίνοντας θετικές τιμές στο x σημαίνει μετακίνηση της γραφίδας προς τα δεξιά και οριζόντια ενώ αρνητικές τιμές στο x σημαίνει μετακίνηση της γραφίδας προς τα αριστερά και οριζόντια.

Δίνοντας θετικές τιμές στο y σημαίνει μετακίνηση της γραφίδας κατακόρυφα και προς τα κάτω, ενώ αρνητικές τιμές στο y σημαίνει μετακίνηση της γραφίδας κατακόρυφα και προς τα επάνω.

Σχεδιάζοντας στο canvas

Εφόσον έχουμε δημιουργήσει το canvas Για να σχεδιάσουμε σε αυτό θα γράψουμε κώδικα ανάμεσα στις ετικέτες <script> και </script> εντός του στοιχείου body.

Αρχικά θα πρέπει να αποκτήσουμε πρόσβαση στο canvas και να επιλέξουμε το συγκεκριμένο canvas που δημιουργήσαμε. Για να αποκτήσουμε πρόσβαση θα γράψουμε την παρακάτω γραμμή:

```
<script>
var canvas = document.getElementById("αναγνωριστικό_Canvas");
```

θα πρέπει να προσέξουμε να γράψουμε σωστά το αναγνωριστικό όνομα του canvas εντός των εισαγωγικών. Επίσης θα πρέπει να προσέξουμε τη διάκριση πεζών κεφαλαίων.

Για παράδειγμα, αν γράψουμε:

```
var canvas = document.getElementByid("canvas1");
```

δηλαδή id αντί για Id, δε θα σχεδιαστεί κάτι στο canvas.

Στη συνέχεια για να δημιουργηθεί ένα αντικείμενο σχεδίασης θα γράψουμε:

```
ctx = canvas.getContext("2d");
```

Αφού γράψουμε τον κώδικα για τη σχεδίαση θα γράψουμε και την ετικέτα λήξης </script>

Επιλέγοντας χρώματα και τρόπο σχεδίασης στο canvas

Εάν επιθυμούμε σχεδίαση απλών γραμμών και σχημάτων χωρίς γέμισμα θα γράφουμε:

1. για επιλογή χρώματος

```
ctx.strokeStyle = "…";
```

2. για να γίνει η σχεδίαση

```
ctx.stroke();
```

Εάν επιθυμούμε σχεδίαση συμπαγών σχημάτων με γέμισμα θα γράφουμε:

1. για επιλογή χρώματος γεμίσματος

```
ctx.fillStyle = "…";
```

2. για να γίνει η σχεδίαση

```
ctx.fill();
```

Για να επιλέξουμε χρώμα για γέμισμα θα γράψουμε:

```
ctx.fillStyle = "χρώμα";
```

Για να επιλέξουμε χρώμα για σχεδίαση όπως για μια περιφέρεια κύκλου θα γράψουμε:

```
ctx.strokeStyle="χρώμα";
```

Για την τιμή στο χρώμα μπορούμε να επιλέξουμε όποια συμβατή μορφή χρώματος με την Html.

2.3 Ορθογώνια και γραμμές στο html Canvas

Για να σχεδιάσουμε μόνο το περίγραμμα ενός ορθογωνίου παραλληλογράμμου θα γράψουμε:

1. για να επιλέξουμε το χρώμα σχεδίασης

```
ctx.strokeStyle="…";
```

2. για να σχεδιάσουμε το ορθογώνιο παραλληλόγραμμο

```
ctx.strokeRect(a,b,c,d);
```

Οι τιμές των παραμέτρων a,b αντιπροσωπεύουν τις συντεταγμένες (x,y) της επάνω αριστερής γωνίας του ορθογωνίου παραλληλογράμμου. Οι παράμετροι c,d αντιπροσωπεύουν τις διαστάσεις πλάτος και ύψος του παραλληλογράμμου (σε pixels) αντίστοιχα.

Παράδειγμα

```
ctx.strokeStyle="green";
ctx.strokeRect(300,650,250,150);
```

Έστω για παράδειγμα ότι επιθυμούμε να τοποθετήσουμε ένα παραλληλόγραμμο στην επάνω αριστερή γωνία του canvas θα γράψουμε:

```
ctx.strokeRect(0,0,c,d);
```

Για να σχεδιάσουμε ένα ορθογώνιο παραλληλόγραμμο με γέμισμα χρώματος τότε θα γράψουμε:

1. για να επιλέξουμε το χρώμα γεμίσματος

```
ctx.fillStyle = "…";
```

2. για να σχεδιάσουμε το ορθογώνιο παραλληλόγραμμο με γέμισμα χρώματος

```
ctx.fillRect(a,b,c,d);
```

οι παράμετροι a, b, c, d

a → τετμημένη x για την επάνω αριστερή γωνία του ορθογώνιου παραλληλογράμμου

b → τεταγμένη y για την επάνω αριστερή γωνία του ορθογώνιου παραλληλογράμμου

c → πλάτος σε pixels του ορθογωνίου

d → ύψος σε pixels του ορθογωνίου

Σημείωση

Οι παράμετροι είναι ίδιες είτε πρόκειται για απλό σχεδιασμό του ορθογωνίου είτε για σχεδιασμό με γέμισμα χρώματος.

Από τη στιγμή που θα ορίσουμε χρώμα για το strokeStyle οτιδήποτε σχεδιασθεί στη συνέχεια (γραμμές σχήματα κτλ.) με τη μέθοδο stroke θα έχει το ίδιο χρώμα εφόσον αυτό δεν αλλάξει με μία νέα strokeStyle. Το ίδιο ισχύει με τη μέθοδο fill και το fillStyle.

Παράδειγμα

Για την καλύτερη κατανόηση του χώρου του Canvas καθώς και των συντεταγμένων του θα σχεδιάσουμε ένα ορθογώνιο παραλληλόγραμμο στις τέσσερις γωνίες του canvas. Τα βήματα που θα ακολουθήσουμε θα είναι τα παρακάτω:

1. Δημιουργία canvas με διαστάσεις 800X800 pixels και περίγραμμα κόκκινο με πάχος 2 pixels, background-color:bisque και όνομα canvas1:

```
<canvas id="canvas1" width="800" height="800"
style="border:3px solid red;background-color:bisque">
</canvas>
```

2. Θα γράψουμε την ετικέτα <script>
3. Θα γράψουμε τις απαραίτητες γραμμές για την επιλογή του canvas και τη δημιουργία drawing object:

```
var canvas = document.getElementById("canvas1");
ctx = canvas.getContext("2d");
```

4. Για να έχουμε μπλε χρώμα γεμίσματος, θα γράψουμε

```
ctx.fillStyle = "blue";
```

5. Θα τοποθετήσουμε το πρώτο ορθογώνιο στην επάνω αριστερή γωνία του canvas. Στο ορθογώνιο θα δώσουμε διαστάσεις πλάτος Χ ύψος 200Χ150 pixels.

```
 // επάνω αριστερά
ctx.fillRect(0,0,200,150);
```

Σχόλια

a=0

b=0

Άρα οι συντεταγμένες της επάνω αριστερής γωνία του ορθογωνίου παραλληλογράμμου θα είναι (0,0).

c=200

το πλάτος θα είναι ίσο με 200 Pixels

d=150

το ύψος θα είναι ίσο με 150 Pixels.

Στα επόμενα ορθογώνια οι διαστάσεις 200Χ150 Θα είναι ίδιες. Επίσης και το χρώμα γεμίσματος θα είναι το ίδιο εφόσον δεν το αλλάξουμε.

6. Για την κάτω αριστερή γωνία θα γράψουμε

```
 // κάτω αριστερά
ctx.fillRect(0,650,200,150);
```

Σε σχέση με το προηγούμενο ορθογώνιο η τιμή α δεν άλλαξε εφόσον δεν η τετμημένη x παραμένει η ίδια. Το y όμως θα είναι όσο το μήκος του κατακόρυφου άξονα 800 μείον το ύψος του ορθογωνίου 150 pixels. Άρα

y=800-150=650

17. Για την κάτω δεξιά γωνία θα γράψουμε

```
// κάτω δεξιά
ctx.fillRect(600,650,200,150);
```

για την τιμή x θα γράψουμε το μήκος του οριζόντιου άξονα μείον το πλάτος του ορθογωνίου:

800-200=600

Για την τιμή του y θα γράψουμε το μήκος του άξονα y μείον το ύψος του ορθογωνίου.

800-150=650

8. Για την επάνω δεξιά γωνία θα γράψουμε

```
// επάνω δεξιά
ctx.fillRect(600,0,200,150);
```

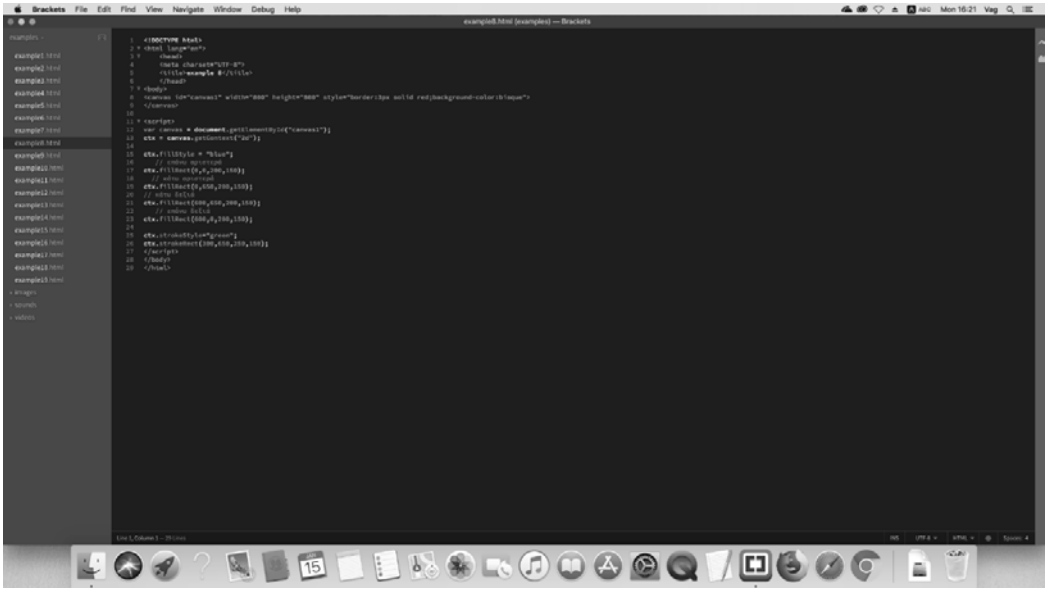
Για την τιμή x θα γράψουμε το μήκος του άξονα x μείον το πλάτος του ορθογωνίου. Για την τιμή y θα γράψουμε την τιμή 0 έτσι ώστε το ορθογώνιο να εφάπτεται στην επάνω πλευρά του canvas.

Σημείωση

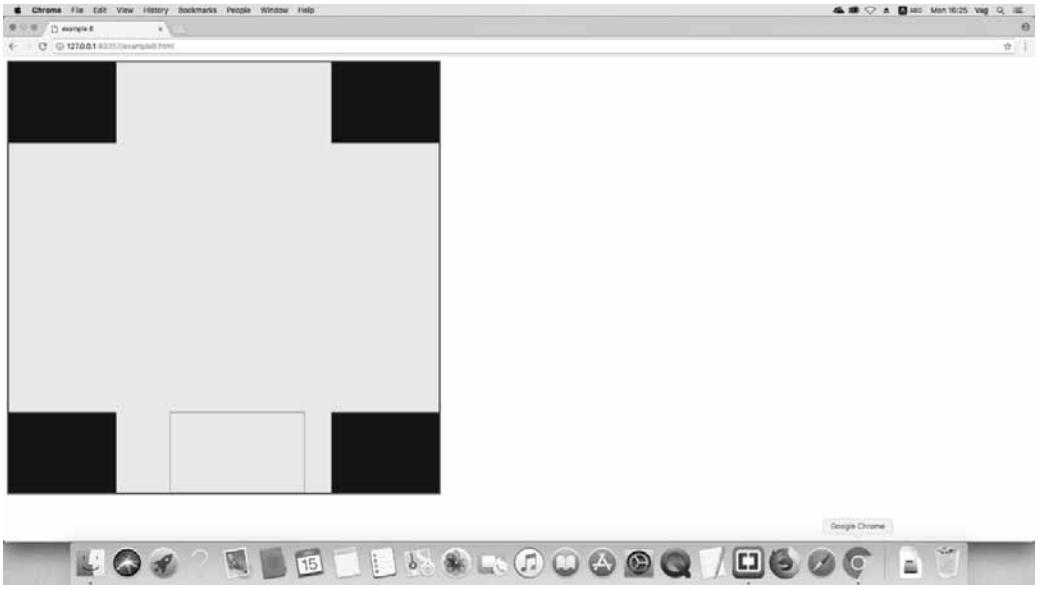
Εντός του script με δύο διαδοχικά // slash μπορούμε να γράψουμε σχόλια που τελειώνουν στο τέλος μιας γραμμής.

Παράδειγμα 8. Html Canvas- ορθογώνια

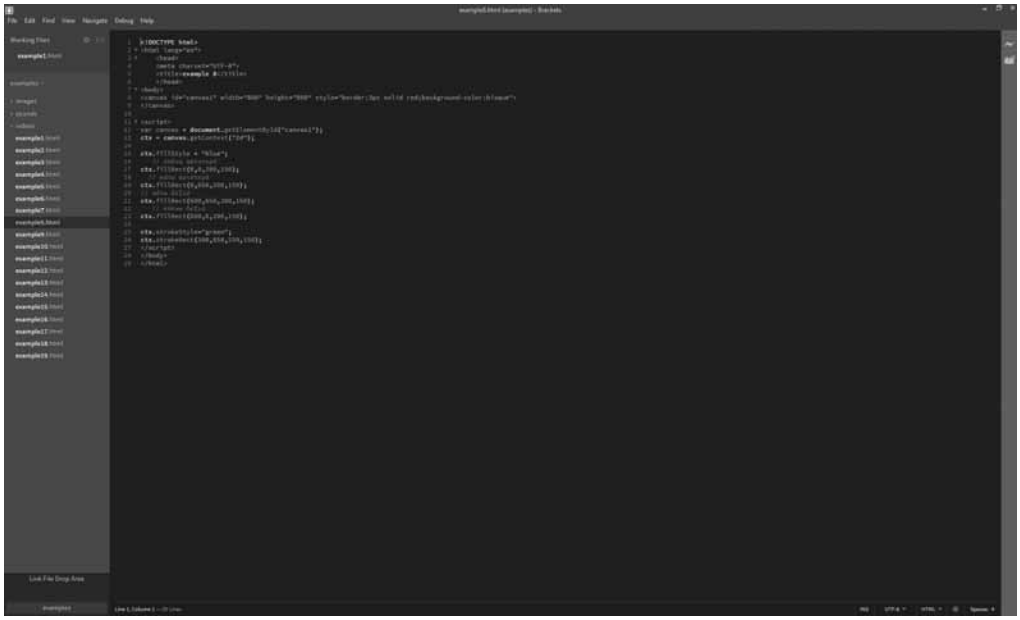
Κώδικας(Mac): Html Canvas- ορθογώνια



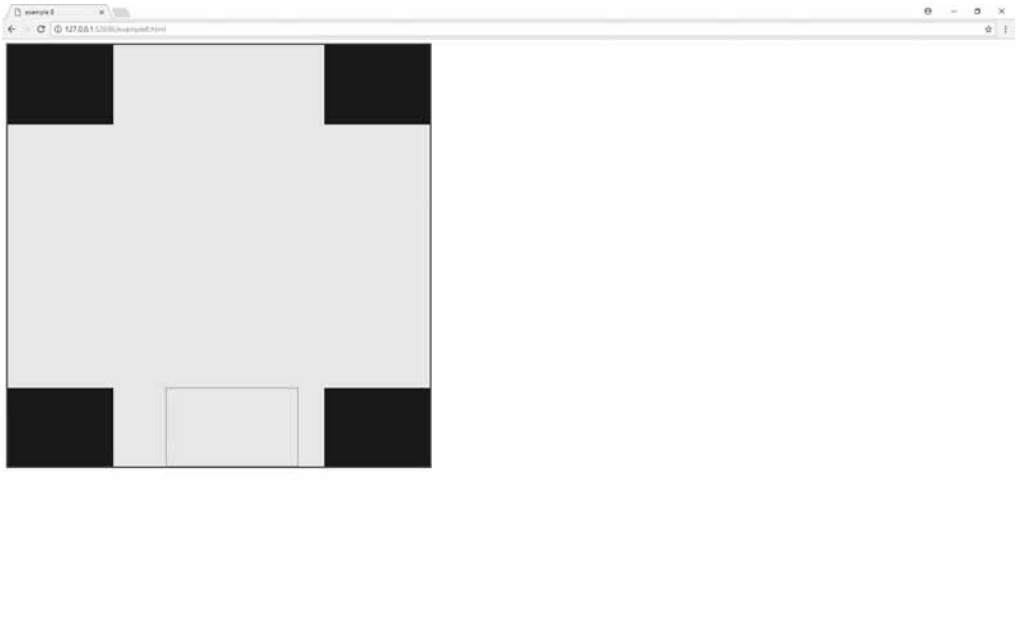
Εικόνα Browser(Mac): Html Canvas- ορθογώνια



Κώδικας(Win): Html Canvas- ορθογώνια



Εικόνα Browser(Win): Html Canvas- ορθογώνια



Σημείωση

Στην περίπτωση που χρησιμοποιούμε το Live Preview του Brackets για κάθε αλλαγή στο canvas, θα χρειάζεται πρώτα να επιλέγουμε save στο brackets και μετά να πατάμε το refresh button του browser.

Σχεδιάζοντας γραμμές

Για να σχεδιάσουμε ένα ευθύγραμμο τμήμα θα πρέπει να ορίσουμε τις συντεταγμένες της αρχής και του τέλους του ευθύγραμμου τμήματος και στη συνέχεια ενώνουμε τα δύο σημεία ορίζοντας ένα ευθύγραμμο τμήμα.

Βήματα για το σχεδιασμό γραμμών

1. Για να ξεκινήσουμε ένα καινούργιο σχεδιασμό γράφουμε

```
ctx.beginPath();
```

2. τιμή είναι το πάχος της γραμμής σε pixels

```
ctx.lineWidth = τιμή;
```

3. Επιλέγουμε το χρώμα για τη σχεδίαση της γραμμής:

```
ctx.strokeStyle = "…";
```

4. Γράφουμε τις συντεταγμένες του ενός άκρου της γραμμής:

```
ctx.moveTo(_,_);
```

5. Γράφουμε τις συντεταγμένες του άλλου άκρου της γραμμής:

```
ctx.lineTo(_,_);
```

6. Σχεδιάζουμε τη γραμμή:

```
ctx.stroke();
```

lineCap

Με την ιδιότητα lineCap μπορούμε να ορίσουμε το σχήμα στις άκρες μια γραμμής σύμφωνα με την παρακάτω σύνταξη:

```
lineCap = "τιμή";
```

Όπου τιμή μπορούμε να γράψουμε μία από τις παρακάτω:

- butt → είναι η προεπιλογή.
- round → για στρογγυλεμένες άκρες
- square → προσθέτει τετράγωνα στις άκρες των γραμμών

Παράδειγμα

```
    ctx.strokeStyle = " red";
    ctx.beginPath();
 ctx.lineWidth=15;
 ctx.lineCap = "butt";
 ctx.moveTo(700, 300);
 ctx.lineTo(700, 600);
 ctx.stroke();
```

Σχόλια

Έχουμε ορίσει τελείωμα στις άκρες της γραμμής butt. Στη συνέχεια μετακινήσαμε τη γραφίδα στο σημείο (700,300) και ορίσαμε για επόμενο σημείο της γραμμής το (700, 600). Με τη μέθοδο stroke ενώσαμε τα δύο σημεία.

Τεθλασμένη γραμμή

Μία τεθλασμένη γραμμή αποτελείται από συνεχόμενα ευθύγραμμα τμήματα όπου στο σημείο που τελειώνει ένα ευθύγραμμο τμήμα συμπίπτει με το σημείο που αρχίζει το επόμενο, εκτός από το τελευταίο ευθύγραμμο τμήμα της τεθλασμένης γραμμής.

Για να σχεδιάσουμε μία τεθλασμένη γραμμή θα κάνουμε τα παρακάτω βήματα:

1. Αρχικά θα τοποθετήσουμε τη γραφίδα του canvas στο πρώτο σημείο, εκεί που θα ξεκινάει το πρώτο ευθύγραμμο τμήμα της τεθλασμένης γραμμής με

```
ctx.moveTo(_,_);
```

2. Κάθε επόμενο σημείο της γραμμής θα το ορίζουμε με

```
ctx.lineTo(_,_);
```

3. Στο τέλος θα σχεδιάζουμε την τεθλασμένη γραμμή με

```
ctx.stroke();
```

Πιο αναλυτικά, τα βήματα για τη σχεδίαση της τεθλασμένης γραμμής είναι τα παρακάτω:

1. Ξεκινάμε τη σχεδίαση γράφοντας

```
ctx.beginPath();
```

2. Όπου τιμή είναι το πάχος της γραμμής σε pixels

```
ctx.lineWidth = τιμή;
```

3. Επιλέγουμε το χρώμα για τη σχεδίαση της γραμμής:

```
ctx.strokeStyle = "…";
```

4. Τοποθετούμε τη γραφίδα του Canvas στο πρώτο σημείο της τεθλασμένης γραμμής γράφοντας

```
ctx.moveTo(_,_);
```

5. Για τα υπόλοιπα σημεία εκτός από το πρώτο τα ορίζουμε γράφοντας

```
ctx.lineTo(_, _);
```

Τα σημεία στον κώδικα θα γραφούν με η σειρά που θα τα ενώσει η γραφίδα.

6. Για να σχεδιαστεί η τεθλασμένη γραμμή γράφουμε

```
ctx.stroke();
```

beginPath()

Γράφοντας

```
ctx.beginPath();
```

ξεκινάμε ένα καινούργιο μονοπάτι σχεδίασης ή κάνουμε reset-επαναφορά σε ένα μονοπάτι που υπάρχει ήδη. Κάθε φορά που ξεκινάμε εκ νέου μια γραμμή θα γράφουμε

```
ctx.beginPath();
```

Παράδειγμα

Θα δημιουργήσουμε ένα στοιχείο canvas με διαστάσεις 800X800:

```
<canvas id="canvas1" width="800" height="800"
style="border:3px solid red;background-color: #55ff00">
</canvas>
```

Θα αποκτήσουμε πρόσβαση στο στοιχείο canvas με id canvas1 και θα δημιουργήσουμε ένα αντικείμενο σχεδίασης:

```
<script>
var canvas = document.getElementById("canvas1");
ctx = canvas.getContext("2d");
ctx.beginPath();
ctx.strokeStyle = "blue";
ctx.lineWidth=5;
```

Εφόσον είμαστε έτοιμοι να σχεδιάσουμε τη γραμμή θα τοποθετήσουμε τη γραφίδα στην αρχή της τεθλασμένης γραμμής και πιο συγκεκριμένα στο σημείο (100,400)

```
ctx.moveTo(100, 400);
```

Έπειτα θα ορίσουμε τα υπόλοιπο σημεία της τεθλασμένης γραμμής με τη σειρά που θα ενωθούν

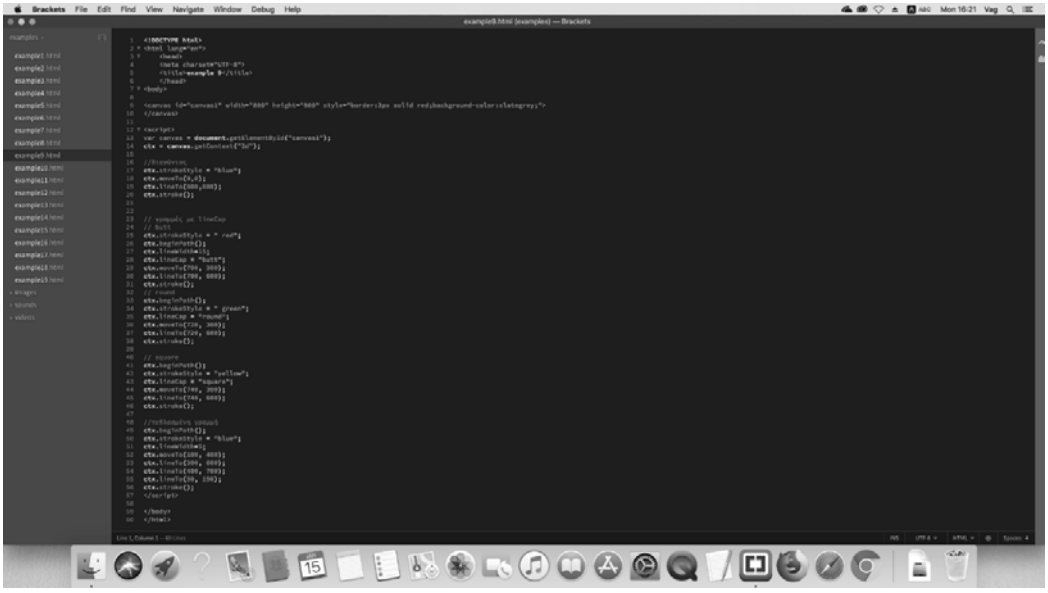
```
ctx.lineTo(300, 600);
ctx.lineTo(50, 150);
```

Τέλος γράφουμε την παρακάτω γραμμή για να ενώσει η γραφίδα όλα τα σημεία.

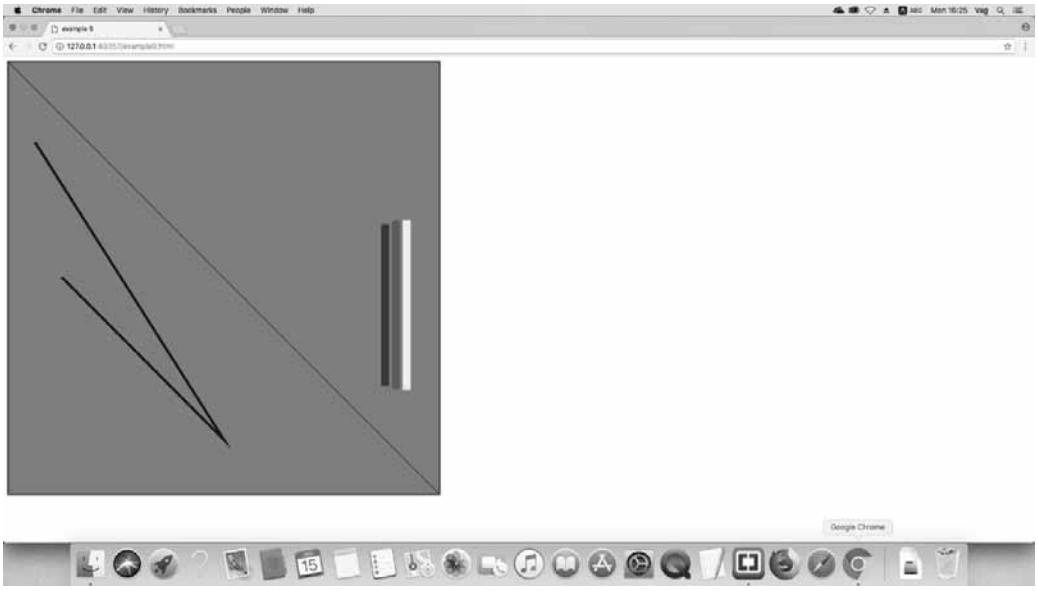
```
ctx.stroke();
```

Παράδειγμα 9. Html Canvas- lines

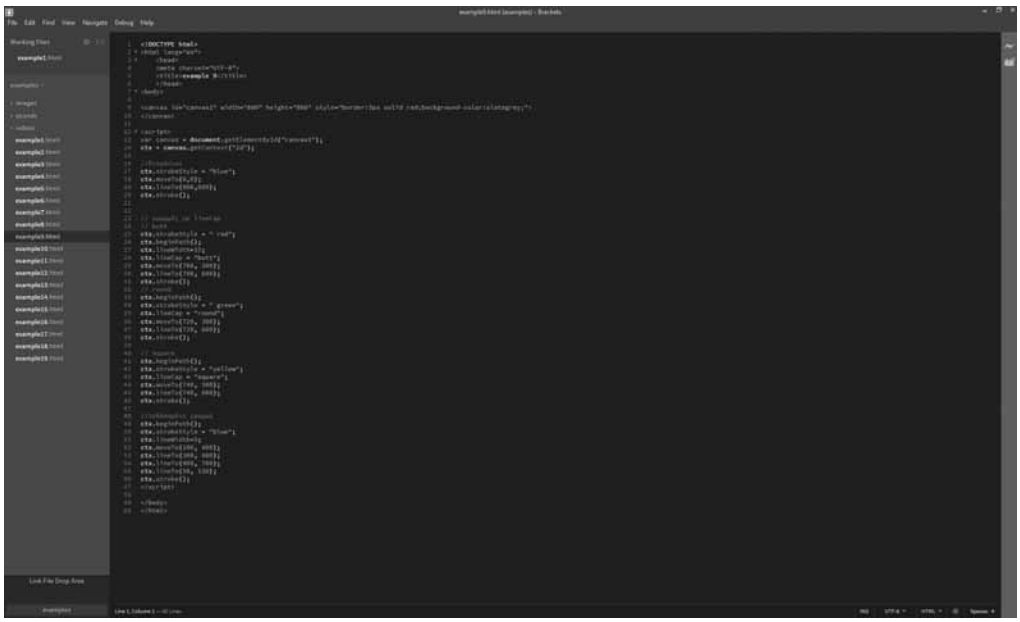
Κώδικας(Mac): Html Canvas- lines



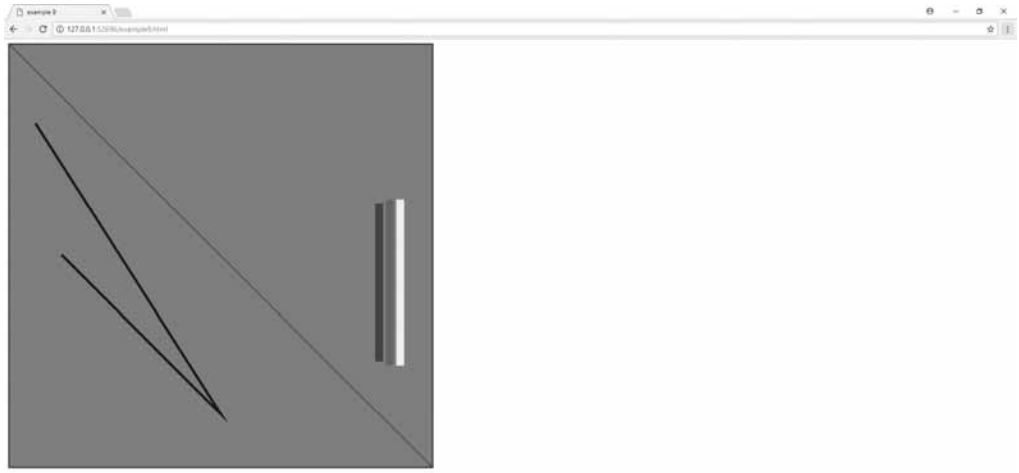
Εικόνα Browser(Mac): Html Canvas- lines



Κώδικας(Win): Html Canvas- lines



Εικόνα Browser(Win): Html Canvas- lines



ΥΠΟΕΝΟΤΗΤΑ 3. ΚΑΜΠΥΛΕΣ ΣΤΟ HTML CANVAS ΚΑΙ LINKS

3.1 Γράφοντας τόξα και κύκλους (μέθοδος arc)

Για να σχεδιάσουμε κύκλο θα χρειαστούμε τις μονάδες rad (ακτίνια). Ολόκληρη η περιφέρεια κύκλου (360°) χωρίζεται σε 2π (περίπου 6.28) ακτίνια.

Σχεδιάζοντας ένα κύκλο ή γενικά ένα τόξο κύκλου στο canvas μπορούμε να έχουμε στο μυαλό μας ότι κρατάμε ένα διαβήτη. Αρχικά ρυθμίζουμε την απόσταση των άκρων του διαβήτη έτσι ώστε να ορίσουμε την ακτίνα του κύκλου. Στη συνέχεια τοποθετούμε τη μύτη του διαβήτη στο κέντρο του κύκλου ή του τόξου.

Για το σχεδιασμό τοποθετούμε τη γραφίδα του διαβήτη στο σημείο που ξεκινάει το τόξο και ανάλογα με τη φορά σχεδιάζουμε μέχρι το σημείο τερματισμού του τόξου.

Για να σχεδιάσουμε ένα τόξο η μία περιφέρεια κύκλου θα πρέπει να γράψουμε:

ctx.arc(x,y,a,b,c,d);
x-> Τετμημένη x του κέντρου του κύκλου ή του τόξου
y-> Τεταγμένη y του κέντρου του κύκλου ή του τόξου
a→ακτίνα σε pixels
b→ γωνία σε rad έναρξης του τόξου
c→ γωνία σε rad τερματισμού του τόξου
d→ Φορά σχεδιασμού τόξου.

Η παράμετρος d είναι προαιρετική. Αποδεκτές τιμές είναι:

- false(προεπιλογή) → σύμφωνα με τη φορά των δεικτών του ρολογιού
- true → αντίθετα με τη φορά των δεικτών του ρολογιού

Παράδειγμα κύκλου

Θα σχεδιάσουμε μια περιφέρεια κύκλου και 3 κυκλικούς δίσκους (με χρώμα γεμίσματος όπου θα εφάπτονται στις τέσσερις γωνίες ενός στοιχείου canvas με διατάσεις 800X800 Pixels.

Αφού γράψουμε τις ετικέτες για τη δημιουργία του html canvas και τις αντίστοιχες γραμμές για πρόσβασή στο canvas:

```
<script>
var canvas = document.getElementById("canvas1");
ctx = canvas.getContext("2d");
```

Για να ξεκινήσουμε νέο σχεδιασμό θα γράψουμε

```
ctx.beginPath();
```

Για την επιλογή χρώματος σχεδιασμού θα γράψουμε

```
ctx.strokeStyle = "blue";
```

Αμέσως μετά θα γράψουμε την παρακάτω γραμμή κώδικα για να ορίσουμε το πάχος της περιφέρειας κύκλου στα 5 pixels:

```
ctx.lineWidth=5;
```

Με την παρακάτω γραμμή θα σχεδιάσουμε κύκλο που να εφάπτεται στην επάνω αριστερή γωνία του canvas

```
ctx.arc(50,50,50,0,2*Math.PI);
```

Οι δύο πρώτες τιμές 50,50 προσδιορίζουν το κέντρο του κύκλου. Οι συντεταγμένες του κέντρου θα είναι (50,50). Εφόσον και η ακτίνα (η τρίτη τιμή είναι ίση με 50) ο κύκλος θα εφάπτεται στην επάνω αριστερή γωνία.
Οι δύο τελευταίες τιμές 0 και 2*Math.PI προσδιορίζουν ένα πλήρη κύκλο.
Για το δεύτερο κύκλο θα γράψουμε:

```
ctx.beginPath();
ctx.fillStyle = "yellow";
ctx.lineWidth=5;
ctx.arc(50,750,50,0,2*Math.PI);
ctx.fill();
```

Σχόλια

Ο συγκεκριμένος κύκλος θα έχει

- κέντρο με συντεταγμένες (50,750)
- ακτίνα 50 pixels
- γωνία έναρξης 0 rad
- γωνία τερματισμού 2*Math.PI rad
- συμπαγές γέμισμα χρώματος κίτρινου

ο τρίτος κύκλος έχει γραφεί με τις ίδιες γραμμές κώδικα μόνο που αλλάζει το χρώμα γεμίσματος και η μέθοδος arc

```
ctx.arc(750,750,50,0,2*Math.PI);
```

Τοποθετήσαμε το κέντρο του κύκλου στο σημείο (750,750) έτσι ώστε εφόσον έχει ακτίνα με μήκος 5 pixels να εφάπτεται στην κάτω δεξιά γωνία του canvas.

Ο τέταρτος κύκλος θα έχει κέντρο με συντεταγμένες (750,50) και ακτίνα 50 για να εφάπτεται στη επάνω δεξιά γωνία του canvas

```
ctx.arc(750,50,50,0,2*Math.PI);
```

Παράδειγμα σχεδίασης τόξου-τεταρτοκυκλίου

Θα δημιουργήσουμε ένα τεταρτοκύκλιο που θα έχει κέντρο την κάτω αριστερή γωνία (0,800) και ακτίνα 800 έτσι ώστε να περνάει από το σημείο (0,0) και το σημείο (800,800).

Αυτό που έχει ενδιαφέρον είναι οι παράμετροι για τη μέθοδο arc:

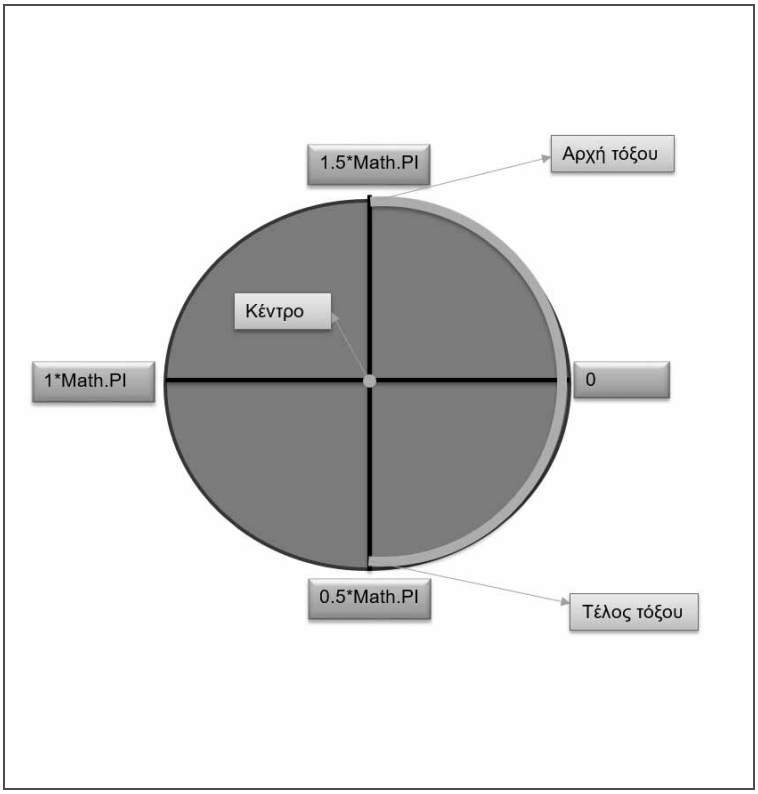
```
ctx.arc(0,800,800,1.5*Math.PI,0,false) ;
```

Σχόλια

εφόσον έχουμε γράψει false που είναι και η προεπιλεγμένη τιμή η φορά σχεδιασμού του τόξου θα είναι σύμφωνη με αυτή των δεικτών του ρολογιού.

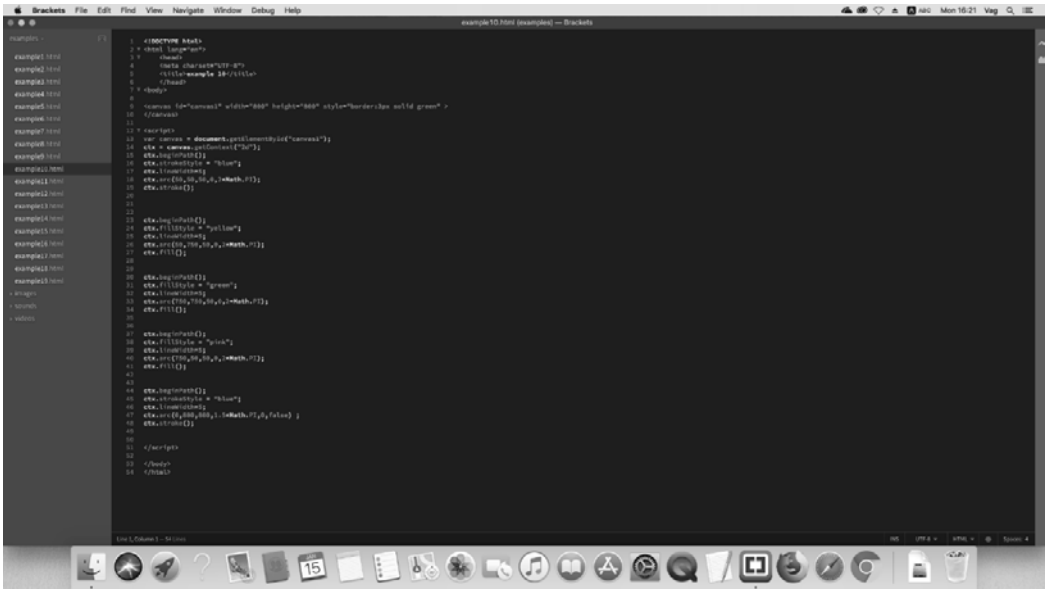
Οπότε το τόξο θα πρέπει να έχει γωνία έναρξης το 1.5*Math.PI και γωνία τερματισμού το 0.

Εικόνα 4. Μέθοδος Arc

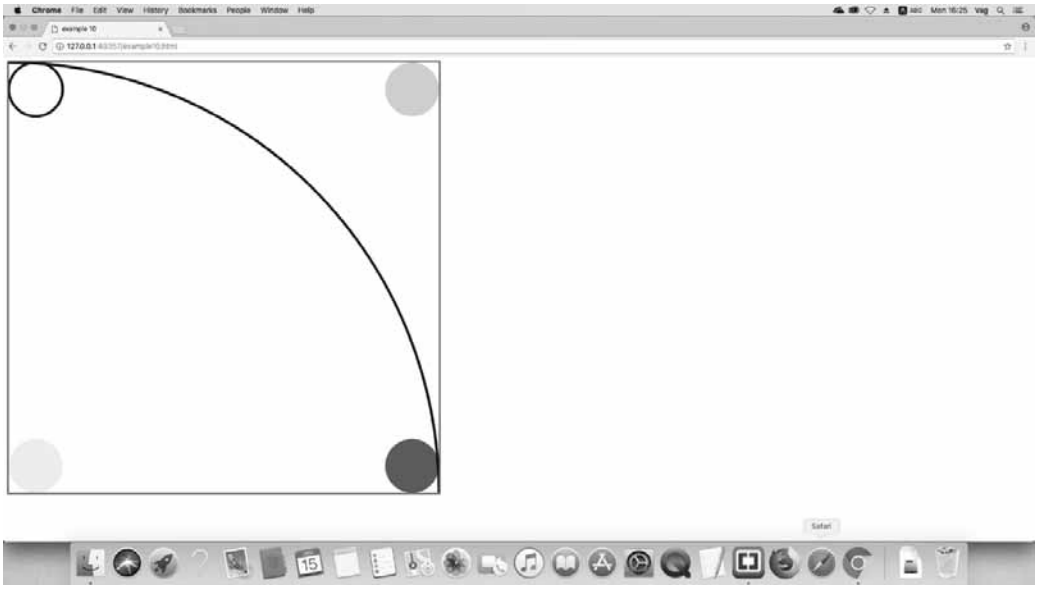


Παράδειγμα 10. Arc

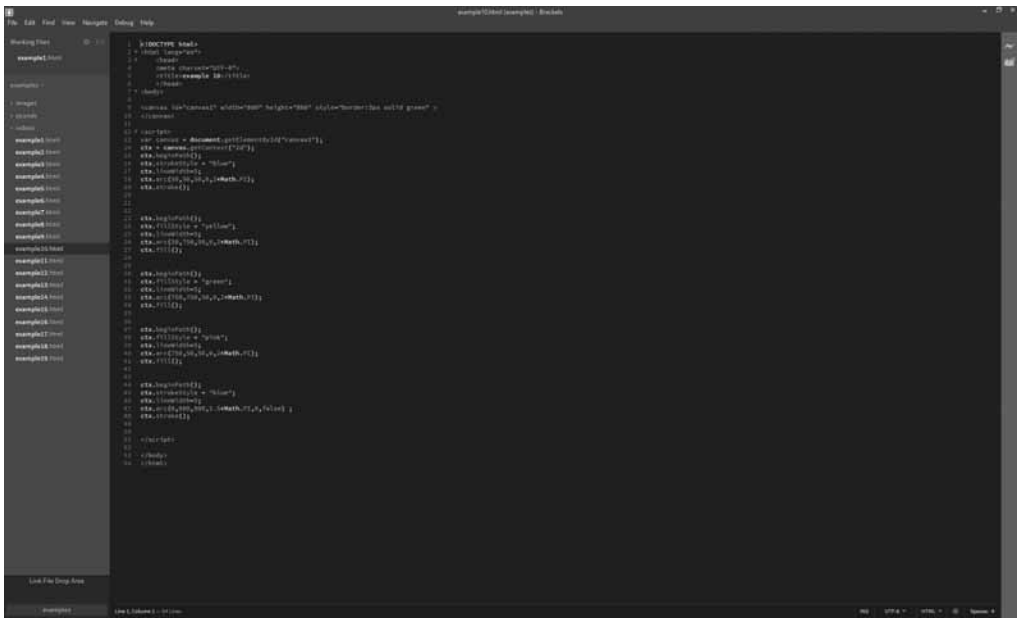
Κώδικας(Mac): arc



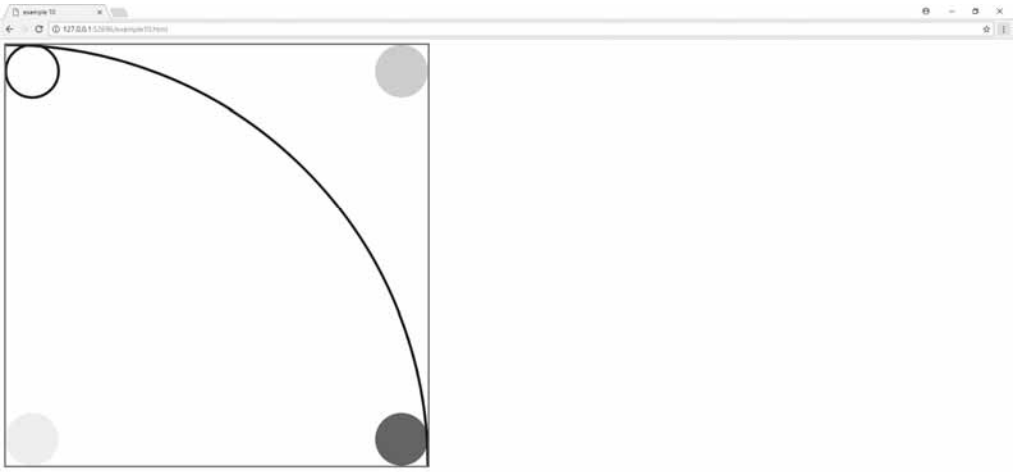
Εικόνα Browser(Mac): arc



Κώδικας(Win): arc



Εικόνα Browser(Win): arc



Σχεδιάζοντας καμπύλες στο Html Canvas

Για να σχεδιάσουμε καμπύλες στο Html Canvas έχουμε τις εξής επιλογές:

1. Καμπύλη με ένα control point την οποία σχεδιάζουμε με τη μέθοδο quadraticCurveTo()
2. Καμπύλη με δύο controls Points τα οποία σχεδιάζουμε με τη μέθοδο bezierCurveTo()

3.2 Καμπύλες, gradient color και Text

quadraticCurveTo()

Για την καμπύλη **quadraticCurve** χρειάζεται να ορίσουμε τρία σημεία:

quadraticCurve

1. Start Point: είναι το σημείο που ξεκινά η καμπύλη. Αφού γράψουμε: ctx.beginPath(); για να ξεκινήσουμε νέα σχεδίαση μετακινώντας τη γραφίδα γράφοντας ctx.moveTo(_,_) ορίζουμε και το σημείο έναρξης της καμπύλης.
2. Control point: είναι το σημείο που ελέγχει την καμπυλότητα της γραμμής και από το σημείο αυτό διέρχονται οι δύο εφαπτομένες στα σημεία καμπύλωσης της γραμμής.
3. End Point: είναι το σημείο τερματισμού της καμπύλης

Πιο αναλυτικά τα βήματα για το σχεδιασμό **quadraticCurve** είναι τα παρακάτω:

1. Για να ξεκινήσουμε μία νέα σχεδίαση γράφουμε

```
ctx.beginPath();
```

2. Για το χρώμα σχεδίασης γράφουμε

```
ctx.strokeStyle = "_";
```

3. Για το πάχος της καμπύλης γράφουμε

```
ctx.lineWidth=5;
```

4. Για να μετακινήσουμε τη γραφίδα στο start point γράφουμε

```
ctx.moveTo(_,_)
```

5. Για να ορίσουμε τη μέθοδο της καμπύλης:

```
ctx.quadraticCurveTo(a,b,c,d);
```

Οι παράμετροι a,b αντιστοιχούν στις συντεταγμένες x,y του σημείου Control Point. Οι παράμετροι c,d αντιστοιχούν στις συντεταγμένες του σημείου τερματισμού της καμπύλης.

6. Για να σχεδιαστεί η καμπύλη γράφουμε:

```
ctx.stroke();
```

quadraticCurveTo(a,b,c,d)
a→ τετμημένη x του σημείου Control Point
b→ τεταγμένη y του σημείου Control Point
c→ τετμημένη x του σημείου End Point
d→ τεταγμένη y του σημείου End Point

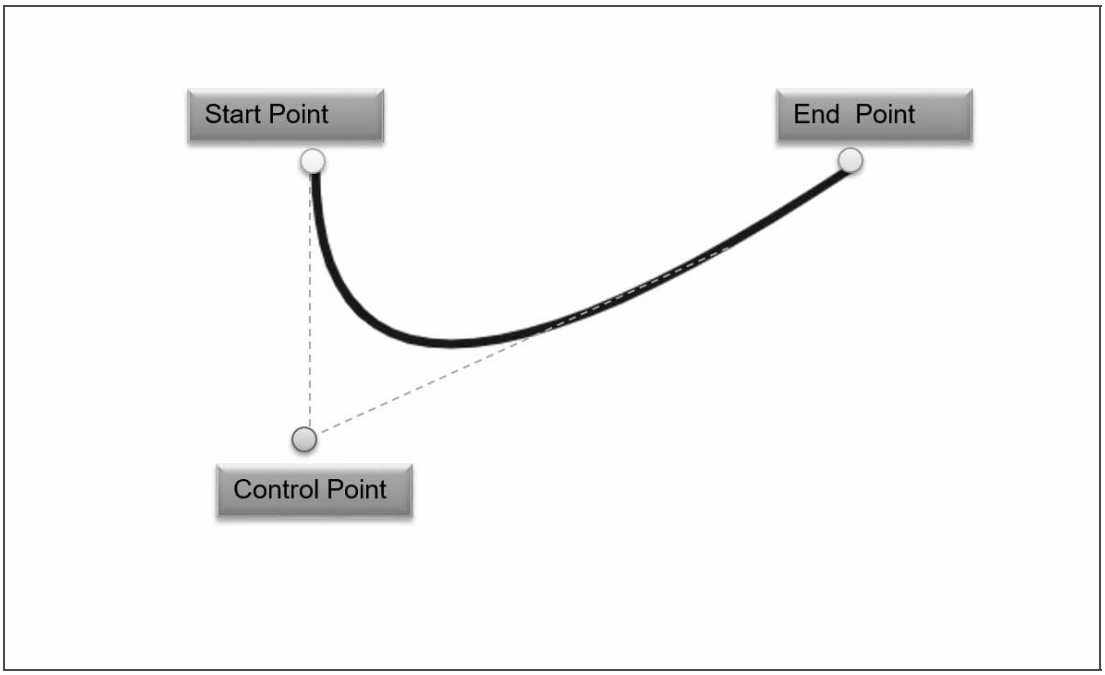
Παράδειγμα

```
ctx.beginPath();
ctx.moveTo(200,100);
ctx.quadraticCurveTo(200,300,500,100);
ctx.stroke();
```

Σχόλια

έχουμε ορίσει καμπύλη με σημείο εκκίνησης το σημείο (200,100), σημείο τερματισμού (500,100) και control point με συντεταγμένες (200,300)

Εικόνα 5. quadraticCurveTo



bezierCurveTo()

Για το σχεδιασμό της καμπύλης **bezierCurve** χρειάζεται να ορίσουμε τέσσερα σημεία:

1. Start Point: σημείο έναρξης σχεδιασμού της καμπύλης.
2. Control point 1: το πρώτο control point ης καμπύλης
3. Control point 2: το δεύτερο control point ης καμπύλης
4. End Point: το σημείο τερματισμού της καμπύλης

Πιο αναλυτικά τα βήματα για το σχεδιασμό **bezierCurve** είναι τα παρακάτω:

1. Για να ξεκινήσουμε καινούργια σχεδίαση γράφουμε

```
ctx.beginPath();
```

2. Για το χρώμα σχεδίασης γράφουμε

```
ctx.strokeStyle = "_";
```

3. Για το πάχος της γραμμής της καμπύλης γράφουμε

```
ctx.lineWidth=5;
```

4. Για να μετακινήσουμε τη γραφίδα στο start point γράφουμε

```
ctx.moveTo(_,_)
```

5. Για να ορίσουμε τη μέθοδο της καμπύλης:

```
ctx.bezierCurveTo (a,b,c,d,e,f);
```

όπου a,b είναι οι συντεταγμένες για το Control Point 1 c,d είναι οι συντεταγμένες για το Control Point 2 και e,f είναι οι συντεταγμένες για το σημείο τερματισμού.

