



Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης

3.1. Πακέτο Εκπαιδευτικού Υλικού του έργου Code4SP

WP3:
Εκπαιδευτικό Υλικό του
έργου Code4SP

Εκπονητής:



Πληροφορίες αναφορικά με το Έργο

Ακρώνυμο του Έργου: Code4SP

Τίτλος Έργου: Κωδικοποίηση για Κοινωνική Ένταξη/ Ενσωμάτωση

Αριθμός Έργου: 621417-EPP-1-2020-1-PT-EPPKA3-IPI-SOC-IN

Ιστοσελίδα του έργου: www.code4sp.eu

Συνεργάτης εξουσιοδότησης: CEPROF

Εκδοχή/Εκδοση Εγγράφου: 1.0

Ημερομηνία Προετοιμασίας: 25/02/2022

Ιστορικό Εγγράφου

Ημερομηνία	Έκδοση	Συντάκτης	Περιγραφή
11/02/2022	1	CEPROF	Προσχέδιο

Ιστορικό Εγγράφου			
Ημερομηνία	Έκδοση	Συντάκτης	Περιγραφή

2 | Σελίδα

Πίνακας περιεχομένων

Πληροφορίες αναφορικά με το έργο	2
Θέμα : CSS	
Εισαγωγή στην CSS	8
1 - Σύντομη επισκόπηση της CSS	8
1.1 Τι είναι η CSS ?	8
1.2 Προαπαιτούμενες γνώσεις	8
1.3 Πλεονεκτήματα της CSS	9
1.4 Ποιος δημιουργεί και συντηρεί την CSS?	10
1.5 Εκδόσεις CSS (Versions)	10
2 – Σύνταξη κανόνων CSS	11
2.1 Επιλογείς CSS (Selectors)	12
2.1.1 Ο επιλογέας στοιχείου CSS (Element selector)	12
2.1.2 Ο επιλογέας ταυτότητας CSS (Identity selector)	12
2.1.3 Ο επιλογέας κατηγορίας CSS (Class selector)	13
2.1.4 Ο καθολικός επιλογέας CSS (Universal selector)	13
2.1.5 Ομαδοποίηση επιλογέων CSS (Grouping selector)	14
3 – Σχόλια CSS (Comments)	14
4 – Χρώματα Css	14
4.1 Δεκαεξαδικά Χρώματα	15
4.2	154.3
	164.4
	174.5
	174.6
	184.7
	194.8
205 – Φόντο CSS (Backgrounds)	19
5.1	205.2
	215.3
	225.4
	225.5
236 – Όρια CSS (Borders)	23

7 – Περιθώρια CSS (Margins)	24
8 – Επένδυση CSS (Padding)	25
9 – Ύψος και πλάτος CSS (Height and Width)	25
10 – Το μοντέλο κουτιού CSS (Box Model)	26
11 – Περίγραμμα CSS (Outline)	27
12 – Κείμενο CSS (Text)	27
13 – Γραμματοσειρές CSS (Fonts)	28
14 – Εικονίδια CSS (Icons)	29
15 – Σύνδεσμοι CSS (Links)	29
16 – Λίστες CSS (Λίστες)	29
17 – Προβολή CSS (Display)	30
18 – Πίνακες CSS (Tables)	31
19 – Μέγιστο Πλάτος CSS (Max-width)	31
20 – Τοποθέτηση στοιχείων CSS (Position)	32
21 - CSS Z-index	32
22 – Ιδιότητα Υπερχείλισης CSS (Overflow)	33
23 - CSS Float	34
24 - CSS Ενσωματωμένα κουτιά μπλοκ (inline block)	34
25 – Ευθυγράμμιση CSS (Align)	35
26 – Συνδυαστές CSS (Combinators)	35
27 – Ψευδό-κλάση CSS (Pseudo-class)	36
28 – Ψευδο-στοιχείο CSS (Pseudo-element)	36
29 – Αδιαφάνεια CSS (Opacity)	37
30 – Γραμμή Πλοήγησης CSS (Navigation Bar)	37
31 – Αναπτυσσόμενο μενού CSS (Dropdowns)	37
32 – Συλλογή Εικόνων CSS (Image Gallery)	38
33 – Εικόνες Sprites CSS	38
34 – Επιλογείς Attr CSS	39
35 – Μορφές CSS (Forms)	39
36 – Μετρητής CSS (Counters)	39
37 – Διάταξη Ιστοσελίδας CSS (Website Layout)	40
38 – Έντυπα CSS (Forms)	40
39 – Ειδικότητα CSS (Specificity)	41