6. Για να σχεδιαστεί η καμπύλη γράφουμε:

```
ctx.stroke();
```

bezierCurveTo (a,b,c,d,e,f);
a→ τετμημένη χ του σημείου Control Point 1
b→ τεταγμένη y του σημείου Control Point 1
c→ τετμημένη χ του σημείου Control Point 2
d→ τεταγμένη y του σημείου Control Point 2
e→ τετμημένη χ του σημείου End Point
f→ τεταγμένη y του σημείου End Point

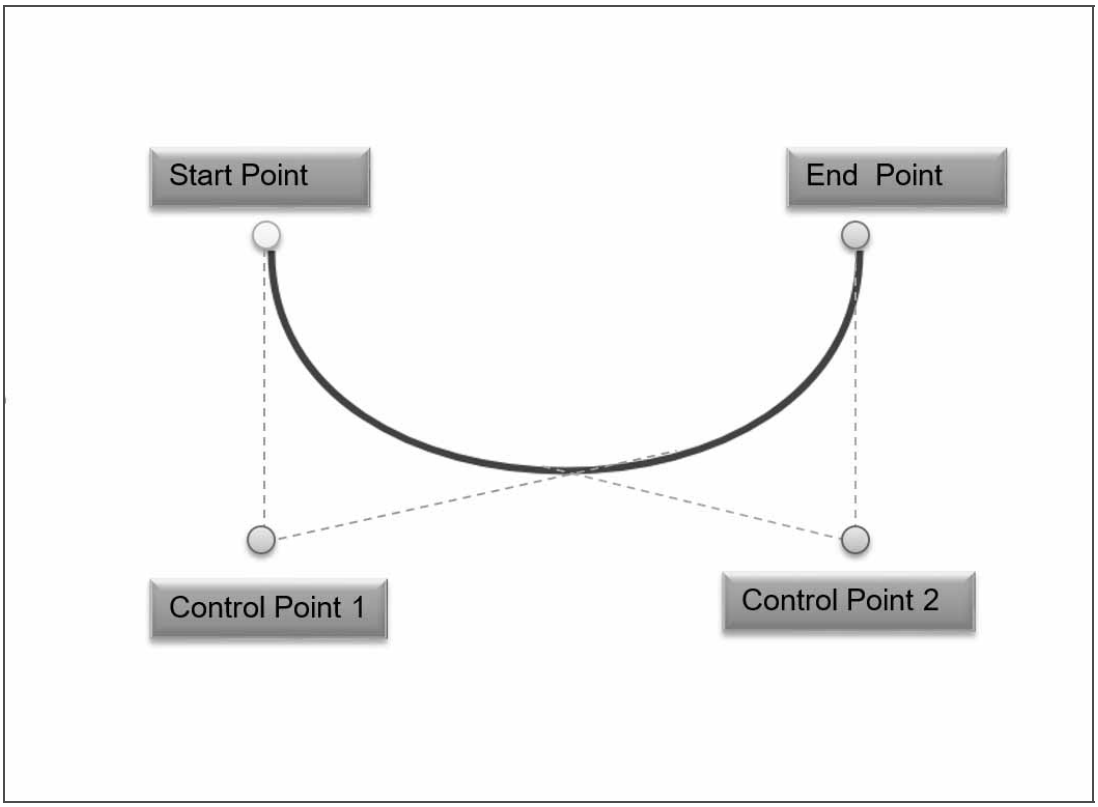
Παράδειγμα

```
ctx.beginPath();
ctx.strokeStyle = "red";
ctx.moveTo(200,400);
ctx.bezierCurveTo(200,600,600,600,600,400);
ctx.stroke();
```

Σχόλια

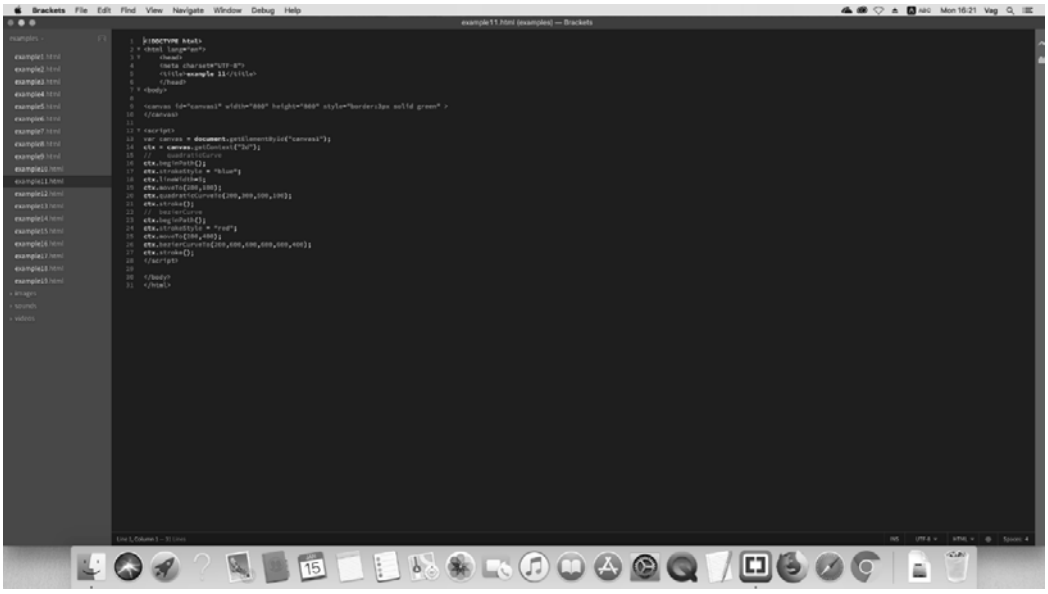
Έχουμε σχεδιάσει μια καμπύλη bezierCurve με σημείο εκκίνησης το σημείο (200,400), συντεταγμένες για το control Point 1 (200,600), συντεταγμένες για το control Point 2 (600,600) και τέλος για το end point έχουμε ορίσει συντεταγμένες (600,400).

Εικόνα 6. bezierCurveTo()

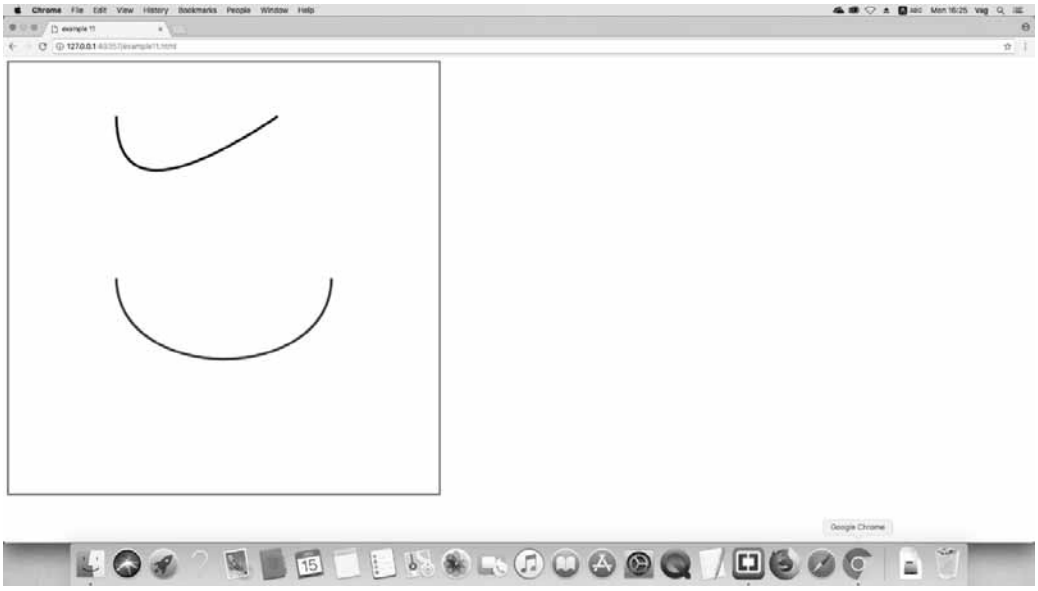


Παράδειγμα 11. Καμπύλες στο Html Canvas

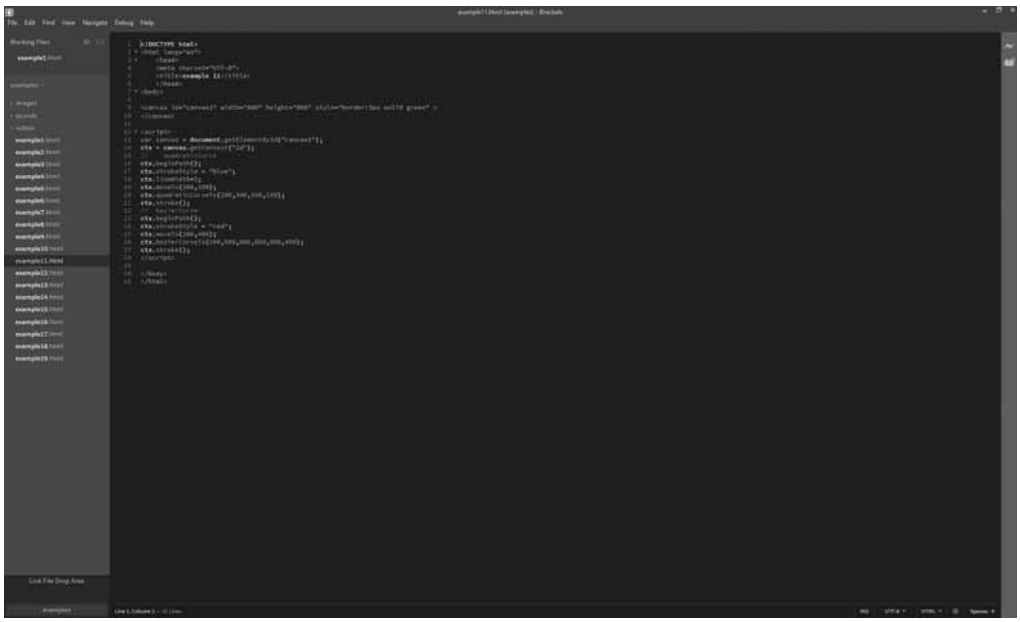
Κώδικας(Mac): Καμπύλες στο Html Canvas



Εικόνα Browser(Mac): Καμπύλες στο Html Canvas



Κώδικας(Win): Καμπύλες στο Html Canvas



Εικόνα Browser(Win): Καμπύλες στο Html Canvas

Gradient color

Το Html canvas μας δίνει τη δυνατότητα να δημιουργήσουμε gradient colors. Το gradient color εφαρμόζεται σε μια περιοχή που έχουμε ορίσει και συνιστά βαθμιαία μετάβαση από ένα χρώμα σε ένα άλλο. Παρόμοια λειτουργία εμπεριέχουν και εφαρμογές τύπου office.

Για να ορίσουμε μία περιοχή που θα εφαρμόζεται gradient color θα πρέπει να κάνουμε τα παρακάτω βήματα:

1. Για να ορίσουμε την gradient περιοχή γράφουμε:

```
var gradient = ctx.createLinearGradient(a,b,c,d);
```

Οι παράμετροι a,b ορίζουν τις συντεταγμένες (x,y) του σημείου έναρξης του gradient, ενώ οι παράμετροι c,d ορίζουν τις συντεταγμένες x,y του σημείου τερματισμού του gradient.

2. Για κάθε χρώμα που συμμετέχει στο gradient color γράφουμε

```
gradient.addColorStop("αριθμός από 0-1", "τιμή χρώματος");
```

Σε κάθε χρώμα γράφουμε ένα αριθμό από 0-1 για να δηλώσουμε τη θέση του χρώματος σε σχέση με το gradient. Το 0 αντιστοιχεί στην αρχή του gradient ενώ το 1 στο τέλος του Gradient.

3. Για σχεδίαση με γέμισμα χρώματος γράφουμε

```
ctx.fillStyle=gradient;
```

4. Για σχεδίαση χωρίς γέμισμα γράφουμε

```
ctx.strokeStyle=gradient;
```

Παράδειγμα

```
// Create gradient
var gradient = ctx.createLinearGradient(0,0,800,800);
```

Σχόλια

Ορίσαμε σημείο έναρξης του gradient την επάνω αριστερή γωνία του Canvas (0,0) και σημείο τερματισμού το σημείο (800,800) δηλαδή την κάτω αριστερή γωνία.

```
gradient.addColorStop("0", "orange");
gradient.addColorStop("0.3", "purple");
gradient.addColorStop("0.7", "blue");
gradient.addColorStop("1.0", "red");
```

Σχόλια

Έχουμε προσθέσει τέσσερα χρώματα τα οποία συμμετέχουν στο gradient. Εφόσον τα σημεία έναρξης και τερματισμού του Gradients ταυτίζονται με τα άκρα της κύριας διαγωνίου του ορθογωνίου η κλιμάκωση του χρώματος θα γίνει επάνω στην κύρια διαγώνιο και σε διάταξη κάθετων ευθειών σε σχέση με την κύρια διαγώνιο.

Μπορούσαμε να ορίσουμε το δεύτερο σημείο στην επάνω δεξιά γωνία και να είχαμε κλιμάκωση από αριστερά προς τα δεξιά.

Στη συνέχεια θα ορίσουμε ένα ορθογώνιο που θα καταλαμβάνει ολόκληρη την περιοχή του Canvas. Για το ορθογώνιο θα ορίσουμε την τιμή gradient για παράμετρο της μεθόδου fillStyle.

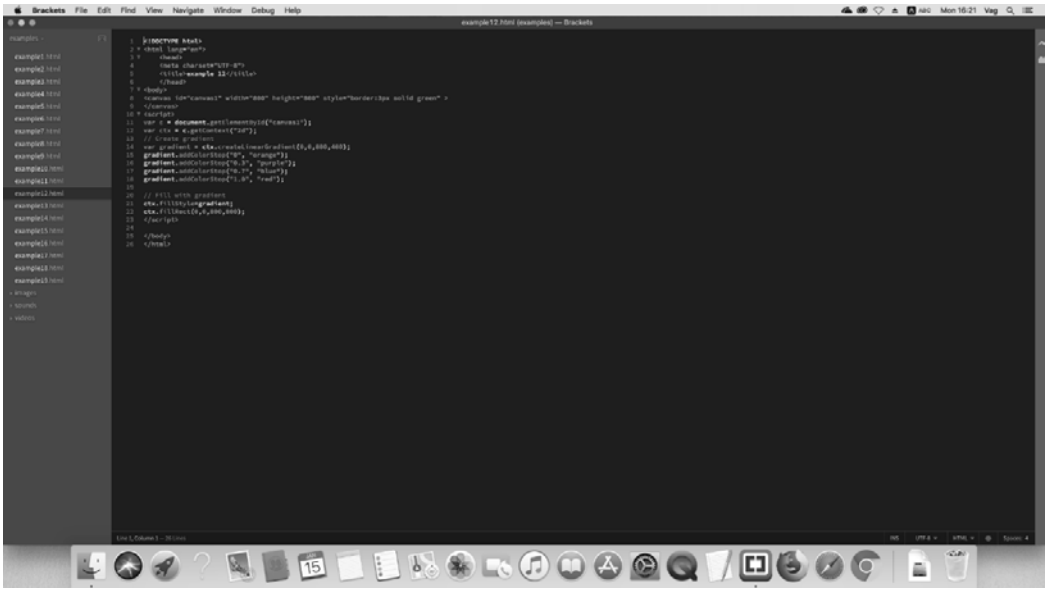
```
ctx.fillStyle=gradient;

ctx.fillRect(0,0,800,800);
```

Σχόλια

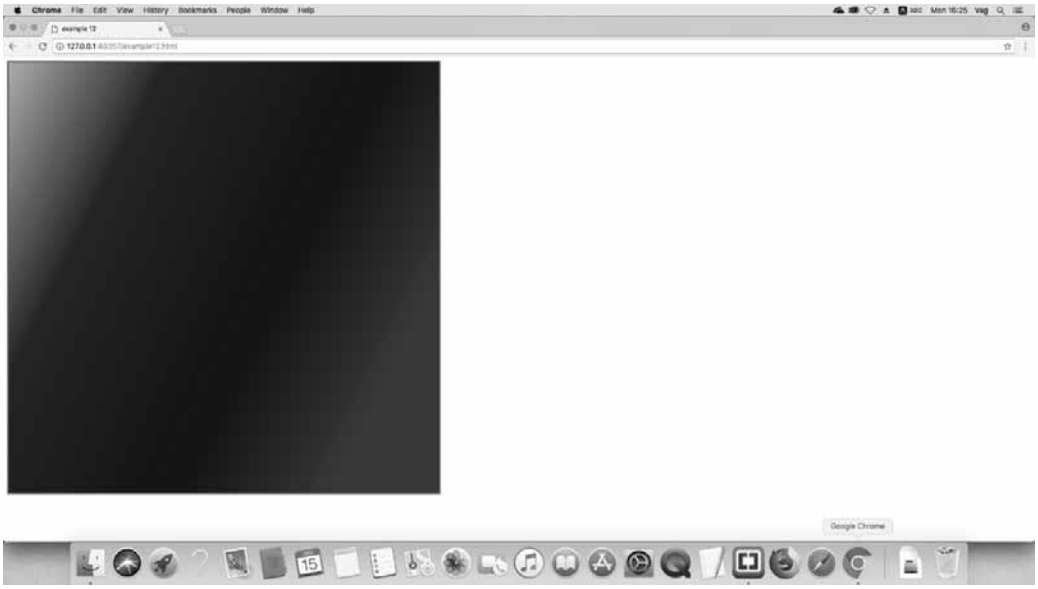
Το ορθογώνιο έχει συντεταγμένες της επάνω αριστερής γωνίας (0,0) και διαστάσεις 800X800, οπότε ταυτίζεται με το Canvas.

Παράδειγμα 12. Gradient color

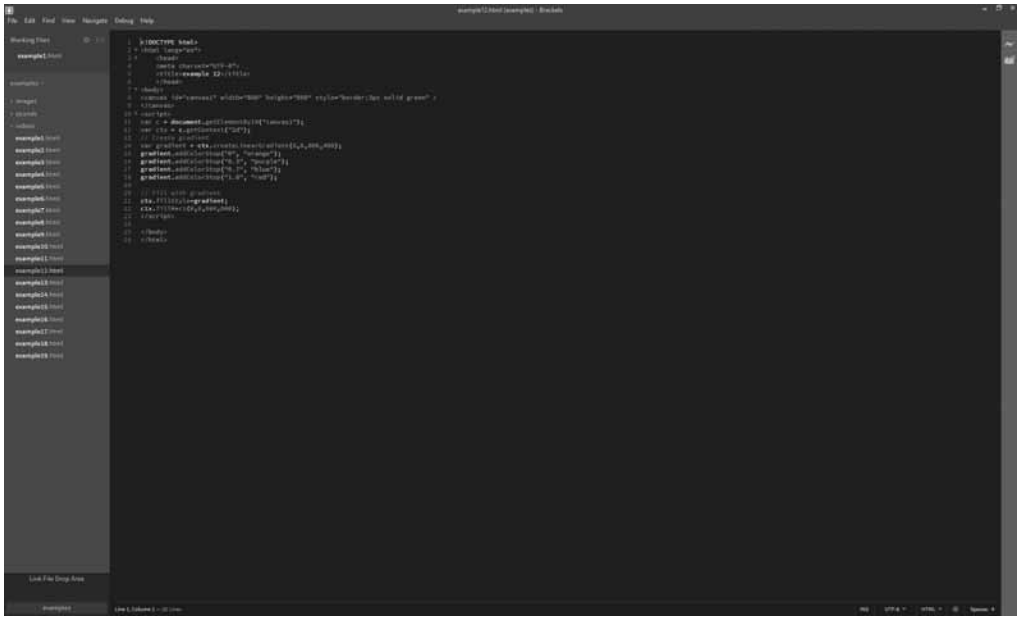
Κώδικας(Mac): Gradient color



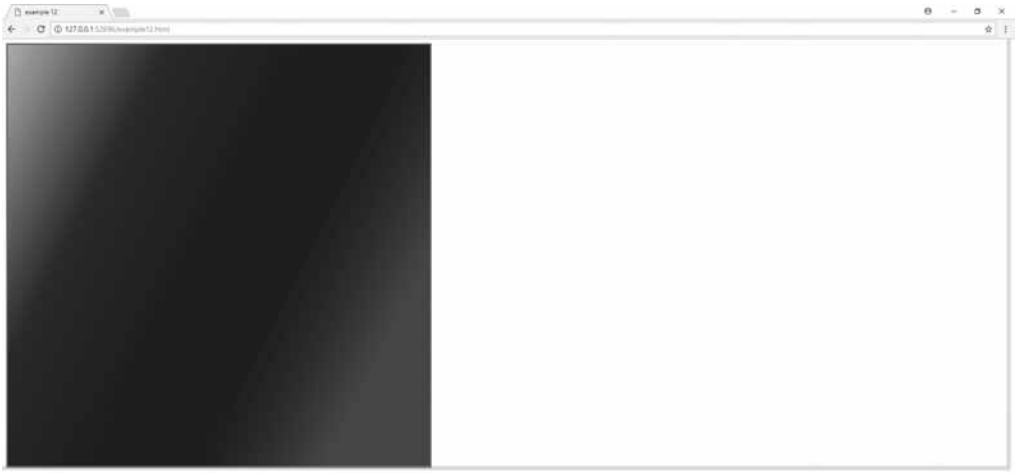
Εικόνα Browser(Mac): Gradient color



Κώδικας(Win): Gradient color



Εικόνα Browser(Win): Gradient color



Εισαγωγή κειμένου στο html canvas

Για να εισάγουμε κείμενο στο html Canvas θα πρέπει να χρησιμοποιήσουμε τη μέθοδο

```
ctx.strokeText(“  text  “, x,y,max)
```

Για να προσθέσουμε κείμενο στο canvas θα πρέπει να κάνουμε τα παρακάτω βήματα:

1. Για να επιλέξουμε το χρώμα του κειμένου γράφουμε

```
ctx.strokeStyle = "…";
```

2. Για να επιλέξουμε γραμματοσειρά και μέγεθος γράφουμε

```
ctx.font="…..";
```

3. Για να εμφανιστεί το κείμενο γράφουμε

```
ctx.strokeText(“  text  “, x,y,max)
```

Ανάμεσα στα διπλά εισαγωγικά θα γράψουμε το κείμενο ενώ οι παράμετροι x,y δηλώνουν τις συντεταγμένες που θα ξεκινάει η εμφάνιση του κειμένου. Τέλος, η παράμετρος max η οποία είναι και προαιρετική δηλώνει το μέγιστο πλάτος του κειμένου.

Σύνταξη

ctx.strokeText(“ text “, x,y,max)
text→ κείμενο που θα εμφανιστεί
x→ τετμημένη της αρχής του κειμένου
y→ τεταγμένη της αρχής του κειμένου
max(προαιρετική παράμετρος) → μέγιστο πλάτος κειμένου

Παράδειγμα

```
ctx.strokeStyle = "orange";

ctx.font="25px San Serif";

ctx.strokeText("Text in Canvas",300,350,30);
```

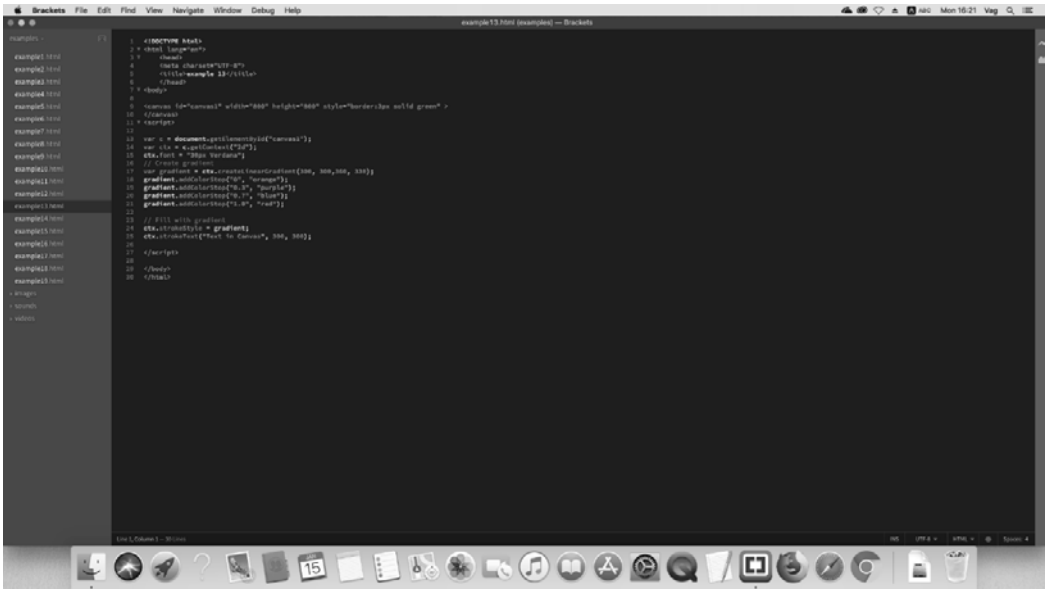
Μπορούμε για τιμή strokeStyle να γράψουμε gradient (που γράψαμε στο προηγούμενο παράδειγμα) και να εισάγουμε κείμενο με gradient.

Παράδειγμα

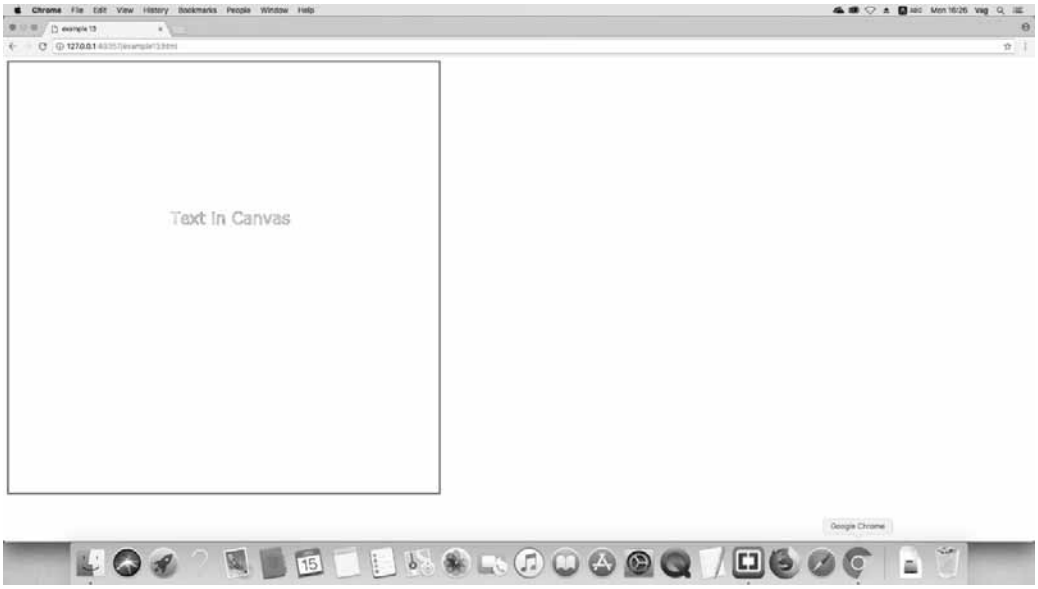
```
ctx.strokeStyle = gradient;

ctx.strokeText("Text in Canvas", 300, 300);
```

Παράδειγμα 13. Εισαγωγή κειμένου στο html Canvas

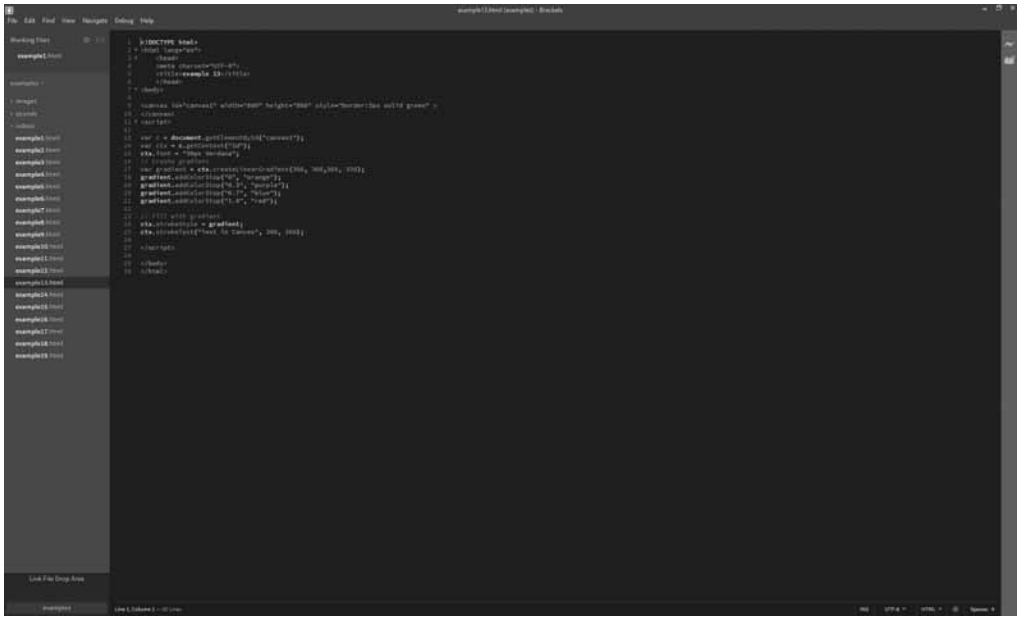
Κώδικας(Mac): Εισαγωγή κειμένου στο html Canvas



Εικόνα Browser(Mac): Εισαγωγή κειμένου στο html Canvas



Κώδικας(Win): Εισαγωγή κειμένου στο html Canvas



Εικόνα Browser(Win): Εισαγωγή κειμένου στο html Canvas

3.3 Μετασχηματισμοί στο Canvas

Πριν ξεκινήσουμε να εφαρμόζουμε μετασχηματισμούς στο Canvas θα πρέπει να γνωρίζουμε ότι πρώτα γράφουμε τον κώδικα για την ενεργοποίηση του οποιουδήποτε μετασχηματισμού και έπειτα γράφουμε τον κώδικα της γραμμής, του σχήματος κτλ. πάνω στο οποίο θα εφαρμοστεί ο μετασχηματισμός.

Αλλαγή κλίμακας

Για να αλλάξουμε την κλίμακα σχημάτων στο Html Canvas θα πρέπει να χρησιμοποιήσουμε τη μέθοδο scale με την παρακάτω σύνταξη:

```
ctx.scale(a,b)
```

Οι παράμετροι a, b καθορίζουν την κλίμακα στον οριζόντιο και τον κάθετο άξονα αντίστοιχα.

Για να διατηρήσουμε τις αναλογίες του σχήματος σε σχέση με τις διαστάσεις πλάτος και ύψος αρκεί να δώσουμε ίσες τιμές για a,b.

Ενδεικτικά ισχύουν τα παρακάτω για τις τιμές των a,b:

1=100%

2=200%

3=300%

0.5=50%

0.3=30%

0.2=20%

Παράδειγμα

Στο παράδειγμα που ακολουθεί θα σχεδιάσουμε δύο ορθογώνια με την επάνω αριστερή γωνία τους να ταυτίζεται με την αρχή των αξόνων. Στο δεύτερο ορθογώνιο θα γράψουμε τις ίδιες παραμέτρους για το σχεδιασμό του αλλά προηγουμένως θα έχουμε ορίσει αλλαγή κλίμακας με τη μέθοδο scale έτσι ώστε οι διαστάσεις του να ταυτιστούν με τις διαστάσεις του Canvas.

```
<canvas id="canvas1" width="800" height="800" style="border:3px solid red;">
</canvas>
```

Σχόλια

Έχουμε ορίσει canvas με id canvas1 και διαστάσεις 800X800 Pixels.

```
<script>
var canvas = document.getElementById("canvas1");
ctx = canvas.getContext("2d");
```

Σχόλια

Ενεργοποιήσαμε το canvas και δημιουργήσαμε αντικείμενο σχεδίασης

```
ctx.lineWidth=13;
```

Σχόλια

Ορίσαμε το πάχος της γραμμής σχεδιασμού ίσο με 13 pixels.

```
ctx.strokeRect(0,0,400,200);
```

Σχόλια

Σχεδιάσαμε ένα ορθογώνιο παραλληλόγραμμο τοποθετώντας την επάνω αριστερή γωνία του στο (0,0) και ορίσαμε διατάσεις 400X200 Pixels (πλάτος X ύψος).

```
ctx.lineWidth=7;
ctx.strokeStyle="yellow";
```

Σχόλια

Ορίσαμε πάχος γραμμής 7 pixels και χρώμα κίτρινο. Ορίσαμε διαφορετικό πάχος γραμμής για να μπορέσουμε να παρατηρήσουμε στο Canvas και τα δύο ορθογώνια

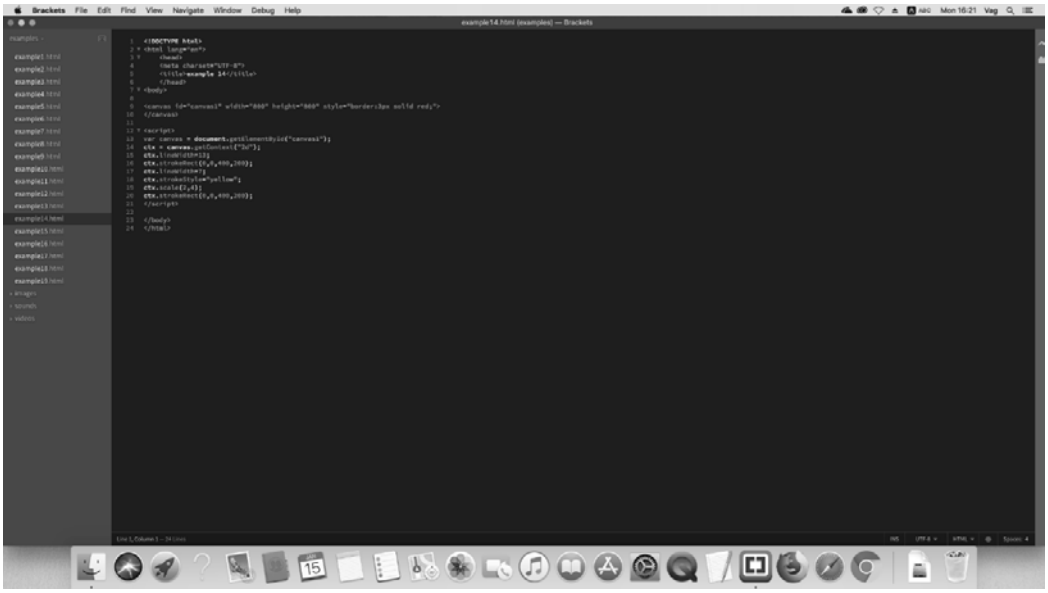
```
ctx.scale(2,4);

ctx.strokeRect(0,0,400,200);

</script>
```

εφόσον έχουμε ορίσει στον άξονα x κλίμακα 2 σημαίνει ότι το πλάτος του ορθογωνίου θα διπλασιαστεί και από 400 θα γίνει 800 Pixels

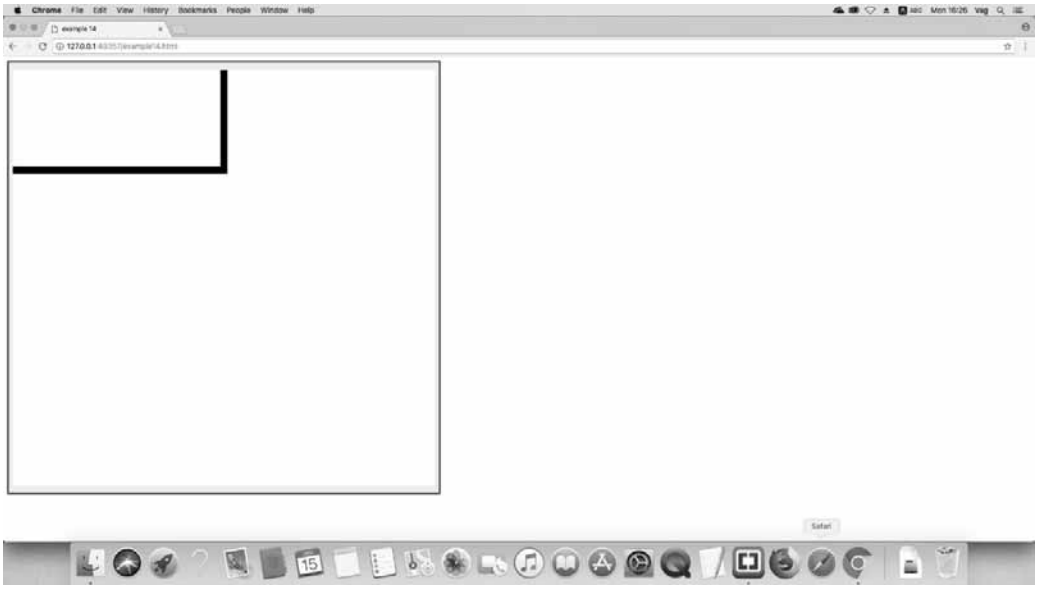
Ομοίως εφόσον στον άξονα y έχουμε ορίσει κλίμακα 4 το ύψος του ορθογωνίου από 200 θα γίνει 800 Pixels. Οπότε το ορθογώνιο θα ταυτίζεται με τις διαστάσεις του canvas.

Παράδειγμα 14. Scale

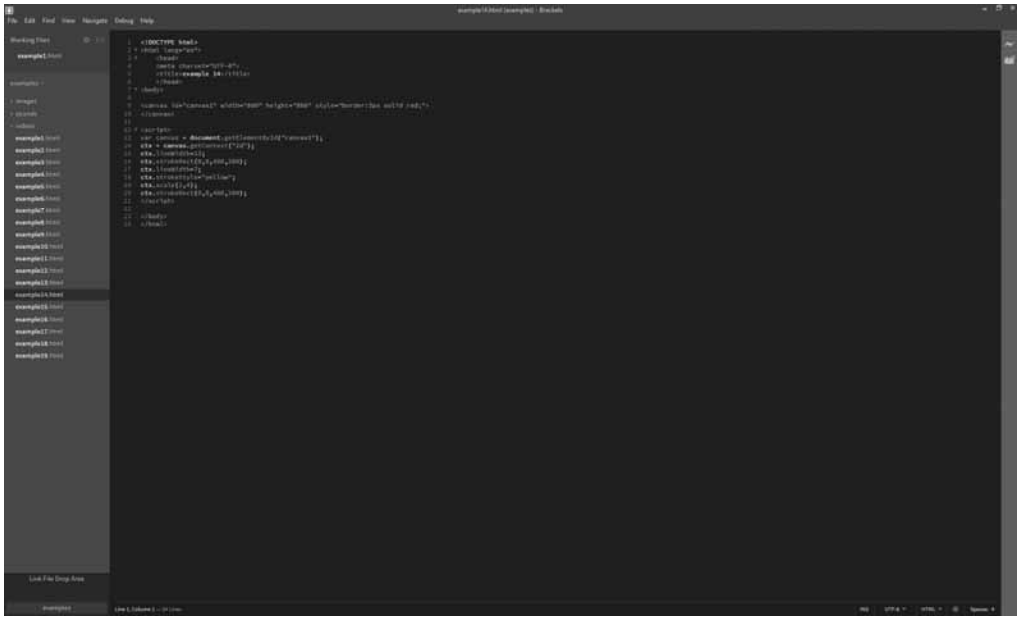
Κώδικας(Mac): scale



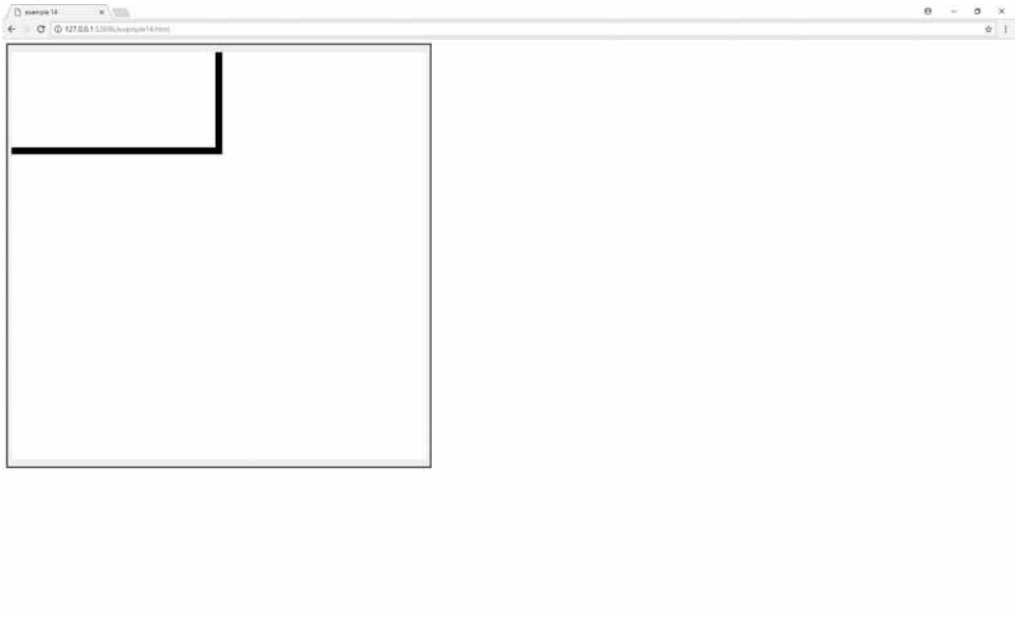
Εικόνα Browser(Mac): scale



Κώδικας(Win): scale



Εικόνα Browser(Win): scale



Περιστροφή (rotate)

Μπορούμε να περιστρέψουμε ένα σχήμα σύμφωνα με την περιστροφή των δεικτών του ρολογιού στο canvas με τη μέθοδο rotate με την ακόλουθη σύνταξη:

rotate(γωνία);

Στη θέση της γωνίας θα γράψουμε τη γωνία σε ακτίνια rad. Στην περίπτωση που επιθυμούμε να γράφουμε τη γωνία σε μοίρες μπορούμε να γράψουμε:

```
rotate(γωνία_σε_μοίρες * Math.PI/180);
```

Παράδειγμα

Θα ορίσουμε ένα canvas 800X800 με αναγνωριστικό canvas1:

```
<canvas id="canvas1" width="800" height="800" style="border:3px solid red;">
</canvas>
```

Στη συνέχεια θα αποκτήσουμε πρόσβαση στο canvas και θα δημιουργήσουμε αντικείμενο σχεδίασης

```
<script>
var canvas = document.getElementById("canvas1");
ctx = canvas.getContext("2d");
```

Θα σχεδιάσουμε ένα ορθογώνιο χωρίς να έχουμε προσθέσει περιστροφή

```
ctx.fillStyle="orange";
ctx.fillRect(0,0,400,200);
```

Στη συνέχεια θα ορίσουμε περιστροφή 40 μοιρών:

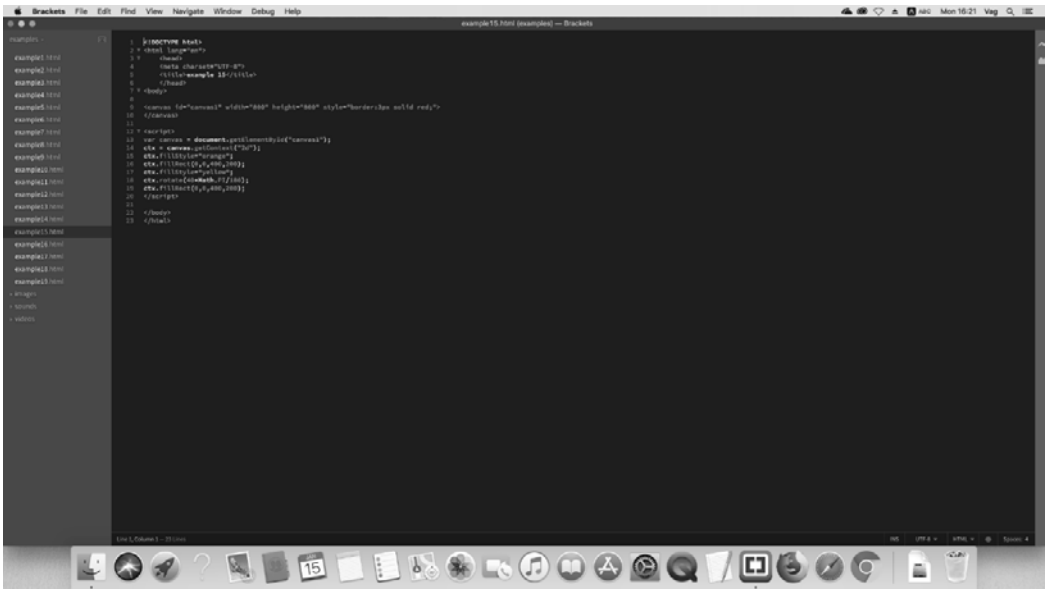
```
ctx.rotate(40*Math.PI/180);
```

Αφού έχει ενεργοποιηθεί η περιστροφή 40 μοιρών θα σχεδιάσουμε το ίδιο ορθογώνιο με γέμισμα χρώματος και θα κλείσουμε με την ετικέτα </script>

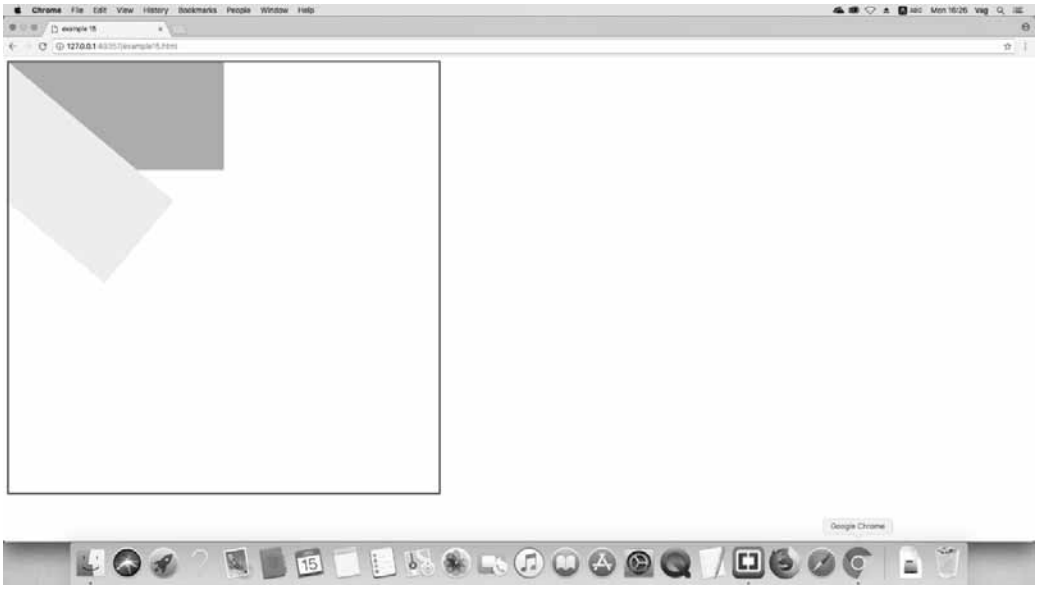
```
ctx.fillRect(0,0,400,200);
</script>
```

Παράδειγμα 15. Rotate

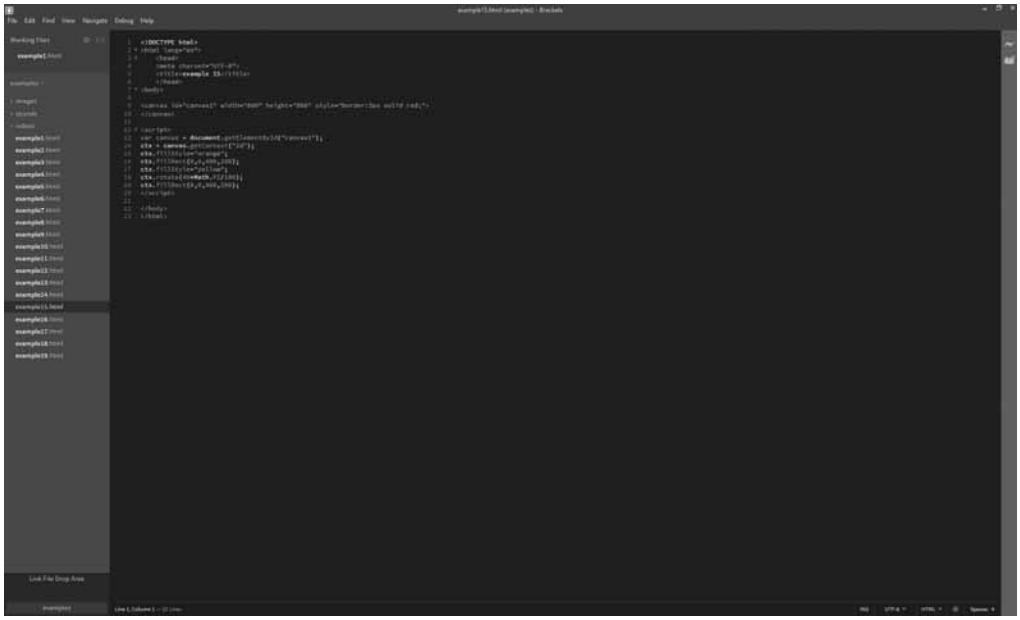
Κώδικας(Mac): rotate



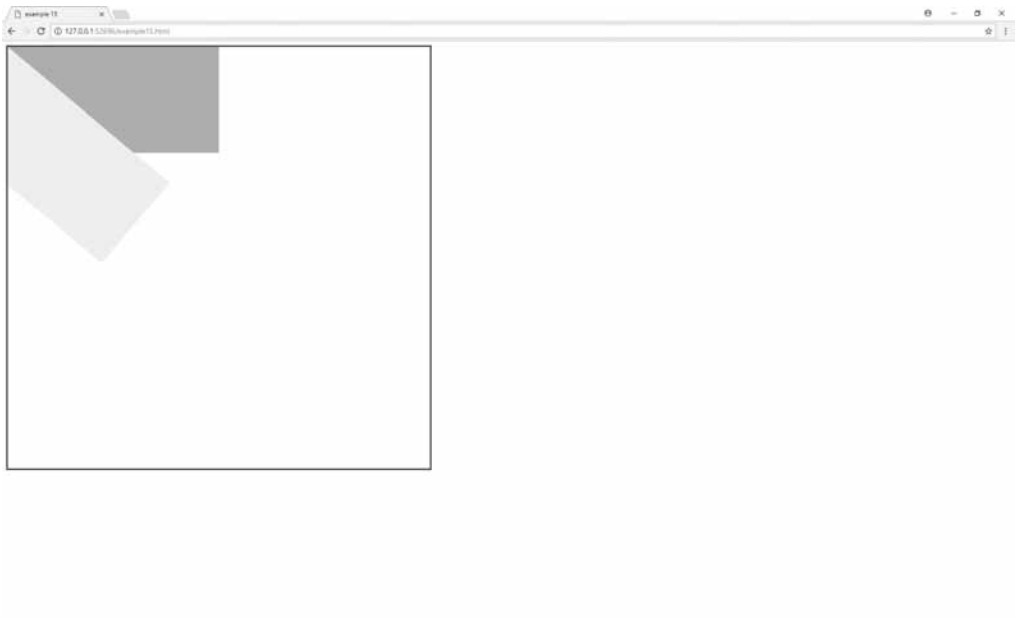
Εικόνα Browser(Mac): rotate



Κώδικας(Win): rotate



Εικόνα Browser(Win): rotate



Μετατόπιση

Μπορούμε να προκαλέσουμε μετατόπιση στις συντεταγμένες ενός σχήματος ή μιας γραμμής με τη βοήθεια της μεθόδου translate με την παρακάτω σύνταξη

```
translate(x,y);
```

Η παράμετρος x αντιστοιχεί στην μετατόπιση στον οριζόντιο άξονα
Η παράμετρος y αντιστοιχεί στην μετατόπιση στον κατακόρυφο άξονα
Οι τιμές μετριούνται σε pixels.

translate(x,y);	
+x	Μετατόπιση προς τα δεξιά
-x	Μετατόπιση προς τα αριστερά
+y	Μετατόπιση προς τα κάτω
-y	Μετατόπιση προς τα πάνω

Έστω για παράδειγμα ότι έχουμε ορίσει συντεταγμένες της επάνω αριστερής γωνίας ενός ορθογωνίου το (50,100).
Αν εφαρμόσουμε translate(150,200); Οι καινούργιες συντεταγμένες για την επάνω αριστερή γωνία του ορθογωνίου είναι:
X=50+150=200
Y=100+200=300
Άρα οι συντεταγμένες του μετατοπισμένου ορθογωνίου θα είναι (200,300).

Σημείωση

Στην ουσία εφόσον έχουμε ενεργοποιήσει τη μέθοδο translate και γράψουμε συντεταγμένες για ένα καινούργιο σχήμα ή μία γραμμή οι τελικές συντεταγμένες προκύπτουν ως αποτέλεσμα της πρόσθεσης των αντίστοιχων συντεταγμένων

translate(x,y)

Συντεταγμένες_σχεδίου(x_1,y_1)

Τελικές συντεταγμένες($x_{τελ}$=x+x_1,$y_{τελ}$=y+y_1)

Παράδειγμα

Εφόσον έχουμε ορίσει το canvas και εντός του script έχουμε αποκτήσει πρόσβασή στο canvas και έχουμε δημιουργήσει νέο αντικείμενο σχεδίασης σχεδιάζουμε το παρακάτω ορθογώνιο:

```
ctx.fillStyle="orange";
ctx.fillRect(0,0,400,200);
ctx.fillStyle="yellow";
```

Στη συνέχεια θα ενεργοποιήσουμε μετατόπιση 400 pixels κατά τον οριζόντιο άξονα και 600 pixels κατά τον κάθετο άξονα.

```
ctx.translate(400,600);
ctx.fillRect(0,0,400,200);
</script>
```

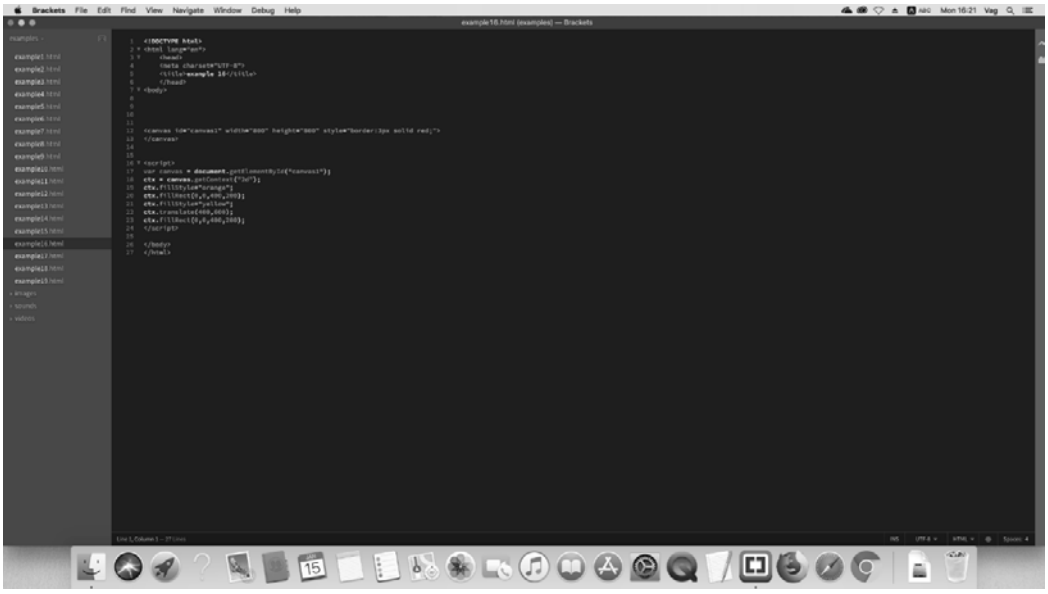
Σχόλια

οι πραγματικές συντεταγμένες της επάνω αριστερής γωνίας του ορθογωνίου θα είναι:

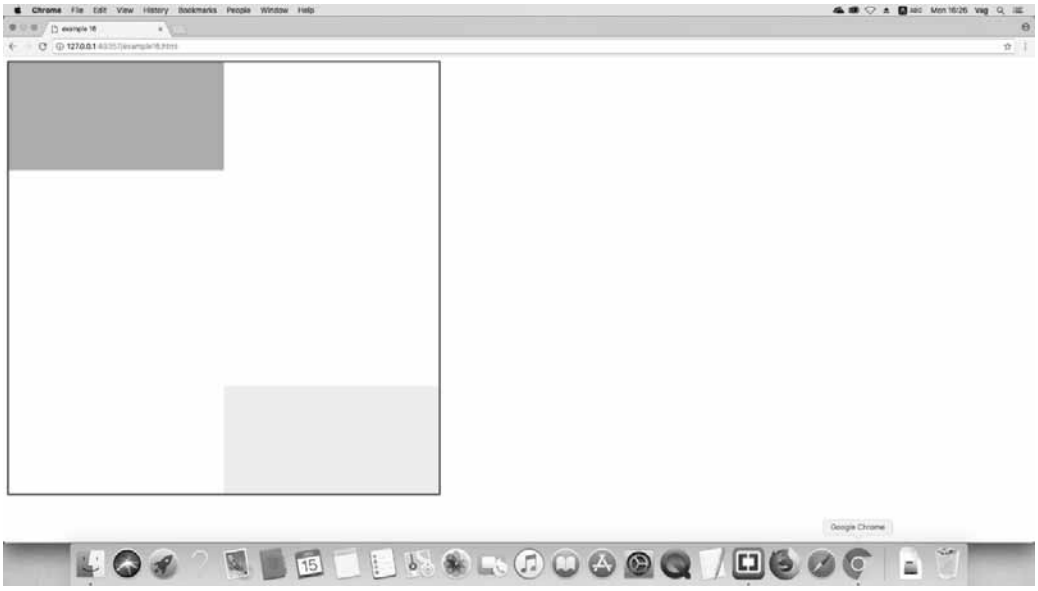
(400+0,600+0)=(400,600)

Παράδειγμα 16. Translate

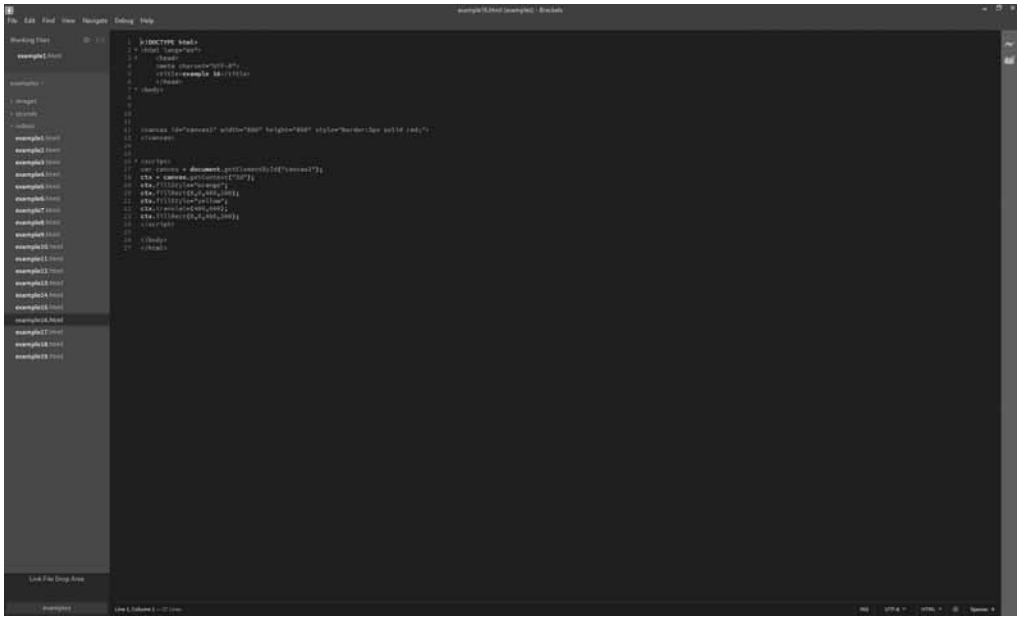
Κώδικας(Mac): translate



Εικόνα Browser(Mac): translate



Κώδικας(Win): translate



Εικόνα Browser(Win): translate

Μέθοδος transform

Η μέθοδος transform μας επιτρέπει να εφαρμόσουμε ένα σύνολο μετασχηματισμών σύμφωνα με την παρακάτω σύνταξη:

transform(a,b,c,d,e,f)
a→ αλλαγή κλίμακας στον οριζόντιο άξονα
b→ οριζόντια λοξή παραμόρφωση (horizontal scew)-τιμές tan()
c→ κάθετη λοξή παραμόρφωση (vertical scew)- τιμές tan()
d→ αλλαγή κλίμακας στον κάθετο άξονα
e→οριζόντια μετατόπιση +x pixels για δεξιά κατεύθυνση και -x pixels για αριστερή κατεύθυνση
f→ κάθετη μετατόπιση:-y pixels προς τα επάνω και +y pixels προς τα κάτω

Οι τιμές για τις μεθόδους transform λειτουργούν προσθετικά. Αν για παράδειγμα γράψουμε μία μέθοδο transform και υπάρχει ήδη άλλη μία, η δεύτερη μέθοδος transform λειτουργεί λαμβάνοντας σαν σημείο αναφοράς την πρώτη κοκ.