40 - CSS !important	42
41 – Μαθηματικές Συναρτήσεις CSS (Math Function)	42
Προηγμένοι Κανόνες Σύνταξης CSS	43
1. Στρογγυλεμένες γωνίες 452.	Εικόνες περιγράμματος 463.
Φόντο CSS (Backgrounds)	44
4. Χρώματα CSS (Colors)	44
5. Λέξεις-Κλειδιά Χρώματος 486.	Κλίση 497.
Σκίαση 498.	Εφέ κειμένου 509.
Γραμματοσειρές Ιστού 5110.	Μετασχηματισμοί CSS 2D (Transforms)
49	
11. Μετασχηματισμοί CSS 3D (Transforms)	50
12. Μεταβάσεις CSS (Transitions)	50
13. Κινούμενα Σχέδια CSS (Animations)	51
14. Συμβουλές Εργαλείων CSS (Tooltips)	52
15. Εικόνες Στυλ CSS (Style images)	53
16. Εικόνες με Εφέ Αντανάκλασης CSS (Image Reflection)	53
17. CSS Object-fit	54
18. Ιδιότητα Θέσης Αντικειμένου CSS (Object-position)	55
19. Ιδιότητα Μάσκας - Στρώσης CSS (Masking)	56
20. Κουμπιά CSS (Buttons)	57
21. Σελιδοποίηση CSS (Pagination)	57
22. Πολλαπλές Στήλες CSS (Multiple Columns)	58
23. Διεπαφή Χρήση (CSS User Interface)	58
24. Μεταβλητές CSS (Variables)	59
25. Ιδιότητα Ορισμού Μεγέθους CSS (Box sizing)	60
26. Ερωτήματα Μέσων CSS (Media Queries)	61
27. Παραδείγματα MQ CSS	61
28. Ιδιότητα Ευέλικτης Ενότητας Διάταξης CSS (Flexbox)	62
Σχεδιασμός Ιστοσελίδων Άμεσης Ανταπόκρισης (RWD)	62
1. Εισαγωγή στο RWD:	62
2. Θύρα προβολής RWD (Viewport)	63
Ρύθμιση Θύρας Προβολής	64
3. Προβολή Πλέγματος RWD (Grid-View)	65
Κατασκευή ενός αποκριτού πλέγματος προβολής (responsive Grid-View)	66

4. Ερωτήματα Πολυμέσων RWD (Media Queries)	67
Προσθήκη ενός σημείου διακοπής (breakpoint)	68
5. Εικόνες RWD	70
Η χρήση της ιδιότητας μέγιστου πλάτους (the max-width property):	70
6. Βίντεο RWD	71
Η χρήση της ιδιότητας μέγιστου πλάτους (max-width property):	72
7. Πλαίσια RWD (Frameworks)	72
Πλέγμα CSS (Grid)	73
2. Κοντέινερ Πλέγματος (Grid Container)	73
3. Αντικείμενο Πλέγματος (Grid Item)	74
SASS	75
1. Εισαγωγή στο Sass	75
Τι είναι το Sass?	75
Γιατί να χρησιμοποιήσετε το Sass;	75
Ένα απλό παράδειγμα γιατί το SASS είναι χρήσιμο	75
Πώς λειτουργεί το SASS;	77
Τύποι Αρχείων Sass (File Type)	77
Σχόλια Sass (Comments)	77
Μεταβλητές Sass (Variables)	77
Πεδίο Εφαρμογής Μεταβλητών Sass (Variable Scope)	79
Η χρήση του !global	80
Ένθετοι Κανόνες Sass (Nested Rules)	81
Ένθετες Ιδιότητες Sass (Nested Properties)	82
Sass @import and Partials	83
Εισαγωγή αρχείων Sass (Importing Files)	83
Sass Partials	85
Sass @mixin and @include	86
Sass Mixins	86
Ο ορισμός ενός Mixin	86
Η χρήση ενός Mixin	87
Η μεταφορά μεταβλητών σε ένα Mixin	88
Οδηγία Sass @extend	89