Αν για παράδειγμα η πρώτη transform δίνει +200 pixels δεξιά κατεύθυνση και η επόμενη -300 pixels προς τα αριστερά το συνολικό αποτέλεσμα θα είναι -100 Pixels μετατόπιση προς τα αριστερά.

Παράδειγμα

```
ctx.transform(1,0,0,1,150,0);

ctx.transform(1,0,0,1,150,0);
```

Σχόλια

Οι δύο μέθοδοι transform έχουν τις ίδιες παραμέτρους και προκαλούν μόνο μετατόπιση 150 pixels προς τα δεξιά η κάθε μία. Η συνολική μετατόπιση είναι 150+150=300 pixels προς τα δεξιά.

Σημείωση

Αν δεν χρειαζόμαστε αλλαγή κλίμακας στους άξονες x και y οι τιμές οι τιμές για a,d είναι ίσες με ένα.

Αν δε χρειαζόμαστε λοξή μετατόπιση στους άξονες x και y οι τιμές για b,c είναι ίσες με μηδέν εφόσον η εφαπτομένη στις 0 μοίρες είναι ίση με μηδέν.

Παράδειγμα

Εντός canvas με διαστάσεις 800X800 θα γράψουμε κώδικα για να σχεδιασθούν τρία ορθογώνια με ακριβώς τις ίδιες παραμέτρους. Το πρώτο ορθογώνιο θα σχεδιασθεί χωρίς να προηγείται μέθοδος transform, το δεύτερο θα σχεδιασθεί αφού έχει προηγηθεί μία ιδιότητα transform, ενώ το τρίτο θα σχεδιασθεί αφού έχουν προηγηθεί δύο μέθοδοι transform με τις ίδιες παραμέτρους.

Αρχικά εισάγουμε το στοιχείο canvas:

```
<canvas id="canvas1" width="800" height="800"
style="border:3px solid blue;">
</canvas>
```

Στη συνέχεια αποκτούμε πρόσβαση στο στοιχείο canvas και δημιουργούμε αντικείμενο σχεδίασης.

```
<script>
var canvas = document.getElementById("canvas1");
ctx = canvas.getContext("2d");
```

Ορίζουμε γέμισμα χρώματος κίτρινο και σχεδιάζουμε το πρώτο ορθογώνιο

```
ctx.fillStyle="yellow";
ctx.fillRect(100,100,150,100);
```

Ενεργοποιούμε το πρώτο transform:

```
ctx.transform(1,1,-0.2,1,50,60);
```

Σχεδιάζουμε το δεύτερο ορθογώνιο που θα επηρεαστεί από το πρώτο transform

```
ctx.fillStyle="red";
ctx.fillRect(100,100,150,100);
```

Γράφουμε ένα ίδιο transform

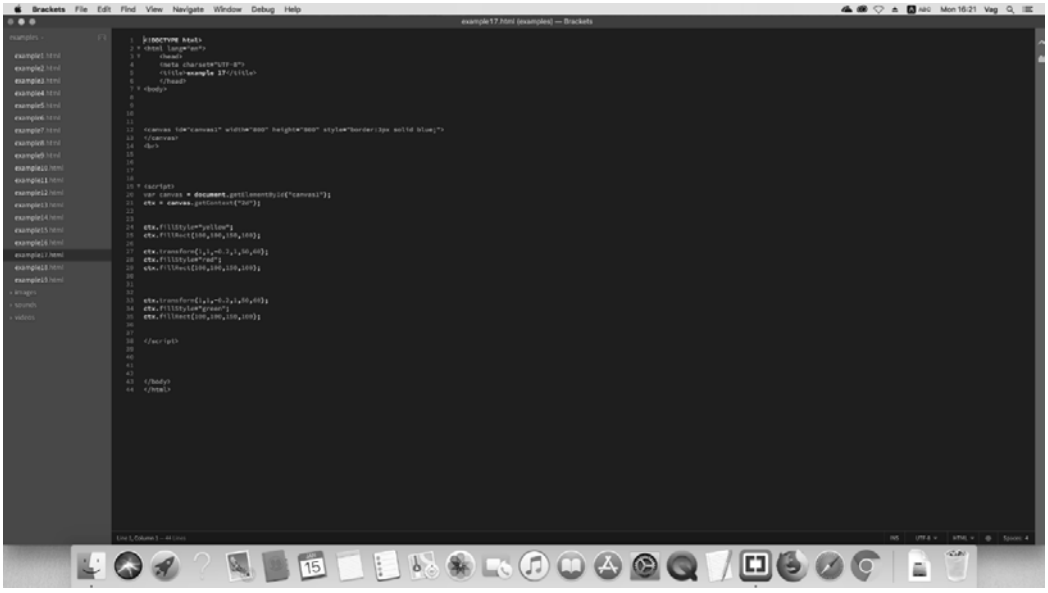
```
ctx.transform(1,1,-0.2,1,50,60);
```

Σχεδιάζουμε με γέμισμα χρώματος πράσινο το τρίτο ορθογώνιο που θα επηρεαστεί και από τα δύο transforms:

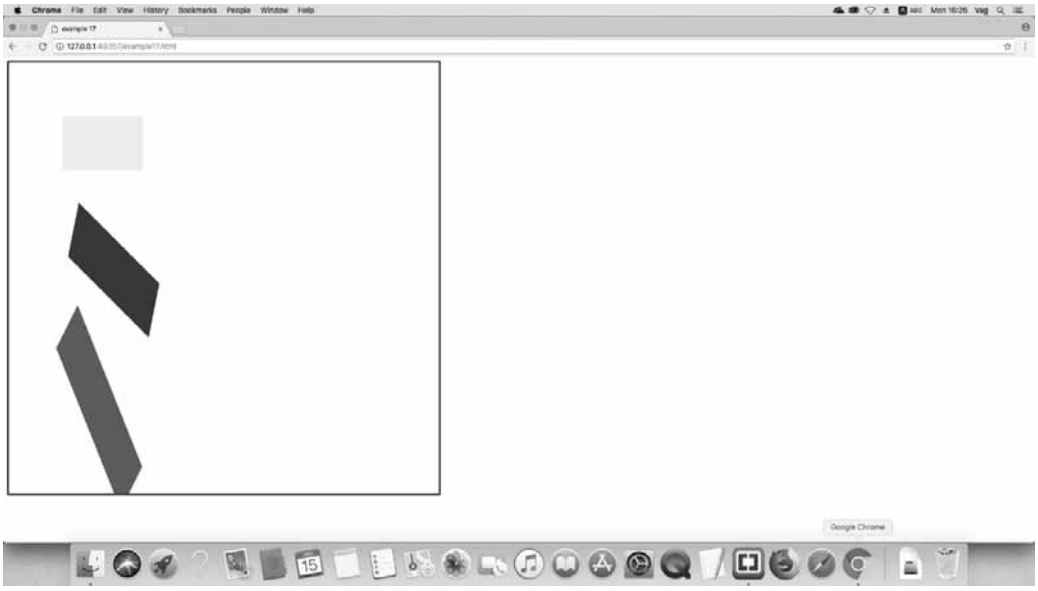
```
ctx.fillStyle="green";
ctx.fillRect(100,100,150,100);
```

Παράδειγμα 17. transform()

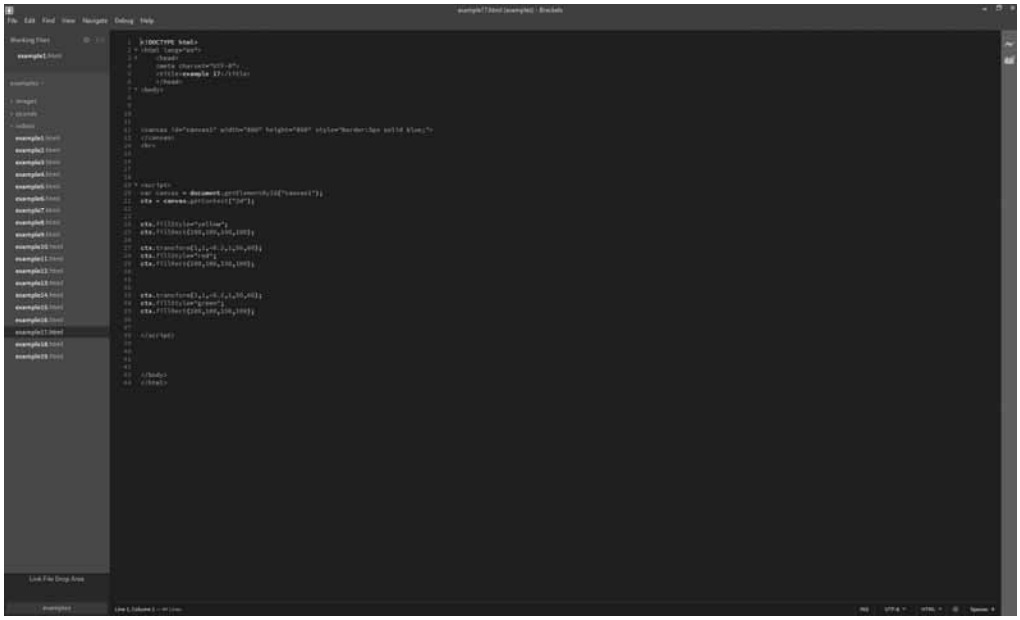
Κώδικας(Mac): transform()



Εικόνα Browser(Mac): transform()



Κώδικας(Win): transform()



Εικόνα Browser(Win): transform()

Μέθοδος setTransform()

Η μέθοδος setTransform() έχει ακριβώς τα ίδια αποτελέσματα, την ίδια σύνταξη και τις ίδιες παραμέτρους με την μέθοδο transform με μία ειδοποιό διαφορά. Πριν ενεργοποιήσει του μετασχηματισμούς κάνει επαναφορά των προεπιλεγμένων τιμών σε όλες τις ιδιότητες μετασχηματισμού που επηρεάζει.

Οπότε όσες φορές και αν γράψουμε μία μέθοδο setTransform() με τις ίδιες τιμές για τις παραμέτρους, οι μετασχηματισμοί που θα εφαρμοσθούν μετά το πρώτο ή το δεύτερο setTransform() θα είναι οι ίδιοι, εφόσον κάθε φορά γίνεται επαναφορά στις προεπιλεγμένες τιμές.

setTransform(a,b,c,d,e,f)
a→ αλλαγή κλίμακας στον οριζόντιο άξονα
b→ Οριζόντια λοξή παραμόρφωση (horizontal scew)-τιμές tan()
c→ Κάθετη λοξή παραμόρφωση (vertical scew)- τιμές tan()
d→ αλλαγή κλίμακας στον κάθετο άξονα
e→Οριζόντια μετατόπιση: +x pixels για δεξιά κατεύθυνση και -x pixels για αριστερή κατεύθυνση
f→ Κάθετη μετατόπιση:-y pixels προς τα επάνω και +y pixels προς τα κάτω

Παράδειγμα

Θα εισάγουμε δύο στοιχεία canvas με διαστάσεις 600X600 και με αναγνωριστικά canvas1 και canvas2 ως εξής:

```
<canvas id="canvas1" width="600" height="600" style="border:3px solid blue;">
</canvas>
<br>
<canvas id="canvas2" width="600" height="600" style="border:3px solid red;">
</canvas>
```

Στο πρώτο στοιχείο canvas θα γράψουμε

```
<script>
var canvas = document.getElementById("canvas1");
ctx = canvas.getContext("2d");
ctx.fillStyle="yellow";
ctx.fillRect(100,100,150,100);
```

Σχόλια

Έχουμε σχεδιάσει με γέμισμα χρώματος κίτρινο ένα ορθογώνιο με διαστάσεις 150X100, έχοντας τοποθετήσει την επάνω αριστερή γωνία του στο σημείο (100,100).

Στη συνέχεια ενεργοποιούμε μια σειρά μετασχηματισμών και στη συνέχεια σχεδιάζουμε το ίδιο ορθογώνιο με χρώμα κόκκινο.

```
ctx.setTransform(1,1,-0.2,1,50,60);
ctx.fillStyle="red";
ctx.fillRect(100,100,150,100);
```

Τέλος, γράφουμε την ίδια setTransform() και σχεδιάζουμε πάλι το ίδιο ορθογώνιο με γέμισμα χρώματος πράσινο.

```
ctx.setTransform(1,1,-0.2,1,50,60);
ctx.fillStyle="green";
ctx.fillRect(100,100,150,100);
```

Αποτέλεσμα

Θα εμφανιστεί στο canvas το πρώτο ορθογώνιο και το τελευταίο (πράσινο). Ο λόγος που γίνεται αυτό είναι ότι εφόσον το πράσινο έχεις τις ίδιες διαστάσεις και συντεταγμένες καθώς και τους ίδιους μετασχηματισμούς με το κόκκινο ορθογώνιο καλύπτει 100% το κόκκινο.

Στο δεύτερο canvas θα αλλάξουμε τη σειρά του δεύτερου και του τρίτου ορθογωνίου:

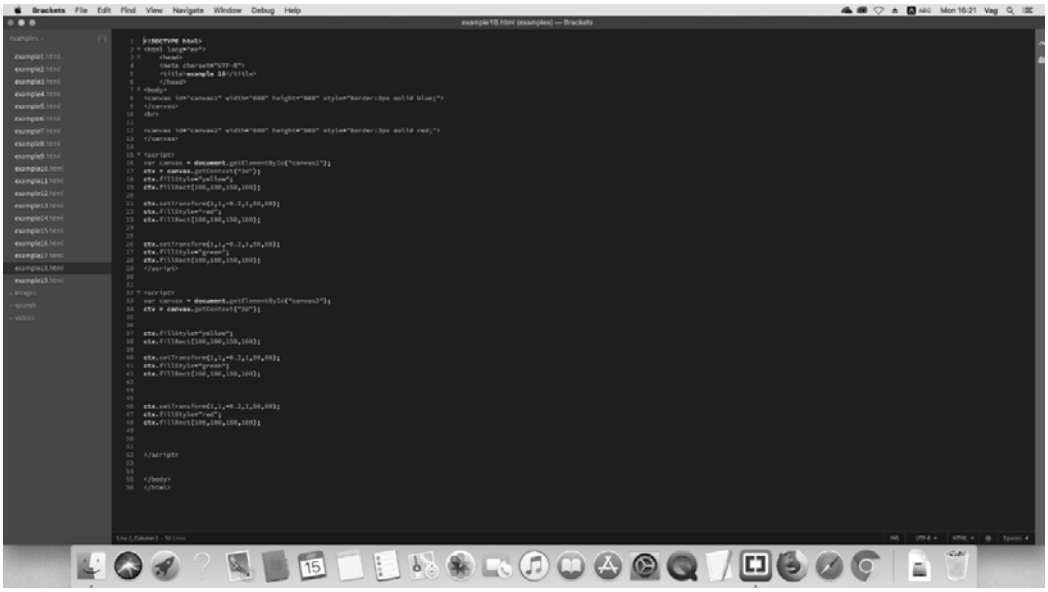
```
<script>
var canvas = document.getElementById("canvas2");
ctx = canvas.getContext("2d");
ctx.fillStyle="yellow";
ctx.fillRect(100,100,150,100);
ctx.setTransform(1,1,-0.2,1,50,60);
ctx.fillStyle="green";
ctx.fillRect(100,100,150,100);
ctx.setTransform(1,1,-0.2,1,50,60);
ctx.fillStyle="red";
ctx.fillRect(100,100,150,100);
</script>
```

Αποτέλεσμα

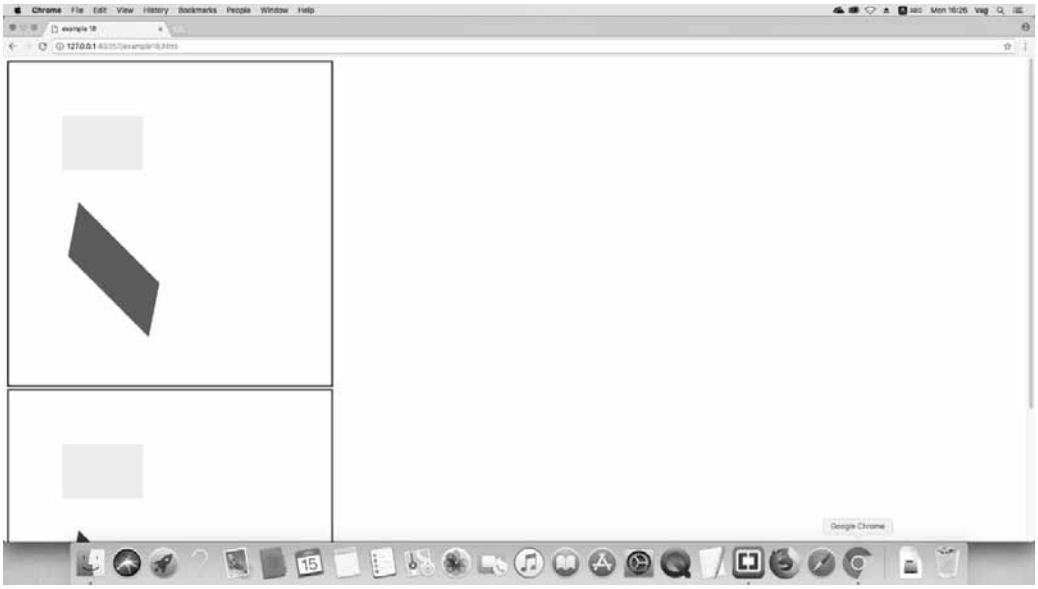
Θα εμφανιστεί το πρώτο ορθογώνιο και το τελευταίο (κόκκινο)

Παράδειγμα 18. setTransform()

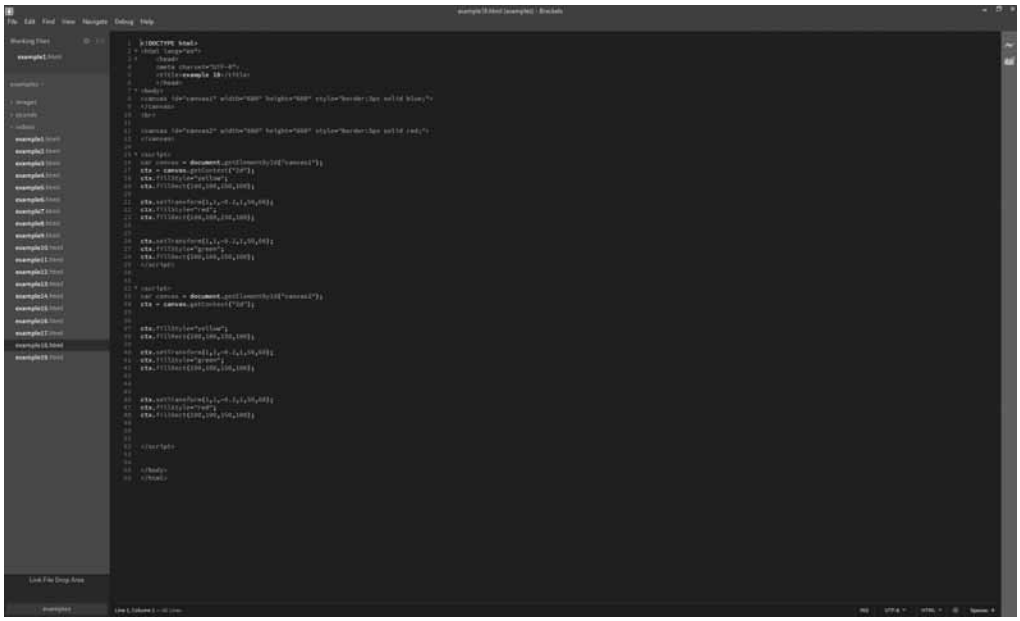
Κώδικας(Mac): setTransform()



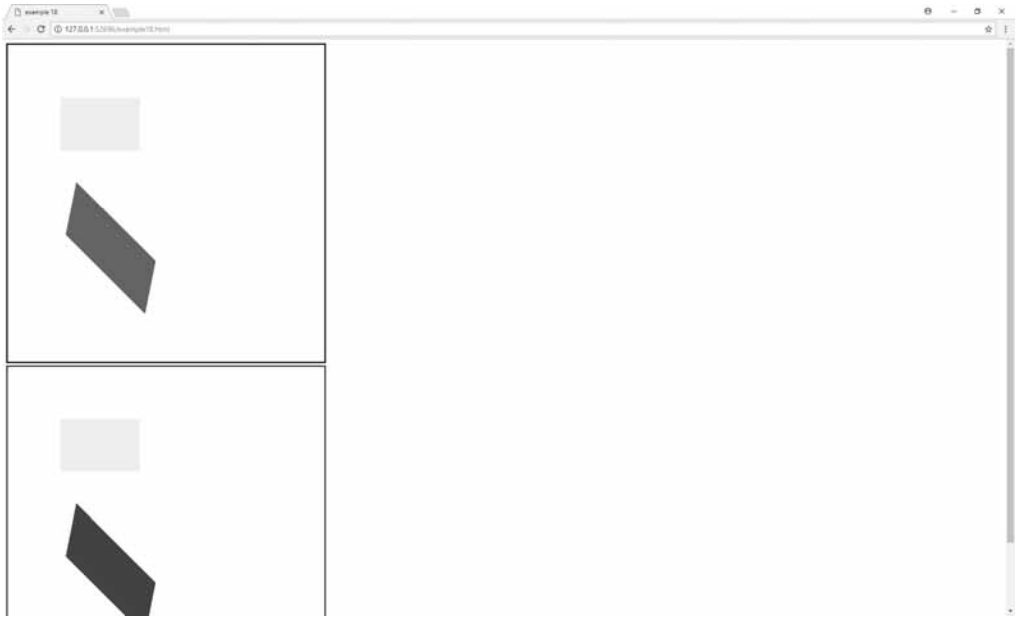
Εικόνα Browser(Mac): setTransform()



Κώδικας(Win): setTransform()



Εικόνα Browser(Win): setTransform()



3.4 Links

Links

Μπορούμε να ορίσουμε σύνδεση προς μία άλλη σελίδα με ένα στοιχείο <a>. Εντός του στοιχείου a θα γράψουμε ένα στοιχείο href και στη συνέχεια θα γράψουμε τη διαδρομή και το όνομα της σελίδας όπως παρακάτω:

```
<a href="path/όνομα_αρχείου">Κείμενο για το link </a>
```

Σελιδοδείκτες-bookmarks

Η HTML υποστηρίζει τη δημιουργία bookmarks. Το bookmark επιτρέπει τη μετάβαση του χρήστη σε κάποιο άλλο σημείο της σελίδας.

Αρχικά εντός στοιχείου a γράφουμε ως τιμή href τον τελεστή hash και στη συνέχεια χωρίς κενό το αναγνωριστικό του σημείου όπου θα μεταβεί ο χρήστης.

```
<a href="#αναγνωριστικό">…</a></
```

Στη συνέχεια εντός κάπου στοιχείου στην ετικέτα έναρξης δηλώνουμε το αναγνωριστικό ως τιμή της ιδιότητας id.

```
id="…"
```

Παράδειγμα

```
<p><a href="#p7">go to paragraph 7</a></p>
```

Σχόλια

Αρχικά δηλώσαμε στη σύνδεση το αναγνωριστικό p7

```
  <h5>paragraph 1</h5>
<p>  …  </p>
   <h5>paragraph 2</h5>
<p>  …  </p>
   <h5>paragraph 3</h5>
<p>  …  </p>
   <h5>paragraph 4</h5>
<p>  …  </p>
   <h5>paragraph 5</h5>
<p>  …  </p>
   <h5>paragraph 6</h5>
<p>  …  </p
   <h5 id="p7">paragraph 7</h5><p>  …  </p>
```

Σχόλια

Στην ετικέτα έναρξης του στοιχείου h5 όρισμα τιμή p7 για την ιδιότητα id. Έτσι λοιπόν μόλις ο χρήστης πατήσει το link με bookmark p7 θα μεταβεί σε αυτό το σημείο.

Σημείωση

Εναλλακτικά μπορούμε να δηλώσουμε bookmark σε άλλη σελίδα γράφοντας πριν τον τελεστή hash στην href τιμή το path και το όνομα του αρχείου

```
<a href=" path/όνομα_αρχείου #αναγνωριστικό">…</a>
```

Παράδειγμα

```
<a href=" page2.html#p7">...</a>
```

Παράδειγμα 19. Bookmarks

Κώδικας(Mac): bookmarks

Εικόνα Browser(Mac): bookmarks

Κώδικας(Win): bookmarks

Εικόνα Browser(Win): bookmarks

3.4 Βίντεο

Η ενότητα κλείνει με ορισμένα βίντεο που παρουσιάζουν όσα αναλύθηκαν στην παρούσα ενότητα, και τα οποία μπορείτε να δείτε στο αντίστοιχο σημείο του ηλεκτρονικού υλικού. Τα βίντεο 1-19 αφορούν mac και τα βίντεο 20-38 αφορούν pc.

Σύνοψη

Μετά την ολοκλήρωση της συγκεκριμένης διδακτικής ενότητας γνωρίζετε ότι:

- Μπορούμε να εισάγουμε εικόνες σε μια Html σελίδα με τη βοήθεια του στοιχείου img
- Μπορούμε με την ιδιότητα position να ορίσουμε τη θέση της εικόνας εντός της ιστοσελίδας
- Μπορούμε να ορίσουμε την σειρά εμφάνισης στοιχείων όπου το ένα καλύπτει το άλλο με την ιδιότητα z-index
- Μπορούμε να εισάγουμε video στην ιστοσελίδα μας με την ετικέτα <video>.
- Μπορούμε να εισάγουμε audio στην ιστοσελίδα μας με την ετικέτα <audio>
- Μπορούμε να εισάγουμε Html canvas στη σελίδα μας με τη βοήθεια της ετικέτας <canvas>
- Μπορούμε να σχεδιάζουμε γραμμές στο canvas με τη μέθοδο lineTo()
- Μπορούμε να σχεδιάζουμε ορθογώνια με γέμισμα χρώματος με τη μέθοδο fillRect()

- Μπορούμε να σχεδιάζουμε τόξα και κύκλους με τη βοήθεια της μεθόδου arc()
- Μπορούμε να σχεδιάζουμε καμπύλες με ένα control point με τη μέθοδο quadraticCurveTo()
- Μπορούμε να σχεδιάζουμε καμπύλες με δύο control points με τη μέθοδο **bezierCurve()**
- Με τη μέθοδο strokeText() μπορούμε να εισάγουμε κείμενο στο Canvas
- Με τη μέθοδο createLinearGradient() μπορούμε να ορίσουμε διαβάθμιση χρωμάτων
- Με τη μέθοδο transform() μπορούμε να εφαρμόζουμε μετασχηματισμούς
- Με τη μέθοδο setTransform() μπορούμε να εφαρμόζουμε μετασχηματισμούς αφού προηγουμένως γίνει επαναφορά των ιδιοτήτων των μετασχηματισμών στις αρχικές προεπιλεγμένες τιμές
- Με τη βοήθεια του στοιχείου a μπορούμε να δημιουργήσουμε συνδέσεις προς άλλες σελίδες ή και να δημιουργήσουμε bookmarks

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Προγράμματα Συμπληρωματικής Εκπαίδευσης

Με τη χρήση καινοτόμων μεθόδων εξ αποστάσεως εκπαίδευσης

1937 - ΔΗΜΙΟΥΡΓΙΑ RESPONSIVE ΣΕΛΙΔΩΝ ΜΕ HTML5 ΚΑΙ CSS

ΔΙΔΑΚΤΙΚΗ ΕΝΟΤΗΤΑ 4. CSS3, FLEXBOX ΚΑΙ CSS GRID SYSTEM

Εισαγωγικές Παρατηρήσεις

Στην υποενότητα 1 θα αναλυθούν οι τρόποι ενσωμάτωσης και εφαρμογής CSS σε μία σελίδα. Επίσης θα γίνει ανάλυση των CSS selectors για την επιλογή στοιχείων της σελίδας έτσι ώστε να εφαρμοστεί CSS.

Στην υποενότητα 2 θα παρουσιαστεί και θα αναλυθεί το CSS box model και οι ιδιότητες που είναι απαραίτητες για τη διαμόρφωσή του. Επίσης θα αναλυθεί βήμα προς βήμα η δημιουργία navigation. Τέλος θα γίνει εισαγωγή στο Flex-Box και στις δυνατότητές του.

Στην υποενότητα 3 θα παρουσιαστούν και θα αναλυθούν οι ιδιότητες για το Flex box. Επίσης θα αναλυθεί το CSS Grid System και οι ιδιότητές του που χρησιμεύουν στην παραμετροποίηση του.

ΥΠΟΕΝΟΤΗΤΑ 1. CSS

Για να εισάγουμε κώδικα CSS στις σελίδες μας υπάρχουν οι παρακάτω τρόποι:

- External style sheet
- Internal style sheet
- Inline style

1.1 Τρόποι ενσωμάτωσης CSS

External Style Sheet

Ο συγκεκριμένος τρόπος εισαγωγής κώδικα CSS για τον καθορισμό της εμφάνισης των σελίδων προσφέρει πολλά πλεονεκτήματα. Σύμφωνα με αυτή τη μέθοδο γράφουμε τον CSS κώδικα σε ένα αρχείο και σε όποια σελίδα επιθυμούμε, δημιουργούμε μία σύνδεση της σελίδας προς το αρχείο CSS. Με αυτή τη μέθοδο εξοικονομούμε χρόνο, εφόσον διορθώνοντας ένα CSS αρχείο οι αλλαγές μεταφέρονται σε όλες τις σελίδες που χρησιμοποιών τον συγκεκριμένο κώδικα CSS. Επίσης μπορούμε να έχουμε στο αρχείο μας δικά μας CSS αρχεία και με κατάλληλες τροποποιήσεις να τα χρησιμοποιούμε σε όποιο website και σε όποιες σελίδες επιθυμούμε.

Πιο αναλυτικά εντός του στοιχείου head θα γράψουμε ανάμεσα στις ετικέτες <link> και </link> κώδικα με την παρακάτω σύνταξη:

```
<head>
<link rel="stylesheet" type="text/css" href="path/όνομα_αρχείου.css">
</head>
```

Για να χρησιμοποιήσουμε τη μέθοδο External Style Sheet θα πρέπει να κάνουμε τα ακόλουθα βήματα:

1. Θα δημιουργήσουμε ένα αρχείο στο Brackets ή σε οποιοδήποτε Text Editor
2. Θα γράψουμε τον CSS κώδικα στο αρχείο χωρίς να γράψουμε πρόσθετες ετικέτες π.χ. <style>
3. Αφού γράψουμε τις CSS ιδιότητες θα αποθηκεύσουμε το αρχείο δίνοντας κατάληξη .css. Παράδειγμα: style1.css

4. Θα γράψουμε εντός του στοιχείου head το link για να συνδέσουμε το αρχείο CSS με τη συγκεκριμένη σελίδα.

```
<link rel="stylesheet" type="text/css" href="path/όνομα_αρχείου.css">
```

Παράδειγμα

Γράφουμε σε ένα αρχείο τον παρακάτω κώδικα CSS με τις ιδιότητες για font-family, font-size και color:

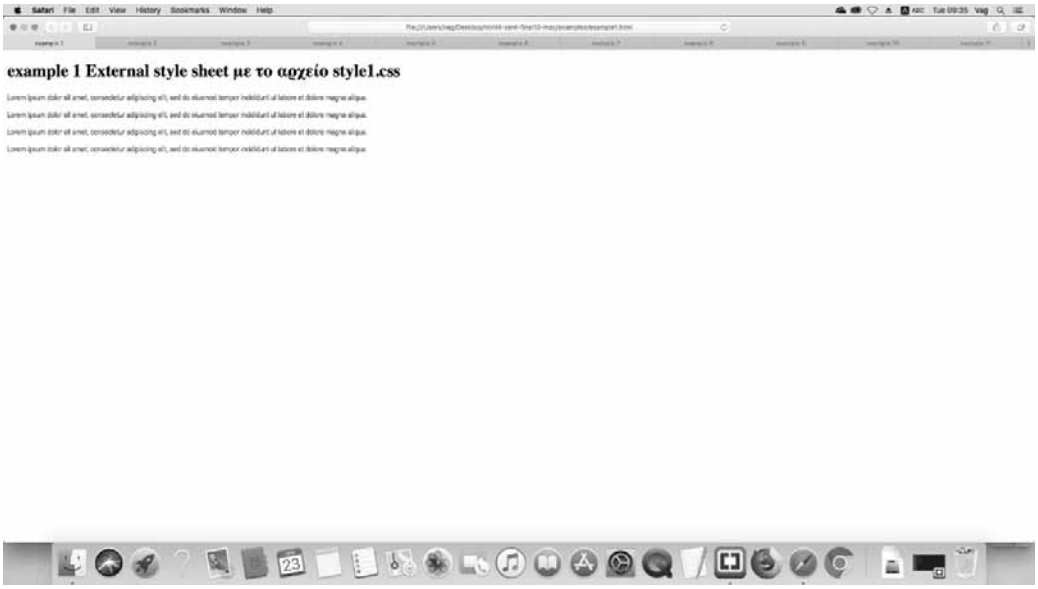
```
p {
font-family: Arial Narrow, sans-serif ;
    font-size: 15px;
    color:red;
}
```

Αποθηκεύουμε το αρχείο ως style1.css

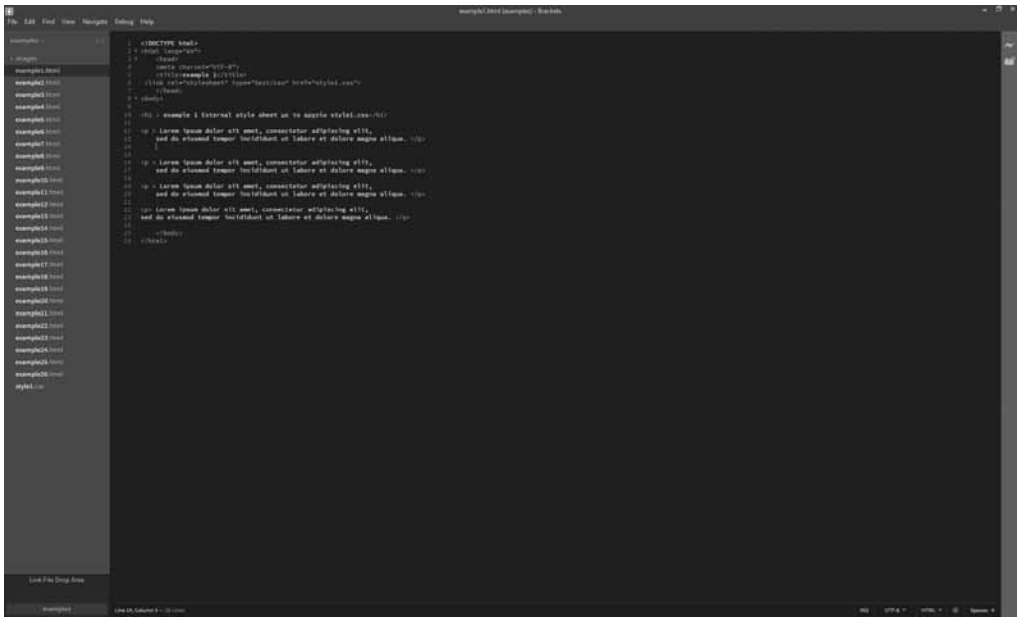
Παράδειγμα 1. External style sheet

Κώδικας(Mac): External style sheet

Εικόνα Browser(Mac): External style sheet



Κώδικας(Win): External style sheet



Εικόνα Browser(Win): External style sheet

Γράφουμε τον κώδικα για τη σελίδα προσθέτοντας παραγράφους με δοκιμαστικό κείμενο με στοιχεία p ή όποιο άλλο στοιχείο επιθυμούμε.

Εντός του στοιχείου head της σελίδας και εφόσον η σελίδα και το αρχείο css βρίσκονται στο ίδιο folder γράφουμε:

```
<head>
   <meta charset="UTF-8">
   <title>example 1</title>
 <link rel="stylesheet" type="text/css" href="style1.css">
   </head>
```

Αν η σελίδα μας βρίσκεται στο κεντρικό folder και το αρχείο css είναι αποθηκευμένο στο subfolder css τότε για το link θα γράψουμε:

```
<link rel="stylesheet" type="text/css" href="css/style1.css">
```

Σημείωση

Τα css αρχεία μπορούμε να τα δημιουργούμε και να τα επεξεργαζόμαστε σε ένα οποιοδήποτε Text Editor

Internal style sheet

Με αυτή τη μέθοδο γράφουμε τον κώδικα για τις ιδιότητες CSS στο στοιχείο head της σελίδας και μεταξύ των ετικετών <style> και </style>.

Παράδειγμα

```
<style>
        p {
   font-family: Arial Narrow, sans-serif ;
        font-size: 15px;
        color:red;
          }
      </style>
```

Σχόλια

Εφαρμόσαμε το ίδιο CSS γράφοντας τις ιδιότητες για το στοιχείο p εντός του στοιχείου head και ανάμεσα στις ετικέτες <style> και </style>.

Inline style sheet

Το Inline style sheet χρησιμοποιείται όταν επιθυμούμε να εφαρμόσουμε CSS style σε ένα συγκεκριμένο στοιχείο, όπως για παράδειγμα σε μία συγκεκριμένη παράγραφο (στοιχείο p).

Για να εφαρμόσουμε Inline style θα πρέπει να γράψουμε στην ετικέτα έναρξης του στοιχείου τη δήλωση style=" " και εντός των εισαγωγικών να γράψουμε τις CSS ιδιότητες.

Η μέθοδος Inline style συνιστάται μόνο για να εφαρμόσουμε CSS σε κάποιο στοιχείο ως εξαίρεση. Δεν είναι η καλύτερη λύση σε κάθε ετικέτα έναρξης να γράφουμε CSS μόνο για ένα στοιχείο.

Παράδειγμα

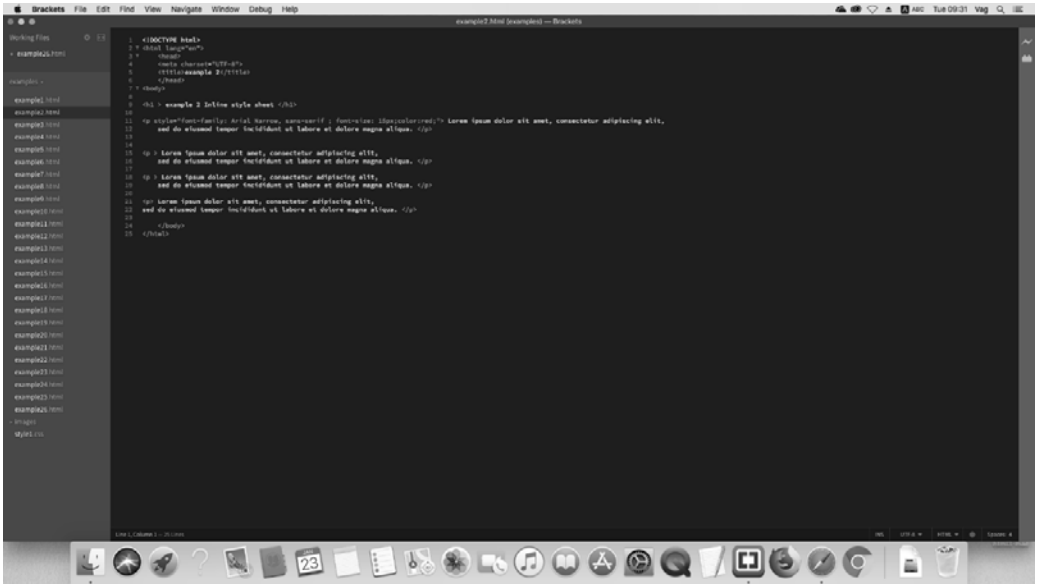
```
<p style="  font-family: Arial Narrow, sans-serif ;        font-size:     15px;color:red;
"> Lorem ipsum dolor sit amet, consectetur adipiscing elit,

   sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. </p>
```

Σχόλια

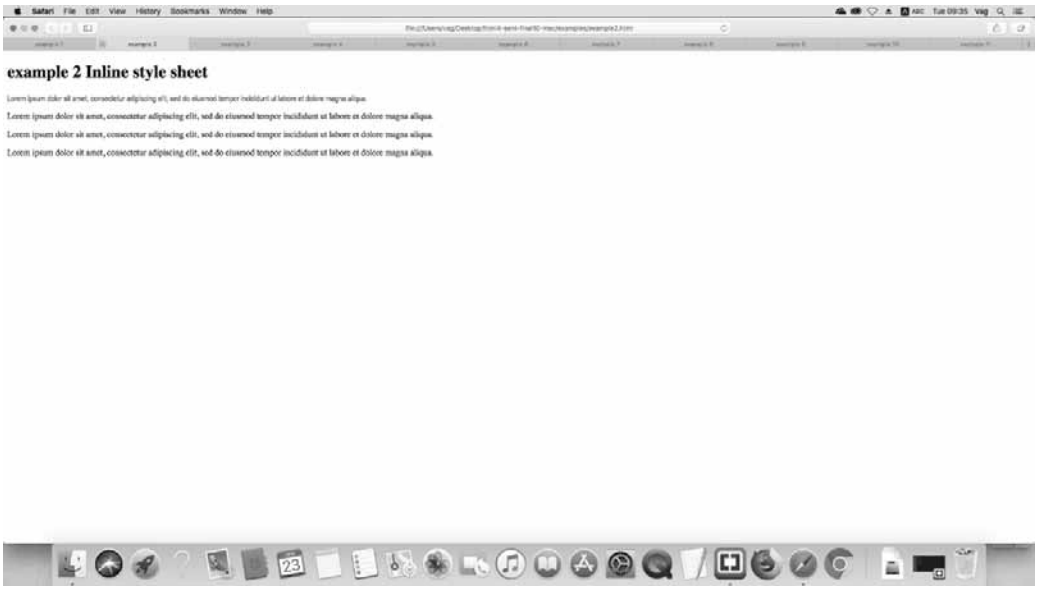
Εντός την ετικέτας έναρξης του στοιχείου p γράψαμε τις CSS ιδιότητες ανάμεσα στα διπλά εισαγωγικά της δήλωσης style. Στο Inline style δεν προσθέτουμε και τα άγκιστρα για να ορίσουμε την αρχή και το τέλος των ιδιοτήτων.

Παράδειγμα 2. Inline Style Sheet

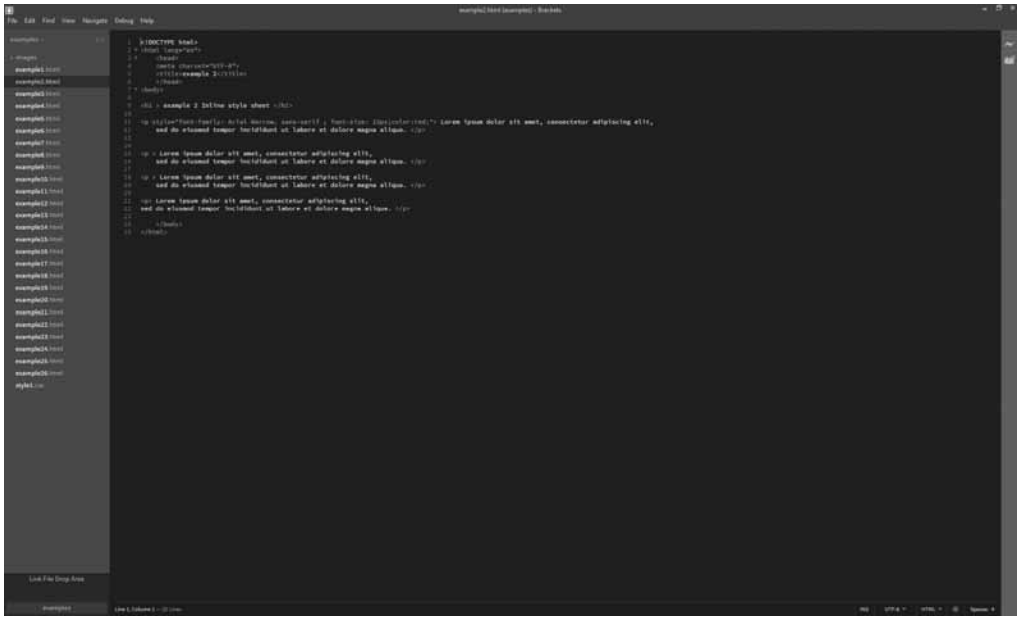
Κώδικας(Mac): inline style sheet



Εικόνα Browser(Mac): inline style sheet



Κώδικας(Win): inline style sheet



Εικόνα Browser(Win): inline style sheet

Multiple Style Sheets

Στην περίπτωση αυτή χρησιμοποιούνται δύο από τις τρεις ή και όλες οι μέθοδοι εισαγωγής CSS ιδιοτήτων. Όταν γίνεται επικάλυψη ιδιοτήτων CSS τότε εφαρμόζονται οι κανόνες προτεραιότητας. Αν για παράδειγμα σε ένα External sheet έχει οριστεί color:red για μία παράγραφο και σε ένα internal style sheet έχει οριστεί χρώμα για την παράγραφο color:blue θα πρέπει να εφαρμοστούν

οι κανόνες προτεραιότητας έτσι ώστε ο Browser να εμφανίσει την παράγραφο με ένα από τα δύο χρώματα.

Γενικά υπάρχει ο κανόνας για το CSS ότι σε μία ιδιότητα αν υπάρχουν για την ίδια ιδιότητα παραπάνω από μία τιμές μετράει αυτή που θα διαβαστεί και θα εκτελεστεί τελευταία.

Τα inline style sheet έχουν πάντα προτεραιότητα.

Παράδειγμα

Έστω ένα Internal style sheet εντός του στοιχείου head μιας σελίδας:

```
<style>
        p {
        color:red;
    color:blue;
    color:green;
          }
      </style>
```

Το χρώμα που θα εφαρμοστεί στα στοιχεία p θα είναι το πράσινο εφόσον η τιμή red αρχικά θα αντικατασταθεί από την τιμή blue και η τιμή blue θα αντικατασταθεί από την τιμή green.

Προτεραιότητες

1. Πρώτη προτεραιότητα έχουν τα inline style sheets
2. Δεύτερη προτεραιότητα έχουν τα internal ή τα external style sheets ανάλογα με το ποιο από τα δύο θα διαβαστεί τελευταίο.

Επειδή και τα δύο style sheets έμμεσα ή άμεσα δηλώνονται στο στοιχείο Head της σελίδας υπερισχύει αυτό που έχει γραφεί τελευταίο.

Οπότε αν στο στοιχείο head γράψουμε πρώτα το internal και έπειτα το link για το external θα μετρήσει το external.

```
<head>
<style>
...
...
...
</style>
<link rel="stylesheet" type="text/css" href="path/όνομα_αρχείου.css">
</head>
```

Αν γράψουμε τα style sheets με ανάποδη σειρά θα μετρήσει το internal

```
<head>
<link rel="stylesheet" type="text/css" href="path/όνομα_αρχείου.css">
<style>
...
...
...
</style>
</head>
```

1.2 CSS selectors

Για να επιλέξουμε στοιχεία στα οποία θα εφαρμοστούν οι CSS κανόνες που έχουμε δημιουργήσει θα πρέπει να χρησιμοποιήσουμε τους CSS selectors.

Class selector

Επιλέγει όλα τα στοιχεία που έχουν τη συγκεκριμένη κλάση.

Τα βήματα που θα πρέπει να ακολουθήσουμε είναι τα παρακάτω:

1. Αρχικά στο CSS κώδικα γράφουμε τον χαρακτήρα τελεία κα αμέσως μετά το όνομα της κλάσης. Στη συνέχεια εντός αγκίστρων γράφουμε το σύνολο των CSS ιδιοτήτων της κλάσης.

```
<style>
.όνομα_κλάσης {… }
</style>
```

Παράδειγμα

```
.class1 {
        color:orangered;
          }
      </style>
```

2. Εντός του στοιχείου body σε κάθε στοιχείο που επιθυμούμε να εφαρμόσουμε την κλάση θα γράψουμε τη δήλωση class="όνομα_κλάσης" χωρίς να γράψουμε την τελεία πριν.

```
<body>
<h1 class="class1"> Class Selector</h1>
<p class="class1"> Lorem ipsum dolor sit amet, consectetur adipiscing elit,
  sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. </p>
</body>
```

Σημείωση

Σε ένα στοιχείο μπορούμε να εφαρμόσουμε παραπάνω από μία κλάσεις γράφοντας στην ετικέτα έναρξης στη δήλωση class=" " τις κλάσεις που επιθυμούμε αφήνοντας τουλάχιστον ένα κενό χαρακτήρα ανάμεσά τους.

class="όνομα_κλάσης1 όνομα_κλάσης2 όνομα_κλάσης3 "

Επίσης μπορούμε να επιλέξουμε από όλα τα στοιχεία του ίδιου τύπου μόνο τα στοιχεία που έχουν τη συγκεκριμένη κλάση.

Στον κώδικα CSS θα γράψουμε:

```
<style>
όνομα_στοιχείου._όνομα_κλάσης {
…
…
…
}
</style>
```

Στη συνέχεια σε ίδιου τύπου στοιχεία θα γράψουμε στην ετικέτα έναρξης:

class= “όνομα_κλάσης”

Παράδειγμα

Έστω ότι θέλουμε να γράψουμε CSS ιδιότητες για να μορφοποιήσουμε μόνο τα στοιχεία h1 που έχουν κλάση .style4 και να μετατρέψουμε το χρώμα της γραμματοσειράς σε πράσινο.

```
<style>
        h1.style4 {
        color:green;
          }
    </style>
```

Στο στοιχείο body θα εφαρμοστεί η συγκεκριμένη κλάση στα στοιχεία που πληρούν τις παρακάτω προϋποθέσεις:

- να είναι στοιχεία h1 και
- να έχουν κλάση style4

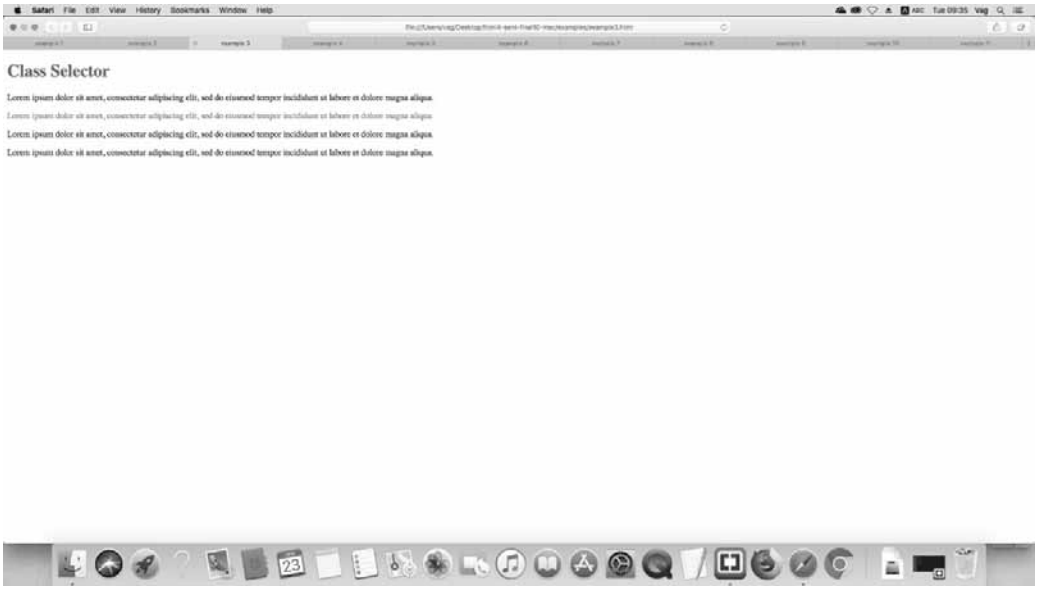
Παράδειγμα

```
<h1 class="style4"> Class Selector</h1>
```

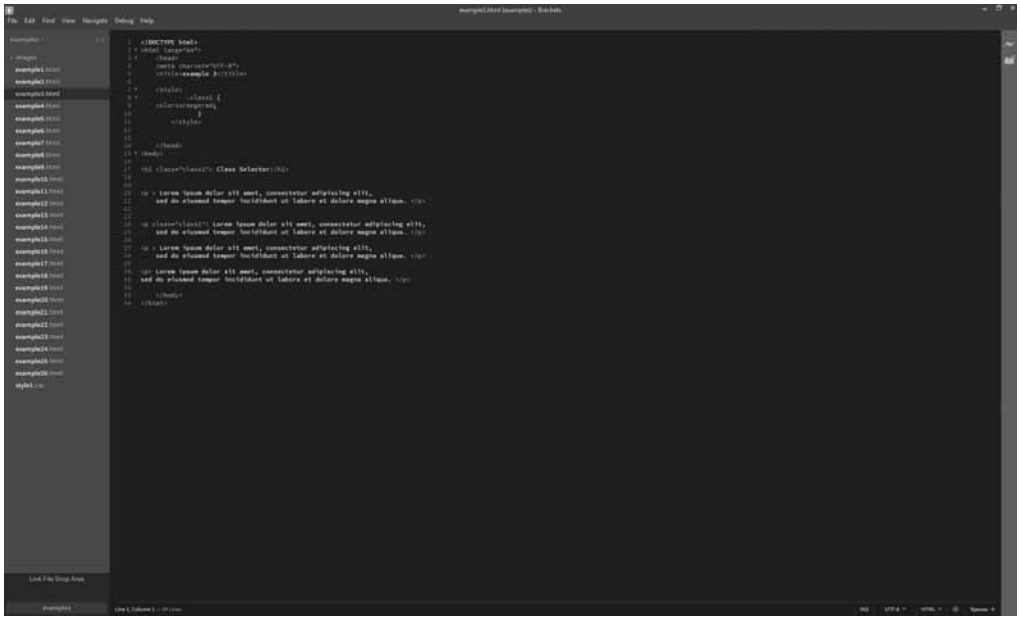
Παράδειγμα 3. Class Selector

Κώδικας(Mac): class selector

Εικόνα Browser(Mac): class selector



Κώδικας(Win): class selector



Εικόνα Browser(Win): class selector

Element selectors

Γράφοντας ένα element selector μπορούμε να επιλέξουμε όλα τα στοιχεία ίδιου τύπου που υπάρχουν εντός του στοιχείου Body. Για να γράψουμε ένα element selector γράφουμε το όνομα του στοιχείου και έπειτα μεταξύ των αγκίστρων γράφουμε τις ιδιότητες του κώδικα:

```
στοιχείο {
...
...
...
}
```

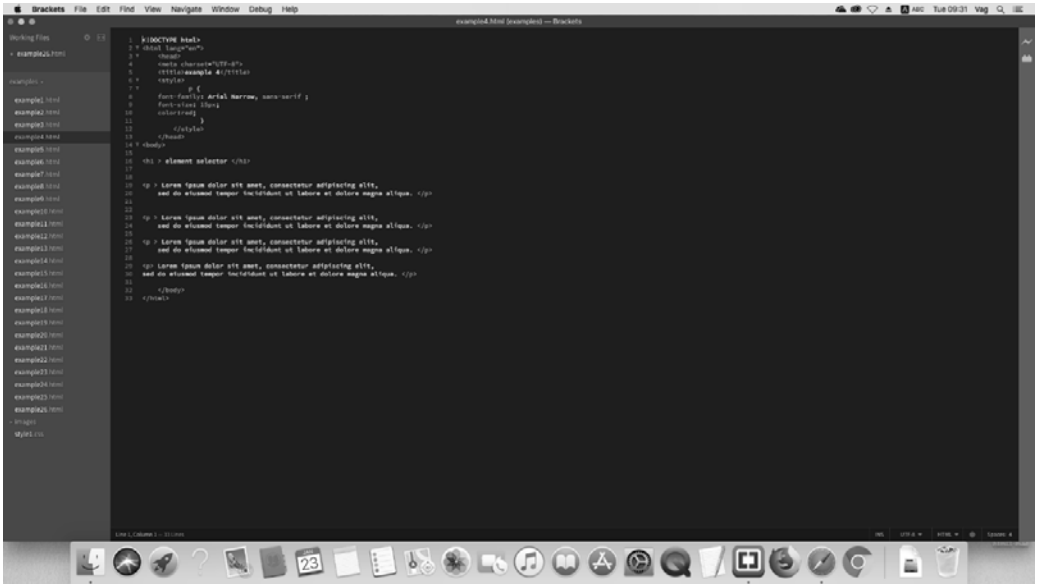
Παράδειγμα

```
h4 {color:yellow}
```

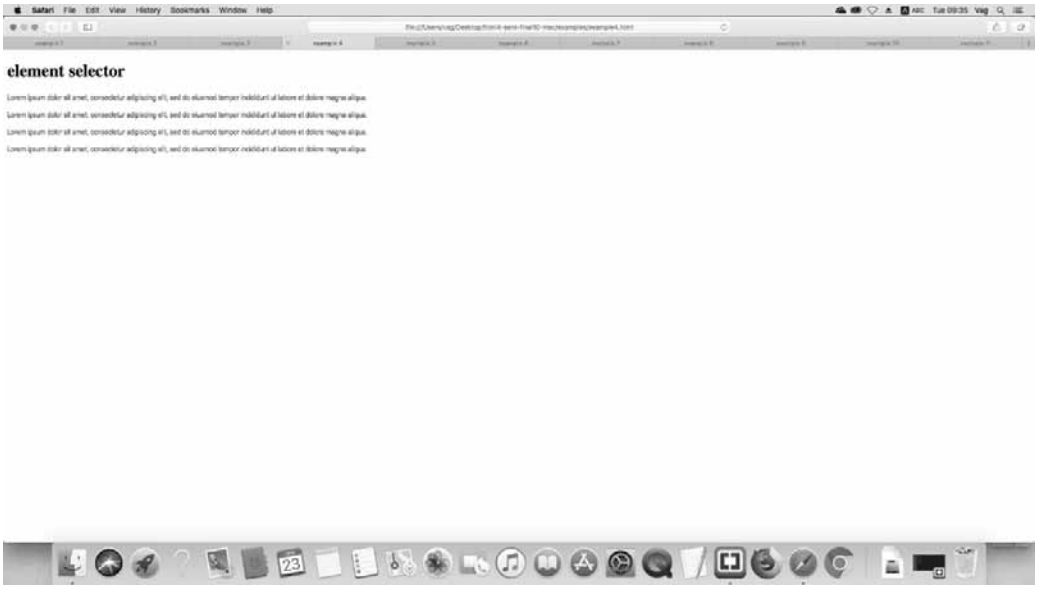
Ο συγκεκριμένος επιλογέας θα επιλέξει όλα τα στοιχεία h4 και θα εφαρμόσει το κίτρινο χρώμα στη γραμματοσειρά.

Παράδειγμα 4. Element Selector

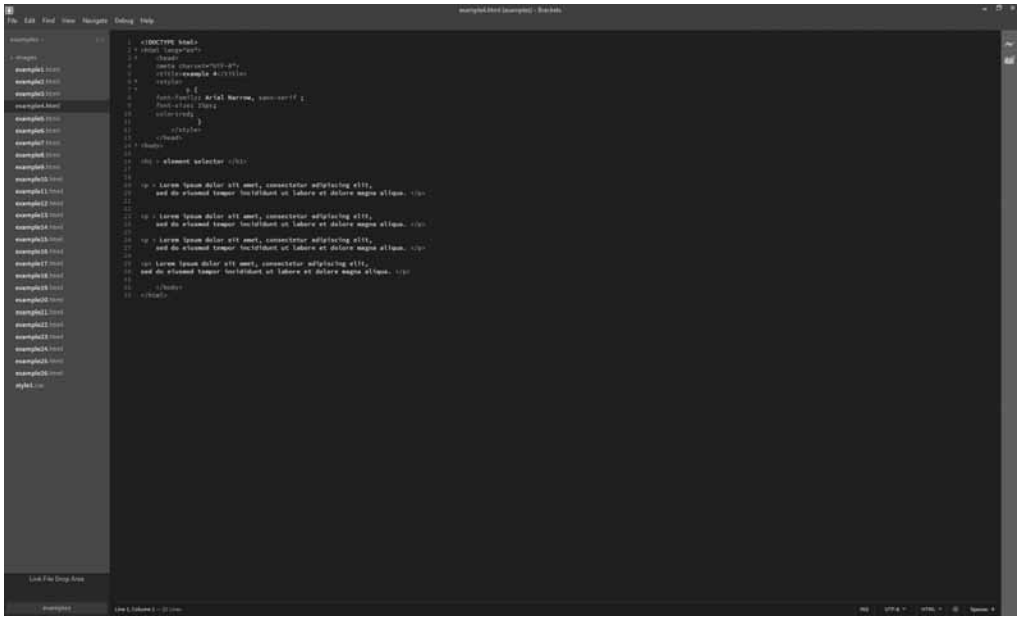
Κώδικας(Mac): element selector



Εικόνα Browser(Mac): element selector



Κώδικας(Win): element selector



Εικόνα Browser(Win): element selector

Element, element selectors

Μπορούμε να γράψουμε το ίδιο set ιδιοτήτων για παραπάνω από ένα στοιχεία:

```
στοιχείο1, στοιχείο2 {
...
...
...
}
```

Pseudoclass

Μία ψευδοκλάση χρησιμοποιείται για να επιλέξει μια ειδική κατάσταση ενός στοιχείου και να εφαρμόσει CSS style όπως για παράδειγμα:

- αιώρηση του mouse επάνω από το στοιχείο
- links που έχει πατήσει ή όχι ο χρήστης
- όταν κάνουμε κλικ σε κάποιο στοιχείο (focus)

Η σύνταξη της ψευδοκλάσης είναι η παρακάτω:

```
selector: state {
…
}
```

Ο επιλογέας μπορεί να είναι στοιχείο, κλάση κτλ.

Ειδικά για τα στοιχεία α θα πρέπει τις ψευδοκλάσεις να τις γράφουμε με την ακόλουθη σειρά:

1. unvisited link

```
a :link{…}
```

2. visited link

```
a:visited {…}
```

3. hover κατάσταση, για αιώρηση του Mouse επάνω από το link

```
a:hover {…]
```

4. active link

```
a:active {…}
```

Παράδειγμα

```
        /* unvisited link */
a:link {
   color: darkblue;
}
/* visited link */
a:visited {
   color: orangered;
}
/* mouse over link */
a:hover {
   color: skyblue;
}
/* selected link */
a:active {
   color: lightpink;
}
      </style>
```

Σχόλια

Για την κάθε κατάσταση της σύνδεσης έχουμε γράψει αντίστοιχο CSS.

Για να εφαρμοστεί το CSS με την ψευδοκλάση θα πρέπει να εισάγουμε ένα στοιχείο a εντός του στοιχείου body:

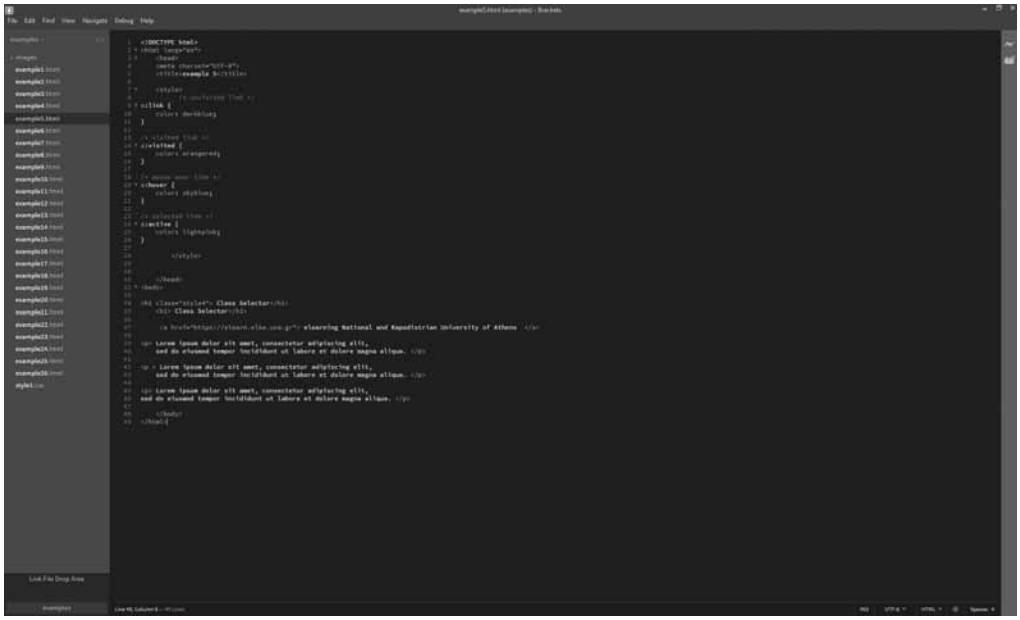
```
<p>   <a href="https://elearn.elke.uoa.gr" target="_blank"a> elearning National and Kapodistrian University of Athens  </a>  </p>
```

Παράδειγμα 5. Ψευδοκλάση

Κώδικας(Mac): Ψευδοκλάση

Εικόνα Browser(Mac): Ψευδοκλάση

Κώδικας(Win): Ψευδοκλάση



Εικόνα Browser(Win): Ψευδοκλάση

Αιώρηση επάνω από div

Μπορούμε με ψευδοκλάση να ορίσουμε ιδιότητες για την hover κατάσταση επάνω από κάποιο div σύμφωνα με την παρακάτω σύνταξη:

```
div:hover {
    background-color: yellow;
}
```

Παράδειγμα

```
div:hover {
    background-color: yellow;
}
```

Σημείωση

Μπορούμε να ορίσουμε σε ένα div ιδιότητες για κατάσταση hover αλλά σε συγκεκριμένο στοιχείο σύμφωνα με την παρακάτω σύνταξη:

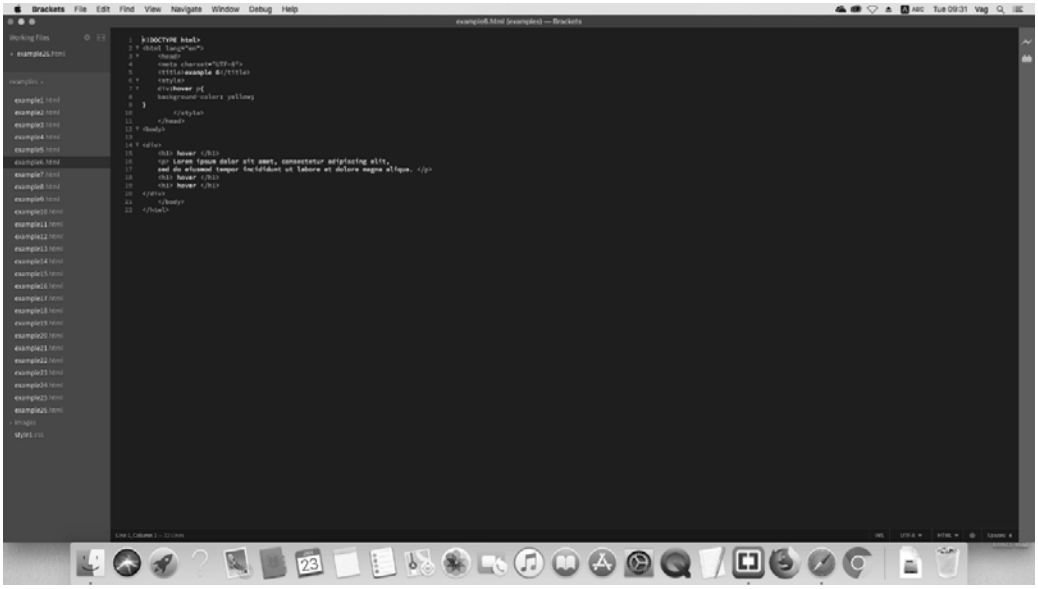
```
div:hover όνομα_στοιχείου {
...
}
```

Παράδειγμα

```
div:hover  p{
    background-color: yellow;
}
```

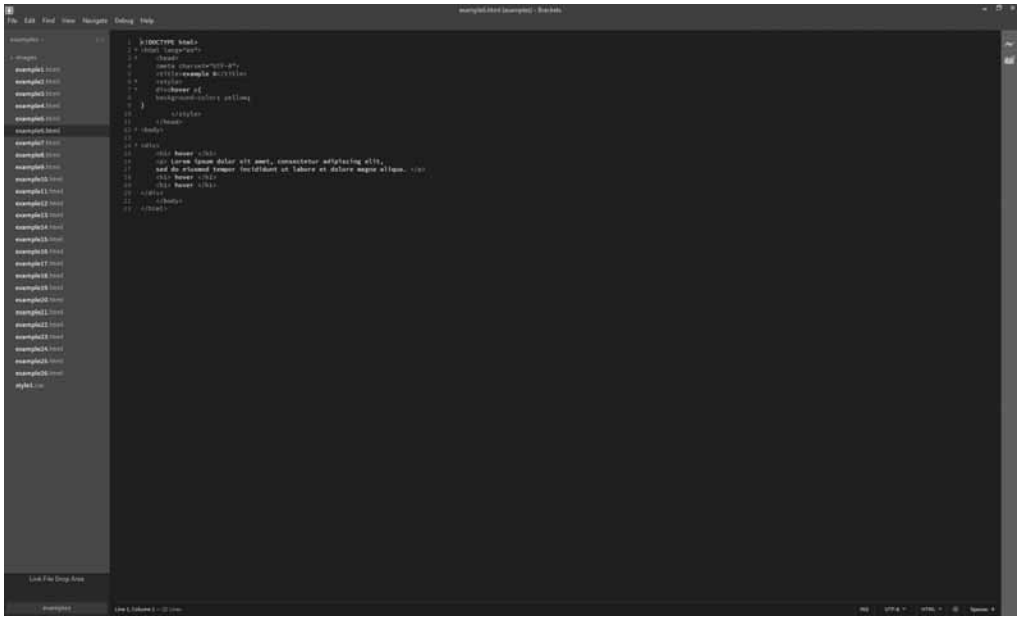
Παράδειγμα 6. div - hover

Κώδικας(Mac): div - hover



Εικόνα Browser(Mac): div - hover

Κώδικας(Win): div - hover



Εικόνα Browser(Win): div - hover

Id selector

Για να εφαρμόσουμε CSS με id selector θα πρέπει να κάνουμε τα παρακάτω βήματα!

Στον κώδικα CSS θα γράψουμε:

```
#αναγνωριστικό { }
```

Μεταξύ του χαρακτήρα δίεση και του αναγνωριστικού δεν θα πρέπει να υπάρχει κενός χαρακτήρας.

Εντός του στοιχείου Body για να εφαρμοστούν οι ιδιότητες CSS στην ετικέτα έναρξης του στοιχείου θα γράψουμε:

id="αναγνωριστικό"

έτσι ώστε να εφαρμοστούν οι ιδιότητες. Επίσης μπορούμε να γράψουμε και σετ ιδιοτήτων για περισσότερα από ένα id αρκεί να διαχωρίζονται μεταξύ τους από τον χαρακτήρα κόμμα.

Παράδειγμα

Θα γράψουμε κώδικα CSS και θα ορίσουμε ιδιότητες για το αναγνωριστικό style1:

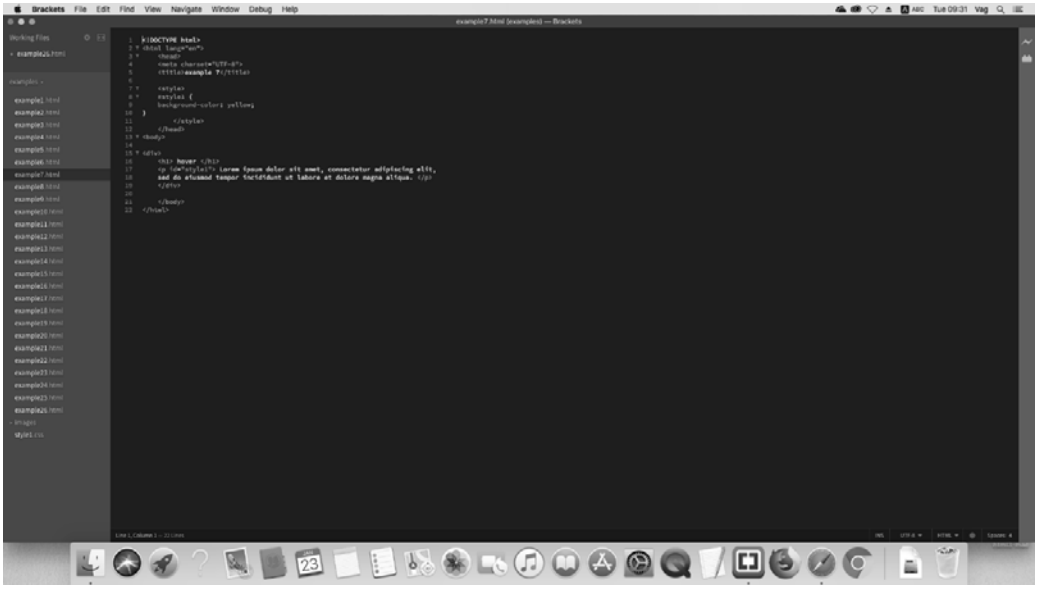
```
<style>
   #style1 {
   background-color: yellow;
}
      </style>
```

Στη συνέχεια εντός του στοιχείου Body για να εφαρμόσουμε το σετ κανόνων του CSS σε στοιχείο p θα πρέπει να γράψουμε:

```
<p id="style1"> Lorem ipsum dolor sit amet, consectetur adipiscing elit,
   sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. </p>
```

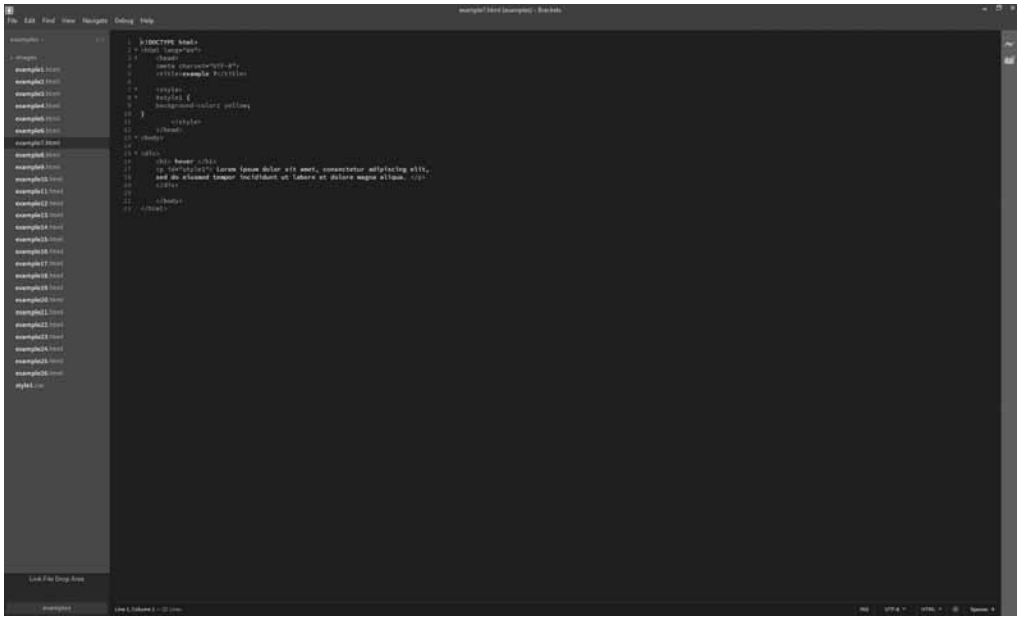
Παράδειγμα 7. id Selector

Κώδικας(Mac): id selector



Εικόνα Browser(Mac): id selector

Κώδικας(Win): id selector



Εικόνα Browser(Win): id selector

CSS ιδιότητες

1. color για το χρώμα της γραμματοσειράς. Η ιδιότητα color η οποία έχει ήδη αναλυθεί.
2. background-color. Η συγκεκριμένη ιδιότητα αναφέρεται στο χρώμα του φόντου. Για την επιλογή των χρωμάτων και τους χρωματικούς κώδικες ισχύουν τα ίδια όπως και με την ιδιότητα color.

3. background-image: Μπορούμε να ορίσουμε μια εικόνα ως φόντο για τη σελίδα μας με την παρακάτω σύνταξη:

```
body{
background-image: url("path/όνομα_αρχείου");
}
```

Παράδειγμα

```
body {
    background-image: url("images/maxresdefault.jpg");
    color:pink;
}
```

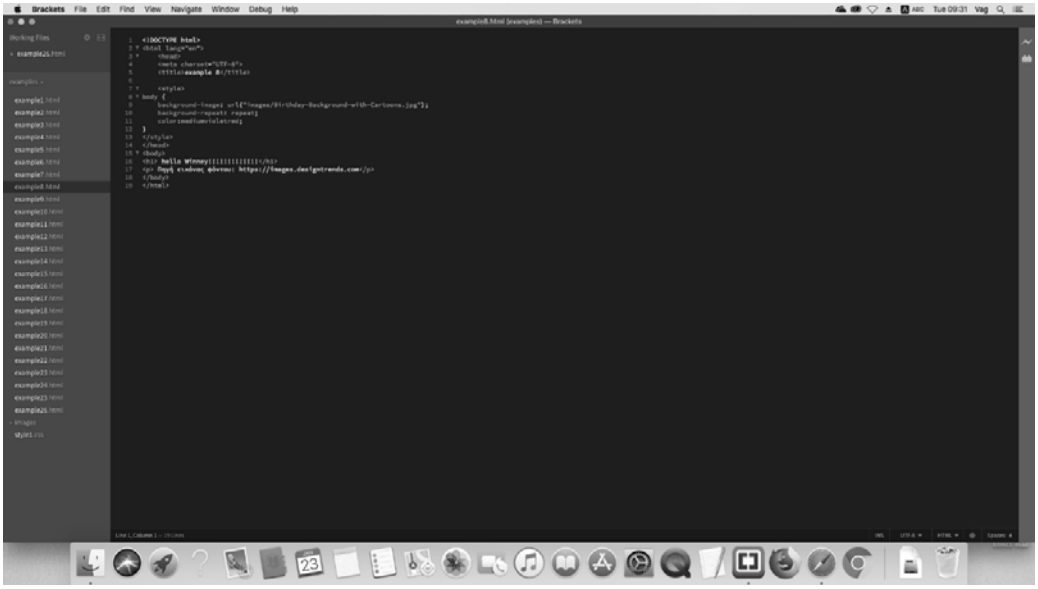
4. background-repeat. Με την ιδιότητα background-repeat εφόσον έχουμε γράψει την ιδιότητα background-image έχουμε επιλογές σε σχέση με την επανάληψη της εικόνας στο background της σελίδας σύμφωνα με την παρακάτω σύνταξη:

```
background-repeat: τιμή;
```

background-repeat	
repeat	Θα γίνει επανάληψη της εικόνας κάθετα και οριζόντια. Η τελευταία εικόνα θα περικοπεί αν δεν χωράει στο background.
repeat-x	Θα γίνει επανάληψη της εικόνας οριζόντια. Η τελευταία εικόνα θα περικοπεί αν δεν χωράει στο background.
repeat-y	Θα γίνει επανάληψη της εικόνας κάθετα. Η τελευταία εικόνα θα περικοπεί αν δεν χωράει στο background.
no-repeat	Η εικόνα στο background θα εμφανιστεί μόνο μία φορά.
space	Θα γίνει επανάληψη της εικόνας κάθετα και οριζόντια. Η τελευταία εικόνα αν δεν χωράει στο background θα αφαιρεθεί. Δηλαδή θα έχουμε το μέγιστο αριθμό επαναλήψεων χωρίς περικοπή εικόνας.
round	Συμπίεση και επέκταση της εικόνας για να καλύψει τον απαιτούμενο χώρο.
initial	Θέτει την προεπιλεγμένη τιμή.
inherit	Κληρονομεί την ιδιότητα από το πατρικό στοιχείο.

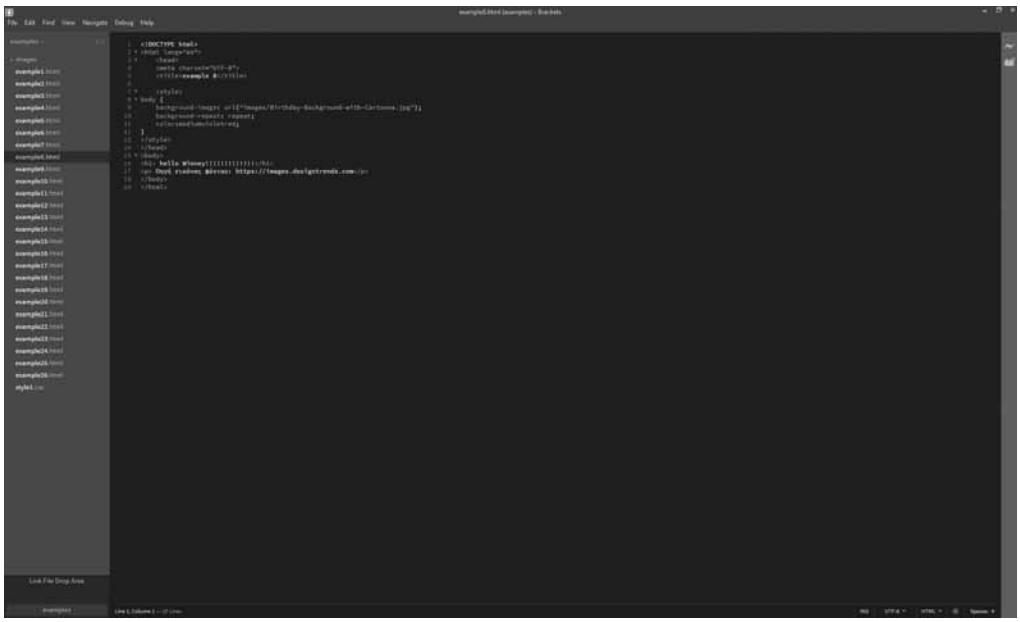
Παράδειγμα 8. Background image

Κώδικας(Mac): Background image

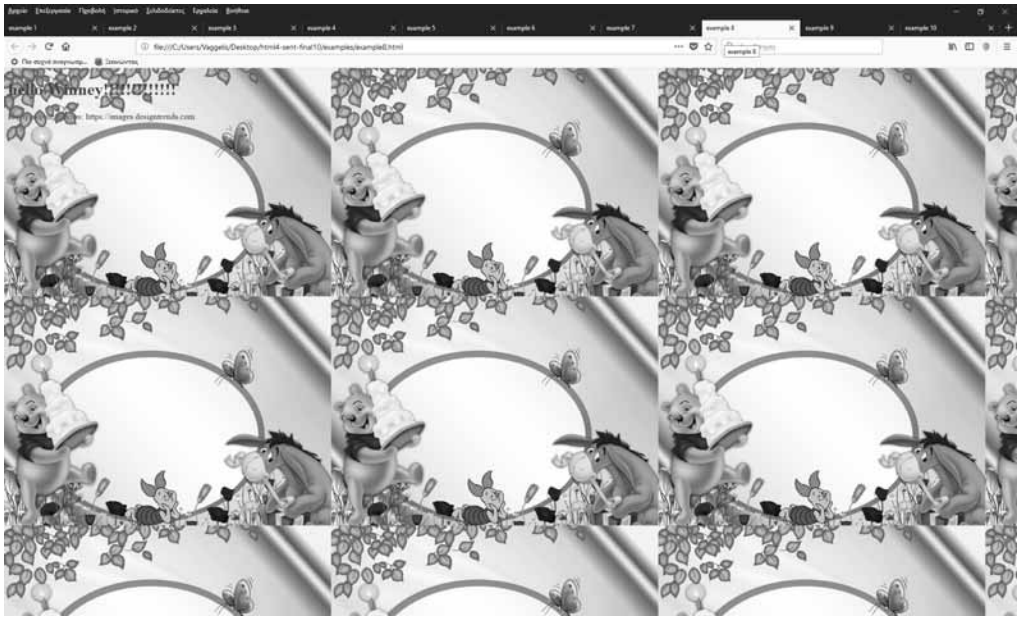


Εικόνα Browser(Mac): Background image

Κώδικας(Win): Background image

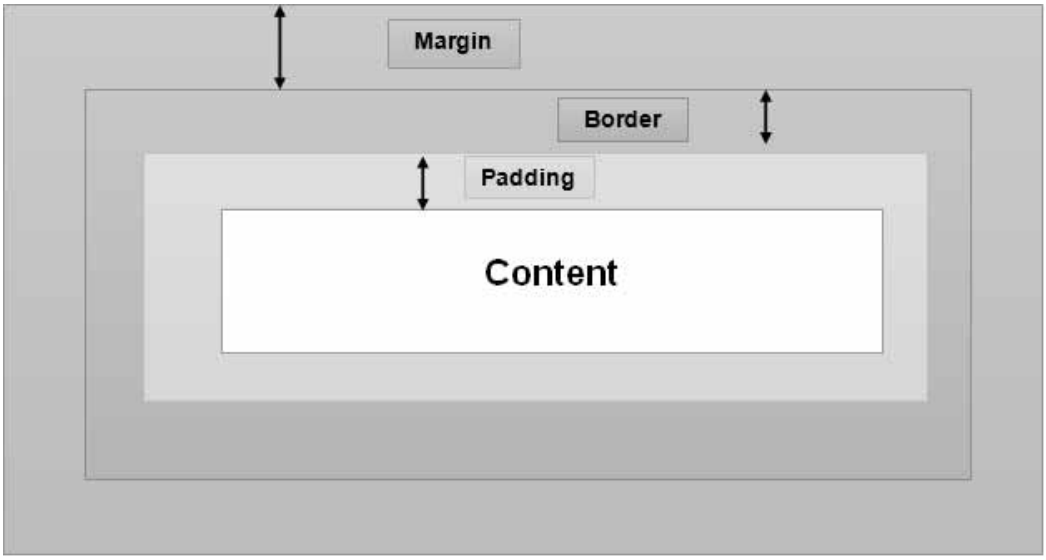


Εικόνα Browser(Win): Background image



2.1 CSS Box Model

Εικόνα 1. CSS Box Model



Το CSS Box Model είναι ένα μοντέλο που ορίζει αποστάσεις μεταξύ των στοιχείων της ιστοσελίδας και προσθέτει περίγραμμα γύρω από τα στοιχεία.

Η δομή του CSS Box model χωρίζεται στις παρακάτω περιοχές από μέσα προς τα έξω:

1. Content. Είναι η περιοχή που καταλαμβάνει το κάθε είδους περιεχόμενο όπως εικόνες, κείμενο κτλ.
2. Padding. Είναι μία περιοχή γύρω από την περιοχή Content. Η περιοχή padding είναι διαφανής.
3. Border. Είναι το περίγραμμα γύρω από την περιοχή padding η οποία περιβάλει την περιοχή padding και την περιοχή Content.
4. Margin. Είναι η περιοχή γύρω από το Border και είναι και αυτή διαφανής όπως και η περιοχή Padding.

Σημείωση

Όταν ορίζουμε διαστάσεις με τις ιδιότητες width και height αναφερόμαστε στις διαστάσεις της περιοχής Content. Για να υπολογίσουμε τις πραγματικές

διαστάσεις που καταλαμβάνει ένα τμήμα της σελίδας θα πρέπει να συνυπολογιστούν οι τιμές για padding, border και margin.

Padding

Η ιδιότητα padding έχει ως αποδεκτές τις παρακάτω τιμές:

- Length όπου μετράμε το padding σε μονάδες px, cm κτλ.
- % προσδιορίζει ποσοστό % του πλάτους
- inherit ορίζει padding που κληρονομείται από τους γονείς.

Αν επιθυμούμε να προσδιορίσουμε padding με ίδιες τιμές για όλες τις πλευρές τότε γράφουμε:

padding: _px

Αν επιθυμούμε να ορίσουμε padding με διαφορετικές τιμές για την κάθε πλευρά γράφουμε:

Padding ξεχωριστό για κάθε πλευρά	
padding-top:	για padding στην επάνω πλευρά
padding-right:	για padding στην δεξιά πλευρά
padding-bottom:	για padding στην κάτω πλευρά
padding-left:	για padding στην αριστερή πλευρά

Padding Shorthand

Με το padding shorthand έχουμε τη δυνατότητα να ορίσουμε padding για όλες τις πλευρές σε μία μόνο ιδιότητα.

Padding shorthand με τέσσερις τιμές

padding: a b c d;

Οι αντιστοιχίσεις των παραμέτρων σε σχέση με τις τιμές είναι οι ακόλουθες:

padding ξεχωριστό για κάθε πλευρά	
a	για padding στην επάνω πλευρά
b	για padding στην δεξιά πλευρά
c	για padding στην κάτω πλευρά
d	για padding στην αριστερή πλευρά

Ένας καλός τρόπος να θυμόμαστε τη λειτουργία στο shorthand με τις τέσσερις τιμές είναι ένα ρολόι όπου οι δείκτες του ξεκινάνε από τις 12:00 και δείχνει την επάνω πλευρά του padding και στη συνέχεια με τη φορά των δεικτών του ρολογιού δείχνει τις υπόλοιπες.

Σημείωση

Τι θα πρέπει να προσέξουμε στο shorthand:

1. ανάμεσα στην τιμή και τη μονάδα δεν θα πρέπει να υπάρχει κενός χαρακτήρας. Για παράδειγμα το σωστό είναι 12px και όχι 12 px.
2. Ανάμεσα στις ξεχωριστές τιμές αφήνουμε κενό χαρακτήρα π.Χ. 12px 14px.
3. Δεν γράφουμε τον χαρακτήρα ',' ανάμεσα στις διαδοχικές τιμές.

Παράδειγμα

padding 12px 14px 10px 8px;

Padding shorthand με τρεις τιμές

padding: k l m;

Οι αντιστοιχίσεις των παραμέτρων σε σχέση με τις τιμές είναι οι ακόλουθες:

	Padding ξεχωριστό για κάθε πλευρά
k	για padding στην επάνω πλευρά
l	για padding στην δεξιά και αριστερή πλευρά
m	για padding στην κάτω πλευρά

Στο συγκεκριμένο shorthand η μεσαία τιμή ορίζει και τα δύο πλευρικά padding

Παράδειγμα

Padding: 12px 14px 10px;

Padding shorthand με δύο τιμές

Padding: s t;

Οι αντιστοιχίσεις των παραμέτρων σε σχέση με τις τιμές είναι οι ακόλουθες:

	Padding ξεχωριστό για κάθε πλευρά
s	για padding στην επάνω και κάτω πλευρά
t	για padding στην δεξιά και αριστερή πλευρά

Όταν το Padding shorthand έχει δύο τιμές η πρώτη ορίζει το επάνω και το κάτω padding, ενώ η δεύτερη ορίζει τα πλευρικά padding.

Padding shorthand με μία τιμή

padding: r;

Το padding με μία τιμή ορίζει padding για όλες τις πλευρές.

Margin

Η ιδιότητα margin ορίζεται ακριβώς όπως και η ιδιότητα padding και padding shorthand.

Η ιδιότητα margin έχει ως αποδεκτές τις παρακάτω τιμές:

- Length όπου μετράμε το margin σε μονάδες px, cm κτλ.
- % προσδιορίζει ποσοστό % του πλάτους
- inherit ορίζει margin που κληρονομείται από τους γονείς.

Αν επιθυμούμε να προσδιορίσουμε margin με ίδιες τιμές για όλες τις πλευρές τότε γράφουμε:

margin: _px

Αν επιθυμούμε να ορίσουμε margin με διαφορετικές τιμές για την κάθε πλευρά γράφουμε:

margin ξεχωριστό για κάθε πλευρά	
margin-top:	για margin στην επάνω πλευρά
margin-right:	για margin στην δεξιά πλευρά
margin-bottom:	για margin στην κάτω πλευρά
margin-left:	για margin στην αριστερή πλευρά

Margin Shorthand

Με το margin shorthand έχουμε τη δυνατότητα να ορίσουμε margin για όλες τις πλευρές σε μία μόνο ιδιότητα.

Margin shorthand με τέσσερις τιμές

Margin: a b c d;

Οι αντιστοιχίσεις των παραμέτρων σε σχέση με τις τιμές είναι οι ακόλουθες:

Margin ξεχωριστό για κάθε πλευρά	
a	για margin στην επάνω πλευρά
b	για margin στην δεξιά πλευρά
c	για margin στην κάτω πλευρά
d	για margin στην αριστερή πλευρά

Παράδειγμα

margin 12px 14px 10px 8px;

Margin shorthand με τρεις τιμές

margin: k l m ;

Οι αντιστοιχίσεις των παραμέτρων σε σχέση με τις τιμές είναι οι ακόλουθες:

Margin ξεχωριστό για κάθε πλευρά	
k	για margin στην επάνω πλευρά
l	για margin στην δεξιά και αριστερή πλευρά
m	για margin στην κάτω πλευρά

Στο συγκεκριμένο shorthand η μεσαία τιμή ορίζει και τα δύο πλευρικά margin.

Παράδειγμα

margin: 12px 14px 10px;

margin shorthand με δύο τιμές

margin: s t;

Οι αντιστοιχίσεις των παραμέτρων σε σχέση με τις τιμές είναι οι ακόλουθες:

margin ξεχωριστό για κάθε πλευρά	
s	για margin στην επάνω και κάτω πλευρά
t	για margin στην δεξιά και αριστερή πλευρά

Όταν το margin shorthand έχει δύο τιμές η πρώτη ορίζει το επάνω και το κάτω margin, ενώ η δεύτερη ορίζει τα πλευρικά margin.

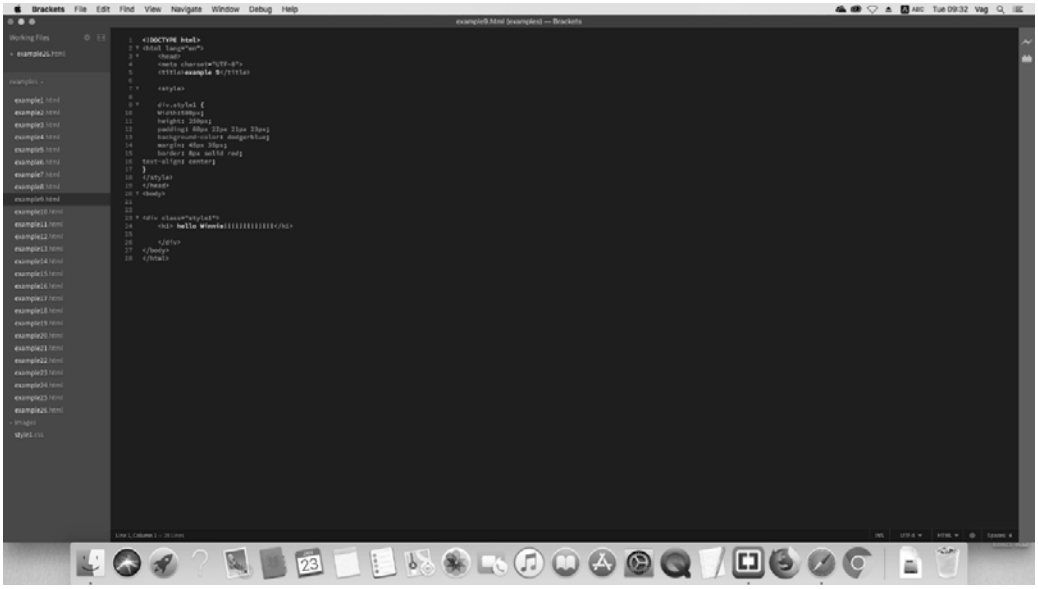
Margin shorthand με μία τιμή

margin: r;

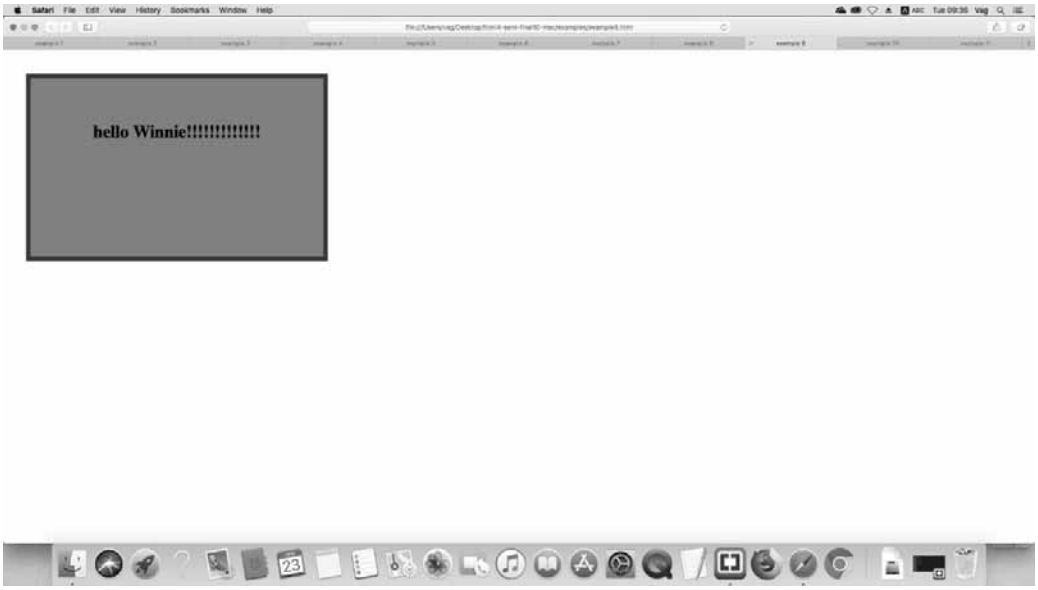
Το margin με μία τιμή ορίζει margin για όλες τις πλευρές.

Παράδειγμα 9. CSS Box Model

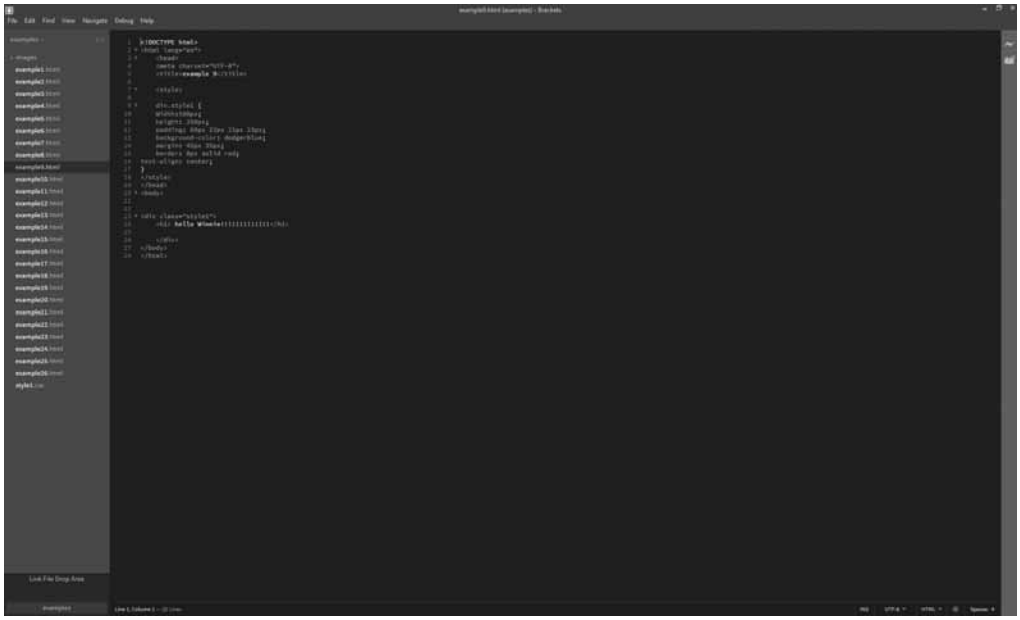
Κώδικας(Mac): CSS Box Model



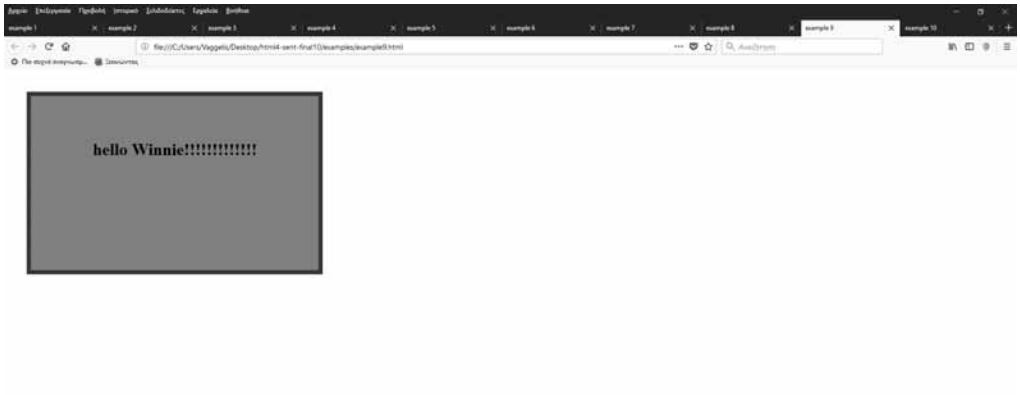
Εικόνα Browser(Mac): CSS Box Model



Κώδικας(Win): CSS Box Model



Εικόνα Browser(Win): CSS Box Model



2.2 CSS layouts και navigation

Block elements

Ένα block element της html ξεκινάει σε μία καινούργια γραμμή και καταλαμβάνει ολόκληρο το διαθέσιμο πλάτος της σελίδας.

Παραδείγματα block elements:

-
- <h1> - <h6>
- <p>
- <form>
- <header>
- <footer>

Inline elements

Τα inline elements δεν ξεκινούν από καινούργια γραμμή και καταλαμβάνουν μόνο το απαραίτητο πλάτος:

Παραδείγματα Inline elements:

- <a>
- <img>

Display

Με την ιδιότητα display μπορούμε να ορίσουμε τον τρόπο εμφάνισης των στοιχείων γράφοντας display:τιμή; και στη θέση της τιμής γράφουμε block ή inline.

Παράδειγμα:

```
img{
display:block;
}
```

Σχόλια

Εφαρμόσαμε display:block; σε ένα inline element με εφαρμοσμένη ιδιότητα display και την τιμή block.

Σημείωση

Θα πρέπει να γίνει κατανοητό ότι με τη μέθοδο display αλλάζουμε μόνο τον τρόπο εμφάνισης του στοιχείου για να ορίσουμε μία συγκεκριμένη εμφάνιση στη σελίδα μας. Ο τύπος του στοιχείου αν αλλάξουμε την προεπιλεγμένη τιμή της ιδιότητας display παραμένει ο ίδιος. Αν σε ένα inline element

εφαρμόσουμε display: block ; δεν μπορούμε να ενσωματώσουμε άλλα block elements μέσα σε αυτό.

Έτσι για παράδειγμα αν γράψουμε τρία στοιχεία a συνεχόμενα στον κώδικα θα εμφανιστεί το ένα δίπλα στο άλλο:

```
<a href="…">  </a>

<a href="…">  </a>

<a href="…">  </a>
```

Ενώ αν αλλάξουμε την ιδιότητα display γράφοντας `a{ display:block;}` θα εμφανιστούν το ένα δίπλα στο άλλο.

Σχόλια

Το στοιχείο a ως προεπιλεγμένη τιμή για την ιδιότητα display έχει την τιμή inline δηλαδή καταλαμβάνει μόνο το απαραίτητο πλάτος. Όταν με την ιδιότητα display και την τιμή block αλλάζουμε τον τρόπο εμφάνισης του στοιχείου, τότε θα ξεκινάει σε μία καινούργια γραμμή και θα καταλαμβάνει ολόκληρο το διαθέσιμο πλάτος της σελίδας. Έτσι λοιπόν θα δούμε τα στοιχεία a με εφαρμοσμένο το display:block θα εμφανίζονται το ένα κάτω από το άλλο.

Οι πιο σημαντικές τιμές για την ιδιότητα display είναι οι παρακάτω:

ιδιότητα display	
inline	εμφανίζει ένα στοιχείο όπως ένα inline element. Οι ιδιότητες width και height δεν έχουν κάποια επίδραση.
block	εμφανίζει ένα στοιχείο ως block element. ξεκινάει σε μία νέα γραμμή και καταλαμβάνει ολόκληρο το διαθέσιμο πλάτος της σελίδας.
flex	εμφανίζει ένα στοιχείο ως block-level flex container.
inline-block	εμφανίζει ένα στοιχείο ως inline-level block container. Το στοιχείο έχει format ενός inline element με τη διαφορά ότι μπορούμε να ορίσουμε διαστάσεις με τις ιδιότητες width και height.
inline-flex	εμφανίζει το στοιχείο ως inline-level flex container

Navigation-Bar

Μπορούμε εύκολα και γρήγορα να δημιουργήσουμε ένα κάθετο navigation bar. Η λογική που θα ακολουθήσουμε είναι να δημιουργήσουμε μία μη διατεταγμένη λίστα στην οποία θα αφαιρέσουμε τους σημειωτές και θα προσθέσουμε συνδέσεις.

Γράφουμε CSS κώδικα όπως για παράδειγμα εντός του στοιχείου head και ανάμεσα στις ετικέτες <style> και </style>.

Ιδιότητες για τη λίστα ul

Θα αφαιρέσουμε τους σημειωτές γράφοντας

```
list-style-type: none;
```

Θα γράψουμε τιμές μηδέν για margin και padding

```
margin: 0;
padding: 0;
```

Θα γράψουμε τιμές για πλάτος, background-color και border:

```
width: …;
background-color: …;
border: … ;
```

Παράδειγμα

```
ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
    width: 250px;
    background-color: #428ff4;
    border: 2px solid red;
}
```

Ιδιότητες για τα στοιχεία a της λίστας

Για να ορίσουμε μία clickable περιοχή και να προσθέσουμε padding, margin κτλ. θα γράψουμε:

```
display:block;
```

Για να μην έχουμε υπογράμμιση των συνδέσεων θα γράψουμε:

```
text-decoration: none;
```

Θα γράψουμε ιδιότητες για χρώμα γραμματοσειράς padding κτλ.

```
color: ... ;
padding: .... ;
```

Παράδειγμα

```
li a {
    display: block;
    color: white;
    padding: 14px 15px;
    text-decoration: none;
}
```

Ιδιότητες για τα list items

Θα γράψουμε style για όλα τα li elements γράφοντας text-align και border-bottom για να ορίσουμε το border στο κάτω μέρος του κάθε στοιχείου:

```
li {
    text-align: center;
    border-bottom: ...;
}
```

Παράδειγμα

```
li {
    text-align: center;
    border-bottom: 1px solid red;
}
```

Για να επιλέξουμε μόνο το τελευταίο στοιχείο για να εφαρμόσουμε να μην έχει border-bottom όπως τα υπόλοιπα θα γράψουμε:

```
li:last-child {
    border-bottom: none;
}
```

Για να γράψουμε style για το στοιχείο a το οποίο θα είναι active δηλαδή για να μας δείχνει σε ποια σελίδα βρισκόμαστε θα γράψουμε ιδιότητες όπως color και background-color:

```
li a.active {
    background-color: ...;
    color: white  ... ;
}
```

Παράδειγμα

```
li a.active {
    background-color: pink;
    color: white;
}
```

Τέλος για να προσδιορίσουμε style για την κατάσταση hover αλλά εξαιρώντας το στοιχείο a που θα είναι active γράφουμε:

```
li a:hover:not(.active) {
    background-color: ...;
    color: ...;
}
```

Παράδειγμα

```
li a:hover:not(.active) {
    background-color: grey;
    color: white;
}
```

Αφού ολοκληρώσουμε το CSS style θα γράψουμε τη μη διατεταγμένη λίστα στο στοιχείο Body δίνοντας την κλάση active στο κατάλληλο li στοιχείο.

Παράδειγμα

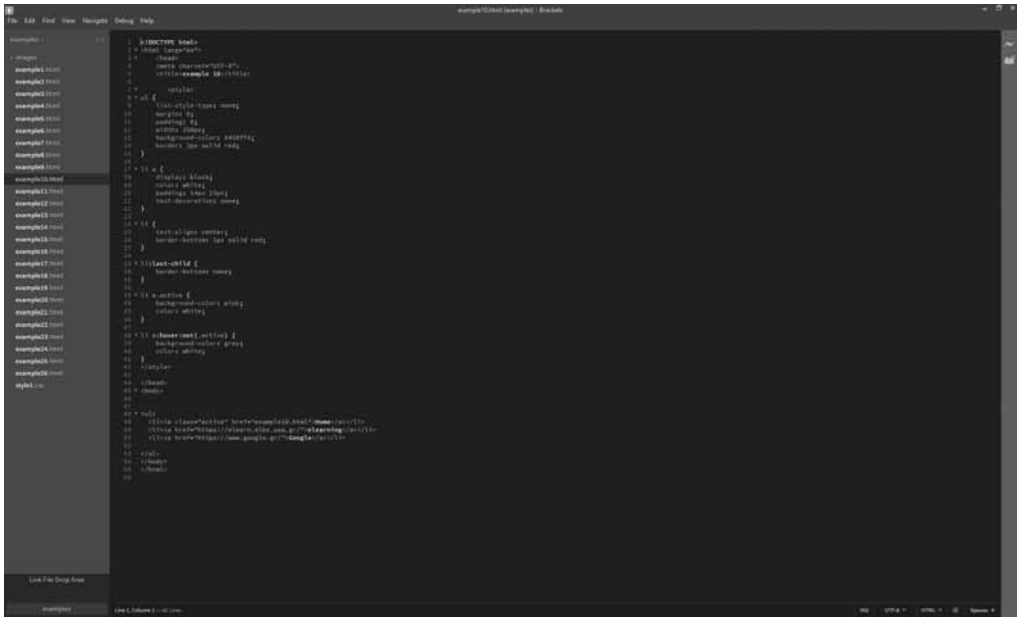
```
<ul>
  <li><a class="active" href="#home">Home</a></li>
  <li><a href="https://elearn.elke.uoa.gr/">elearning</a></li>
  <li><a href="https://www.google.gr/">Google</a></li>
</ul>
```

Παράδειγμα 10. Navigation

Κώδικας(Mac): navigation

Εικόνα Browser(Mac): navigation

Κώδικας(Win): navigation



Εικόνα Browser(Win): navigation

2.3 Εισαγωγή στο Flex Box

Το Flex Box είναι μία πολύ καλή λύση για να υλοποιούμε responsive σελίδες χωρίς να χρησιμοποιούμε ιδιότητες float καθώς και positioning,

Ένα flexbox layout αποτελείται από:

- ένα γονικό στοιχείο (flex-container)
- ένα ή περισσότερα παιδιά (child elements) τα οποία περιέχονται μέσα στο γονικό στοιχείο.

Αρχικά για να κατανοήσουμε τη λειτουργία του flexbox μπορούμε να φανταστούμε ένα εύκαμπτο κουτί (flex-container) και εντός του κουτιού ένα παχύρευστο υγρό που περιέχει ένα η περισσότερα αντικείμενα(child elements).

Καθώς το flex-container προσαρμόζεται στο διαθέσιμο χώρο(οι διαστάσεις του μεταβάλλονται) και μέσω του παχύρευστου υγρού μεταβάλλονται οι θέσεις και οι διαστάσεις των αντικειμένων που περιέχονται μέσα σε αυτό.

Το σύνολο των κανόνων που υποδεικνύουν πως θα μεταβάλλονται οι διαστάσεις και οι θέσεις του γονικού στοιχείου και των αντικειμένων που περιέχονται σε αυτό αποτελεί και την τεχνική του flex-box.

Τι πετυχαίνουμε με το flex-box

Με το flex-box πετυχαίνουμε την προσαρμοστικότητα της σελίδας σε διαφορετικούς τύπους συσκευών με οθόνες διαφορετικών διαστάσεων. Πολλές φορές θα έχει τύχει να ανοίξουμε μια σελίδα στο desktop και να εμφανίζεται άψογα αλλά όταν ελαχιστοποιήσουμε το παράθυρο του browser ή όταν δοκιμάσουμε τη σελίδα σε μια mobile συσκευή με μικρή οθόνη η δομή της σελίδας να εμφανίζεται προβληματική.

Αυτή τη δυσλειτουργία έρχεται να καλύψει το flex-box.

Ξεκινώντας με το Flex-box

Για να υλοποιήσουμε web σελίδες με flex layout αρχικά θα πρέπει να ενεργοποιήσουμε το flexbox και να δημιουργήσουμε ένα flex container εντός του οποίου θα ενσωματώσουμε τα αντικείμενα.

Για να δημιουργήσουμε το αρχικό flex-container θα γράψουμε μία κλάση όπως παρακάτω:

```
<style>
.flex-container {
 display: flex;
}
</style>
```

Το όνομα της κλάσης .flex-container είναι ενδεικτικό και όχι δεσμευτικό.

Η τιμή flex για την ιδιότητα display (display: flex ;) είναι αυτή που καθορίζει το layout της σελίδας και δίνει τη μορφή flex και ουσιαστικά ενεργοποιεί το flex-box.

Εντός του στοιχείου Body θα γράψουμε ένα div στο οποίο θα εφαρμόσουμε την κλάση .flex-container όπως παρακάτω:

```
<div class="flex-container">
Child1
Child2
Child3
...
</div>
```

Το αρχικό div με τη συγκεκριμένη κλάση θεωρείται το πατρικό στοιχείο. Κάθε παιδί που περιέχεται στο flex-container θα γράφεται ως div μέσα στο αρχικό πατρικό div.

Πριν όμως ενσωματώσουμε τα παιδιά θα πρέπει να γράψουμε και μία κλάση ορίζοντας margin, padding κτλ.

Όταν γράφουμε CSS το οποίο θα εφαρμοστεί σε όλα τα flex children θα γράφουμε το όνομα της αρχικής κλάσης του flex-container και στη συνέχεια θα γράφουμε >div.

Αν θεωρήσουμε ότι το όνομα κλάσης για το αρχικό flex-container είναι .flex-container τότε η κλάση που θα εφαρμοστεί σε όλα τα παιδιά είναι:

```
.flex-container>div {
background-color: #f1f1f1;
  margin: ...;
  padding: ... ;
  font-size: ...;
}
```

Σημείωση

Οι ιδιότητες για τα παιδιά είναι προαιρετικές.

Εντός του div που ορίζει το αρχικό flex container κάθε div που θα γράφουμε θεωρείται απόγονος-παιδί του αρχικού flex container και θα μορφοποιηθεί από την κλάση:

```
.flex-container>div
```

Για την καλύτερη κατανόηση έχουμε γράψει ένα στοιχείο h5 σε κάθε div χωρίς αυτό να είναι υποχρεωτικό.

```
<div class="flex-container">
<div>
<h5> Child 1</h5>
</div>
<div>
<h5> Child 2</h5>
</div>
<div>
<h5> Child 3</h5>
</div>
…
</div>
```

Με τη σειρά που θα γράψουμε στον κώδικα το κάθε παιδί (div) εντός του flex-container με την ίδια σειρά θα οριστούν και ως παιδιά του flex container. Έτσι λοιπόν το πρώτο div θα είναι το child 1, το δεύτερο div θα είναι το child 2 κοκ.

Η σειρά έχει διπλή σημασία γιατί:

- Με τη σειρά που θα γραφούν τα div με την ίδια σειρά θα τοποθετηθούν εντός του flex box container
- Εφόσον η σειρά ορίζει και την ταυτότητα τους child 1, child 2 κτλ. μπορούμε να δημιουργήσουμε κλάσεις οι οποίες θα αναφέρονται ειδικά στο πρώτο παιδί ή στο δεύτερο παιδί κτλ.

Για να εφαρμόσουμε ιδιότητες κλάσεων για συγκεκριμένα παιδιά του flex container θα γράψουμε:

```
.flex-container>div:nth-child(αριθμός_παιδιού)
```

Παράδειγμα

```
.flex-container>div:nth-child(3)
```

Σχόλια

Εφόσον η κλάση .flex-container εφαρμόζεται στο flex container τότε η παραπάνω κλάση θα εφαρμοστεί στο τρίτο παιδί του flex container.

Τα βήματα που πρέπει να κάνουμε για να δημιουργήσουμε ένα flex box layout είναι:

1. Δημιουργία του κεντρικού flex container.

Για να ενεργοποιήσουμε το CSS flexbox θα χρειαστούμε μια κλάση στην οποία απαραίτητα θα περιέχεται η δήλωση

```
display: flex;
```

```
<style>
.flex-container {
  display: flex;
}
</style>
```

2. Για να ολοκληρωθεί η δημιουργία του flex box container θα πρέπει εντός του στοιχείου Body να γράψουμε ένα div στο οποίο θα εφαρμόσουμε την παραπάνω κλάση

```
div class="flex-container">
…
…
…

</div>
```

3. Γράφοντας στοιχεία div εντός του Container ορίζουμε τα παιδιά του div και κατά επέκταση και του κεντρικού flex-container

```
div class="flex-container">
<div>
…
</div>

<div>
…
</div>

…
…

</div>
```

2.4 Selectors χρήσιμοι για το flex-box

Για να επιλέξουμε στοιχεία του flex box container χρησιμοποιούμε τους παρακάτω selectors.

Flex box Selectors	
κλάση	επιλογή
.flex-container	επιλέγει το flex container και όλα τα παιδιά του
.flex-container>div	επιλέγει όλα τα παιδιά του flex container
.flex-container>div:nth-child(n) όπου n ο αριθμός του παιδιού	επιλέγει το nth κατά σειρά παιδί.

Σημείωση

Η κλάση για το flex container μπορεί να έχει οποιοδήποτε όνομα όπως για παράδειγμα .flex1. Αν .flex1 είναι το όνομα της κλάσης, τότε ο επιλογέας για όλα τα παιδιά θα είναι:

.flex1>div

ενώ για ένα συγκεκριμένο παιδί ο επιλογέας θα είναι:

.flex1>div:nth-child(n)

Μεθοδολογία

Συνοπτικά η μεθοδολογία για τη δημιουργία flexbox layout βασίζεται στα παρακάτω βήματα:

1. Αρχικά γράφουμε την κλάση για το flex-container

.flex-container

2. Στη συνέχεια γράφουμε ιδιότητες που θα ισχύουν για όλα τα παιδιά

.flex-container>div

3. Τέλος γράφουμε ιδιότητες που θα ισχύσουν σε συγκεκριμένα παιδιά

.flex-container>div:nth-child(n)

Σημείωση

Γράφοντας CSS ιδιότητες για ένα συγκεκριμένο παιδί αν υπάρχει τιμή για κάποια κοινή ιδιότητα στην κλάση:

.flex-container>div

και στην κλάση

.flex-container>div:nth-child(n)

όπου n ο αριθμός του παιδιού υπερισχύει η τιμή της κλάσης

.flex-container>div:nth-child(n)

Παράδειγμα

Έστω ότι γράφουμε CSS για τα παιδιά του flex container

```
flex-container>div {
  background-color: #f1f1f1;
  margin: 10px;
  padding:20px;
  font-size: 30px;
color:blue;
}
```

Σχόλια

Με αυτή την κλάση όλα τα παιδιά του flex container θα έχουν μπλε χρώμα γραμματοσειράς.

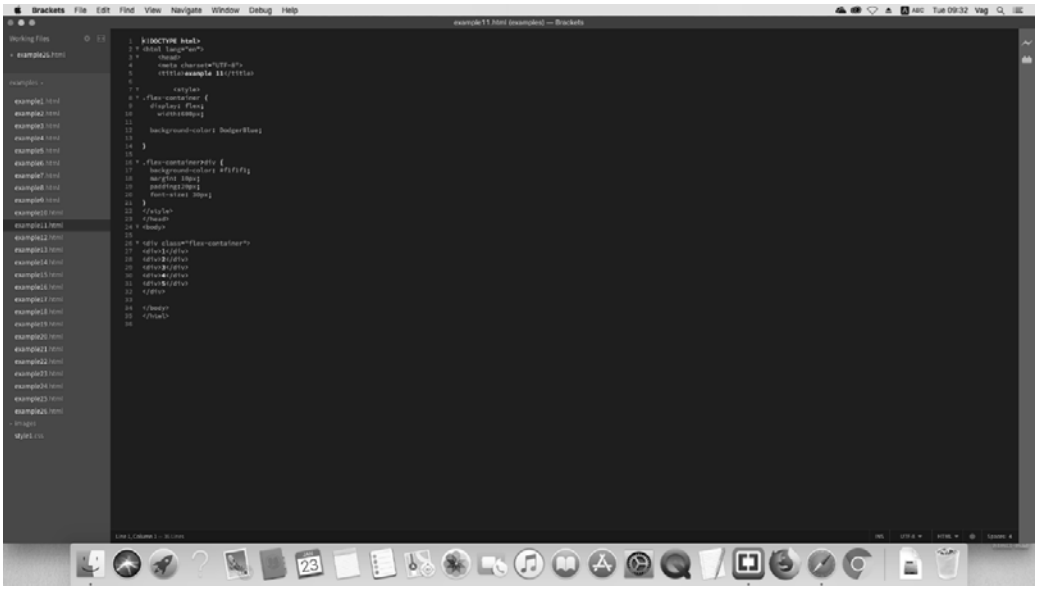
```
.flex-container>div:nth-child(3) {
color:firebrick;
  margin: 10px;
  padding:20px;
  font-size: 40px;
}
```

Σχόλια

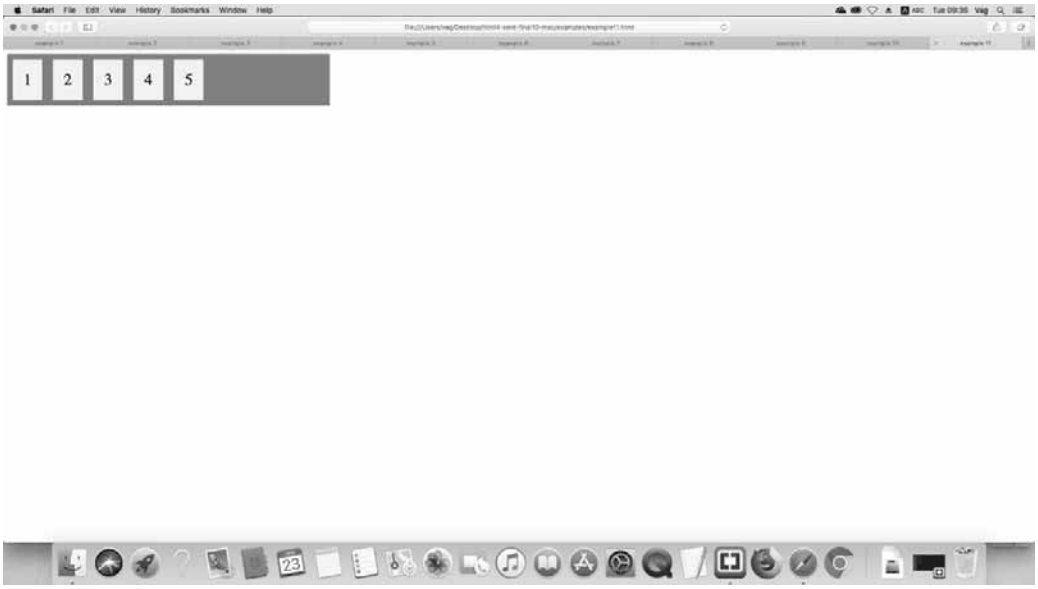
Μα αυτή την κλάση όλα τα παιδιά θα εξακολουθούν να έχουν font color μπλε εκτός από το παιδί 3 το οποίο θα έχει χρώμα γραμματοσειράς firebrick.

Παράδειγμα 11. Flexbox container και selectors

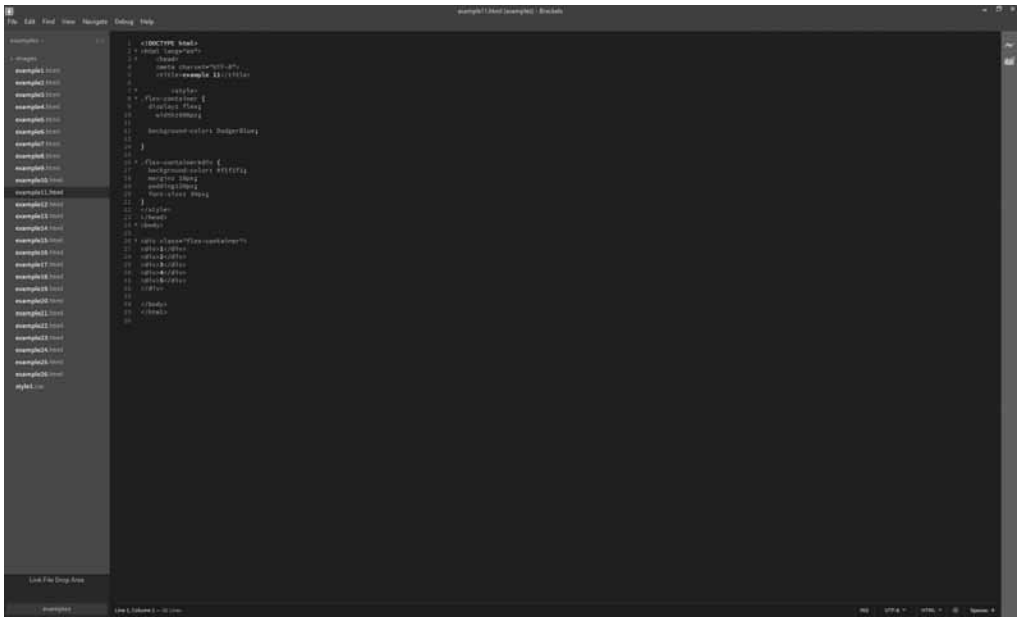
Κώδικας(Mac): Flexbox container και selectors



Εικόνα Browser(Mac): Flexbox container και selectors



Κώδικας(Win): Flexbox container και selectors



Εικόνα Browser(Win): Flexbox container και selectors

ΥΠΟΕΝΟΤΗΤΑ 3. FLEXBOX ΚΑΙ GRID SYSTEM

3.1 Flex Ιδιότητες για το Parent element

Flex-direction

Η ιδιότητα flex-direction ορίζει την κατεύθυνση στην οποία θα τοποθετηθούν τα παιδιά στο flex container

Για να τοποθετήσουμε τα στοιχεία σε στήλη και με κατεύθυνση από επάνω προς τα κάτω γράφουμε:

```
flex-direction:column;
```

Για να τοποθετήσουμε τα στοιχεία σε στήλη και με κατεύθυνση από κάτω προς τα επάνω γράφουμε:

```
flex-direction:column-reverse;
```

Για να τοποθετήσουμε τα στοιχεία σε γραμμή και με κατεύθυνση από αριστερά προς τα δεξιά γράφουμε:

```
flex-direction:row;
```

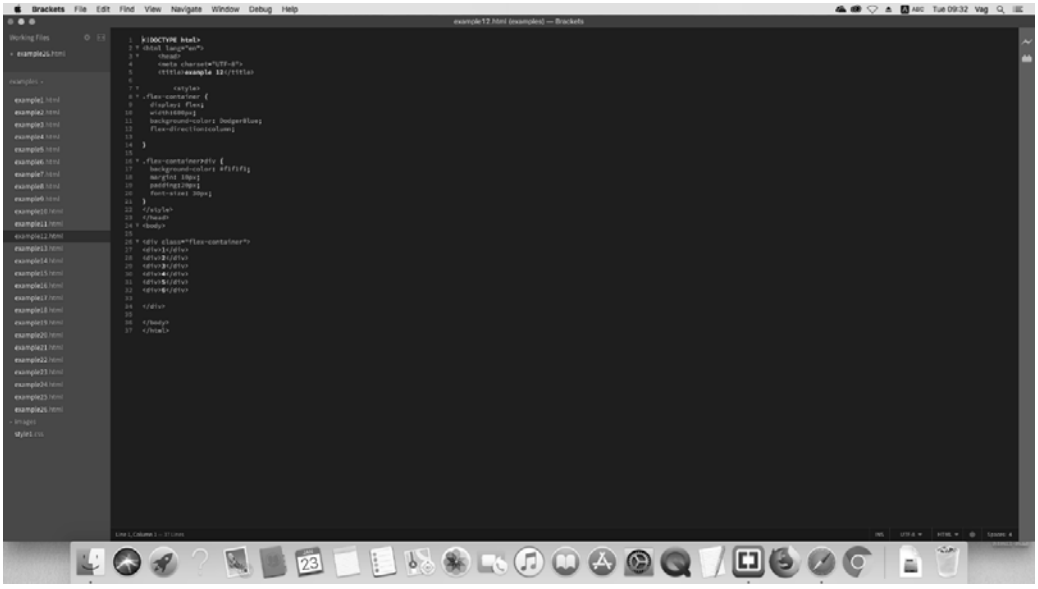
Για να τοποθετήσουμε τα στοιχεία σε γραμμή και με κατεύθυνση από δεξιά προς τα αριστερά γράφουμε:

```
flex-direction:row-reverse;
```

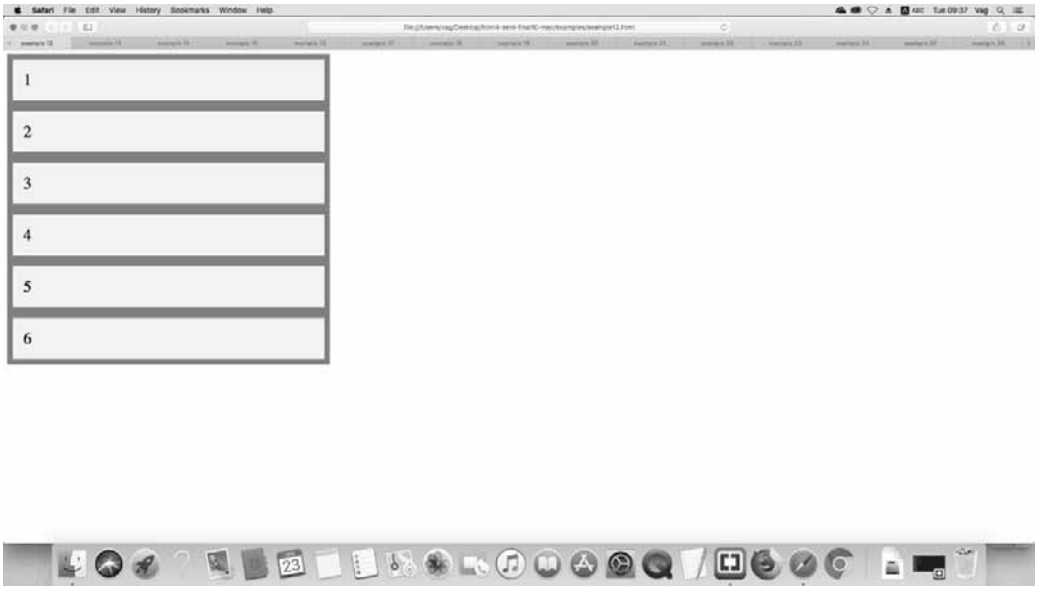
flex-direction	
flex-direction:column;	στήλη και με κατεύθυνση από επάνω προς τα κάτω
flex-direction:column-reverse;	στήλη και με κατεύθυνση από κάτω προς τα επάνω
flex-direction:row;	γραμμή και με κατεύθυνση από αριστερά προς τα δεξιά
flex-direction:row-reverse;	γραμμή και με κατεύθυνση από δεξιά προς τα αριστερά

Παράδειγμα 12. flex-direction

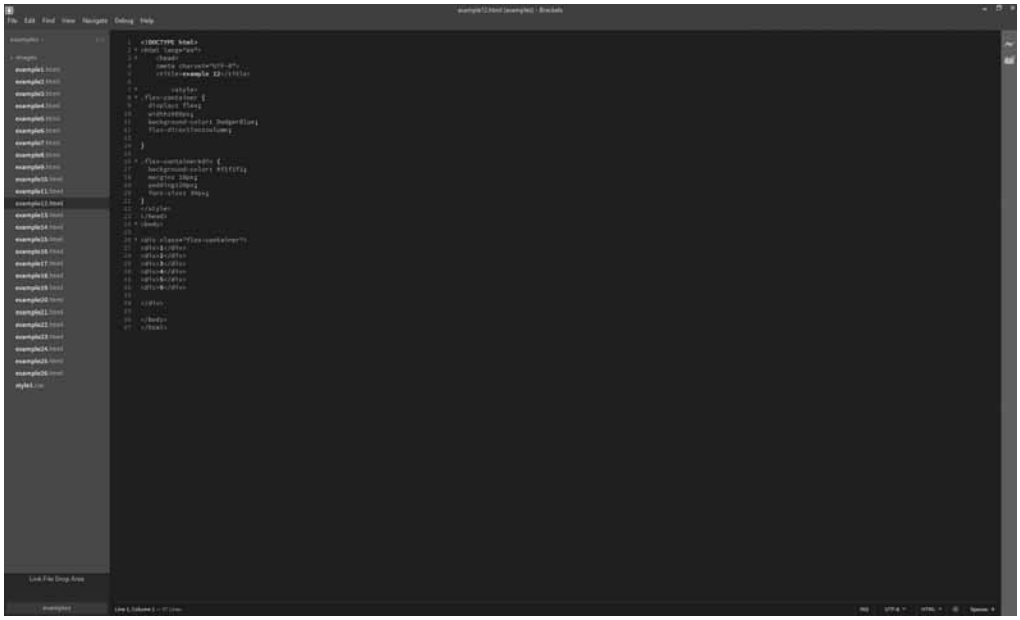
Κώδικας(Mac): flex-direction



Εικόνα Browser(Mac): flex-direction



Κώδικας(Win): flex-direction



Εικόνα Browser(Win): flex-direction

flex-wrap

Με την ιδιότητα flex-wrap μπορούμε να πετύχουμε αναδίπλωση των παιδιών στην περίπτωση που οι διαστάσεις του παραθύρου του Browser μικραίνουν.

Για να εμφανίζει τα παιδιά του flex box container σε πολλαπλές γραμμές θα γράψουμε

```
flex-wrap: wrap;
```

Για να εμφανίζει τα παιδιά του flex box container σε μία γραμμή θα γράψουμε

flex-wrap: nowrap;

Για να εμφανίζει τα παιδιά του flex box container σε πολλαπλές γραμμές ξεκινώντας όμως από την γραμμή του πυθμένα του δοχείου γράφουμε

flex-wrap: wrap-reverse;

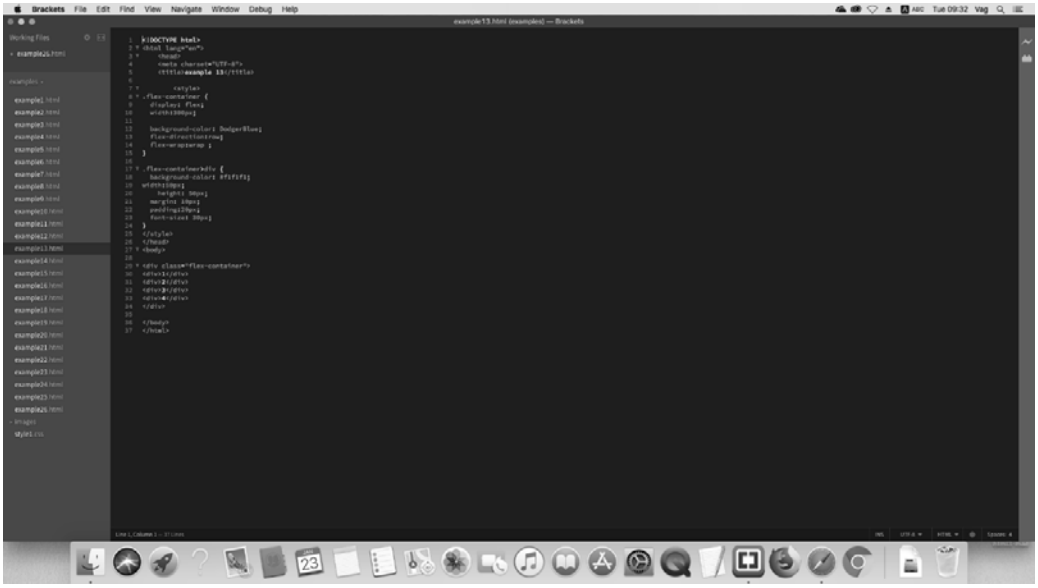
Έτσι λοιπόν για την ιδιότητα flex-wrap έχουμε τις παρακάτω αποδεκτές τιμές:

flex-wrap: τιμή;

flex-wrap	
wrap	πολλαπλές γραμμές
nowrap	μία γραμμή
wrap-reverse;	πολλαπλές γραμμές ξεκινώντας από την τελευταία

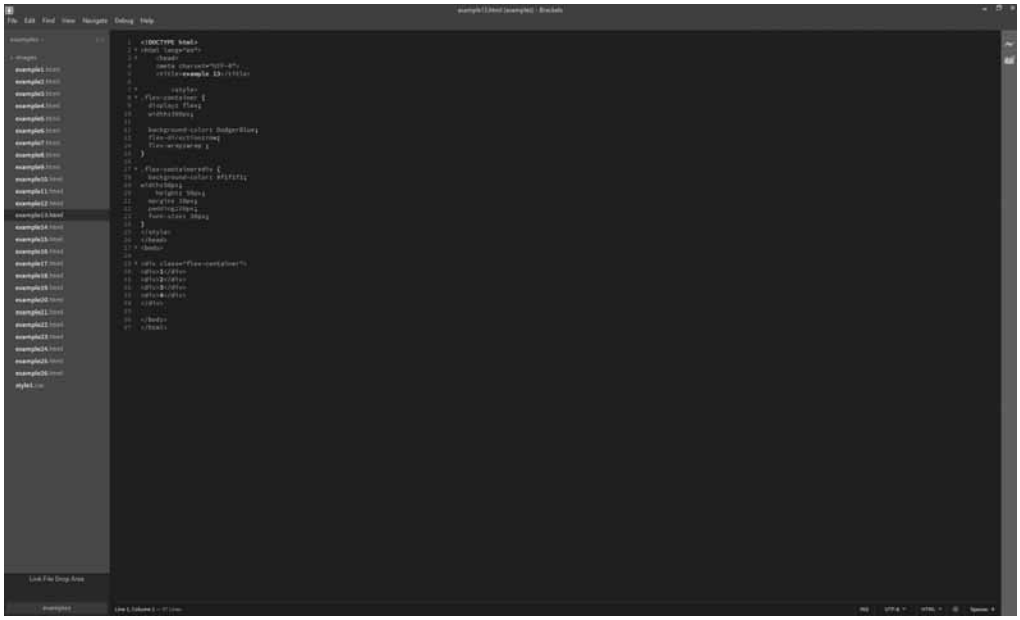
Παράδειγμα 13. flex-wrap

Κώδικας(Mac): flex-wrap

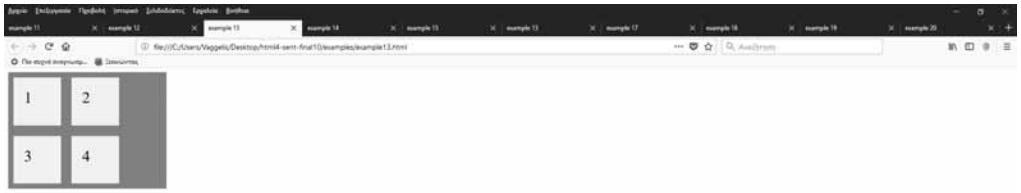


Εικόνα Browser(Mac): flex-wrap

Κώδικας(Win): flex-wrap



Εικόνα Browser(Win): flex-wrap



flex-flow

Η ιδιότητα flex flow είναι ένα shorthand που μας δίνει τη δυνατότητα να ορίσουμε ταυτόχρονα τις τιμές για flex-direction και flex-wrap. Το μόνο που χρειάζεται είναι στην ιδιότητα flex flow να γράψουμε τις δύο τιμές συνεχόμενα με κενό χαρακτήρα ανάμεσα σύμφωνα με την παρακάτω σύνταξη.

flex-flow: τιμή_flex-direction τιμή_flex-wrap ;

Παράδειγμα

flex-flow: column wrap ;

Στοίχιση αντικειμένων στον οριζόντιο άξονα (justify-content)

Για την στοίχιση των flex items στην οριζόντια διεύθυνση υπεύθυνη είναι η ιδιότητα justify-content.

Για να στοιχίσουμε τα flex items στον οριζόντιο άξονα και στο κέντρο γράφουμε

justify-content: center;

Για να στοιχίσουμε τα flex items στον οριζόντιο άξονα και στην αρχή του container γράφουμε

justify-content: flex-start;

Για να στοιχίσουμε τα flex items στον οριζόντιο άξονα και στο τέλος του container γράφουμε

justify-content: flex-end;

Για να εμφανίσουμε τα αντικείμενα με ίσες αποστάσεις περιμετρικά (γύρω από αυτά) γράφουμε

justify-content: space-around;

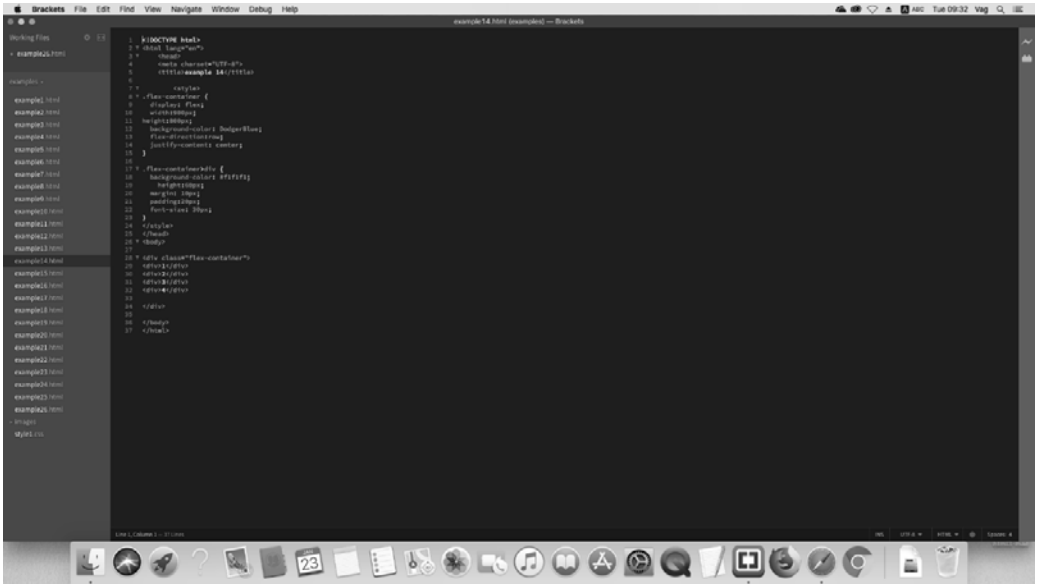
Για να εμφανίσουμε τα αντικείμενα με ίσες αποστάσεις μεταξύ τους γράφουμε

justify-content: space-between;

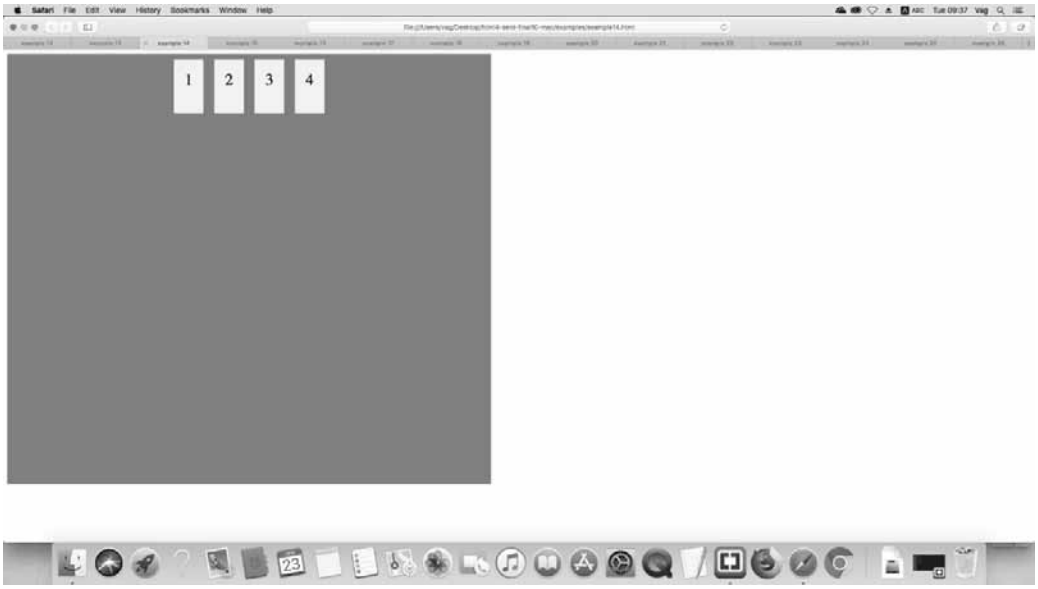
justify-content (οριζόντιος άξονας)	
center	κέντρο
flex-start	αρχή του container
flex-end	τέλος του container
space-arround	ίσος κενός χώρος περιμετρικά των αντικειμένων
space-between	ίσος κενός χώρος ανάμεσα στα αντικείμενα

Παράδειγμα 14. justify-content

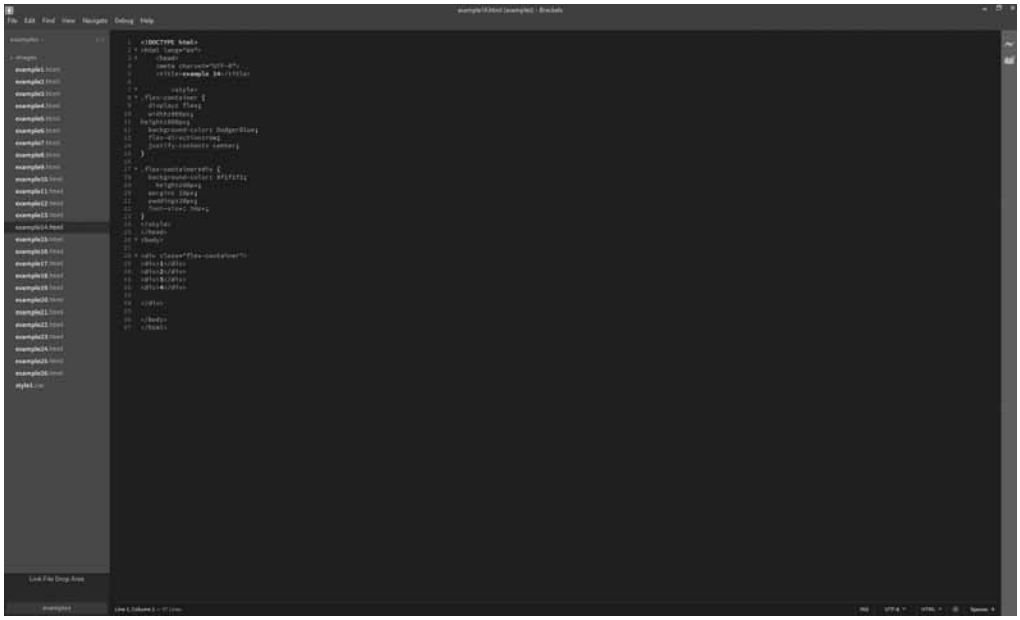
Κώδικας(Mac): justify-content



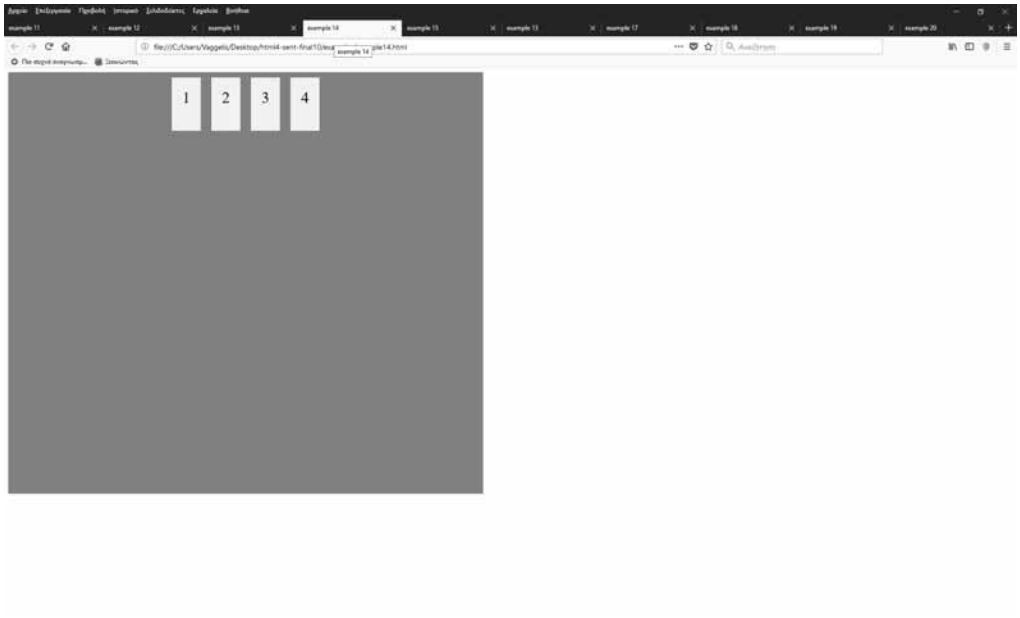
Εικόνα Browser(Mac): justify-content



Κώδικας(Win): justify-content



Εικόνα Browser(Win): justify-content



Στοίχιση αντικειμένων στον κάθετο άξονα με την ιδιότητα align-items

Η ιδιότητα align-items στοιχίζει τα αντικείμενα στον κατακόρυφο άξονα.

Έτσι λοιπόν για να στοιχίσουμε τα αντικείμενα στο κέντρο του κατακόρυφου άξονα γράφουμε

```
align-items:center;
```

Για να στοιχίσουμε τα αντικείμενα στην αρχή του container γράφουμε

```
align-items:flex-start;
```

Για να στοιχίσουμε τα αντικείμενα στο τέλος του container γράφουμε

```
align-items:flex-end;
```

Για να προκαλέσουμε τη μέγιστη δυνατή επέκταση των αντικειμένων έτσι ώστε να καταλαμβάνουν το διαθέσιμο ύψους του flex-container γράφουμε

```
align-items:stretch;
```

Για να στοιχίσουμε τα αντικείμενα σύμφωνα με το text baseline γράφουμε

```
align-items:baseline;
```

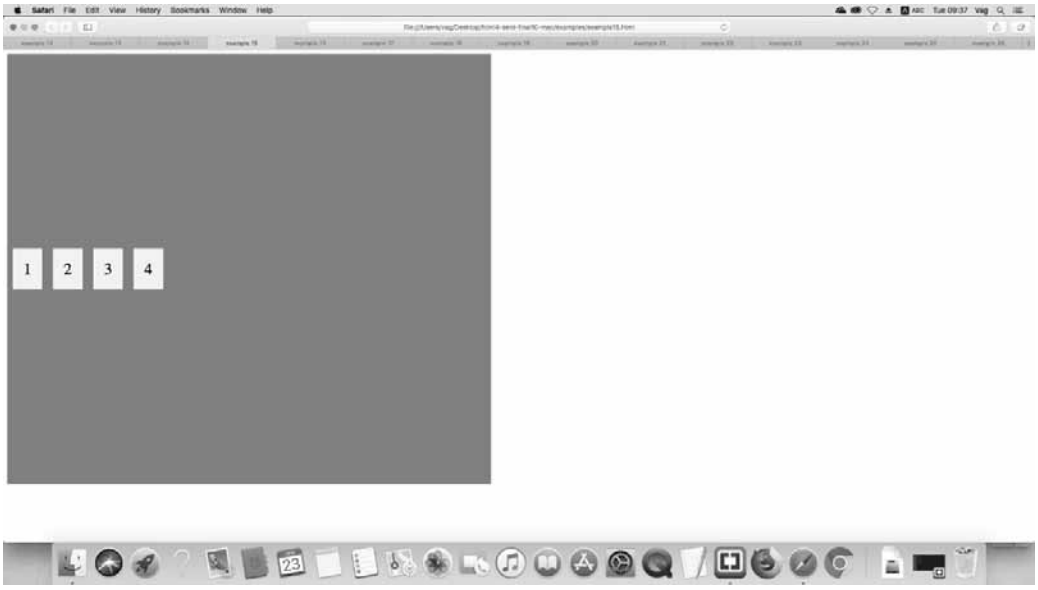
Η γραμμή text baseline ουσιαστικά είναι η κάτω γραμμή όπως αυτή ενός τετραδίου, όπου αποτελεί και τη βάση για τον κάθε χαρακτήρα που γράφουμε. Έτσι λοιπόν όταν έχουμε παιδιά στο flexbox container και στο κάθε παιδί έχουμε διαφορετικό μέγεθος γραμματοσειράς, με την τιμή baseline το κείμενο στα διαφορετικά παιδιά θα είναι στοιχισμένο.

	align-items
center	κέντρο
flex-start	αρχή του container
flex-end	τέλος του container
stretch	επέκταση κατά ύψος
baseline	στοίχιση κατά κείμενο

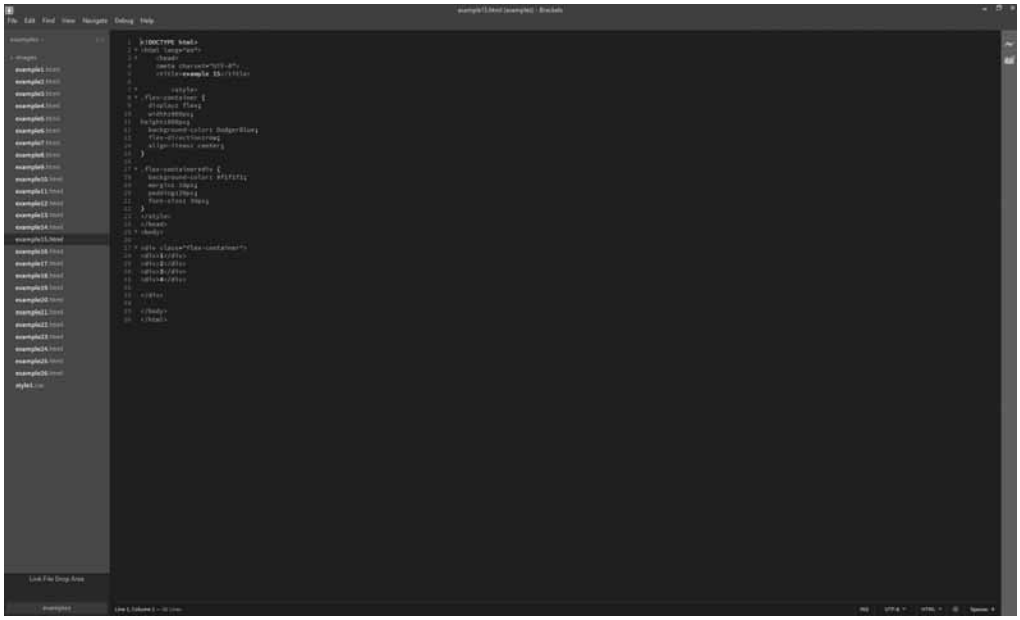
Παράδειγμα 15. align-items

Κώδικας(Mac): align-items

Εικόνα Browser(Mac): align-items



Κώδικας(Win): align-items



Εικόνα Browser(Win): align-items

align-content

Με την ιδιότητα align-content εφαρμόζουμε στοίχιση όχι σε όλα τα αντικείμενα, αλλά στις γραμμές των αντικειμένων. Για παράδειγμα αν σε ένα συγκεκριμένο πλάτος οθόνης έχουμε δύο οριζόντιες σειρές αντικειμένων η ιδιότητα align-content «βλέπει» την κάθε σειρά-γραμμή αντικειμένων ως ένα αντικείμενο. Άρα εφαρμόζει ιδιότητες στις δύο αυτές σειρές.

Για να εφαρμόσουμε κενό διάστημα μεταξύ των σειρών των αντικειμένων γράφουμε

```
align-content:space-between;
```

Για να εφαρμόσουμε κενό διάστημα περιμετρικά των σειρών των αντικειμένων γράφουμε

```
align-content:space-arround;
```

Για να προκαλέσουμε επέκταση των γραμμών για να καταλαμβάνουν το διαθέσιμο ύψος γράφουμε

```
align-content:stretch;
```

Για να στοιχίσουμε τις γραμμές στο κέντρο του flex-container γράφουμε

```
align-content:center;
```

Για να στοιχίσουμε τις γραμμές στην αρχή του flex-container γράφουμε

```
align-content:flex-start;
```

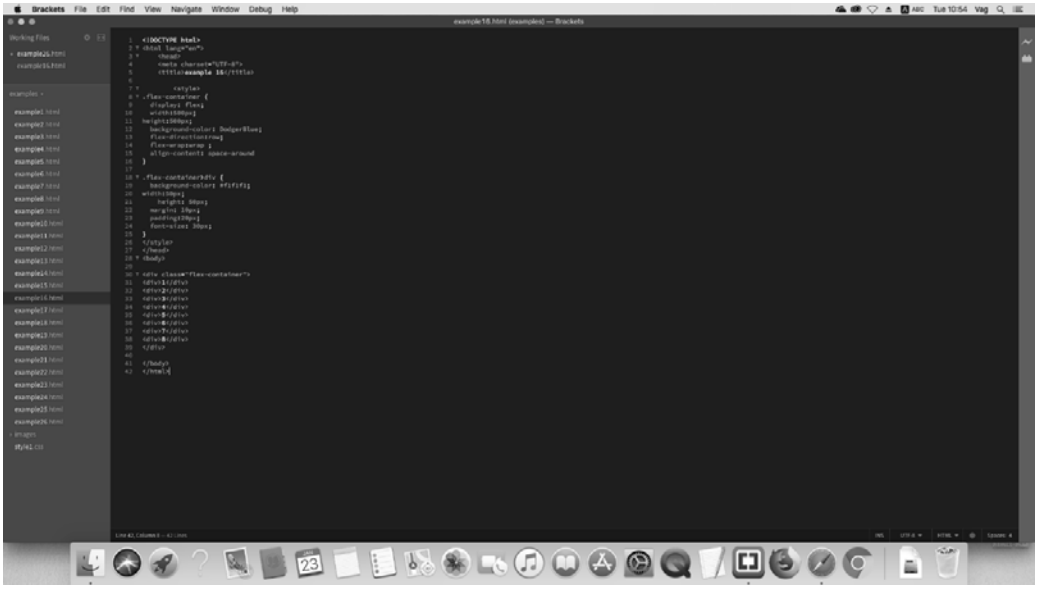
Για να στοιχίσουμε τις γραμμές στο τέλος του flex-container γράφουμε

```
align-content:flex-end;
```

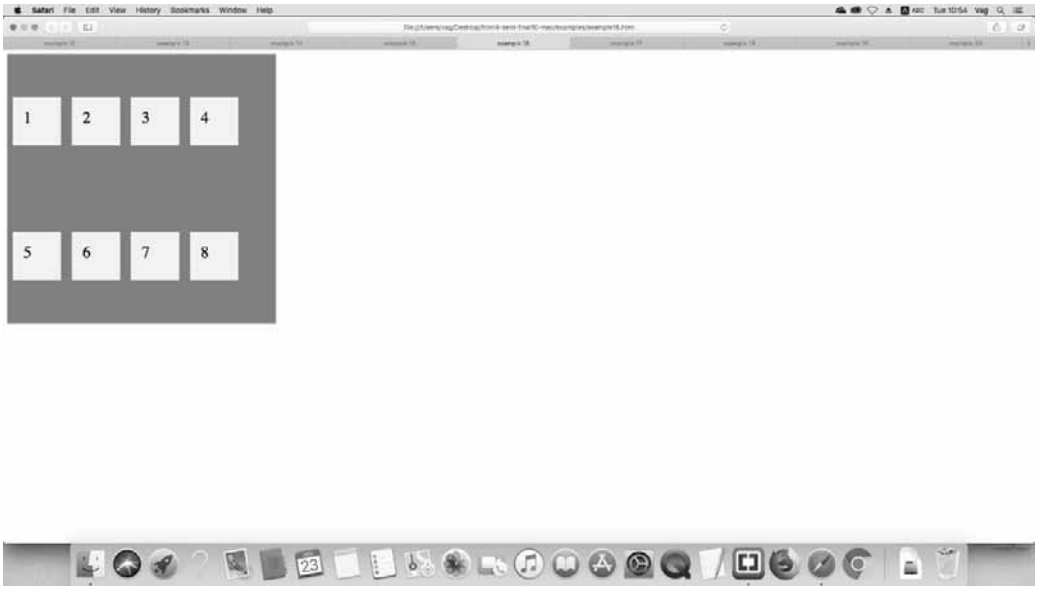
align-content (οριζόντιος άξονας)	
center	κέντρο
flex-start	αρχή του container
flex-end	τέλος του container
space-arround	ίσος κενός χώρος περιμετρικά των αντικειμένων
space-between	ίσος κενός χώρος ανάμεσα στα αντικείμενα
stretch	επέκταση των γραμμών για να καταλαμβάνουν το διαθέσιμο ύψος

Παράδειγμα 16. align-content

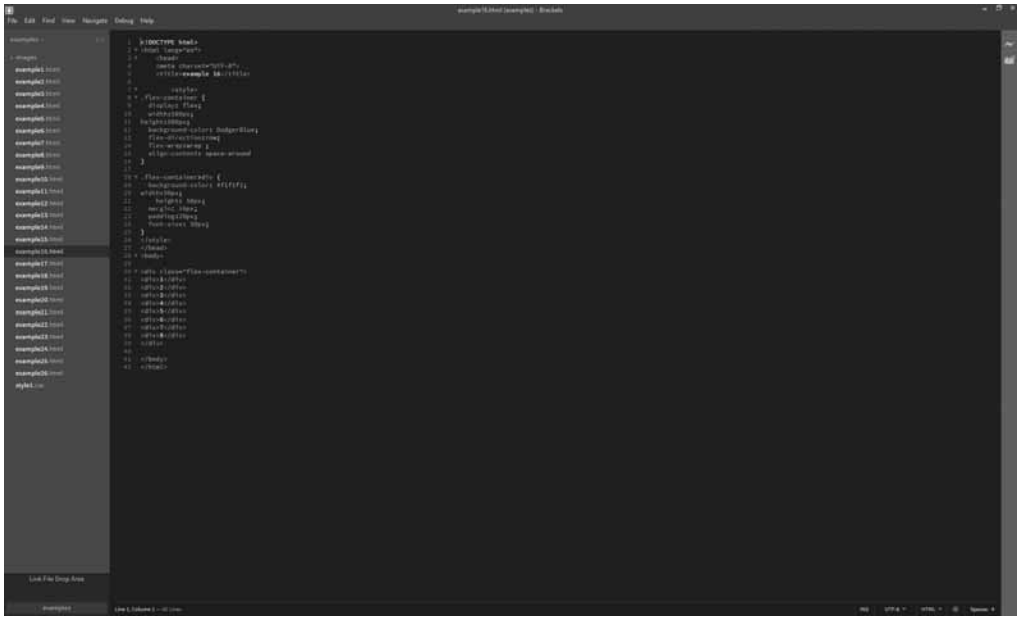
Κώδικας(Mac): align-content



Εικόνα Browser(Mac): align-content



Κώδικας(Win): align-content



Εικόνα Browser(Win): align-content

3.2 Flex Ιδιότητες για τα child elements

order

Η εισαγωγή των παιδιών εντός του flex container γίνεται σύμφωνα με τη σειρά που τα παιδιά είναι γραμμένα στον κώδικα. Το πρώτο παιδί (στοιχείο div) που θα γραφεί μετά το div του flex container θα είναι και το πρώτο που θα εισαχθεί στο flex container.

Στην περίπτωση που επιθυμούμε να ορίσουμε εμείς τη σειρά με την οποία θα εμφανιστούν τα παιδιά στο flexbox layout γράφουμε την ιδιότητα order σύμφωνα με την παρακάτω σύνταξη:

order: τιμή;

Αποδεκτές τιμές για την ιδιότητα order είναι οι φυσικοί αριθμοί 0, 1, 2, 3…. Η τιμή μηδέν είναι και η προεπιλογή.

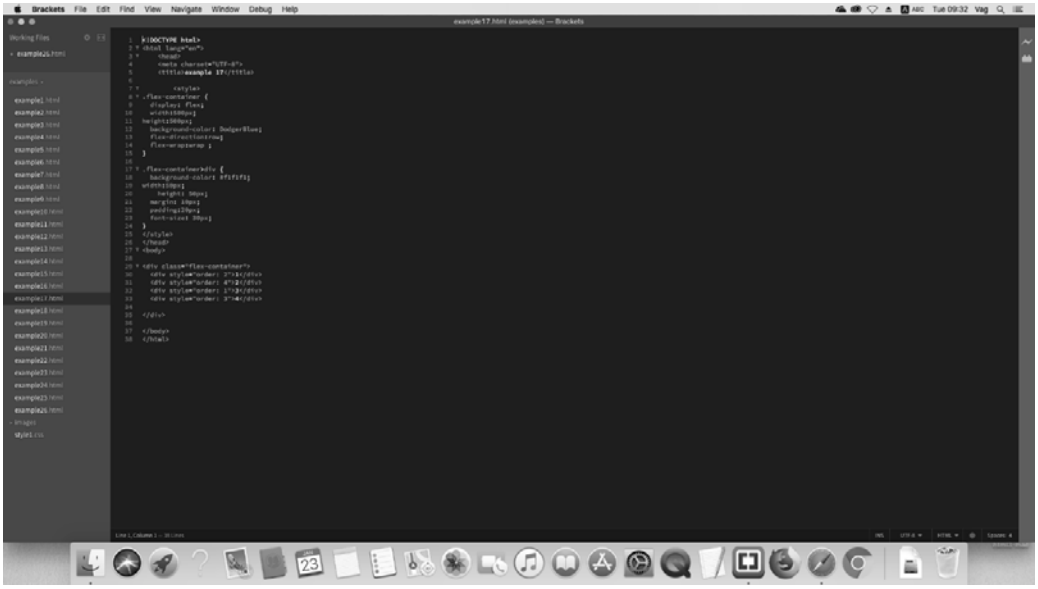
Έτσι λοιπόν το στοιχείο με μικρότερη τιμή στην ιδιότητα order θα εμφανιστεί πρώτο, ενώ αυτό με τη μεγαλύτερη τιμή order θα εμφανιστεί τελευταίο. Στην περίπτωση που δύο παιδιά έχουν την ίδια τιμή order θα ισχύει η σειρά με την οποία έχουν γραφεί στον κώδικα.

Παράδειγμα

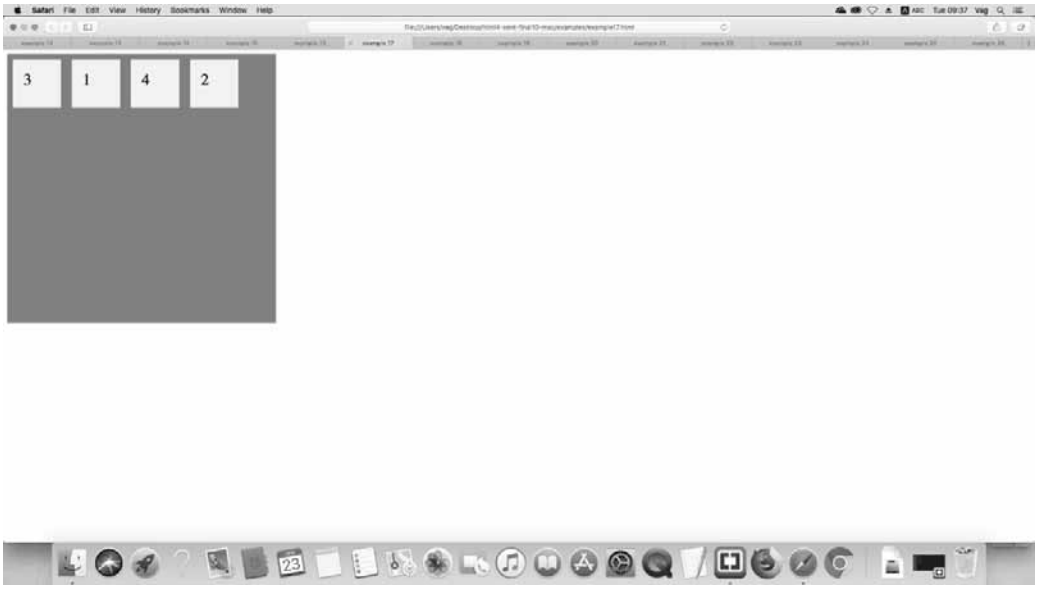
```
<div style="order:5> …  </div>
```

Παράδειγμα 17. Order

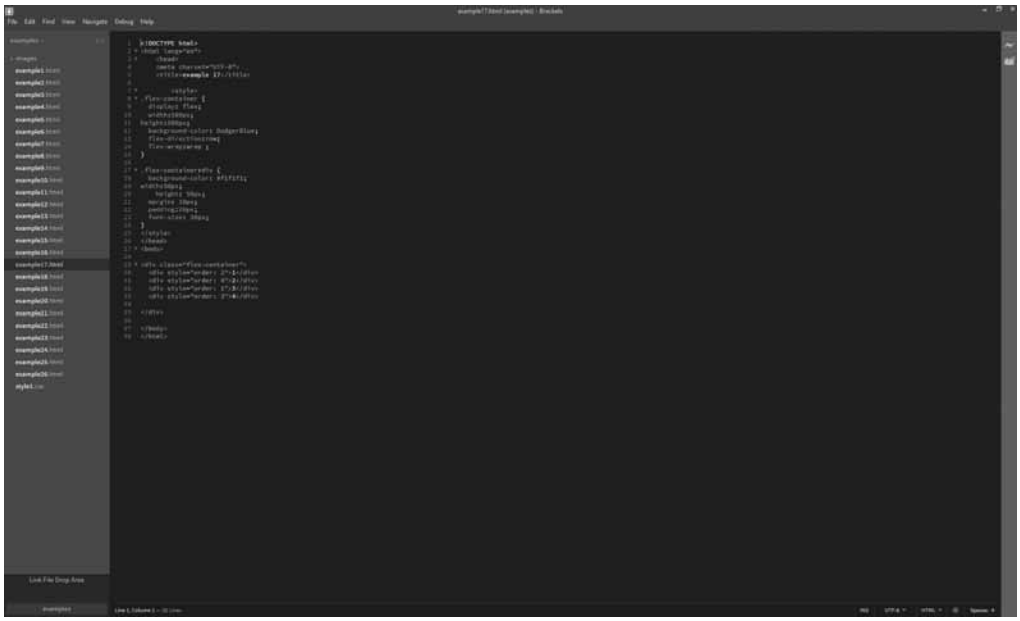
Κώδικας(Mac): order



Εικόνα Browser(Mac): order



Κώδικας(Win): order



Εικόνα Browser(Win): order

flex-grow

πολλές φορές όταν εμφανίζεται μία σελίδα εμφανίζεται κάποιος κενός χώρος, χώρος που είναι διαθέσιμος. Δηλαδή αν για παράδειγμα έχουμε ένα μεγάλο flexbox container και δύο μικρών διαστάσεων child elements αφού τα child elements εμφανιστούν στις κανονικές προβλεπόμενες διαστάσεις θα περισσέψει ανεκμετάλλευτος κενός χώρος. Στην περίπτωση που επιθυμούμε να εκμεταλλευτούμε τον διαθέσιμο χώρο επεκτείνοντας τα child elements ως προς το πλάτος τους με αναλογίες που έχουμε ορίσει. Αυτή την επέκταση του χώρου των παιδιών του flexbox container την ελέγχουμε και την ενεργοποιούμε με την ιδιότητα flex-grow σύμφωνα με την παρακάτω σύνταξη:

flex-grow:τιμή ;

στη θέση της τιμής σε κάθε παιδί θα γράψουμε έναν αριθμό από 1,2,3…
Η αναλογία των τιμών για το κάθε παιδί είναι και η αναλογία με την οποία θα μοιραστεί ο παραπάνω διαθέσιμος χώρος.

Παράδειγμα

Αρχικά έστω ότι θα γράψουμε ένα flex-container με πλάτος 800px

```
<style>
.flex-container {
width:800px;
  display: flex;
  background-color: DodgerBlue;
flex-direction: row;
}
</style>
```

Στη συνέχεια για όλα τα παιδιά θα γράψουμε πλάτος ίσο με 250 Pixels.

```
  .flex-container>div {
  background-color: greenyellow;
  width:250px;
  font-size: 30px;
 color:blue;
 }
```

Εάν στις ιδιότητες των παιδιών δεν προσθέσουμε την ιδιότητα flex-grow θα ισχύει το παρακάτω σε σχέση με την εμφάνιση της σελίδας:

- Το συνολικό πλάτος του container είναι 800 pixels.
- Το συνολικό πλάτος που καταλαμβάνουν τα παιδιά είναι 2X250=500 pixels.
- Το πλάτος που «περισσεύει» είναι 800-500=300 Pixels.

Σημείωση

Δεν έχουμε προσθέσει στα παιδιά margin padding κτλ. για την καλύτερη κατανόηση των υπολογισμών στην εφαρμογή της ιδιότητας flex-grow.

Στη συνέχεια στο πρώτο παιδί θα ορίσουμε τιμή 1 στην ιδιότητα flex-grow

```
.flex-container>div:nth-child(1) {
color:firebrick;
background-color: blue;
font-size: 40px;
flex-grow:1;
}
```

Στο δεύτερο παιδί θα γράψουμε την τιμή 2 για την ιδιότητα flex-grow.

```
.flex-container>div:nth-child(2) {
color:red;
background-color: yellow;
font-size: 40px;
flex-grow:2;
}
```

Η αναλογία flex-grow θα είναι 1:2 (child1:child2)

Οπότε εφόσον το πλάτος που περισσεύει είναι 300 pixels και η αναλογία μοιράσματος του περισσευούμενου πλάτους είναι 1 προς 2:

- Το πρώτο παιδί θα καταλαμβάνει έξτρα πλάτος 100 pixels
- Το δεύτερο παιδί θα καταλαμβάνει έξτρα πλάτος 200 pixels

Οπότε συνολικά τα πλάτη των παιδιών θα είναι:

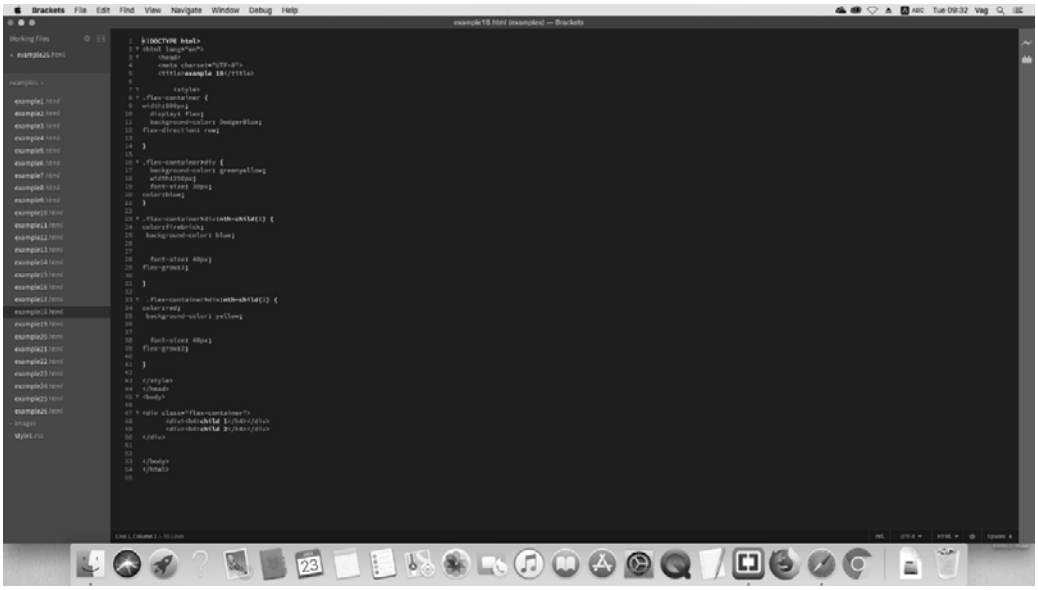
Πλάτος_child1=250+100=350 pixels

Πλάτος_child2=250+200=450 pixels

Οπότε τώρα συνολικά τα δύο παιδιά θα καταλαμβάνουν πλάτος 800 Pixels.

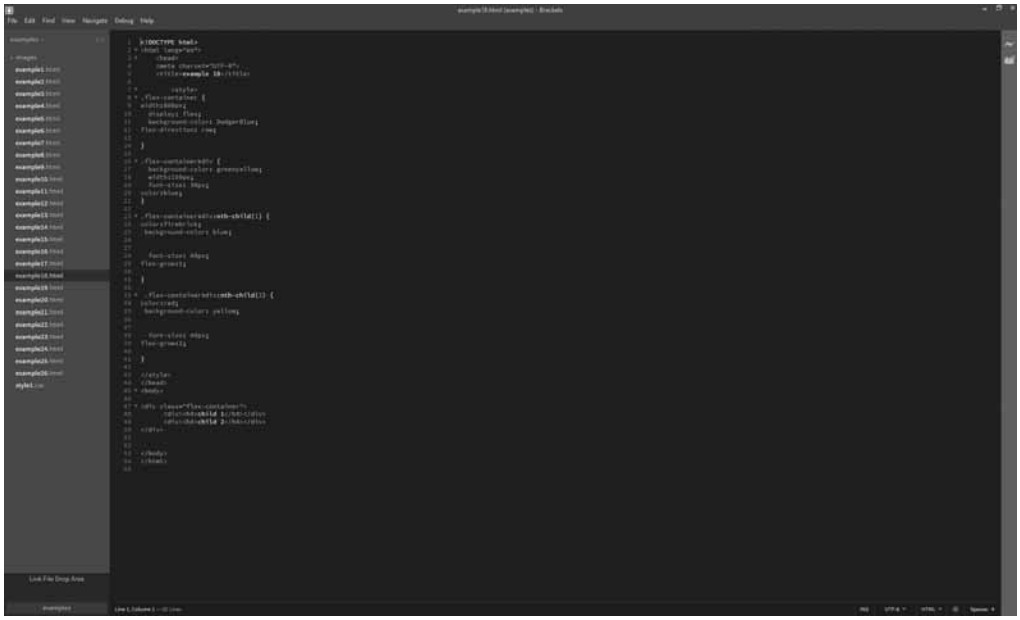
Παράδειγμα 18. flex-grow

Κώδικας(Mac): flex-grow



Εικόνα Browser(Mac): flex-grow

Κώδικας(Win): flex-grow



Εικόνα Browser(Win): flex-grow

flex-shrink

Η ιδιότητα flex-shrink ενεργοποιείται όταν ο διαθέσιμος χώρος δεν είναι αρκετός για να εμφανιστούν τα παιδιά στις κανονικές τους διαστάσεις και συμπιέζει τα παιδιά έτσι ώστε να μπορεί να εμφανιστεί η σελίδα.

Στην περίπτωση αυτή μπορούμε να συμπιέσουμε με ορισμένη αναλογία τα παιδιά. Ο υπολογισμός για τη συμπίεση γίνεται με την ίδια λογική που χρησιμοποιεί το flex- grow και η σύνταξή του είναι η παρακάτω:

flex-shrink: τιμή;

Η προεπιλεγμένη τιμή είναι η τιμή ένα.

Αν επιθυμούμε για ένα στοιχείο να συμπιεστεί τελευταίο (ως τελευταία λύση) θα γράψουμε τιμή ίση με μηδέν.

Δηλαδή το πλάτος του χώρου που λείπει για να εμφανιστούν τα παιδιά στις κανονικές τους διαστάσεις εξοικονομείται από την κατά αναλογία συμπίεση των flex-items.

Παράδειγμα

Έστω flex container με πλάτος ίσο με 500 pixels.

```
.flex-container {
width:500px;
display: flex;
background-color: DodgerBlue;
flex-direction: row;
}
```

Έστω δύο παιδιά με πλάτη 400 Pixels το καθένα. Οπότε για να εμφανιστούν τα παιδιά στις κανονικές τους διαστάσεις χρειάζονται πλάτος 2X400=800 pixels.

```
.flex-container>div {
background-color: greenyellow;
width:400px;
font-size: 30px;
color:blue;
}
```

Για τα πρώτο παιδί θα γράψουμε

```
flex-shrink:1;
```

```
.flex-container>div:nth-child(1) {
color:firebrick;
background-color: blue;
font-size: 40px;
flex-shrink:1;
}
```

Για το δεύτερο παιδί θα γράψουμε flex-shrink:2;

```
.flex-container>div:nth-child(2) {
color:red;
background-color: yellow;
font-size: 40px;
flex-shrink:2;
}
```

Ο χώρος που χρειάζεται να βρεθεί έχει πλάτος ίσο με 300 pixels. Αυτά τα 300 pixels θα δοθούν από τα πλάτη των παιδιών με αναλογία ένα προς δύο. Οπότε το πρώτο παιδί θα χάσει 100 pixels πλάτος ενώ το δεύτερο παιδί θα χάσει 200 pixels.

Τα πλάτη των παιδιών μετά την ενεργοποίηση του flex-shrink θα είναι:

Παιδί 1: πλάτος=400-100=300 pixels

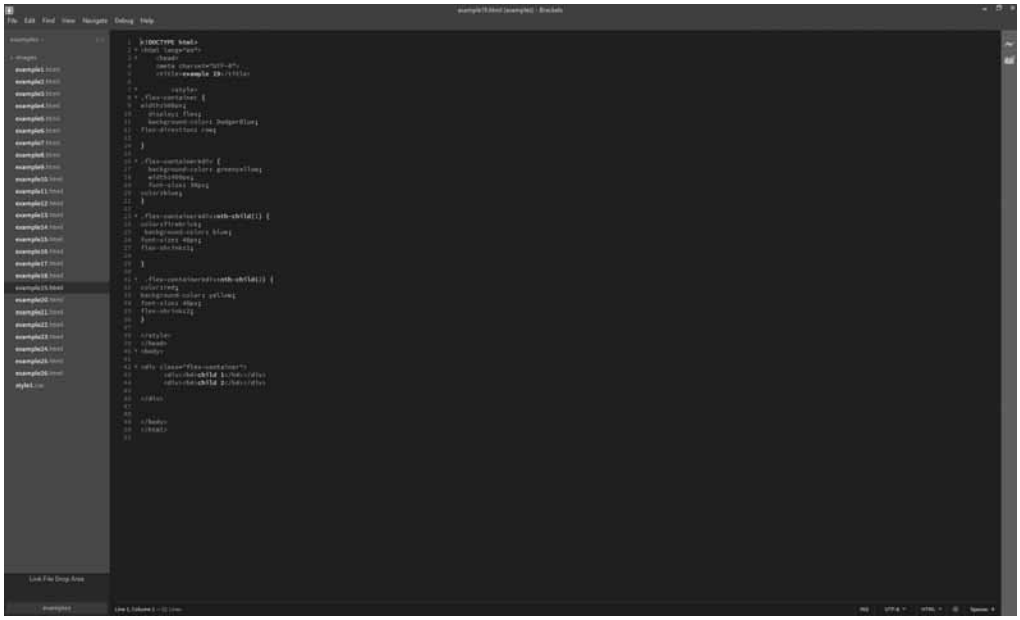
Παιδί 2: πλάτος=400-200=200 pixels

Παράδειγμα 19. flex-shrink

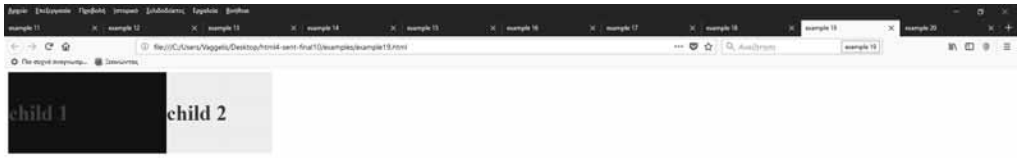
Κώδικας(Mac): flex-shrink

Εικόνα Browser(Mac): flex-shrink

Κώδικας(Win): flex-shrink



Εικόνα Browser(Win): flex-shrink



Flex-basis

Η ιδιότητα flex-basis χρησιμεύει για να ορίζουμε το πλάτος που καταλαμβάνει ένα flex item και η σύνταξή του είναι η παρακάτω:

flex-basis: τιμή ;

Παράδειγμα

```
flex-basis:250px;
```

flex property shorthand

Μπορούμε με την παρακάτω σύνταξη να ορίσουμε ταυτόχρονα τιμές για flex-grow, flex-shrink και flex-basis σύμφωνα με την παρακάτω σύνταξη:

flex: flex-grow_τιμή flex-shrink_τιμή flex-basis_τιμή ;

Σχόλια

Μεταξύ των συνεχόμενων τιμών αφήνουμε κενό.

Παράδειγμα

```
flex: 0 0 250px;
```

Στο παραπάνω παράδειγμα έχουμε απενεργοποιήσει τα flex-grow και flex-shrink και έχουμε ορίσει πλάτος για το flex item με αρχικό πλάτος ίσο με 250 pixels.

Align-self

Έστω ότι έχουμε ορίσει την ιδιότητα align-items για την κατακόρυφη στοίχιση των αντικειμένων. Ένα επιθυμούμε για ένα η περισσότερα παιδιά να ακυρωθεί η συγκεκριμένη στοίχιση και να ισχύσει κάποια άλλη τότε γράφουμε την ιδιότητα align-self σύμφωνα με την παρακάτω σύνταξη:

```
Align self: τιμή;
```

Για τιμή του align-self μπορούμε να γράψουμε:

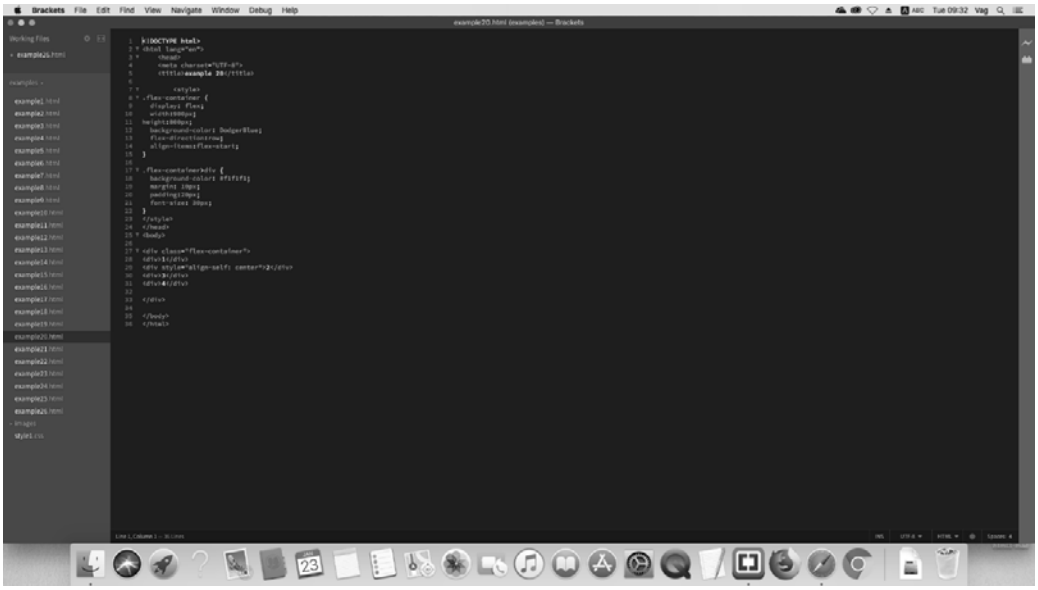
align-self	
center	κέντρο
flex-start	αρχή του container
flex-end	τέλος του container

Παράδειγμα

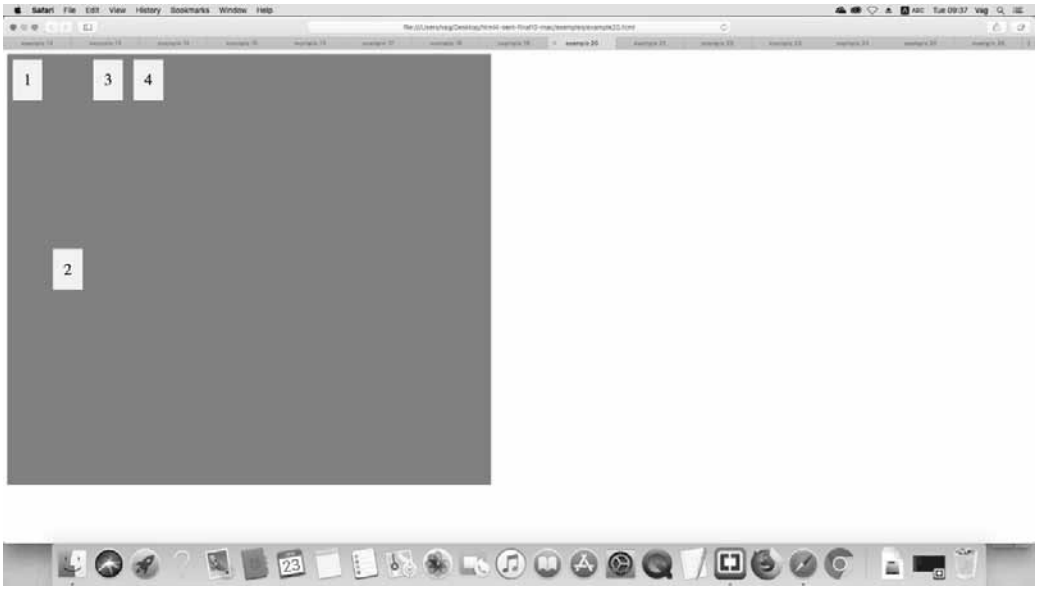
```
<div style="align-self: flex-start ">2</div>
```

Παράδειγμα 20. align-self

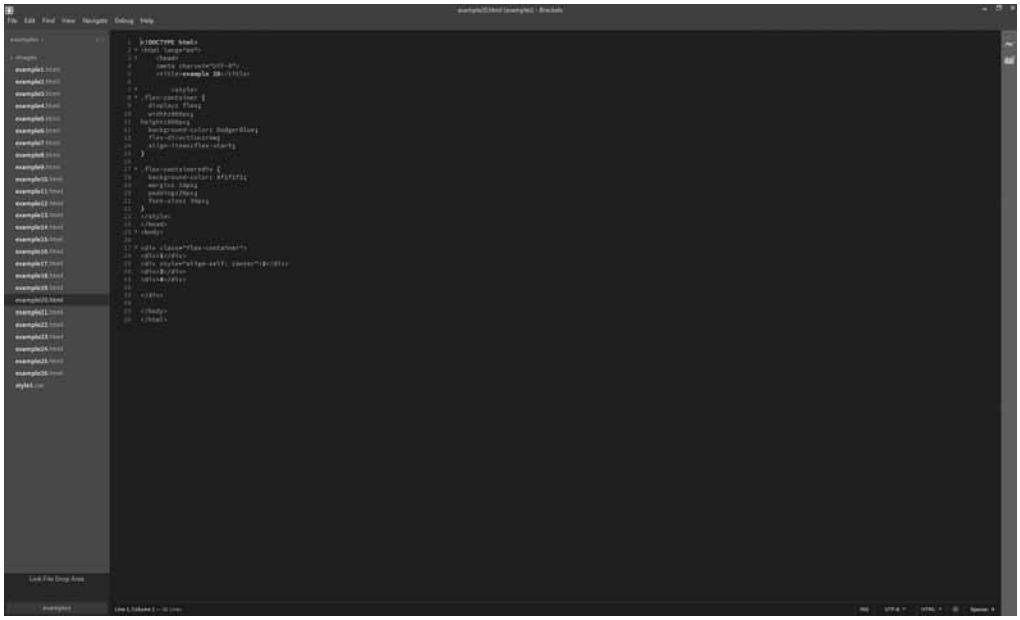
Κώδικας(Mac): align-self



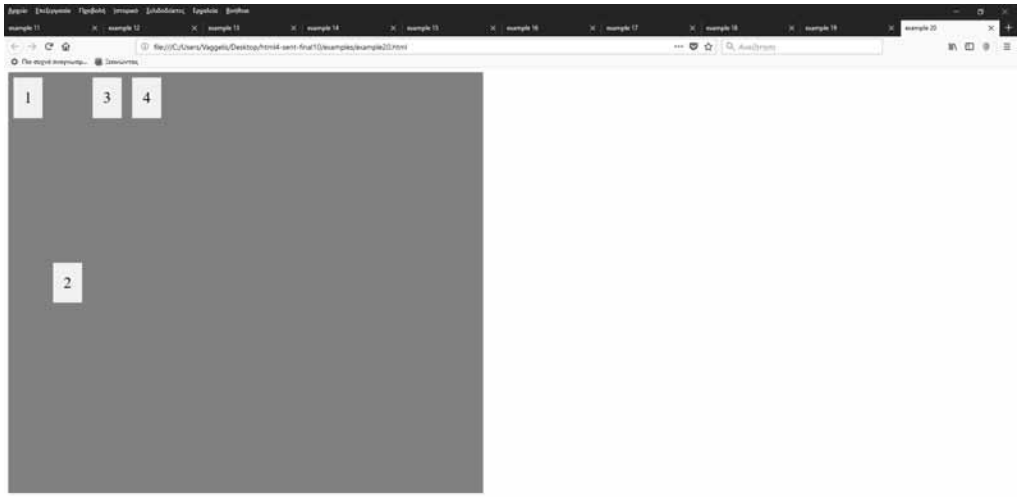
Εικόνα Browser(Mac): align-self



Κώδικας(Win): align-self



Εικόνα Browser(Win): align-self



3.3 CSS Grid Layout

Με το CSS Grid Layout μπορούμε να δημιουργήσουμε responsive δομές σελίδων αποτελούμενες από στήλες και γραμμές χωρίς να χρησιμοποιούμε float και Positioning.

Η λογική που δουλεύουμε με το CSS grid system είναι παρόμοια με αυτή στο flex box. Δημιουργούμε ένα grid container το οποίο θα παίζει το ρόλο του πατρικού στοιχείου.

Στη συνέχεια εντός του Grid container εισάγουμε ένα ή περισσότερα child elements.

Για να ενεργοποιήσουμε το grid container χρησιμοποιούμε πάλι την ιδιότητα display. Έχουμε δύο επιλογές σε σχέση με το grid container:

Γράφουμε display:grid; για να δημιουργήσουμε ένα block grid container το οποίο θα καταλαμβάνει ολόκληρο το πλάτος της σελίδας

Γράφουμε display:inline-grid; για να δημιουργήσουμε ένα inline grid container

Οπότε για το block grid container γράφουμε μία κλάση όπως η παρακάτω:

```
.grid-container{
display:grid; }
```

Στη συνέχεια εντός του στοιχείου Body γράφουμε ένα div στο οποίο θα εφαρμόσουμε τη συγκεκριμένη κλάση

```
<div class="grid-container">
…
</div>
```

Εντός του grid-container προσθέτουμε τα παιδιά με τον ίδιο τρόπο όπως και στο flexbox container:

```
<div class="grid-container">
<div>    </div>

    <div>  </div>
…
…
</div>
```

Σημείωση

Σε σχέση με τους selectors για τα παιδιά και το grid container ισχύουν τα ίδια ακριβώς που ισχύουν στο flexbox layout.

Για να εφαρμόσουμε CSS style στο grid container γράφουμε μια κλάση όπως η παρακάτω:

```
.grid-container {

}
```

Για να επιλέξουμε τα παιδιά του Grid system αν η κλάση του container είναι η κλάση .grid-container τότε επιλέγουμε τα παιδιά γράφοντας:

```
.grid-container>div {

}
```

Αν επιθυμούμε να επιλέξουμε ένα συγκεκριμένο παιδί τότε γράφουμε:

```
.grid-container>div:nth-child(n) {

}
```

Όπου n ο αριθμός του παιδιού σε σχέση με τη σειρά που έχει εντός του Grid container

Grid columns

Τις κατακόρυφες σειρές των grid items τις ονομάζουμε columns (στήλες)

Grid Rows

Τις οριζόντιες γραμμές των grid items τις ονομάζουμε rows (γραμμές)

Grid Gaps

Τις αποστάσεις μεταξύ δύο columns τις ονομάζουμε column gaps

Τις αποστάσεις μεταξύ δύο rows τις ονομάζουμε row gaps

Για να ορίσουμε τις αποστάσεις μεταξύ των στηλών γράφουμε

```
grid-column-gap:τιμή;
```

Για να ορίσουμε τις αποστάσεις μεταξύ των γραμμών γράφουμε

```
grid-row-gap:τιμή;
```

Παράδειγμα

```
grid-column-gap:12px;

grid-row-gap: 11px;
```

grid-gap shorthand

Με το grid-gap shorthand μπορούμε να ορίσουμε ταυτόχρονα τιμές για γράφουμε grid-column-gap και grid-row-gap με την παρακάτω σύνταξη:

grid-gap: τιμή_grid-column-gap τιμή_grid-row-gap;

Επίσης γράφοντας μία τιμή στο grid-gap shorthand μπορούμε να ορίσουμε μία τιμή που θα ισχύει και για τις δύο ιδιότητες grid-column-gap grid-row-gap.

Παράδειγμα

```
grid-gap: 25px 35px;
```

Σχόλια

Ορίσαμε grid-column-gap ίσο με 25 pixels και grid-row-gap ίσο με 35 pixels.

Παράδειγμα

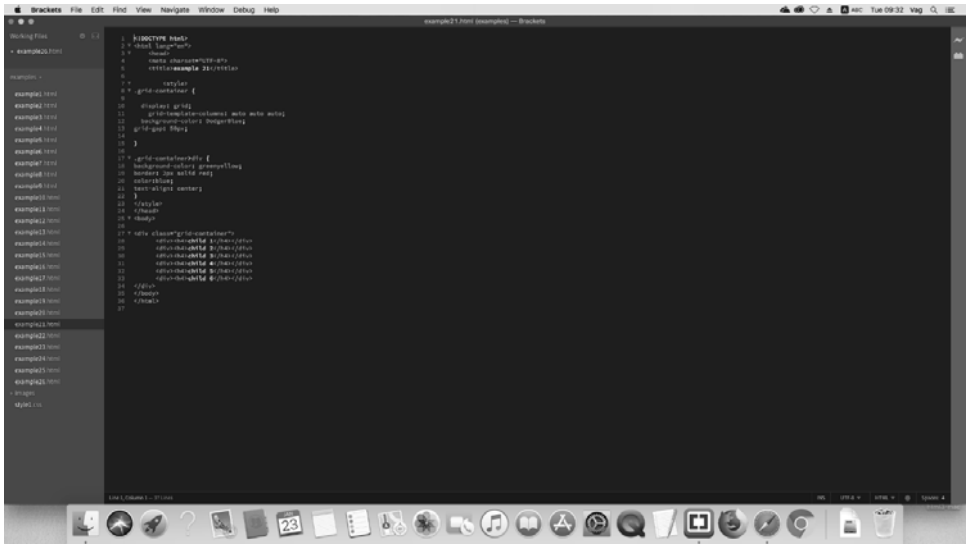
```
grid-gap: 15px ;
```

Σχόλια

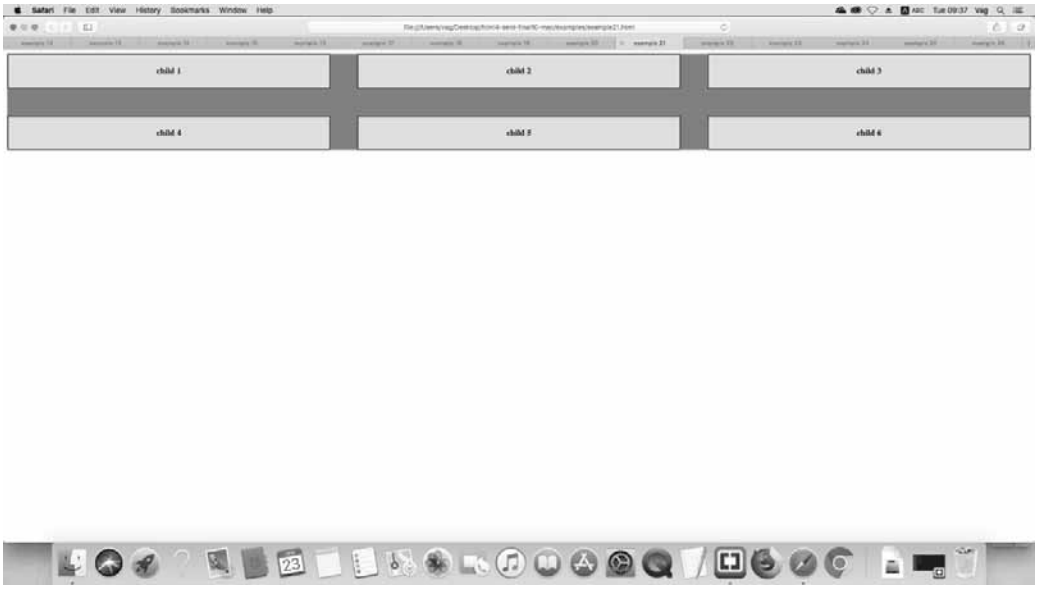
Ορίσαμε grid-column-gap ίσο με 15 pixels και grid-row-gap ίσο με 15 pixels.

Παράδειγμα 21. grid-gap

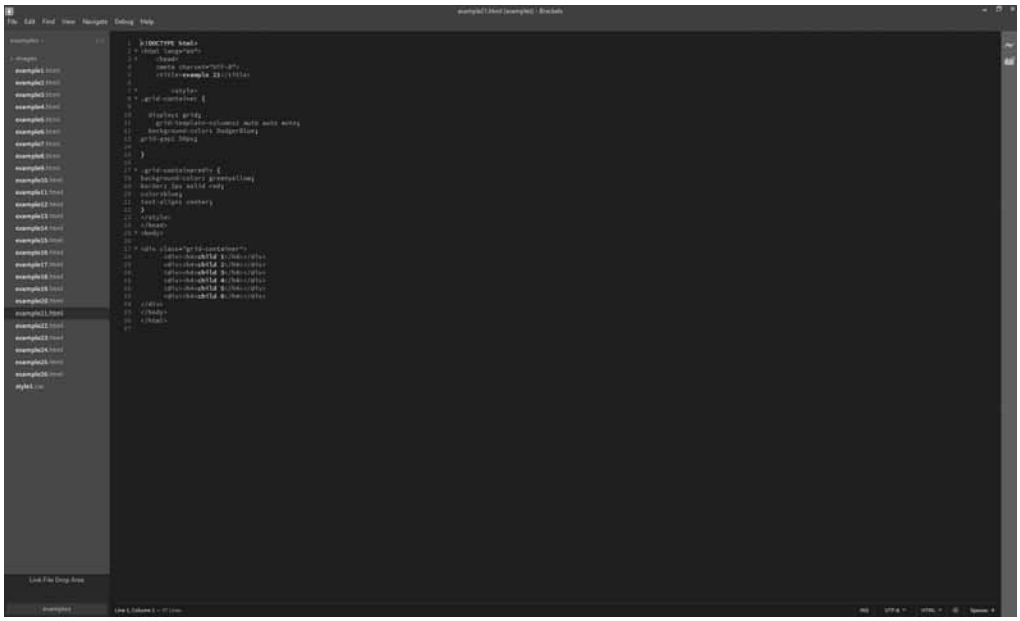
Κώδικας(Mac): grid-gap



Εικόνα Browser(Mac): grid-gap

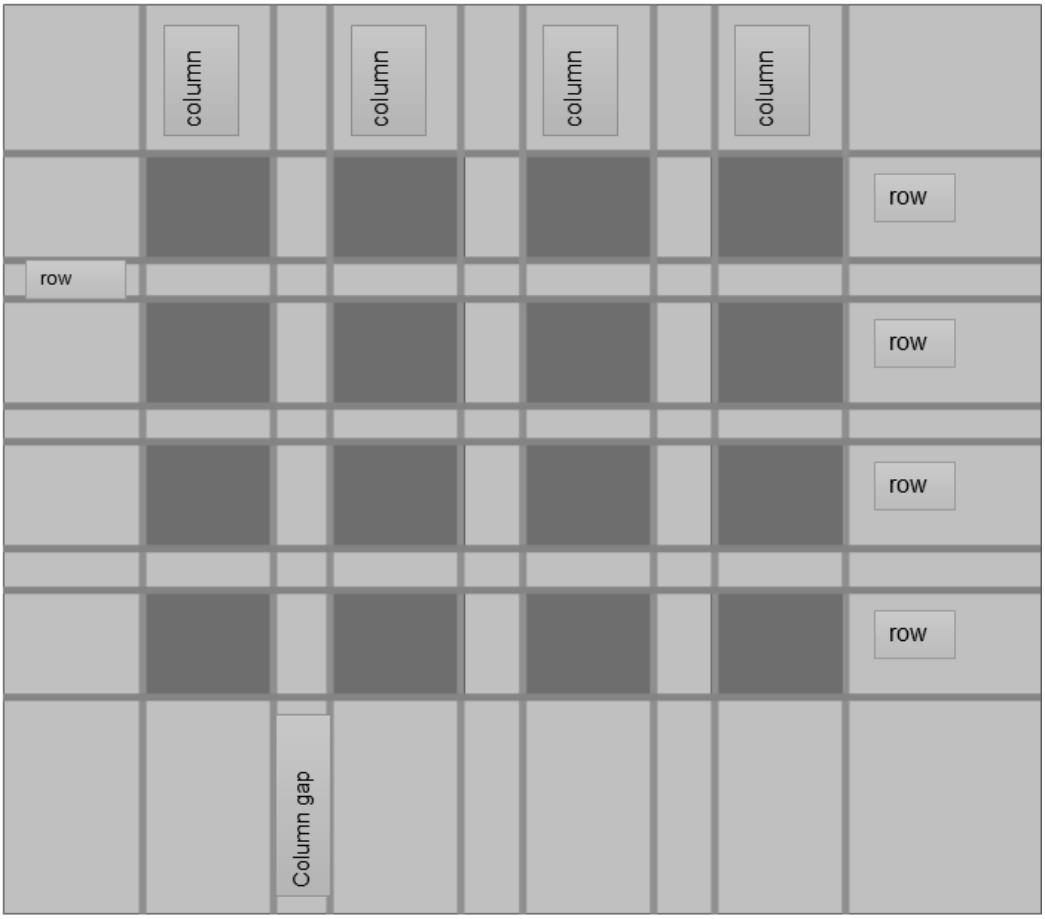


Κώδικας(Win): grid-gap



Εικόνα Browser(Win): grid-gap

Εικόνα 2. Grid System



Grid Lines

Οι γραμμές ανάμεσα στις στήλες ονομάζονται column lines

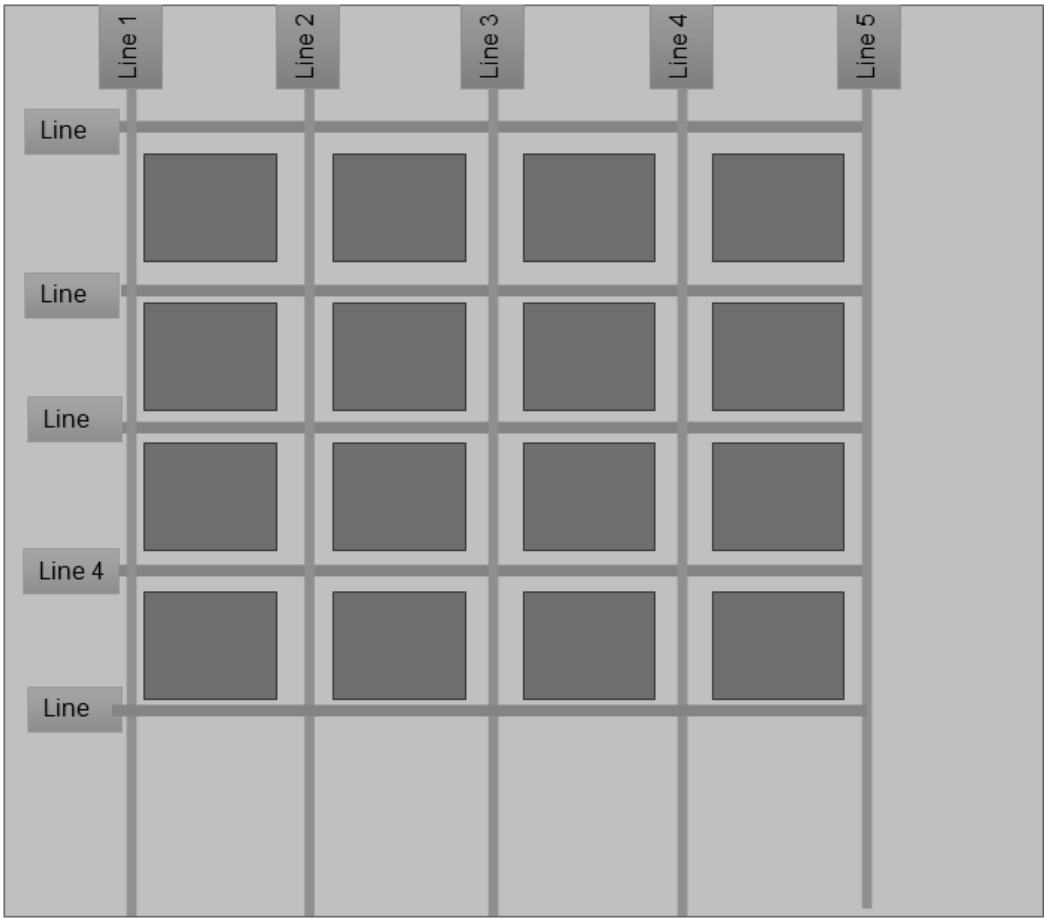
Οι γραμμές ανάμεσα στις οριζόντιες σειρές αντικειμένων ονομάζονται row lines

Μπορούμε να τοποθετήσουμε ένα grid-item έτσι ώστε να ξεκινάει από ένα column line και να τελειώνει στο κάποιο άλλο column line με τις ιδιότητες:

grid-column-start;

grid-column-end;

Εικόνα 3. Grid Lines



Παράδειγμα

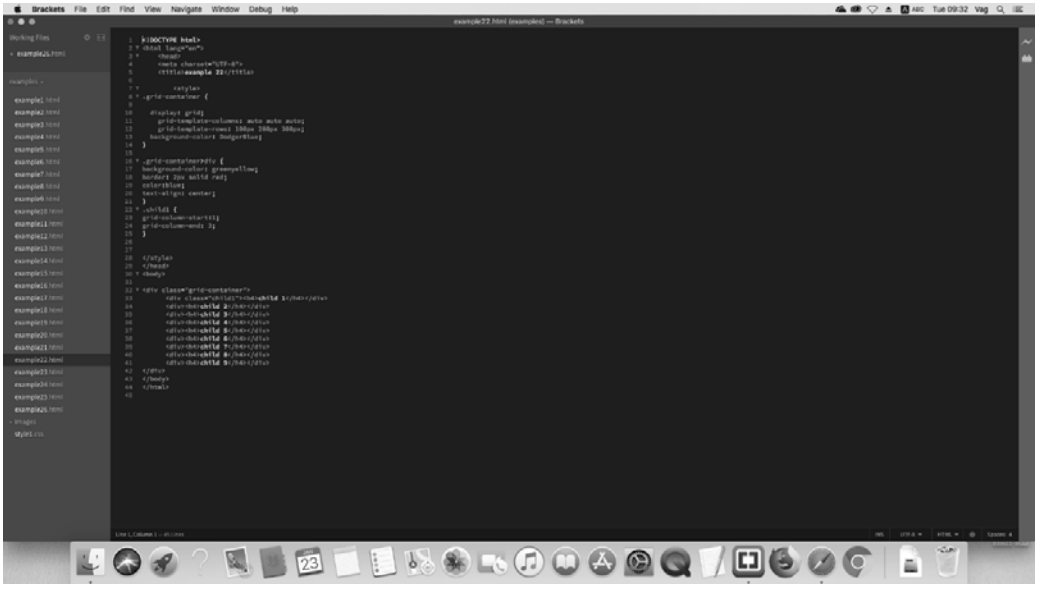
```
.child1 {
grid- column -start:1;
grid-column-end: 3;
}
```

Σχόλια

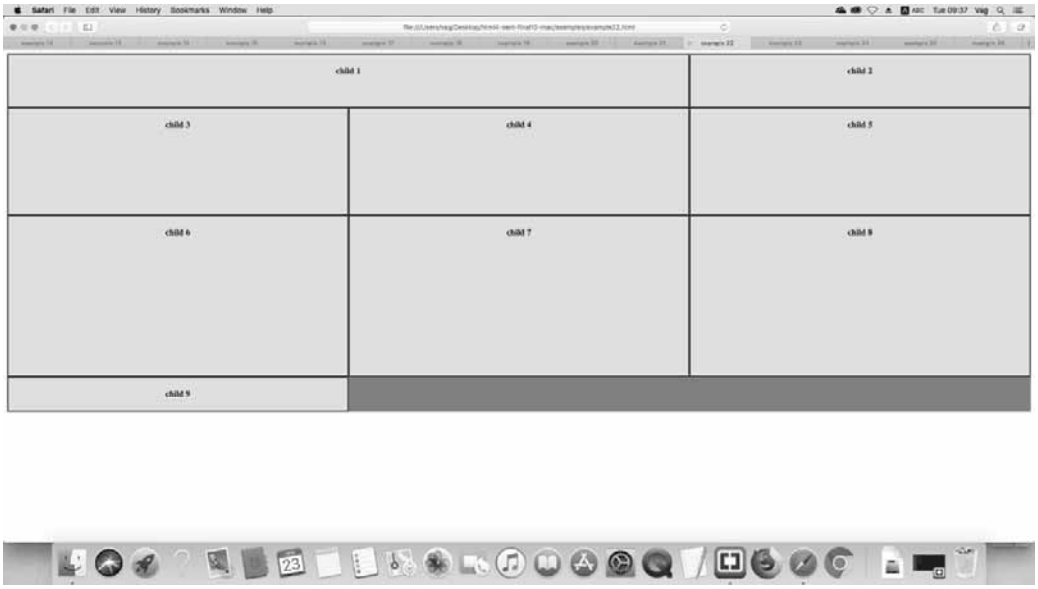
Το child 1 θα ξεκινάει στο πρώτο column line και θα τελειώνει στο 3. Στην ουσία αυτό ισοδυναμεί με τη συγχώνευση των δύο διπλανών κελιών ενός πίνακα.

Παράδειγμα 22. grid-column-start, grid-column-end

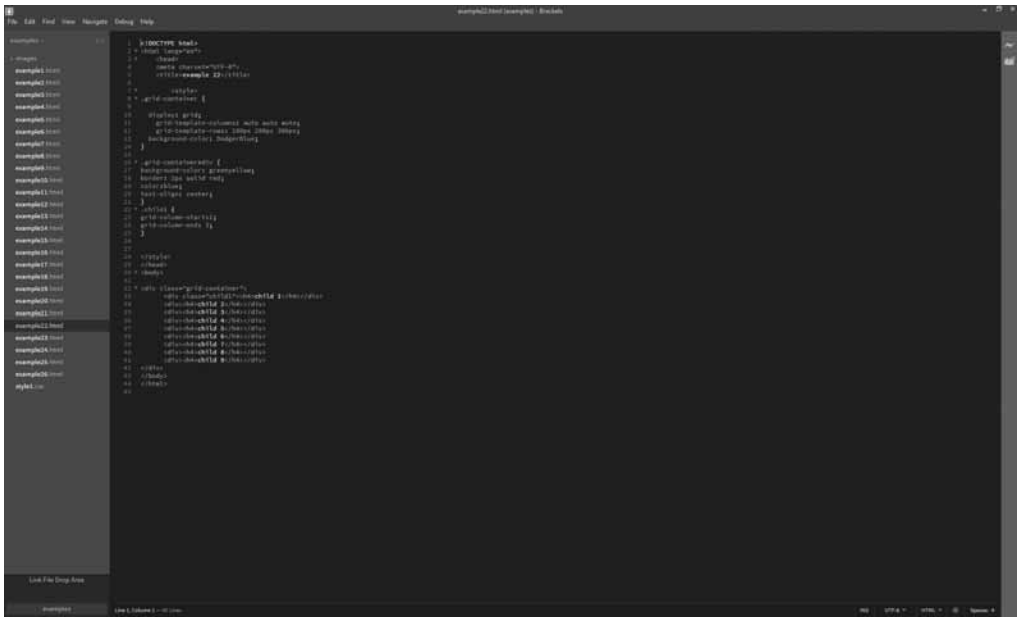
Κώδικας(Mac): grid-column-start, grid-column-end



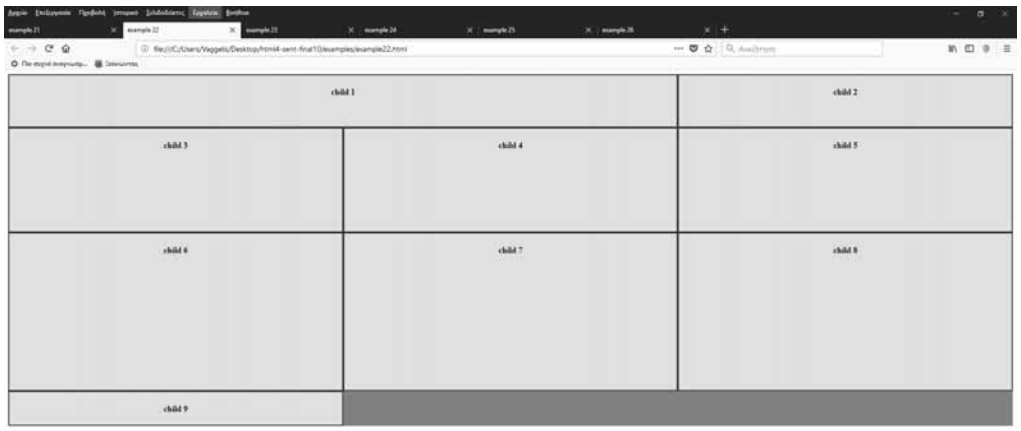
Εικόνα Browser(Mac): grid-column-start, grid-column-end



Κώδικας(Win): grid-column-start, grid-column-end



Εικόνα Browser(Win): grid-column-start, grid-column-end



Μπορούμε να τοποθετήσουμε ένα grid-item έτσι ώστε να ξεκινάει από ένα row line και να τελειώνει στο κάποιο άλλο row column line με τις ιδιότητες:

- grid-row-start;
- grid-row-end;

Παράδειγμα

```
.child2 {
grid- row -start:1;
grid-row-end: 3;
}
```

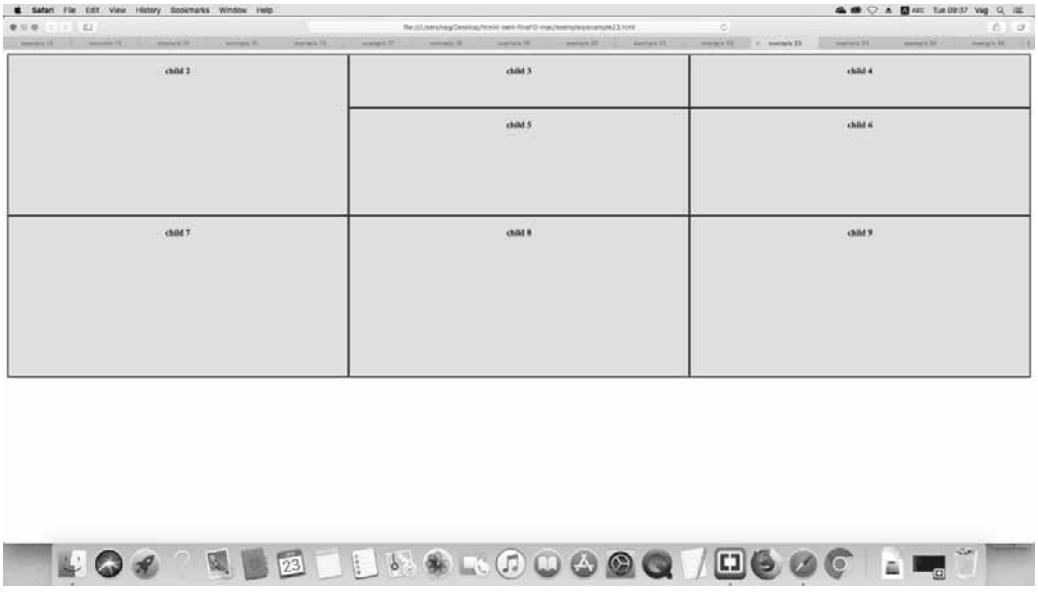
Σχόλια

Το child 2 θα ξεκινάει στο πρώτο row line και θα τελειώνει στο τρίτο. Στην ουσία αυτό ισοδυναμεί με τη συγχώνευση των δύο κελιών ενός πίνακα όπου το ένα βρίσκεται κάτω από το άλλο.

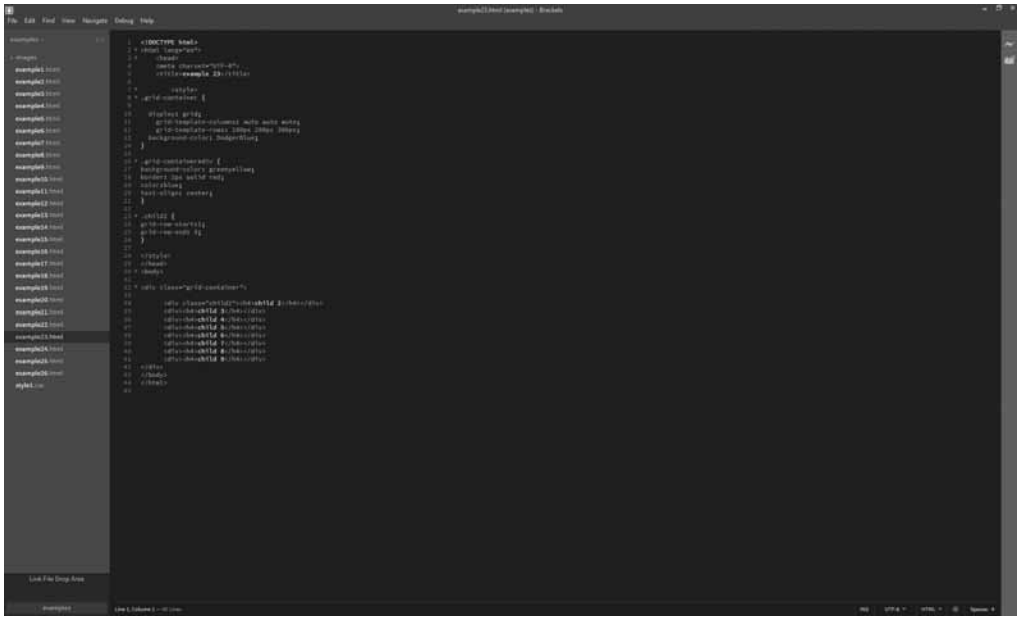
Παράδειγμα 23. grid-row-start, grid-row-end

Κώδικας(Mac): grid-row-start, grid-row-end

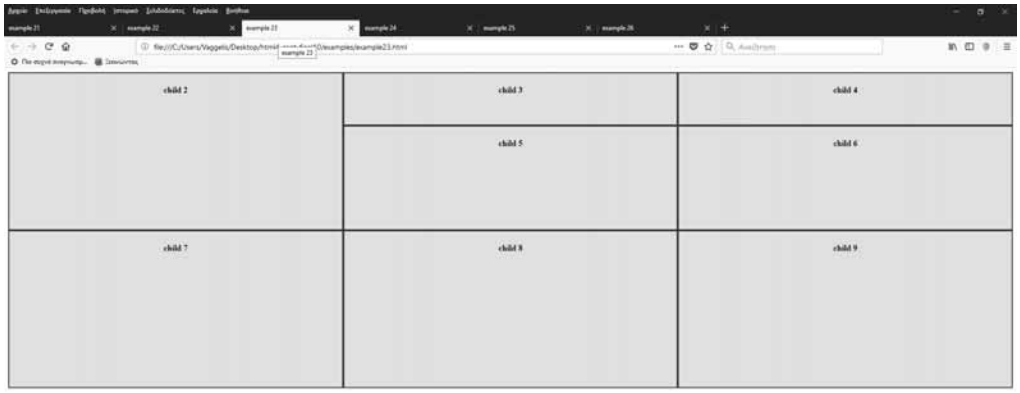
Εικόνα Browser(Mac): grid-row-start, grid-row-end



Κώδικας(Win): grid-row-start, grid-row-end



Εικόνα Browser(Win): grid-row-start, grid-row-end



3.4 Rows και Columns στο Grid System

grid-template-columns

Με την ιδιότητα grid-template-columns εντός της κλάσης του grid container μπορούμε να ορίσουμε τον αριθμό των στηλών του grid container και ταυτόχρονα ορίζουμε και το πλάτος της κάθε στήλης. Για κάθε στήλη γράφουμε το πλάτος της ή γράφουμε auto για να ρυθμιστούν όλες οι στήλες στο ίδιο πλάτος.

Θα πρέπει να διευκρινιστεί ότι ορίζουμε με την ιδιότητα grid-template-columns ***θέσεις*** για τα grid items και δε δημιουργούμε συγκεκριμένο αριθμό από grid items.

Σύνταξη

grid-template-columns: τιμή_1 τιμή_2 …;

Παράδειγμα

Έστω ότι γράφουμε στην κλάση του grid-container

```
.grid-container {
  display: grid;
  grid-template-columns: auto auto auto;
  background-color: DodgerBlue;
}
```

Με την παραπάνω κλάση ορίσαμε τρεις στήλες με τις δηλώσεις auto auto auto.

Οι τρεις στήλες αντιπροσωπεύουν τον αριθμό των διαθέσιμων θέσεων.

Έστω ότι γράφουμε στο στοιχείο body

```
<div class="grid-container">
 <div><h4>child 1</h4></div>
 <div><h4>child 2</h4></div>
 <div><h4>child 3</h4></div>
 <div><h4>child 4</h4></div>
</div>
```

Με τον παραπάνω κώδικα δημιουργήσαμε τέσσερα grid items. Εφόσον οι στήλες είναι τρεις σημαίνει ότι σε κάθε γραμμή θα μπορούμε να τοποθετήσουμε τρία παιδιά.

Οπότε τα παιδιά από 1-3 θα τοποθετηθούν στην πρώτη γραμμή ενώ αυτό που περισσεύει θα τοποθετηθεί στη δεύτερη γραμμή.

Σημείωση

Μπορούμε να ορίσουμε και στήλες με πάχος σε pixels όπως για παράδειγμα τέσσερις στήλες με πλάτη 10px, 12px, 14px και 16px:

grid-template-columns: 10px, 12px, 14px 16px;

grid-template-rows

Με την ιδιότητα grid-template-rows ορίζουμε το ύψος της κάθε γραμμής του grid container. Την ιδιότητα grid-template-rows την ενσωματώνουμε στην κλάση του grid container.

grid-template-rows: ύψος_row1 ύψος_row2 … ;

Παράδειγμα

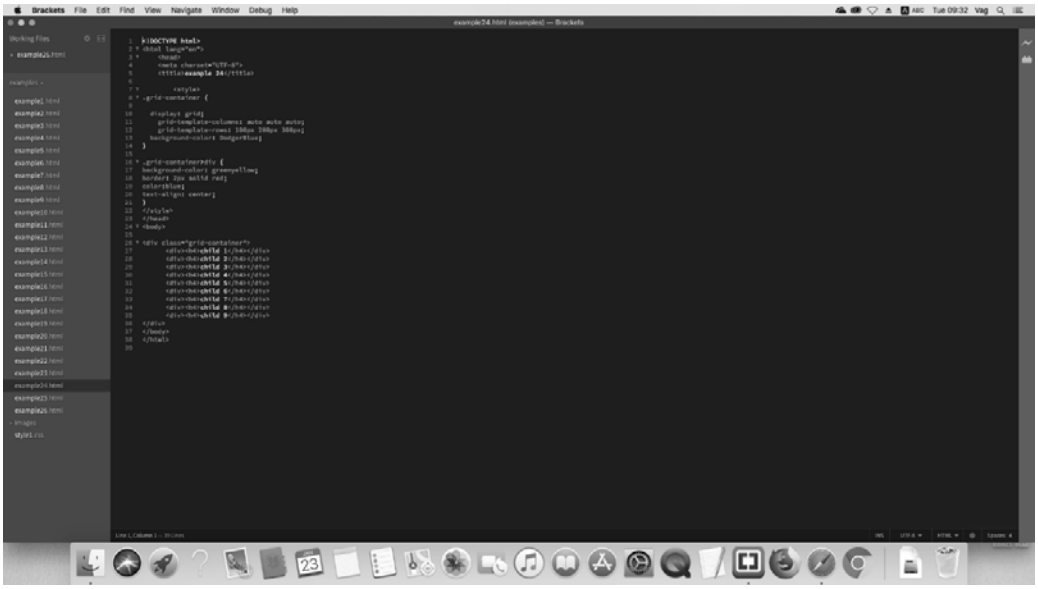
grid-template-rows: 100px 200px ;

Σχόλια

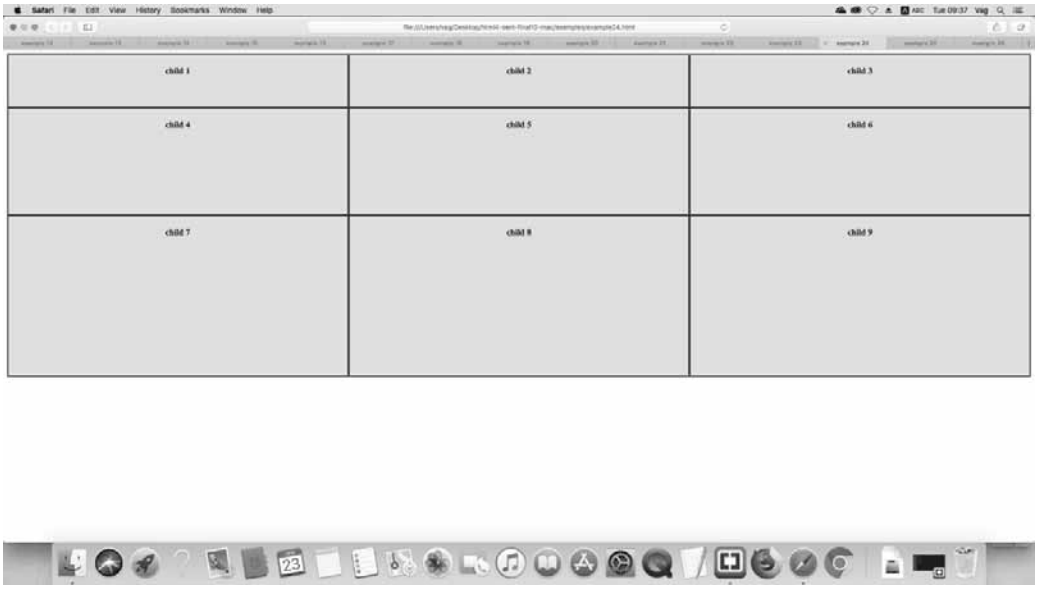
Ορίσαμε ύψος ίσο με 100 pixels για την πρώτη γραμμή από grid items και ύψος 200 Pixels για τη δεύτερη γραμμή.

Παράδειγμα 24. grid-template-columns, grid-template-rows

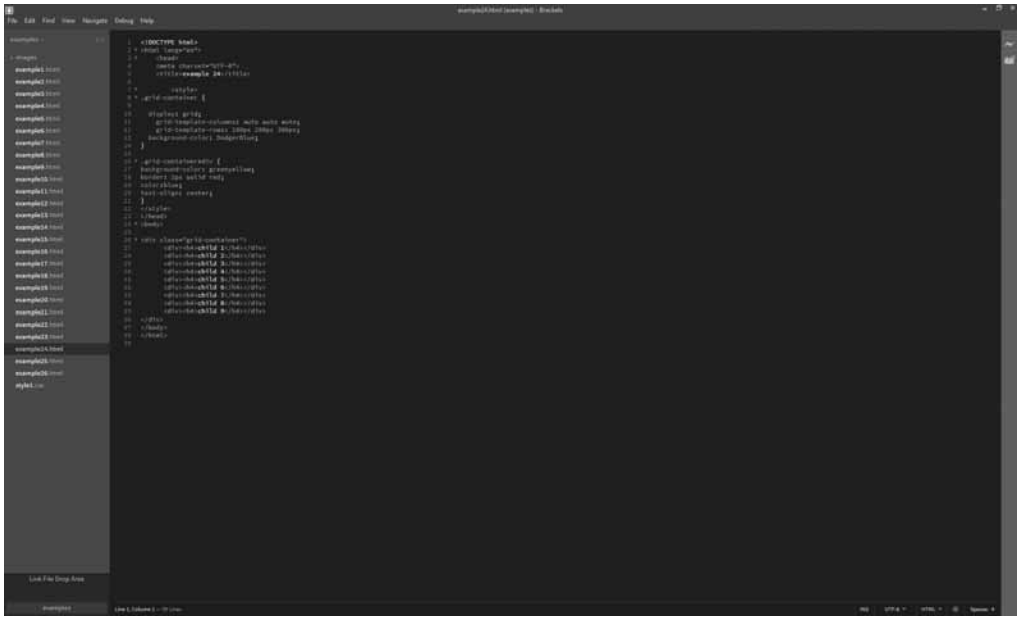
Κώδικας(Mac): grid-template-columns, grid-template-rows



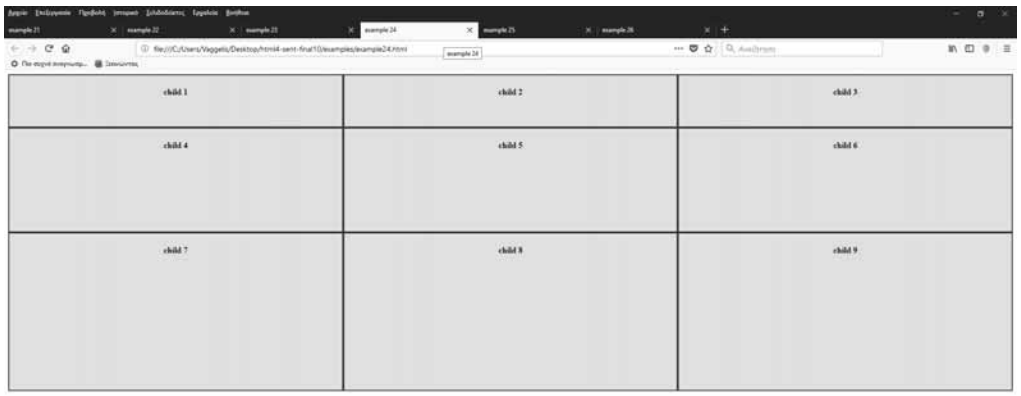
Εικόνα Browser(Mac): grid-template-columns, grid-template-rows



Κώδικας(Win): grid-template-columns, grid-template-rows



Εικόνα Browser(Win): grid-template-columns, grid-template-rows



Στοίχιση στο grid container

Στοίχιση στις στήλες (Columns)

Για να στοιχίσουμε τις στήλες (columns) του Grid System θα πρέπει να χρησιμοποιήσουμε την ιδιότητα justify-content. Τις ιδιότητες για τη στοίχιση εντός του grid container τις γράφουμε στην κλάση του grid container.

Για να ορίσουμε ίσες αποστάσεις μεταξύ των στηλών και γύρω από αυτές θα γράψουμε

```
justify-content: space-evenly;
```

Για να ορίσουμε ίσες αποστάσεις περιμετρικά των στηλών θα γράψουμε

```
justify-content: space-around;
```

Για να ορίσουμε ίσες αποστάσεις μεταξύ των στηλών θα γράψουμε

```
justify-content: space-between;
```

Για να ορίσουμε στοίχιση των στηλών στο κέντρο του grid container θα γράψουμε

```
justify-content: center;
```

Για να ορίσουμε στοίχιση των στηλών στην αρχή του grid container θα γράψουμε

```
justify-content: start;
```

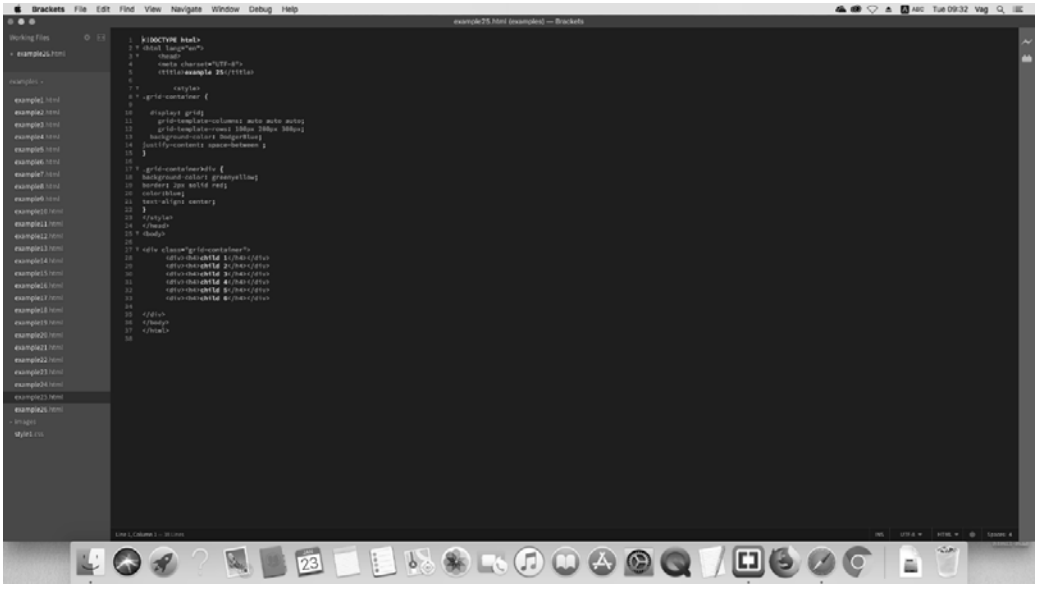
Για να ορίσουμε στοίχιση των στηλών στο τέλος του grid container θα γράψουμε

justify-content: end;

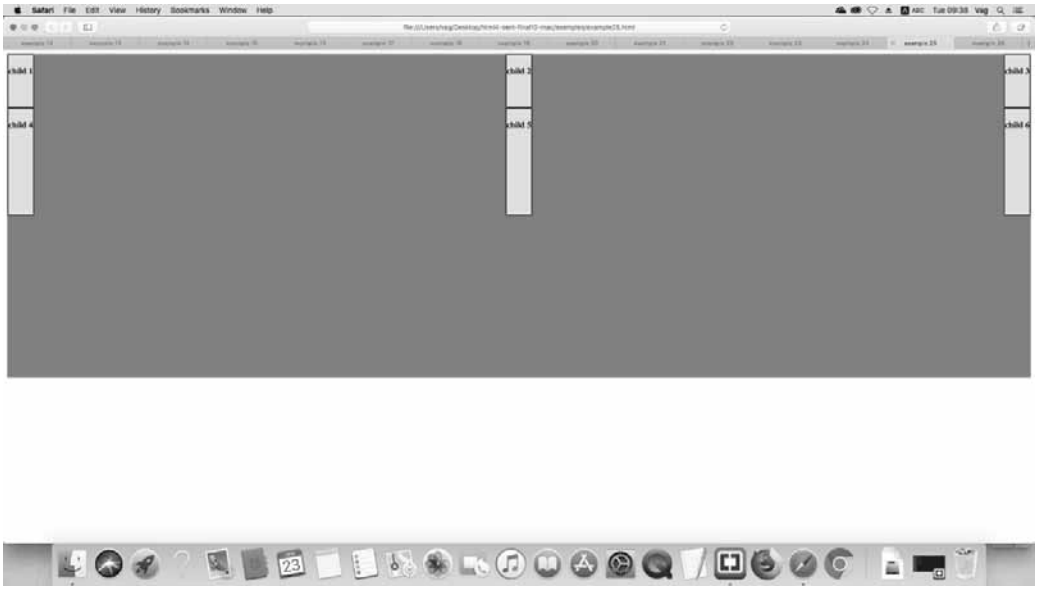
justify-content	
space-evenly	ίσες αποστάσεις μεταξύ των στηλών και γύρω από αυτές
space-around	ίσες αποστάσεις περιμετρικά των στηλών
space-between	ίσες αποστάσεις μεταξύ των στηλών
center	στοίχιση στηλών στο κέντρο
start	στοίχιση στηλών στην αρχή του grid container
end	στοίχιση στηλών στο τέλος του grid container

Παράδειγμα 25. justify-content

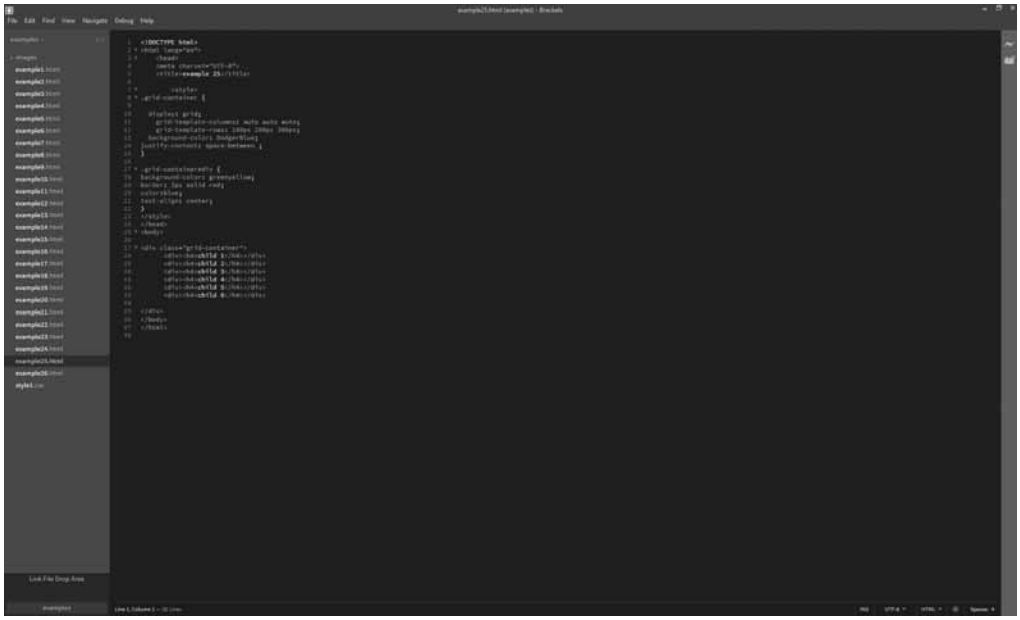
Κώδικας(Mac): justify-content



Εικόνα Browser(Mac): justify-content



Κώδικας(Win): justify-content



Εικόνα Browser(Win): justify-content

Στοίχιση των γραμμών (rows) του grid container

Για τη στοίχιση των γραμμών του grid container υπεύθυνη είναι η ιδιότητα align-content.

Για να στοιχίσουμε τις γραμμές στο κέντρο του grid container θα γράψουμε

```
align-content:center;
```

Για να ορίσουμε ίσες αποστάσεις μεταξύ των γραμμών και γύρω από αυτές θα γράψουμε

```
align-content:space-evenly;
```

Για να ορίσουμε ίσες αποστάσεις γύρω από τις γραμμές θα γράψουμε

```
align-content:space-around;
```

Για να ορίσουμε ίσες αποστάσεις μεταξύ των γραμμών θα γράψουμε

```
align-content:space-between;
```

Για να στοιχίσουμε τις γραμμές στην αρχή του container θα γράψουμε

```
align-content:start;
```

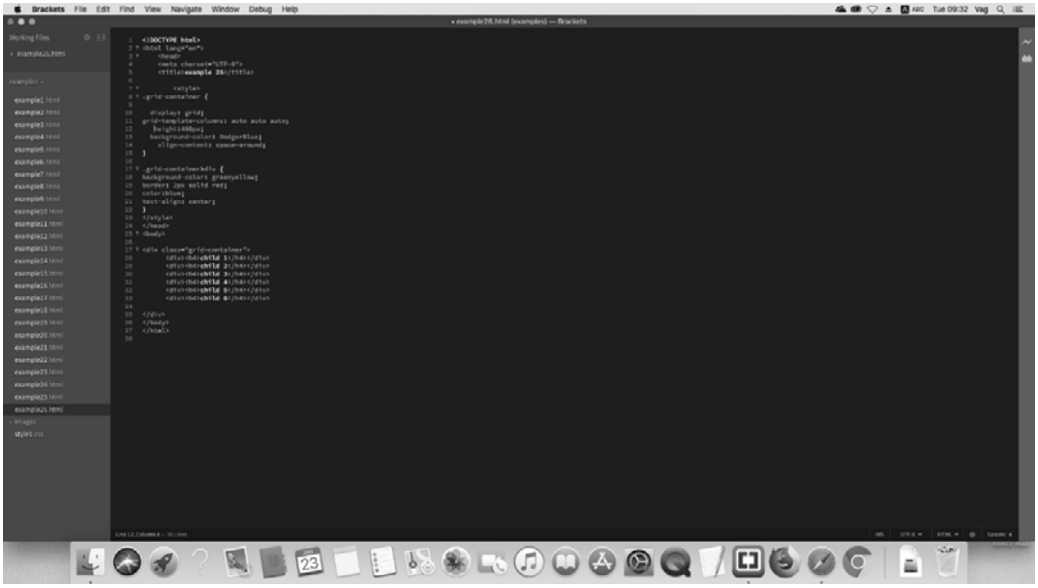
Για να στοιχίσουμε τις γραμμές στο τέλος του container θα γράψουμε

```
align-content:end;
```

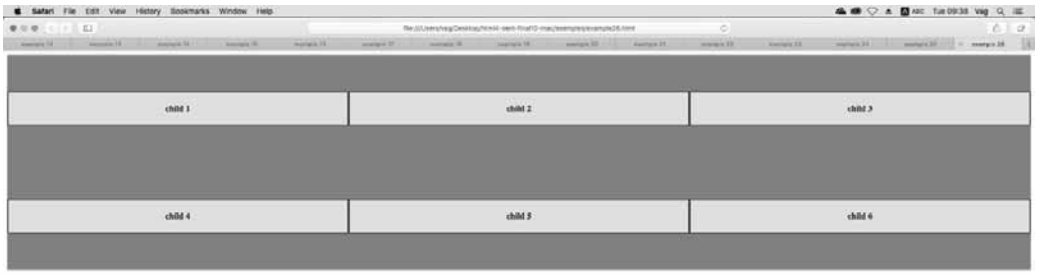
align-content	
space-evenly	ίσες αποστάσεις μεταξύ των γραμμών και γύρω από αυτές
space-around	ίσες αποστάσεις περιμετρικά των γραμμών
space-between	ίσες αποστάσεις μεταξύ των γραμμών
center	στοίχιση γραμμών στο κέντρο
start	στοίχιση γραμμών στην αρχή του grid container
end	στοίχιση γραμμών στο τέλος του grid container

Παράδειγμα 26. align-content

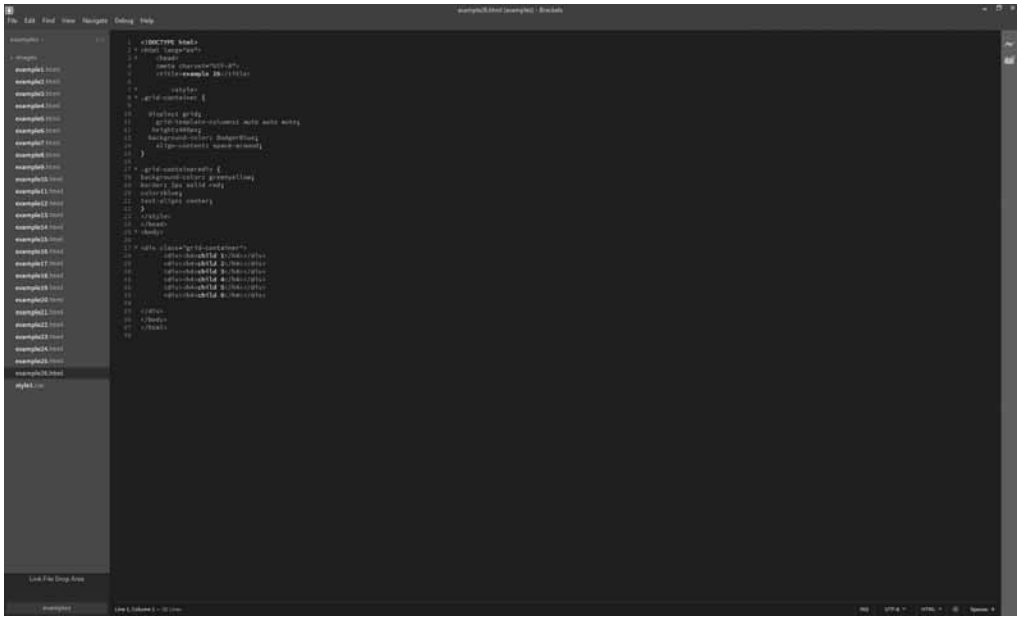
Κώδικας(Mac): align-content



Εικόνα Browser(Mac): align-content



Κώδικας(Win): align-content



Εικόνα Browser(Win): align-content

3.5 Βίντεο

Η ενότητα κλείνει με ορισμένα βίντεο που παρουσιάζουν όσα αναλύθηκαν στην παρούσα ενότητα, και τα οποία μπορείτε να δείτε στο αντίστοιχο σημείο του ηλεκτρονικού υλικού. Τα βίντεο 1-26 αφορούν mac και τα βίντεο 27-52 αφορούν pc.

Σύνοψη

Μετά την ολοκλήρωση της συγκεκριμένης διδακτικής ενότητας γνωρίζετε ότι:

- Μπορούμε να γράψουμε στο στοιχείο Head ένα link προς ένα αρχείο που θα περιέχει CSS για τη σελίδα μας.
- Με μια ψευδοκλάση μπούμε να ορίσουμε style για ειδικές καταστάσεις όπως η αιώρηση του mouse επάνω από ένα στοιχείο a.
- Με την ιδιότητα background-image μπορούμε να ορίσουμε μία εικόνα για background της σελίδας μας.
- Με την ιδιότητα padding ορίζουμε κενό χώρο γύρω από το περιεχόμενο του στοιχείου.
- Με την ιδιότητα border ορίζουμε περίγραμμα γύρω από το padding του στοιχείου.
- Με την ιδιότητα margin ορίζουμε κενό χώρο γύρω από το border του στοιχείου.
- Γράφοντας display:flex; ενεργοποιούμε το flex box.
- Με την ιδιότητα flex-direction ορίζουμε την κατεύθυνση στην οποία θα τοποθετηθούν τα παιδιά στο flex container.
- Με την ιδιότητα flex-wrap ορίζουμε αναδίπλωση στο flex container.
- Με την ιδιότητα justify-content ορίζουμε τη στοίχιση των παιδιών στον οριζόντιο άξονα.
- Με την ιδιότητα align-items ορίζουμε τη στοίχιση των παιδιών στον κατακόρυφο άξονα.
- Με την ιδιότητα order ορίζουμε τη σειρά τοποθέτησης των child elements εντός του flex-container.
- Με την ιδιότητα flex-grow ορίζουμε το κατά αναλογία μοίρασμα περισσευούμενου χώρου στα παιδιά του flex container.
- Με την ιδιότητα flex-shrink ορίζουμε αναλογία αφαίρεσης χώρου στα παιδιά του flex container για να εμφανιστεί σωστά η σελίδα στην περίπτωση που ο χώρος της σελίδας δεν επαρκεί για να εμφανιστούν τα παιδιά.
- Γράφοντας display:grid; ενεργοποιούμε το grid container.
- Γράφουμε grid-column-gap για να ορίσουμε τις αποστάσεις μεταξύ των στηλών του grid container.

- Γράφουμε grid-row-gap για να ορίσουμε τις αποστάσεις μεταξύ των γραμμών του grid container.
- Με τη βοήθεια της ιδιότητας grid-template-columns εντός της κλάσης του grid container μπορούμε να ορίσουμε τον αριθμό των στηλών του grid container και ταυτόχρονα ορίζουμε και το πλάτος της κάθε στήλης.

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

Προγράμματα Συμπληρωματικής Εκπαίδευσης

Με τη χρήση καινοτόμων μεθόδων εξ αποστάσεως εκπαίδευσης

1937 - ΔΗΜΙΟΥΡΓΙΑ RESPONSIVE ΣΕΛΙΔΩΝ ΜΕ HTML5 ΚΑΙ CSS

ΔΙΔΑΚΤΙΚΗ ΕΝΟΤΗΤΑ 5. ΔΗΜΙΟΥΡΓΙΑ TEMPLATES ΜΕ ΤΗ ΒΟΗΘΕΙΑ ΤΗΣ HTML5

Εισαγωγικές Παρατηρήσεις

Στην υποενότητα 1 θα γίνει παρουσίαση των δυνατότητων που προσφέρουν τα media queries και θα αναλυθούν παραδείγματα. Επίσης θα αναλυθεί ο τρόπος που μπορούμε να θέσουμε σε animation κάποιο στοιχείο της σελίδας.

Στην υποενότητα 2 θα παρουσιαστούν ιδιότητες για να δημιουργούμε συγκεκριμένες δομές στο Grid System. Επίσης θα αναλυθεί βήμα προς βήμα η δημιουργία template με την εισαγωγή εικόνας και κειμένου.

Στην υποενότητα 3 θα παρουσιαστούν και θα αναλυθούν οι απαραίτητες ιδιότητες για τη δημιουργία σελίδων, βασιζόμενων στο flexbox model. Επίσης θα αναλυθεί βήμα προς βήμα η δημιουργία σελίδας με τη βοήθεια του flexbox.

ΥΠΟΕΝΟΤΗΤΑ 1. MEDIA QUERIES ΚΑΙ CSS ANIMATION

Με τα Media Queries μπορούμε να εφαρμόζουμε CSS κάτω από ειδικές συνθήκες και παραμέτρους όπως:

- Διαστάσεις width και height του viewport (η περιοχή του Browser που είναι ορατή από το χρήστη)
- Διαστάσεις width και height της συσκευής
- Προσανατολισμός-orientation της συσκευής.
- Ανάλυση

1.1 Media Queries

Η σύνταξη για τα Media Queries είναι η παρακάτω:

```
@media mediatype not ή only ή and (εκφράσεις) {
CSS code;
}
```

Επίσης μπορούμε να γράψουμε σε link εντός του στοιχείου Head ένα media query.

```
<link rel="stylesheet" media="mediatype and|not|only (expressions)" href="όνομα_αρχείου.css">
```

Στη θέση mediatype μπορούμε να αντικαταστήσουμε μία από τις παρακάτω τιμές:

media type	
all	όλες οι συσκευές
print	εκτυπωτές
screen	οθόνες υπολογιστών, tablets, smart-phones κτλ.
speech	screenreaders

Τα screenreaders είναι μία κατηγορία εφαρμογών που διαβάζουν και εκφωνούν το περιεχόμενο για άτομα με πρόβλημα στην όραση.

Για να ορίζουμε συνθήκες με μήκος πλάτους υπάρχουν οι παρακάτω επιλογές:

1. Διαστάσεις παραθύρου του browser

Διαστάσεις παραθύρου του browser	
min-width	ελάχιστο πλάτος
max-width	μέγιστο πλάτος
min-height	ελάχιστο ύψος
max-height	μέγιστο ύψος
width	πλάτος
height	ύψος

2. Διαστάσεις οθόνης συσκευής

Διαστάσεις οθόνης της συσκευής	
min-device-width	ελάχιστο πλάτος
max- device-width	μέγιστο πλάτος
min- device-height	ελάχιστο ύψος
max- device-height	μέγιστο ύψος
width	πλάτος
height	ύψος

Παράδειγμα

Ανάμεσα στις ετικέτες <style> και </style> θα γράψουμε CSS ιδιότητες για ένα div εντός της κλάσης .div1. Πιο συγκεκριμένα θα ορίσουμε διαστάσεις 300X300 pixels και θα ορίσουμε χρώμα φόντου (background-color) καθώς και χρώμα γραμματοσειράς.

```
.div1 {
width:300px;
height:300px;
background-color: dodgerblue;
color:blueviolet;
  }
```

Στη συνέχεια θα γράψουμε πάλι την κλάση .div αλλά αυτή τη φορά εντός του παρακάτω media query:

```
@media screen and (max-width: 600px) {
  .div1 {
width:300px;
height:300px;
background-color: yellow;
color:red;
   }
   }
```

Σχόλια

Το media query θα ενεργοποιηθεί όταν το πλάτος της οθόνης του Browser γίνει μικρότερο ή ίσο με 600 pixels.

Για να γράψουμε το Media query αρχικά γράψαμε @media και έπειτα γράψαμε το media type (screen). Τέλος, προσθέσαμε τη δήλωση and και την ιδιότητα για το μέγιστο πλάτος (max-width: 600px)

@media screen and (max-width: 600px)

Η μετάφραση του Media Query είναι:

Όταν έχουμε οθόνη (screen) και (and) το μέγιστο πλάτος της οθόνης είναι ίσο με 600 pixels, τότε θα εφαρμοστούν οι παρακάτω κανόνες.

Σημείωση

Μπορούμε να χρησιμοποιήσουμε τα min,max height ή width και εντός κανόνων CSS:

Παράδειγμα

```
  .div1 {
min-height:300px;
   }
```

Σχόλια

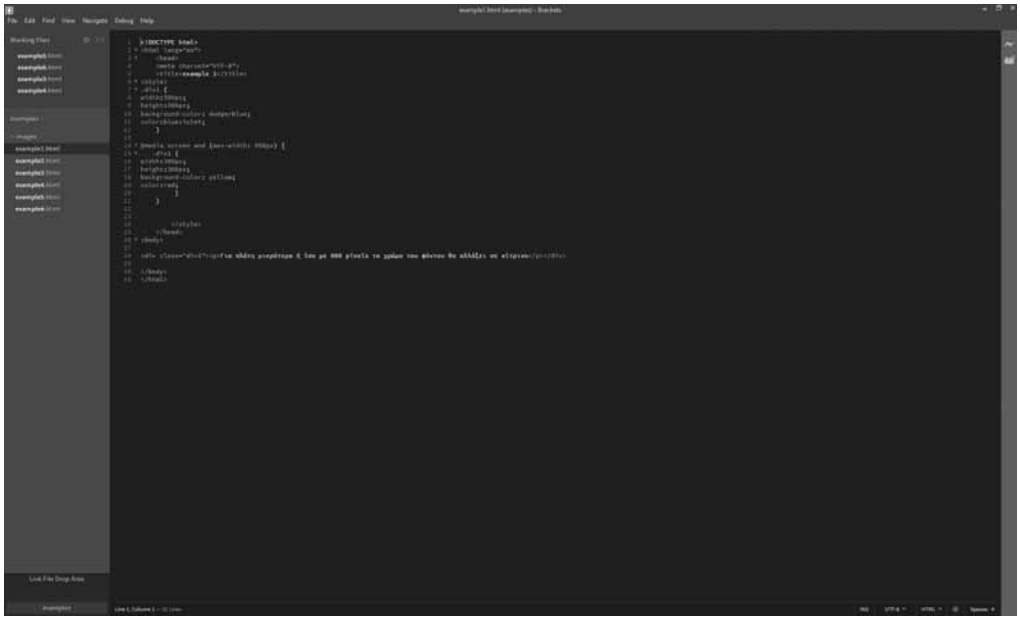
Με τη συγκεκριμένη ιδιότητα εξασφαλίζουμε ότι το ύψος του div δε θα γίνει μικρότερο από 300 pixels.

Παράδειγμα 1. Media Query

Κώδικας(Mac): media query

Εικόνα Browser(Mac): media query

Κώδικας(Win): media query



Εικόνα Browser(Win): media query

1.2 CSS animation

Το CSS animation ορίζεται ως η σταδιακή αλλαγή CSS style ενός στοιχείου. Στο CSS animation μπορούμε να συμπεριλάβουμε όσες ιδιότητες επιθυμούμε και να αλλάξουμε το CSS style πάλι όσες φορές επιθυμούμε.

Βασικό συστατικό στο CSS animation αποτελούν τα @keyframes. Με τα @keyframes μπορούμε να ορίζουμε το style που πρέπει να έχει ένα

συγκεκριμένο στοιχείο σε μία συγκεκριμένη χρονική στιγμή που αντιστοιχεί σε κάποια φάση του animation.
Μπορούμε να ορίσουμε animation με δύο τρόπους.

Α' τρόπος

Αρχικά θα γράψουμε CSS για την normal κατάσταση του στοιχείου. Έτσι λοιπόν μπορούμε με μία κλάση ή ένα id να ορίσουμε τη normal κατάσταση του animation. Μετά το τέλος του animation το στοιχείο που θα εκτελέσει το animation θα επιστρέψει σε αυτή την κατάσταση.

```
.όνομα_κλάσης {
CSS ιδιότητες;
    animation-name: όνομα_animation ;
    animation-duration:  _s;
}
```

Στην κλάση που θα περιλαμβάνει τις ιδιότητες του στοιχείου όταν δε θα εκτελείται το animation θα γράψουμε το όνομα του animation ως τιμή στην ιδιότητα animation-name και θα γράψουμε τη χρονική διάρκεια του animation σε δευτερόλεπτα π.χ.:

```
animation-duration:  7s;
```

Στη συνέχεια θα ορίσουμε @keyframes μόνο για την αρχή και το τέλος του animation με την παρακάτω σύνταξη:

```
@keyframes animation1 {
    from {
CSS code;
    }
    to { CSS code;
    }
}
```

Η αρχή του animation δηλώνεται ως "from" ενώ το τέλος του animation δηλώνεται ως "to".

Στη συνέχεια θα συνδέσουμε ένα στοιχείο με την κλάση που γράψαμε για το style του στοιχείου όταν δεν θα εκτελεί το animation (normal).

```
<div class="…"> </div>
```

Παράδειγμα

Αρχικά θα γράψουμε μία κλάση .anim1 με τιμές για πλάτος, ύψος και χρώμα φόντου. Επίσης θα δηλώσουμε και το όνομα του animation που θα εκτελέσει το στοιχείο καθώς και τη χρονική διάρκεια. Το όνομα του animation θα είναι animation1 και η διάρκεια του animation θα είναι επτά δευτερόλεπτα.

```
.anim1 {
    width: 300px;
    height: 300px;
    background-color: darkcyan;
    animation-name: animation1 ;
    animation-duration: 7s;
    color:red;
}
```

Στη συνέχεια θα γράψουμε @keyframes και δίπλα το όνομα του animation. Εντός του @keyframes θα γράψουμε ιδιότητες για την έναρξη του animation1 (from) καθώς και ιδιότητες για τη λήξη του animation1.

```
@keyframes animation1 {
    from {background-color: darkcyan;
    color:orangered;
    font-size: 20px;
    }
    to {background-color: dodgerblue;
    color:red;
    font-size:30px;
    }
}
```

Τέλος εντός του στοιχείου body θα γράψουμε ένα div και θα γράψουμε το όνομα της κλάσης . anim1.

```
<div class="anim1">
   <p>αλλαγή χρωμάτων</p>
</div>
```

Η λειτουργία του animation στο παράδειγμα είναι η εξής:

Γράψαμε την κλάση .anim για ιδιότητες εκτός animation

Εντός της κλάσης .anim1 γράψαμε το όνομα του animation που είναι "animation1"

Γράψαμε @keyframes με το όνομα animation1

Γράψαμε ένα div με όνομα κλάσης .anim

Οπότε οι συνδέσεις για να δουλέψει το animation είναι.

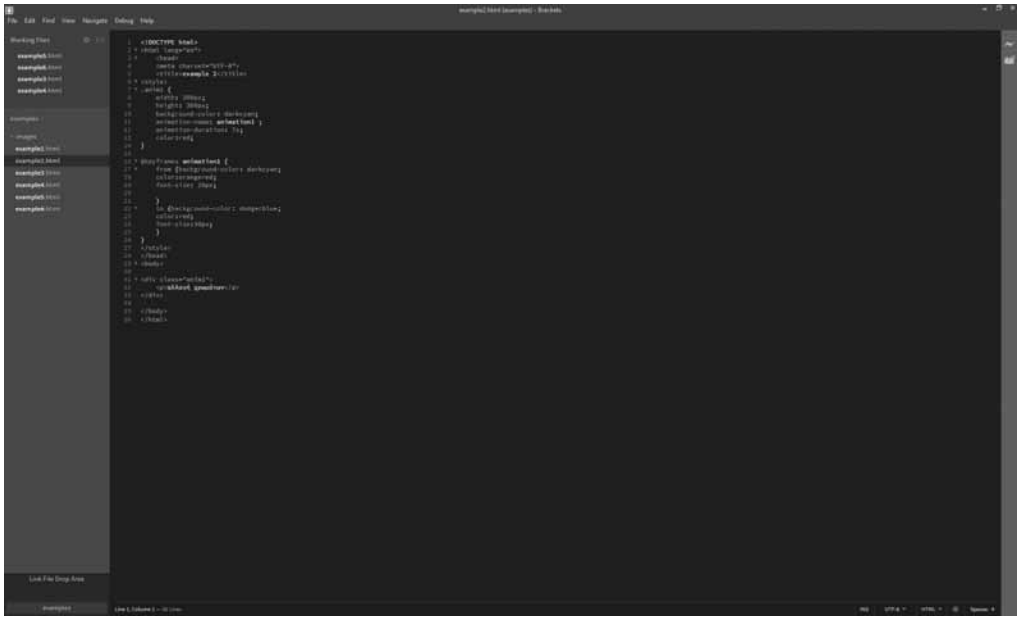
Το div με κλάση .anim1 παίρνει το όνομα του animation που είναι το animation1. Στη συνέχεια συνδέεται το όνομα animation1 με τα @keyframes που έχουμε γράψει για αυτό.

Παράδειγμα 2. CSS animation με δύο keyframes

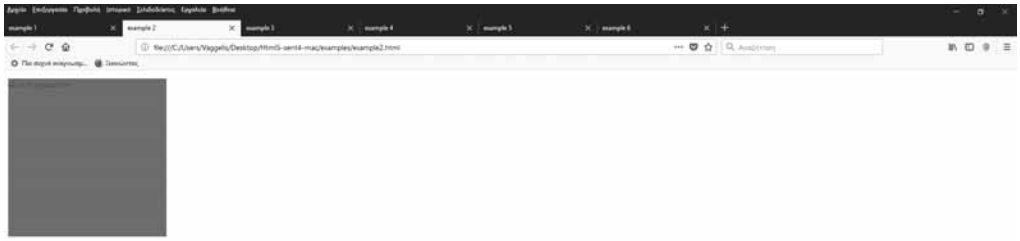
Κώδικας(Mac): CSS animation με δύο @keyframes

Εικόνα Browser(Mac): CSS animation με δύο @keyframes

Κώδικας(Win): CSS animation με δύο @keyframes



Εικόνα Browser(Win): CSS animation με δύο @keyframes



Β' τρόπος

Στον δεύτερο τρόπο αλλάζει μόνο ο τρόπος που ορίζουμε τα @keyframes. Μπορούμε να γράψουμε όσα keyframes επιθυμούμε ξεκινώντας από το 0% και καταλήγοντας στο 100%.

Παράδειγμα

```
@keyframes ...{
0% { ...}
10% { ...}
70% { ...}
100% { ...}
}
```

Παράδειγμα

Βασιζόμενοι στο προηγούμενο παράδειγμα θα δημιουργήσουμε ένα div με διαστάσεις 800X800 και χρώμα φόντου κόκκινο με την κλάση .box1.

```
.box1 {
width: 800px;
height: 800px;
background-color: red;
}
```

Στη συνέχεια θα χρησιμοποιήσουμε το div με το animation1, για να εκτελέσει animation περνώντας από τις τέσσερις γωνίες του γονέα div. Στο μικρό div θα γράψουμε

```
width: 200px;
```

και

```
height: 200px;
```

για τις διαστάσεις του και θα γράψουμε τη δήλωση

```
position:relative;
```

για relative positioning.

Στη συνέχεια θα γράψουμε `@keyframes` έτσι ώστε το μικρό div να περνάει από τις γωνίες του μεγάλου div.

```
Επάνω αριστερή γωνία
0%{left:0px; top:0px;}
Επάνω δεξιά γωνία
25%{  left:600px; top:0px;}
Κάτω δεξιά γωνία
75%{   left:600px;  top:600px;}
Κάτω αριστερή γωνία
85% {   left:0px;  top:600px;}
Επάνω αριστερή γωνία
100%  {   left:0px;  top:0px;  }
}
```

Επιπλέον ιδιότητες για το animation

Animation-delay

Μπορούμε να γράψουμε animation-delay:_s για να καθυστερήσει το animation.

Αν για παράδειγμα επιθυμούμε να καθυστερήσει το animation για 9 δευτερόλεπτα μετά τη φόρτωση της σελίδας θα γράψουμε:

```
animation-delay: 9s;
```

animation-iteration-count

Μπορούμε να ορίσουμε τον αριθμό των επαναλήψεων ενός animation με την ιδιότητα:

```
animation-iteration-count: αριθμός_επαναλήψεων;
```

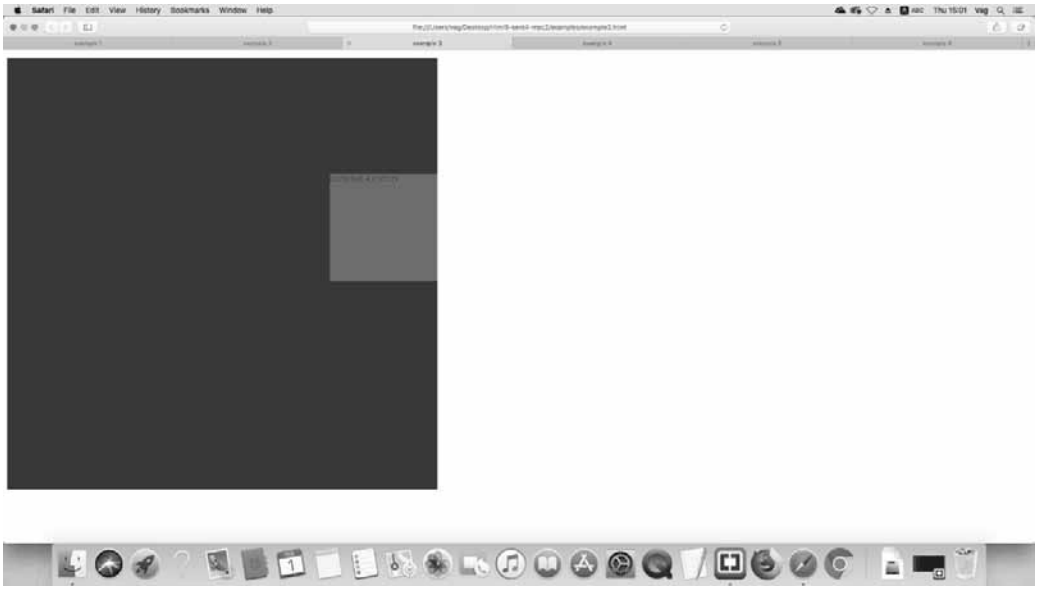
Αν για παράδειγμα επιθυμούμε ένα animation να εκτελεστεί 9 φορές θα γράψουμε:

```
animation-iteration-count: 9 ;
```

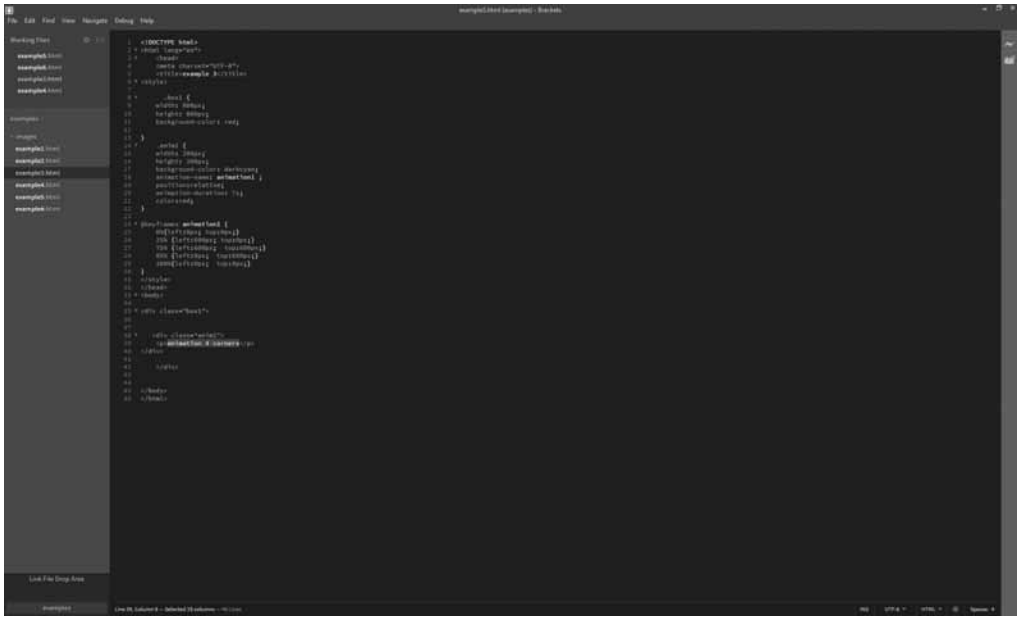
Παράδειγμα 3. CSS animation με πολλαπλά keyframes

Κώδικας(Mac): CSS animation με πολλαπλά @keyframes

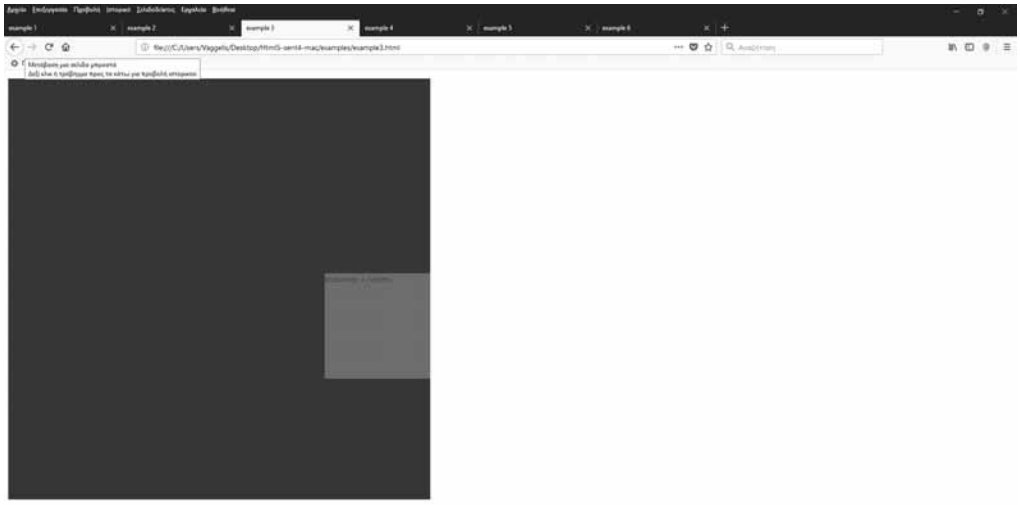
Εικόνα Browser(Mac): CSS animation με πολλαπλά @keyframes



Κώδικας(Win): CSS animation με πολλαπλά @keyframes



Εικόνα Browser(Win): CSS animation με πολλαπλά @keyframes



Ιδιότητα border-radius

Με την ιδιότητα border-radius μπορούμε να προσθέσουμε καμπύλες στις γωνίες κάποιου div.

Μπορούμε να χρησιμοποιήσουμε μονάδες σε px ή σε ποσοστά %. Στην ουσία η τιμή στο border-radius αντιστοιχεί στην ακτίνα με την οποία θα σχεδιαστούν οι καμπύλες στις γωνίες.

border-radius με μία τιμή

Προσθέτει καμπύλες σε όλες τις γωνίες.

```
border-radius: τιμή_1 ;
```

Παράδειγμα

```
border-radius: 15px;
```

border-radius με τέσσερις τιμές

```
border-radius: τιμή_1 τιμή_2 τιμή_3 τιμή_4 ;
```

border-radius	
τιμή_1	επάνω αριστερή γωνία
τιμή_2	επάνω δεξιά γωνία
τιμή_3	κάτω δεξιά γωνία
τιμή_4	κάτω αριστερή γωνία

Παράδειγμα

```
border-radius: 15px 20px 35px 45px;
```

border-radius με τρεις τιμές

```
border-radius: τιμή_1 τιμή_2 τιμή_3
```

border-radius	
τιμή_1	επάνω αριστερή γωνία
τιμή_2	επάνω δεξιά γωνία και κάτω αριστερή γωνία
τιμή_3	κάτω δεξιά γωνία

Παράδειγμα

```
border-radius: 15px 20px 35px;
```

border-radius με δύο τιμές

```
border-radius: τιμή_1 τιμή_2 ;
```

border-radius	
τιμή_1	επάνω αριστερή γωνία και κάτω δεξιά γωνία
τιμή_2	επάνω δεξιά γωνία και κάτω αριστερή γωνία

Παράδειγμα

```
border-radius: 15px 20px ;
```

animation με κυκλικό δίσκο

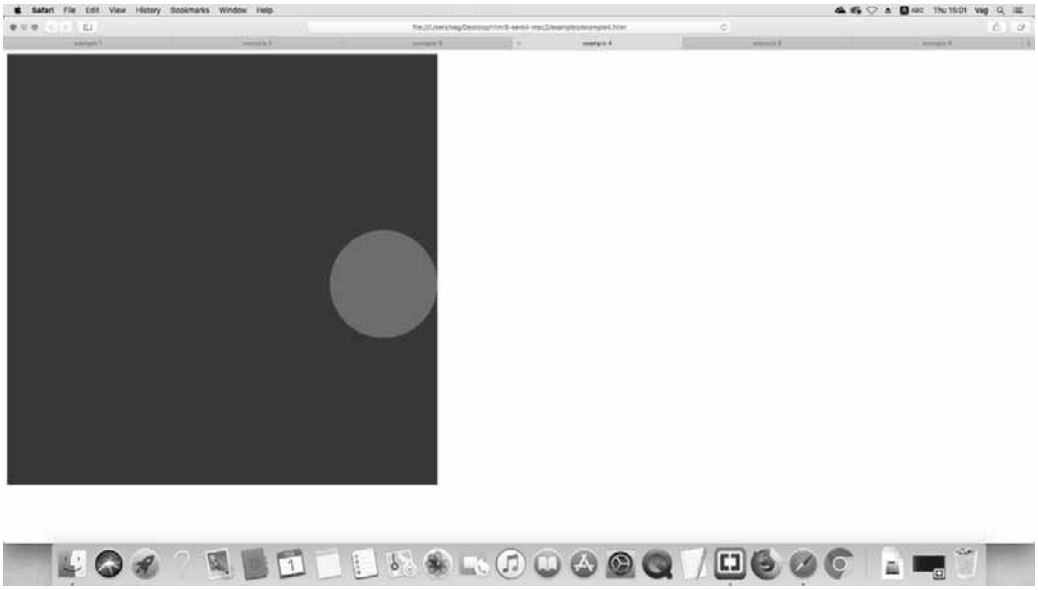
Στο προηγούμενο παράδειγμα μπορούμε εύκολα να αντικαταστήσουμε το τετράγωνο div που εκτελεί animation προσθέτοντας απλά μία ιδιότητα border-radius με 50% σε όλες τις γωνίες.

```
.anim1 {
width: 200px;
height: 200px;
border-radius: 50%;
background-color: darkcyan;
animation-name: animation1 ;
   position:relative;
animation-duration: 7s;
color:red;
}
```

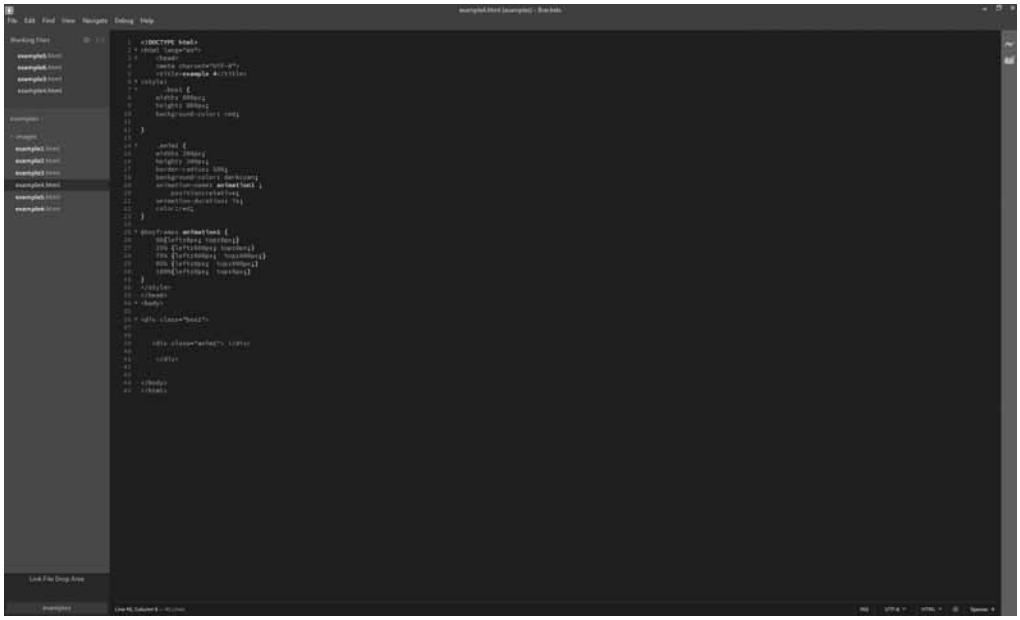
Παράδειγμα 4. CSS animation κυκλικού δίσκου με πολλαπλά keyframes

Κώδικας(Mac): CSS animation κυκλικού δίσκου με πολλαπλά @keyframes

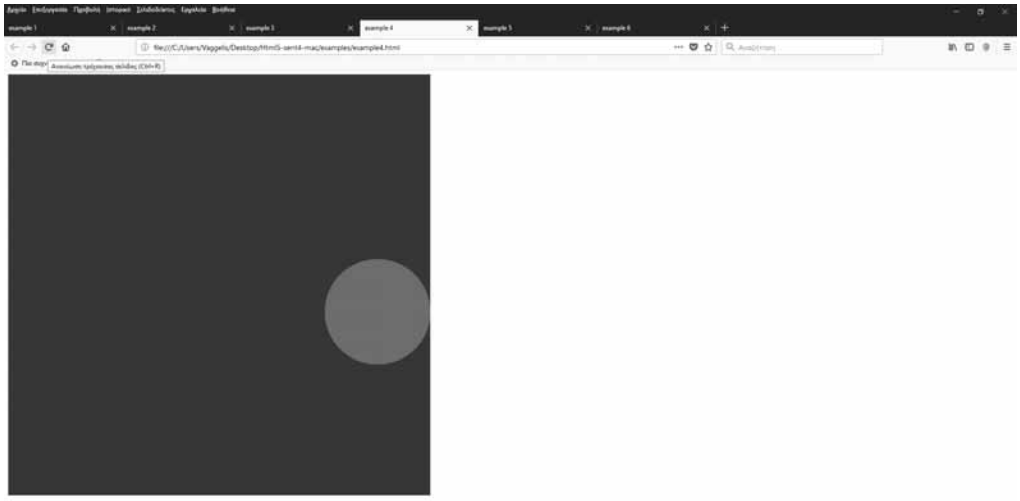
Εικόνα Browser(Mac):CSS animation κυκλικού δίσκου με πολλαπλά @keyframes



Κώδικας(Win): CSS animation κυκλικού δίσκου με πολλαπλά @keyframes



Εικόνα Browser(Win): CSS animation κυκλικού δίσκου με πολλαπλά @keyframes



ΥΠΟΕΝΟΤΗΤΑ 2. GRID SYSTEM TEMPLATE

Σε αυτή την υποενότητα θα παρουσιαστούν διάφορες ιδιότητες του Grid System και Grid Template.

2.1 Συμπληρωματικές ιδιότητες για το Grid System

grid-column shorthand

Το Grid-column shorthand είναι ένα shorthand που περιλαμβάνει τις ιδιότητες grid-column-start και grid-column-end.

Η σύνταξη είναι η παρακάτω:

```
grid-column: αριθμός_γραμμής_έναρξης / αριθμός_γραμμής_λήξης ;
```

Παράδειγμα

```
.item1{
grid-column: 1/ 5 ;}
```

Σχόλια

Με τις παραπάνω τιμές για το grid-column, το grid item θα ξεκινάει από την αρχή της στήλης 1(column line1) και θα τερματίζει στην αρχή της στήλης 5 (column line 5). Άρα το πλάτος του θα ισούται με το πλάτος τεσσάρων items.

Εναλλακτικά στη δεύτερη τιμή μπορούμε να γράψουμε τη δήλωση span έτσι ώστε να δηλώσουμε τον αριθμό των θέσεων που θα καλύπτει.

Έτσι λοιπόν το προηγούμενο παράδειγμα θα γίνει:

```
.item1{
grid-column: 1/ span 4 ;}
```

Σημείωση

Με την ιδιότητα grid-column πετυχαίνουμε να ενώσουμε γειτονικές θέσεις στην ίδια γραμμή.

grid-row shorthand

Με την ιδιότητα grid-row μπορούμε να ορίζουμε ταυτόχρονα τιμές για τις ιδιότητες grid-row-start και grid-row-end με την παρακάτω σύνταξη:

```
grid-row: αριθμός_στήλης_έναρξης / αριθμός_στήλης_λήξης ;
```

Παράδειγμα

```
.item1{
grid-row: 1/ 5 ;}
```

Σχόλια

Το item1 θα ξεκινάει από το row-line 1 και θα καταλήγει στο row-line 5. Άρα θα καταλαμβάνει τέσσερις κανονικές θέσεις item σε κατακόρυφη διεύθυνση.

Σημείωση

Και σε αυτή την ιδιότητα μπορούμε στη δεύτερη τιμή να γράψουμε τη δήλωση span έτσι ώστε να ορίσουμε πόσες κατακόρυφες θέσεις θα καλύπτει το συγκεκριμένο item.

Έτσι λοιπόν για το προηγούμενο παράδειγμα θα γράψουμε:

```
.item1{
grid-row: 1/ span 4 ;}
```

Σημείωση

Με την ιδιότητα grid-row πετυχαίνουμε και ενώνουμε θέσεις items που βρίσκονται σε κατακόρυφη διεύθυνση δηλαδή η μία κάτω από την άλλη.

grid-area

με την ιδιότητα grid-area μπορούμε να γράψουμε ένα shorthand για τις ιδιότητες grid-column και grid-row.

```
grid-area: grid-row-start / grid-column-start/ grid-row-end / grid-column-end
```

Σύνταξη

```
grid-area: αριθμός_γραμμής_έναρξης/ αριθμός_στήλης_έναρξης /
αριθμός_γραμμής_λήξης / αριθμός_στήλης_λήξης ;
```

Παράδειγμα

```
.item1{
grid-area: 1/ 3/ 5/ 4 ;}
```

Σχόλια

Στο παραπάνω παράδειγμα το item1 θα τοποθετηθεί σύμφωνα με τις παρακάτω προδιαγραφές:

- Γραμμή έναρξης 1
- Γραμμή τερματισμού 5
- Στήλη έναρξης 3
- Στήλη τερματισμού 4

Σημείωση

Στην ιδιότητα grid-area είναι αποδεκτή η δήλωση span για να δηλώσουμε τον αριθμό των καλυπτόμενων θέσεων.

Με τις ιδιότητες grid-row, grid-column ή την ιδιότητα grid-area μπορούμε να δημιουργήσουμε ένα responsive grid layout προσαρμοσμένο στις ανάγκες μας κάνοντας τα παρακάτω βήματα:

1. Δημιουργούμε ένα grid-container και ορίζουμε τις αντίστοιχες ιδιότητες στην κλάση που θα εφαρμόσουμε σε αυτό.
2. Γράφουμε την κλάση για τα grid items.
3. Μέσα στο κεντρικό div του grid container γράφουμε τα child elements του grid container όπου το κάθε child element αντιστοιχεί σε ένα div.
4. Με τις ιδιότητες grid-row, grid-column ή την ιδιότητα grid-area δημιουργούμε μία responsive δομή προσαρμοσμένη στις ανάγκες μας.

2.2 Ονοματίζοντας Grid items

Στα προηγούμενα παραδείγματα γράφαμε τιμές για την ιδιότητα grid-area και δημιουργούσαμε την επιθυμητή δομή της σελίδας μας. Εκτός από το να γράφουμε τιμές στην ιδιότητα grid-area, μπορούμε εναλλακτικά να ορίσουμε ονόματα για το κάθε child item και έπειτα με την ιδιότητα grid-template-areas να ορίσουμε το επιθυμητό layout για τη σελίδα μας.

Παράδειγμα

```
.item1{
grid-area: header;
}
```

Σχόλια

Στο προηγούμενο παράδειγμα ορίσαμε το όνομα header για το child item που αντιστοιχεί στην κλάση .item1.

Στη συνέχεια εντός της κλάσης που εφαρμόζουμε στο grid-container θα γράψουμε την ιδιότητα grid-template-areas.

Με την ιδιότητα grid-template-areas ορίζουμε οριζόντιες σειρές από θέσεις. Έτσι θα γράφουμε το όνομα του κάθε παιδιού τόσες φορές όσες είναι και οι γειτονικές θέσεις που καταλαμβάνει στη συγκεκριμένη σειρά.

Για παράδειγμα αν επιθυμούμε να ορίσουμε 8 θέσεις για το στοιχείο που ονομάσαμε header θα γράψουμε:

```
grid-container {
  display: grid;
  grid-template-areas:
    'header header  header header header header header header '}
...
}
```

Αν για παράδειγμα στην ακριβώς από κάτω σειρά έχουμε τρία items menu, main και right:

```
.item2 { grid-area: menu;}
.item3 { grid-area: main; }
.item4 { grid-area: right; }
```

και επιθυμούμε να δημιουργήσουμε μια σειρά θέσεων κάτω από το header, όπου στις δύο πρώτες θέσεις θα είναι για το item menu, οι τέσσερις επόμενες για το item main και οι δύο τελευταίες για το item right θα γράψουμε:

```
.grid-container {
 display: grid;
 grid-template-areas:
 'header header  header header header header header header '
 'menu menu main main main main  right right'
...
}
```

Σχόλια

Με τις παραπάνω γραμμές έχουμε ορίσει οριζόντιες γραμμές από 8 θέσεις. Στην πρώτη σειρά το item header καταλαμβάνει και τις οκτώ θέσεις. Στην από κάτω γραμμή έχουμε ορίσει άλλη μία οριζόντια σειρά με τα items menu, main και right.

Σημείωση

Η σειρά που γράφουμε τα ονόματα στην grid-template-areas αντιστοιχεί και στη σειρά που καταλαμβάνουν τις θέσεις τα items.
Στην περίπτωση όπου υπάρχουν items τα οποία δεν έχουμε ονοματίσει στη θέση του ονόματος μπορούμε να γράψουμε τον χαρακτήρα τελεία.

Παράδειγμα

```
'menu menu main main main main  .  .'
```

Σχόλια

Αν δεν είχαμε δώσει το όνομα right σε κάποιο item τότε μπορούσαμε να ορίσουμε τις αντίστοιχες θέσεις αν αντί για το όνομα right γράφαμε στη θέση του μία τελεία.

Σημείωση

Με την ιδιότητα grid-area μπορούμε να ορίζουμε και τη θέση στο κάθε item. Έτσι λοιπόν δεν είναι απαραίτητη η τοποθέτηση των παιδιών σύμφωνα με τη σειρά που έχουν γραφεί στον κώδικα εντός του div του grid container.

2.3 Grid Template

Στη συγκεκριμένη παράγραφο θα αναλύσουμε βήμα προς βήμα τον τρόπο δημιουργίας ενός grid-template.

Αρχικά θα πρέπει να χωρίσουμε σε περιοχές το layout και να αποφασίσουμε σχετικά με το χώρο και τη θέση της κάθε περιοχής. Για να κάνουμε απλούστερη τη διαδικασία μπορούμε να κάνουμε ένα πρόχειρο σχέδιο σε ένα πίνακα.

Αν για παράδειγμα χρειαζόμαστε σειρές από οκτώ θέσεις, τότε αρχικά μπορούμε να εκτυπώσουμε σε ένα χαρτί τον παρακάτω πίνακα ο οποίος θα έχει οκτώ στήλες:

Στη συνέχεια σχεδιάζουμε την κάθε περιοχή επάνω στο χαρτί μας:

header	header	header	header	header	header	header	header
menu	menu	main	main	main	main	right	right
menu	menu	main	main	main	main	right	right
menu	menu	main	main	main	main	right	right
menu	menu	main	main	main	main	right	right
footer	footer	footer	footer	footer	footer	footer	footer

Οι πληροφορίες σχετικά με τη δομή της σελίδας, ας είναι ότι χρειαζόμαστε 6 γραμμές από θέσεις.
Στην πρώτη γραμμή το item header θα καταλαμβάνει και τις οκτώ θέσεις.
Στις γραμμές από δύο έως και έξι τις δύο πρώτες θέσεις θα καταλαμβάνει το item main, τις τέσσερις επόμενες το item main και τις δύο τελευταίες θα καταλαμβάνει το item right.
Οπότε και η ιδιότητα grid-template-areas θα μπορούσε να γραφεί όπως παρακάτω:

```
grid-template-areas:
    'header header  header header header header header header '
    'menu menu main main main main  right right'
    'menu menu main main main main  right right'
    'menu menu main main main main  right right'
    'menu menu main main main main  right right'
    'footer footer footer footer footer footer footer footer ';
```

Φάση 1- Δημιουργία layout

Αρχικά θα πρέπει με τη βοήθεια του grid system να υλοποιήσουμε το layout της σελίδας μας.

style

Ανάμεσα στις ετικέτες <style> και </style> ή σε κάποιο εξωτερικό αρχείο CSS θα γράψουμε κλάσεις με ονόματα item1, item2 κτλ. για να δώσουμε στο κάθε item ένα όνομα με τη βοήθεια της ιδιότητας grid-area. Έτσι λοιπόν θα δώσουμε τα ονόματα header, menu, main, right και footer στα items με αριθμούς από 1 έως 5 αντίστοιχα.

```
.item1 { grid-area: header; }
.item2 { grid-area: menu}
.item3 { grid-area: main; }
.item4 { grid-area: right; }
.item5 { grid-area: footer; }
```

Στη συνέχεια θα γράψουμε την κλάση που θα εφαρμόσουμε στο κεντρικό grid container το οποίο θα περιέχει και τα παιδιά. Αρχικά θα γράψουμε εντός της κλάσης

.grid-container τιμή grid για την ιδιότητα display έτσι ώστε να ενεργοποιηθεί το grid system.

```
.grid-container {
 display: grid;
```

Στη συνέχεια με την ιδιότητα grid-template-areas θα οριοθετήσουμε τις περιοχές που θα καταλαμβάνουν τα παιδιά του grid container. Εφόσον στο επάνω μέρος της δομής η περιοχή header θα καταλαμβάνει το 100% του πλάτους της δομής και κάθε οριζόντια σειρά από θέσεις περιλαμβάνει οκτώ θέσεις, τότε θα γράψουμε εντός απλώς εισαγωγικών οκτώ φορές τη δήλωση footer.

```
grid-template-areas:
'header header  header header header header header header '
```

Στην επόμενη γραμμή θα ορίσουμε μια οριζόντια σειρά από οκτώ θέσεις. Στις δύο πρώτες από αριστερά θέσεις θα τοποθετήσουμε το item menu, στις τέσσερις επόμενες θα τοποθετήσουμε το item main και για αυτό το λόγο θα γράψουμε τέσσερις φορές το όνομα main. Τέλος εντός των ίδιων εισαγωγικών θα γράψουμε και δύο φορές το όνομα right για να τοποθετήσουμε σε δύο συνεχόμενες θέσεις το item right.

```
'menu menu main main main main  right right'
```

Επειδή οι επόμενες τρεις γραμμές έχουν την ίδια δομή σε σχέση με το πλάτος που καταλαμβάνει το κάθε παιδί δε θα επαναλάβουμε τη δήλωση:

```
'menu menu main main main main  right right'
```

Τέλος εφόσον χρειαζόμαστε μία περιοχή για το item footer η οποία θα καταλαμβάνει ολόκληρο το πλάτος της σελίδας μας θα γράψουμε και οκτώ φορές το όνομα footer εντός απλών εισαγωγικών.

```
'footer footer footer footer footer footer footer footer ';
```

Τέλος στην κλάση του Grid container θα γράψουμε και τιμές για τα κενά μεταξύ των γραμμών και μεταξύ των στηλών του grid container καθώς και τιμές για χρώμα φόντου και padding.

Η τιμή 2px στην ιδιότητα grid-gap αφήνει κενά διαστήματα δύο pixels και μεταξύ των γραμμών και μεταξύ των στηλών του grid system.

```
grid-gap: 2px;
background-color: #2196F3;
```

Η τιμή padding διατηρεί αποστάσεις πέντε pixels γύρω από το περιεχόμενο του grid container.

```
  padding: 5px;
}
```

Στη συνέχεια θα γράψουμε style για όλα τα παιδιά του Grid container:

```
.grid-container > div {
  background-color:white;
  text-align: center;
  padding: 0 0;
  font-size: 20px;
}
```

Σχόλια

Αρχικά γράψαμε background-color λευκό για όλα τα παιδιά του grid container. Το λευκό background σε συνδυασμό με το μαύρο χρώμα της γραμματοσειράς, δίνει ως αποτέλεσμα περιοχές που μπορεί εύκολα κάποιος χρήστης να εστιάσει και είναι μια πολύ καλή συνταγή για ευανάγνωστο κείμενο. Επίσης ο χρήστης μπορεί εύκολα να ανιχνεύσει συγκεκριμένες πληροφορίες που υπάρχουν στη σελίδα.

Εκτός από το χρώμα του φόντου γράψαμε text-align:center για στοίχιση του κειμένου στο κέντρο, καθώς επίσης και padding ο pixels. Τέλος ορίσαμε ως μέγεθος γραμματοσειράς τα 20 pixels.

Το τελευταίο βήμα για να εμφανιστεί η grid δομή είναι να γράψουμε εντός του στοιχείου Body της σελίδας το div με την κλάση grid-container καθώς και τα child elements του grid container. Το κάθε παιδί θα αντιστοιχεί σε ένα div με την αντίστοιχη κλάση και θα γραφεί εντός του div που αντιστοιχεί στο κεντρικό grid container.

```
<div class="grid-container">
  <div class="item1">child 1 </div>
  <div class="item2">child 2</div>
  <div class="item3">child 3</div>
  <div class="item4"> child 4 </div>
  <div class="item5">child 5 </div>
</div>
```

Συνοψίζοντας

Μέχρι στιγμής έχουμε δημιουργήσει μία grid δομή ορίζοντας και οριοθετώντας περιοχές για το κάθε παιδί του κεντρικού grid container. Στην επόμενη φάση θα πρέπει σε κάθε παιδί να εισάγουμε αντίστοιχο περιεχόμενο. Καθώς εμείς θα προσθέτουμε περιεχόμενο το grid system θα προσαρμόζει τις διαστάσεις των παιδιών, έτσι ώστε το περιεχόμενο που αντιστοιχεί στο κάθε παιδί να έχει τον απαραίτητο χώρο έτσι ώστε να εμφανιστεί σωστά.

Δηλαδή, το grid system θα προσαρμόζει το grid container και τα παιδιά του έτσι ώστε να εμφανίζεται σωστά η σελίδα μας. Εκτός από αυτό, το grid container προσαρμόζει τις περιοχές των παιδιών καθώς το πλάτος της οθόνης του browser αυξομειώνεται.

Η αυξομείωση του πλάτους μπορεί να έχει την έννοια της εμφάνισης της σελίδας σε διαφορετικά υπολογιστικά συστήματα με διαφορετικές διαστάσεις οθονών(desktops, tablets, smartphones) ή ακόμη και στον ίδιο υπολογιστή όταν ο χρήστης αλλάζεις τις διαστάσεις του παραθύρου του browser.

Αν θέλαμε λοιπόν να χαρακτηρίσουμε με μία λέξη τα πλεονεκτήματα των λειτουργιών τόσο του grid system όσο και του flexbox αυτή η λέξη θα ήταν η λέξη **προσαρμοστικότητα**.

2.4 Προσθήκη περιεχομένου στο template και προσαρμογή σε mobile συσκευές

Στη δομή που δημιουργήσαμε θα προσθέσουμε περιεχόμενο. Πιο συγκεκριμένα η περιοχή footer θα είναι και η περιοχή – επικεφαλίδα της σελίδας, ενώ το υποσέλιδο θα αντιστοιχεί στην περιοχή footer.

Εντός της περιοχής main θα γράψουμε ένα vertical navigation που είχαμε δημιουργήσει σε προηγούμενη ενότητα, ενώ στην περιοχή main θα προσθέσουμε το κεντρικό περιεχόμενο της σελίδας. Τέλος στην περιοχή right θα είναι μία περιοχή πληροφοριών μιας ενότητας που θα εμφανίζεται ως στήλη στη δεξιά πλευρά της οθόνης ανάμεσα στο Header και το Footer.
Οι περιοχές που αναφέρθηκαν είναι στην ουσία οι περιοχές που καταλαμβάνουν τα παιδιά του grid container.

Item1 (child 1)

Style

```
.item1 { grid-area: header; }
```

Για το παιδί item1 το οποίο αντιστοιχεί στο Header της σελίδας μας θα γράψουμε στο στοιχείο Body και εντός του στοιχείου div του κεντρικού grid container ένα h1 στοιχείο έτσι ώστε να ορίσουμε και τον τίτλο της σελίδας.

```
<div class="item1"> <h1>Grid Layout</h1></div>
```

Item2 (child 2)

Style

```
.item2 { grid-area: menu;min-width: 250px;}
```

Εντός του child 2 θα ενσωματώσουμε ένα vertical navigation. Για να εξασφαλίσουμε ότι το navigation θα είναι ορατό ακόμη και σε μικρά πλάτη οθονών θα γράψουμε την ιδιότητα min-width δίνοντας τιμή 250 pixels. Έτσι λοιπόν το ελάχιστο πλάτος του navigation θα είναι ίσο με 250 pixels.
Επίσης εντός του στοιχείου head θα γράψουμε και το απαραίτητο style για τη δημιουργία του navigation.
Έτσι λοιπόν για τη μη διατεταγμένη λίστα θα ορίσουμε list-style-type: none για να αφαιρέσουμε από τη λίστα του σημειωτές και θα γράψουμε margin 0 έτσι ώστε να μην υπάρχει κενός χώρος περιμετρικά του navigation. Αν επιθυμούμε μπορούμε να αφήσουμε αποστάσεις περιμετρικά του navigation έτσι ώστε το navigation να μην εφάπτεται στην αριστερή πλευρά της σελίδας. Σε αυτή την περίπτωση θα γράψουμε κάποια τιμή για την ιδιότητα margin.

Αν γράψουμε την ιδιότητα margin για το navigation η καλύτερη λύση είναι να γράψουμε την ιδιότητα margin-left έτσι ώστε το navigation να ευθυγραμμίζεται με την αριστερή πλευρά του grid container.

```
ul {
 list-style-type: none;
 margin: 0;
 padding-left: 0;
 width:250 px;
 background-color:  #0099e6;
 border: 2px solid   #0000ff;
}
```

Στη συνέχεια θα γράψουμε ιδιότητες για τα στοιχεία τα οποία περιέχονται σε στοιχεία li της λίστας ul. Για να μπορεί ο χρήστης να κάνει κλικ με το mouse στα buttons που θα δημιουργηθούν θα γράψουμε:

```
display: block

li a {
   display: block;
   color: white;
   padding: 14px 15px;
   text-decoration: none;
}
```

Για τα στοιχεία li Θα γράψουμε text-align center για στοίχιση του κειμένου στο κέντρο και επίσης θα ορίσουμε border στην κάτω πλευρά του κάθε στοιχείου li.

```
li {
   text-align: center;
   border-bottom:  1px solid  #0000ff;
}
```

Για το τελευταίο ειδικά στοιχείο li της λίστας ul θα γράψουμε τιμή none στην ιδιότητα border έτσι ώστε το τελευταίο στοιχείο li να μην έχει στην κάτω πλευρά του περίγραμμα, εφόσον η κάτω πλευρά του συμπίπτει με την κάτω πλευρά ολόκληρης της λίστας ul στη οποία έχουμε ορίσει περίγραμμα στην κάτω πλευρά.

```
li:last-child {
    border-bottom: none;
}
```

Έχουμε ορίσει ξεχωριστές ιδιότητες για το στοιχείο li το οποίο είναι active.

```
li a.active {
    background-color:  #005580;
    color: white;
}
```

Τέλος, για το navigation έχουμε ορίσει για όλα τα στοιχεία li εκτός από το στοιχείο στο οποίο έχει εφαρμοστεί η κλάση .active χρώμα φόντου και χρώμα γραμματοσειράς.

```
li a:hover:not(.active) {
    background-color: #4d4d00;
    color: white;
}
```

Σημείωση

Στην επιλογή των χρωμάτων για το style του navigation θα πρέπει να επιλέγουμε χρώματα τα οποία είναι συγγενή μεταξύ τους.

Στοιχείο body

Για να ολοκληρωθεί η δημιουργία του navigation θα πρέπει εντός του item 2 να γράψουμε και το στοιχείου ul έτσι ώστε να δημιουργηθεί η μη διατεταγμένη λίστα με τις επιθυμητές συνδέσεις και τις clickable περιοχές που παίζουν τον ρόλο των buttons.

```
  <div class="item2">
 <ul>
  <li><a class="active" href="example1.html">Home</a></li>
  <li><a href="https://elearn.elke.uoa.gr/">elearning</a></li>
  <li><a href="https://www.google.gr/">Google</a></li>
 </ul>
    </div>
```

Item3 (child 3)

Το συγκεκριμένο item αντιστοιχεί στην περιοχή όπου εμφανίζεται το κεντρικό περιεχόμενο της σελίδας. Όπως για παράδειγμα ένα άρθρο, εικόνες κτλ.

Style

```
.item3 { grid-area: main; }
```

Επίσης εντός του style θα πρέπει να γράψουμε ιδιότητες για τις εικόνες που θα ενσωματώσουμε στη σελίδα μας.

```
img {
    height: auto;
     float:left;
}
```

Σχόλια

Γράψαμε `height:auto` έτσι ώστε να προσαρμόζεται το ύψος της εικόνας ανάλογα με το διαθέσιμο χώρο. Επίσης έχουμε ορίσει τα στοιχεία img να επιπλέουν αριστερά του κειμένου με την ιδιότητα:

```
float:left;
```

Έτσι λοιπόν θα έχουμε αναδίπλωση κειμένου με την εικόνα να εμφανίζεται στην αριστερή πλευρά του παιδιού, ενώ το κείμενο να περιβάλλει την εικόνα από τη δεξιά πλευρά.

Για τις ανάγκες του παραδείγματος έχουμε δημιουργήσει ένα subfolder με όνομα images και εντός του subfolder έχουμε αποθηκεύσει την εικόνα με όνομα αρχείου dolphin.jpg.

Εντός του παιδιού 3 θα γράψουμε τον παρακάτω κώδικα:

```
<div class="item3">
```

Αρχικά θα γράψουμε ένα στοιχείο img για να εμφανιστεί η εικόνα. Εφόσον το αρχείο της εικόνας βρίσκεται εντός του subfolder images και το όνομα του αρχείου είναι dolphin.jpg η διαδρομή για την πηγή του αρχείου θα είναι:

```
<img src="images/dolphins.jpg" width="230" height="173" alt="dolphins">
```

Επίσης στην ετικέτα img έχουμε γράψει και διαστάσεις 230 pixels για το πλάτος και 173 pixels για το ύψος της εικόνας. Τέλος έχουμε γράψει και εναλλακτικό κείμενο στην περίπτωση όπου ο browser δεν μπορέσει να εμφανίσει τη συγκεκριμένη εικόνα. Το κείμενο για την περιγραφή της εικόνας βρίσκεται ανάμεσα σε διπλά εισαγωγικά μετά τη δήλωση alt.

```
  <p>Lorem ipsum ...
  </p>
</div>
```

Item4 (child 4)

Το child 4 έχει το όνομα right, αντιστοιχεί στην πλευρική στήλη και μπορούμε να προσθέσουμε κείμενο συνδέσεις κτλ.

Style

```
.item4 { grid-area: right; }
```

Στο στοιχείο body προσθέτουμε:

```
<div class="item4">
    <p>Lorem ipsum ...
    </p></div>
```

Item5 (child 5)

Το child 5 αντιστοιχεί στο υποσέλιδο. Για την κλάση .item5 θα ορίσουμε το όνομα footer:

```
.item5 { grid-area: footer;}
```

Στο στοιχείο Body θα γράψουμε:

```
<div class="item5"> &#169; copyright... </div>
</div>
```

Προσαρμογή σε κινητά τηλέφωνα και γενικά σε συσκευές με μικρές οθόνες

Μέχρι στιγμής έχουμε δημιουργήσει μία responsive δομή μέσα στην οποία μπορούμε να προσθέτουμε περιεχόμενο και η δομή της σελίδας να προσαρμόζεται στο καινούργιο περιεχόμενο. Επίσης έχουμε καταφέρει καθώς μειώνεται το πλάτος της σελίδας του Browser το grid container και τα περιεχόμενά του να προσαρμόζονται στις νέες διαστάσεις του παραθύρου του browser.

Τι γίνεται όμως στην περίπτωση όπου η οθόνη της συσκευής ή το γενικά το πλάτος του παραθύρου του browser γίνει πολύ μικρό για παράδειγμα 500 pixels;

Σε αυτή την περίπτωση θα πρέπει να αλλάξουμε τη δομή έτσι ώστε ο χρήστης με απλό scroll down να έχει πρόσβαση στο περιεχόμενο της σελίδας.

Η λύση στην περίπτωση αυτή έρχεται από τα media queries. Αυτό που μπορούμε να κάνουμε είναι να ορίσουμε κλάσεις οι οποίες θα αλλάξουν τη δομή της σελίδας όταν η οθόνη γίνει μικρότερη από ένα συγκεκριμένο πλάτος.

Έστω για παράδειγμα ότι έχουμε ορίσει ιδιότητες για το grid container στην κλάση .grid-container ανάμεσα στις ετικέτες <style> και </style> και με την ιδιότητα grid-template-areas έχουμε ορίσει μία συγκεκριμένη δομή και κάποιες άλλες ιδιότητες.

```
.grid-container {
 display: grid;
 grid-template-areas:...
..
…
}
```

Αν στη συνέχεια γράψουμε:

```
@media only screen and (max-width: 500px) {
  .grid-container {
   display: grid;
  grid-template-areas:…
…
…
}
```

τότε, μόλις το πλάτος της οθόνης γίνει μικρότερο ή ίσο από 500 pixels, θα αλλάξει η δομή της σελίδας σύμφωνα με τις ιδιότητες της ίδιας κλάσης .grid-container που τώρα όμως περιέχεται εντός του media query. Επίσης θα εφαρμοστούν όλες οι ιδιότητες και όλες οι κλάσεις που θα ενσωματωθούν στο media query.

Εντός του media query θα γράψουμε μόνο ιδιότητες που επιθυμούμε να προσθέσουμε ή ιδιότητες με τιμές που πρέπει να αντικαταστήσουν τις προηγούμενες.

Όταν το πλάτος της οθόνης γίνει μικρότερο ή ίσο από 500 pixels στο προηγούμενο παράδειγμα τότε θα γίνει η εφαρμογή των κλάσεων και των ιδιοτήτων του media query. Αν εντός του media query υπάρχουν κλάσεις με το ίδιο όνομα ή υπάρχουν γενικά διαφορετικές τιμές ιδιοτήτων CSS τότε οι τιμές των κοινών ιδιοτήτων εντός του media query θα αντικαταστήσουν τις προηγούμενες.

Επίσης όταν το πλάτος της οθόνης γίνει μεγαλύτερο από 500 Pixels τότε όλες οι ιδιότητες του media query παύουν να ισχύουν.

Μία ασφαλής δομή για πολύ μικρές οθόνες είναι η κατακόρυφη στοίχιση των child elements του grid system και κάθε item να καταλαμβάνει ολόκληρο το πλάτος. Η δομή αυτή μοιάζει με ίδια κουτιά τοποθετημένα το ένα πάνω στο άλλο.

Στο template που δημιουργήσαμε για πολύ μικρά πλάτη μπορούμε να ορίσουμε μία δομή με τρεις θέσεις-πλάτη στην κάθε γραμμή, στοιχίζοντας έτσι τα παιδιά το ένα κάτω από το άλλο, ξεκινώντας με το header και καταλήγοντας στο footer όπως στον παρακάτω πίνακα:

header	header	header
menu	menu	menu
main	main	main
right	right	right
footer	footer	footer

Υλοποίηση

Θα γράψουμε το media query με τις ιδιότητες για τις κλάσεις του grid container και των παιδιών του. Αυτό που θα αλλάξουμε στην ουσία είναι η δομή με την ιδιότητα grid-template-areas. Έτσι λοιπόν θα γράψουμε τρεις φορές το όνομα του κάθε item ξεκινώντας από το header και καταλήγοντας στο footer.

```
@media only screen and (max-width: 500px) {
  .grid-container {
   display: grid;
  grid-template-areas:
    'header header  header'
    'menu menu menu'
    'main main main'
      'right right right '
    'footer footer footer  ';
  grid-gap: 2px;
  background-color: green;
  padding: 5px;
}
     .grid-container > div {
  background-color:white;
  text-align: center;
color:red;
  padding: 0 0;
  font-size: 20px;
}
}
```

Σχόλια

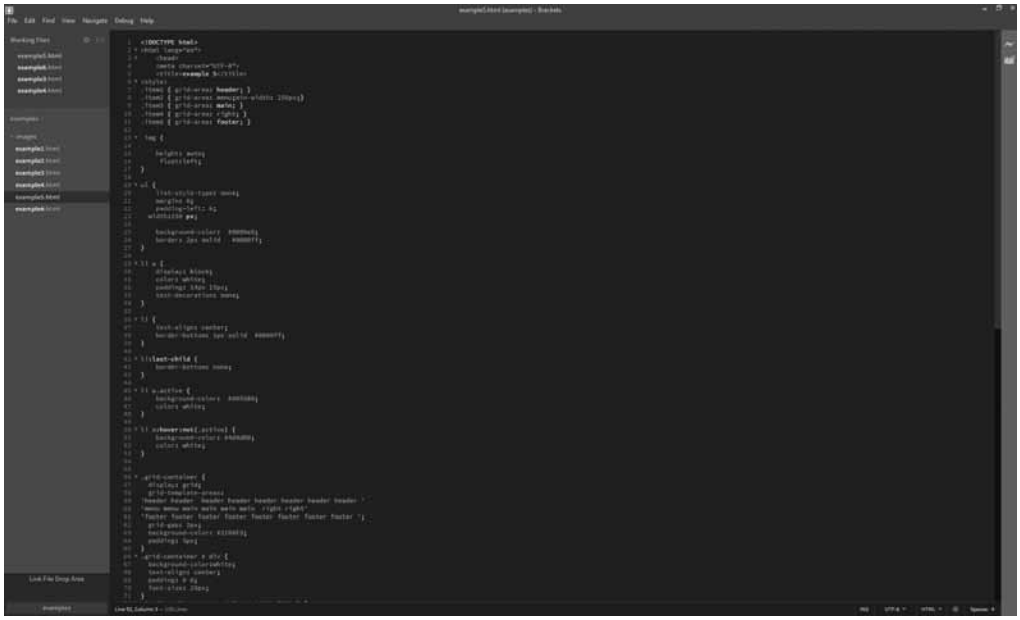
Στην κλάση με τα παιδιά του grid container γράψαμε color:red για τη γραμματοσειρά. Δηλαδή ορίσαμε χρώμα γραμματοσειράς κόκκινο όταν η οθόνη γίνει μικρότερη ή ίση με 500 Pixels. Αυτή η αλλαγή έγινε μόνο για να γίνει αισθητή η διαφορά όταν ενεργοποιείται το media query.

Παράδειγμα 5. Grid Template

Κώδικας(Mac): Grid Template

Εικόνα Browser(Mac): Grid Template

Κώδικας(Win): Grid Template



Εικόνα Browser(Win): Grid Template

ΥΠΟΕΝΟΤΗΤΑ 3. FLEXBOX TEMPLATE

Όπως στο grid system δημιουργήσαμε ένα template με header, footer, menu, main και right περιοχές, θα δημιουργήσουμε και ένα αντίστοιχο template στο flexbox.

3.1 Templates με το flexbox

Για το Flexbox Template θα χρησιμοποιήσουμε επιπλέον δύο ιδιότητες, οι οποίες αναλύονται παρακάτω:

box-sizing

Η ιδιότητα box-sizing αφορά στον τρόπο που υπολογίζονται οι διαστάσεις με τις ιδιότητες width και height.

Η σύνταξη της ιδιότητας box-sizing είναι η παρακάτω:

```
.box1{ box-sizing: τιμή};
```

Όπου τιμή μπορούμε να αντικαταστήσουμε μία από τις παρακάτω επιλογές:

box-sizing	
content-box	Είναι η προεπιλογή. Στον υπολογισμό των width και height, min, max περιλαμβάνεται μόνο το content. Τα border και padding, δεν περιλαμβάνονται.
border-box	Στον υπολογισμό των width και height, min, max περιλαμβάνονται content, padding και border.
initial	θέτει την ιδιότητα στην προεπιλεγμένη τιμή.
inherit	Κληρονομεί την ιδιότητα από το Parent element.

Σημείωση

Υπάρχει και το universal box-sizing όπου εφαρμόζει box-sizing σε όλα τα στοιχεία της σελίδας

```
* {
box-sizing:border-box;
}
```

Παράδειγμα

```
<style>
  * {
box-sizing:border-box;
}
.box1{
width: 300px;
padding: 10px;
border :5px solid green;}
</style>
```

Σχόλια

Εφόσον έχει ενεργοποιηθεί το universal box-sizing αν εφαρμόσουμε την κλάση .box1 σε ένα div τότε το div θα έχει συνολικό πλάτος 300 pixels το οποίο θα περιλαμβάνει και τα 10 pixels padding και τα 5 pixels border.

Viewport

Το viewport αντιπροσωπεύει την ορατή από το χρήστη περιοχή του browser. Στο template που ακολουθεί θα χρησιμοποιήσουμε το <meta> viewport για να ορίσουμε τον τρόπο που θα χειρίζεται ο Browser τις διαστάσεις των σελίδων και το scaling.

Επίσης θα γράψουμε:

width=device-width

έτσι ώστε το πλάτος της σελίδας να προσαρμόζεται στο πλάτος της συσκευής.

Τέλος, στο ίδιο meta tag θα γράψουμε

initial-scale=1

για να ορίσουμε το αρχικό zoom ίσο με 100% όταν φορτώνει η σελίδα στο Browser. Το meta tag που περιέχει τις παραπάνω πληροφορίες είναι το ακόλουθο:

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

Ιδιότητα flex με μία τιμή χωρίς μονάδες

Όταν χρησιμοποιούμε την ιδιότητα flex με μία τιμή και χωρίς μονάδες ορίζουμε τη δομή μιας γραμμής στοιχείων, κάτι αντίστοιχο όπως κάναμε και με την ιδιότητα grid-template-areas στο Grid System.

Έτσι και αντίστοιχα με την ιδιότητα flex αν ορίσουμε δύο κλάσεις για τρία αντικείμενα που θα τοποθετηθούν στην ίδια γραμμή μπορούμε να ορίσουμε την αναλογία πλάτους που θα καταλαμβάνουν.

Παράδειγμα

```
.side {
  flex: 1;
}
.main {
   flex: 2;
}
```

Αν υποθέσουμε ότι θα έχουμε στη γραμμή δύο πλευρικά side και ένα main τότε το συνολικό πλάτος θα περιλαμβάνει 2Χ1 για τα side και 1Χ2 για το main, άρα σύνολο τέσσερις θέσεις. Οπότε:

- το κάθε side θα καταλαμβάνει χώρο ¼=25%
- το main θα καταλαμβάνει χώρο 2/4=50%

3.2 Δημιουργία template με Flexbox

Ο τρόπος που θα αναπτύξουμε την υλοποίηση θα είναι ο ακόλουθος:

Αρχικά θα υπάρχει ένα div που θα παίξει το ρόλο του header και εφόσον θα αντιστοιχεί σε ένα div καταλαμβάνει ολόκληρο το πλάτος της σελίδας.

Στη συνέχεια θα δημιουργήσουμε ένα flex container στο οποίο θα ενσωματώσουμε μία περιοχή menu, μία περιοχή main και μία περιοχή right.

Τέλος εκτός του flexbox Θα ορίσουμε ένα div που θα αντιστοιχεί στην περιοχή footer όπως στον παρακάτω πίνακα:

header	header	header	header	header	header	header	header
menu	menu	main	main	main	main	right	right
menu	menu	main	main	main	main	right	right
menu	menu	main	main	main	main	right	right
menu	menu	main	main	main	main	right	right
footer	footer	footer	footer	footer	footer	footer	footer

Όλα τα παραπάνω θα ισχύουν όταν το πλάτος της οθόνης του Browser είναι μεγαλύτερο από 600 pixels. Όταν το πλάτος της οθόνης του Browser ή το πλάτος της οθόνης της συσκευής γίνει μικρότερο ή ίσο από 600 pixels τότε θα ορίσουμε τη flex διάταξη σε column και όλες οι περιοχές θα στοιχηθούν κατακόρυφα, η μία κάτω από την άλλη όπως κάναμε και με το Grid System.

Αρχικά θα γράψουμε τις πρώτες γραμμές για τη σελίδα προσθέτοντας το meta viewport που περιγράψαμε προηγουμένως.

```
<!DOCTYPE html>
<html lang="en">
    <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
```

Στη συνέχεια θα γράψουμε ανάμεσα στις ετικέτες style και style τις CSS ιδιότητες.

Αρχικά θα γράψουμε το universal box-sizing με την ιδιότητα border-box έτσι ώστε σε όλα τα στοιχεία της σελίδας στις ιδιότητες width, height, min και max να συνυπολογίζονται padding και border.

Στη συνέχεια θα ορίσουμε ένα font-family με τρεις γραμματοσειρές:

```
body {
    font-family: Arial, Helvetica, sans-serif;
}
```

Για το στοιχείο img και το navigation θα γράψουμε ακριβώς τις ίδιες ιδιότητες που γράψαμε και στο Grid Template.

Στη συνέχεια θα γράψουμε δύο κλάσεις με την ιδιότητα flex για να ορίσουμε την αναλογία πλάτους στη γραμμή που θα τοποθετηθούν οι περιοχές menu,main και right.

```
.side {
  flex: 1;
}
.main {
   flex: 2;
}
```

Στις περιοχές menu και right θα εφαρμόζουμε την κλάση .side ενώ στην περιοχή main θα εφαρμόσουμε την κλάση .main.

Έτσι λοιπόν θα χωρίσουμε τη γραμμή αυτή σε τέσσερα ίσα πλάτη. Οι δύο πλευρικές περιοχές θα καταλαμβάνουν από 25% ενώ η περιοχή main θα καταλαμβάνει πλάτος 50%.

Για την περιοχή header θα γράψουμε την παρακάτω κλάση:

```
.header {
   background-color: floralwhite;
   padding: 30px;
   text-align: center;
   font-size: 15px;
     border: 3px solid blue;
}
```

Για την περιοχή footer θα γράψουμε την παρακάτω κλάση:

```
   .footer {
   background-color: floralwhite;
   padding: 10px;
   text-align: center;
     border: 3px solid blue;
       color:red;
}
```

Η κλάση για το κεντρικό flex container είναι η παρακάτω και περιλαμβάνει την ιδιότητα display με την τιμή flex.

```
.flex {
    display: flex;
}
```

Για τα παιδιά του flex container θα γράψουμε:

```
.flex > div {
  background-color:floralwhite;
  text-align: center;
  padding: 0 0;
  font-size: 20px;
min-height:300px;
border: 3px solid blue;
}
```

Σχόλια

Έχουμε ορίσει τιμές για το background-color τη στοίχιση κειμένου στο κέντρο και padding 0 pixels.

Επίσης έχουμε ορίσει μέγεθος γραμματοσειράς 20 pixels και περίγραμμα 3 Pixels. Για να εξασφαλίσουμε ότι το ύψος του κάθε παιδιού δε θα γίνει μικρότερο από 300 Pixels έχουμε γράψει

```
min-height:300px;
```

Για την περίπτωση όπου η οθόνη αποκτήσει πλάτος μικρότερο από 600 pixels θα γράψουμε:

```
@media (max-width: 600px) {
    .flex {
      flex-direction: column;
    }
```

Σχόλια

Όταν η οθόνη γίνει μικρότερη ή ίση με 600 pixels γράψαμε την κλάση του κεντρικού flex container αλλάζοντας σε column το flex-direction.

Επίσης για να εξασφαλίσουμε ότι θα υπάρχει διαθέσιμο ύψος το οποίο θα είναι επαρκές για την εμφάνιση του περιεχομένου θα γράψουμε:

```
.flex > div {
min-height: 400px;
color:red;
}
```

Σχόλια

Έχουμε ορίσει ελάχιστο ύψος 400 pixels για κάθε child element όταν γίνει η κατακόρυφη στοίχιση των παιδιών. Επίσης αλλάξαμε και το χρώμα της γραμματοσειράς για να γίνει αισθητή η αλλαγή όταν μικρύνει το πλάτος της οθόνης και ενεργοποιηθεί το media query.

Αυτό που απομένει είναι να γράψουμε στο στοιχείο Body τα στοιχεία με τις κατάλληλες κλάσεις έτσι ώστε να έχουμε το επιθυμητό layout.

Για το header θα γράψουμε:

```
<div class="header">
<h2>Header</h2>
</div>
```

Για το κεντρικό flex container θα γράψουμε:

```
<div class="flex">
```

Για το navigation θα γράψουμε το στοιχείο ul εφαρμόζοντας την κλάση .side

```
<div class="side" >
      <ul>
  <li><a class="active" href="example1.html">Home</a></li>
  <li><a href="https://elearn.elke.uoa.gr/">elearning</a></li>
  <li><a href="https://www.google.gr/">Google</a></li>
</ul>
     </div>
```

Για την κεντρική περιοχή main θα γράψουμε:

```
<div class="main" >
      <img src="images/dolphins.jpg" width="230" height="173">
     <p>tester ipsum dolor sit amet, massa neque ac cras sit     </p>
    </div>
```

Για την περιοχή right θα γράψουμε πάλι την κλάση side:

```
<div class="side" >
     <p>tester ipsum dolor sit amet,     </p>
    </div>
```

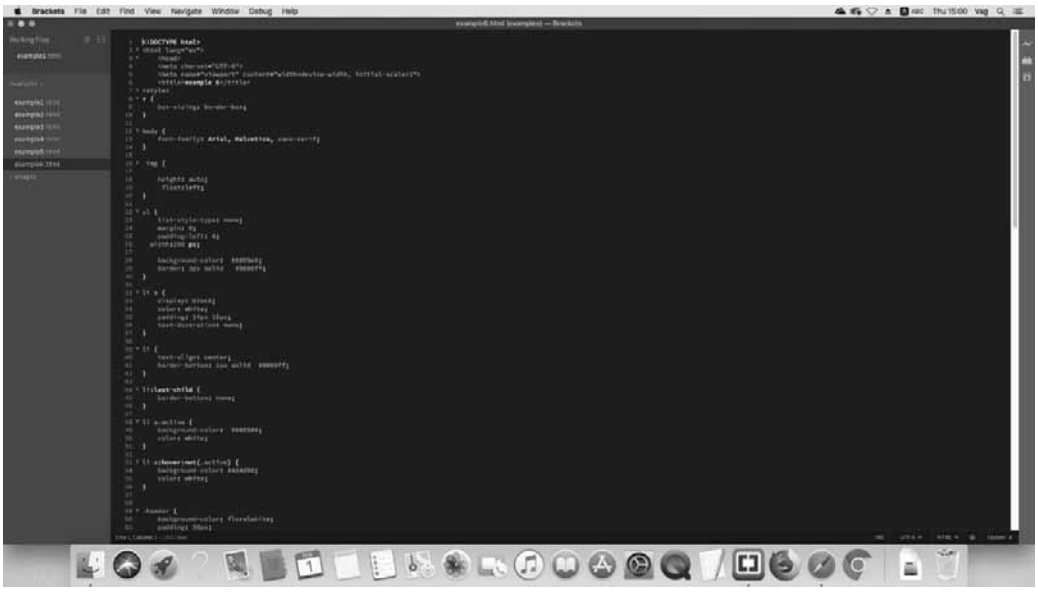
Και τέλος θα κλείσουμε το flex container με την ετικέτα λήξης ** **

Για το footer θα γράψουμε ένα div με κλάση .footer:

```
<div class="footer">
   <h3>&#169; copyright... </h3>
</div>
```

Παράδειγμα 6. Flex Template

Κώδικας(Mac): Flex Template



Εικόνα Browser(Mac): Flex Template

Κώδικας(Win): Flex Template

Εικόνα Browser(Win): Flex Template

3.3 Βίντεο

Η ενότητα κλείνει με ορισμένα βίντεο που παρουσιάζουν όσα αναλύθηκαν στην παρούσα ενότητα, τα οποία μπορείτε να δείτε στο αντίστοιχο σημείο του ηλεκτρονικού υλικού. Τα βίντεο 1-6 αφορούν mac και τα βίντεο 7-12 αφορούν pc.

Σύνοψη

Μετά την ολοκλήρωση της συγκεκριμένης διδακτικής ενότητας γνωρίζετε ότι:

- Μπορούμε να ορίσουμε media queries για να αλλάξουμε τη δομή μιας σελίδας όταν το πλάτος της οθόνης του Browser γίνει μικρότερο ή ίσο με κάποια τιμή.
- Μπορούμε με την ιδιότητα min-height να ορίσουμε το ελάχιστο ύψος που μπορεί να έχει κάποιο στοιχείο της σελίδας.
- Μπορούμε με την ιδιότητα animation-name: να ορίσουμε όνομα για το animation κάποιου στοιχείου.
- Μπορούμε με τα @keyframes να ορίσουμε style για συγκεκριμένο στοιχείο και σε συγκεκριμένη χρονική στιγμή που αντιστοιχεί σε μια φάση του animation.
- Με την ιδιότητα animation-iteration-count μπορούμε να επαναλάβουμε το animation όσες φορές επιθυμούμε.
- Με το Grid-column shorthand μπορούμε να ορίζουμε grid-column-start και grid-column-end.
- Με την ιδιότητα grid-area μπορούμε να δίνουμε ονόματα σε στοιχεία του Grid System.
- Με την ιδιότητα grid-template-areas μπορούμε να ορίζουμε την οριζόντια σειρά θέσεων που καταλαμβάνουν στοιχεία του Grid System.
- Με την ιδιότητα box-sizing ορίζουμε τον τρόπο υπολογισμού των διαστάσεων με τις ιδιότητες width και height.
- Γράφοντας display: flex σε μια κλάση εφαρμοζόμενη σε ένα div μετατρέπουμε το συγκεκριμένο div σε ένα flex container.
- Γράφοντας flex-direction: column ορίζουμε κατακόρυφη στοίχιση των flex items.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- **Castro, E & Hyslop, B**., (2012), *HTML5 & CSS3 Visual Quickstart Guide, 7th edition*, USA: Pearson Education
- **Lemay, L. & Colburn, R. & Kyrnin. J** (2016), Πλήρες εγχειρίδιο *HTML5, CSS και JavaSript, 7th edition*, GREECE: Εκδόσεις Μ. Γκιούρδας
- **Hudson, Ch. & Leadbetter, T**., (2012), *HTML5 Developer's book Guide, 1st edition*, USA: Pearson Education