Με συγχρηματοδότηση από το πρόγραμμα «Erasmus+» της Ευρωπαϊκής Ένωσης



Κεφάλαιο 1: CSS

1- Σύντομη επισκόπηση στην CSS

1.1 Τι είναι η CSS;

Η Cascading Style Sheets (Διαδοχικά Φύλλα Ύφους ή Επικαλυπτόμενα Φύλλα Στυλ), γνωστή με το ακρώνυμο CSS, είναι μια απλή γλώσσα σχεδιασμού ηλεκτρονικών υπολογιστών που βοηθά στην απλοποίηση της διαδικασίας που καθιστά τις ιστοσελίδες ευπαρουσίαστες.

Η CSS διαχειρίζεται την εμφάνιση και την γενική εντύπωση που έχει μια ιστοσελίδα. Με τη χρήση της CSS, μπορείτε να ελέγξετε το χρώμα του κειμένου, το στυλ των γραμματοσειρών, το διάστημα μεταξύ των παραγράφων, το μέγεθος και τη διάταξη που έχουν οι στήλες, τις εικόνες φόντου ή τα χρώματα γενικά, καθώς και διάφορα άλλα εφέ.

Η CSS είναι συνήθως εύληπτη και κατανοητή στους χρήστες, ενώ τους δίνει παράλληλα τη δυνατότητα να έχουν ουσιαστικό έλεγχο επί της εμφάνισης ενός εγγράφου HTML. Συνήθως, η CSS συνδυάζεται με τις γλώσσες σήμανσης HTML ή XHTML.

1.2 Προαπαιτούμενες γνώσεις

Οι εκπαιδευόμενοι θα πρέπει:

- Να έχουν βασικές γνώσεις στην επεξεργασία κειμένου με τη βοήθεια οποιουδήποτε επεξεργαστή κειμένου.
- Να γνωρίζουν πως να δημιουργούν καταλόγους και αρχεία.
- Να γνωρίζουν πώς να περιηγούνται σε διαφορετικούς τύπους καταλόγων/ευρετηρίων.

- Να γνωρίζουν πώς να περιηγούνται στο διαδίκτυο χρησιμοποιώντας γνωστά προγράμματα περιήγησης όπως το Internet Explorer ή το Firefox.
- Να είναι εξοικειωμένοι με τον σχεδιασμό απλών ιστοσελίδων μέσω της χρήσης HTML ή XHTML.

1.3 Πλεονεκτήματα της CSS

- **Η CSS εξοικονομεί χρόνο** - Μπορείτε να γράψετε σε γλώσσα CSS μία φορά και στη συνέχεια να επαναχρησιμοποιήσετε το ίδιο φύλλο σε πολλές σελίδες HTML. Μπορείτε να ορίσετε ένα στυλ για κάθε στοιχείο HTML και να το εφαρμόσετε σε όσες ιστοσελίδες θέλετε.
- **Οι σελίδες φορτώνονται γρηγορότερα** - Εάν χρησιμοποιείτε CSS, δεν χρειάζεται να γράφετε κάθε φορά τις ιδιότητες ετικέτας HTML. Απλά ορίστε έναν κανόνα CSS σε μια ετικέτα και εφαρμόστε τον σε όλες τις εμφανίσεις αυτής της ετικέτας. Επομένως, όσο λιγότερη είναι η κωδικοποίηση τόσο γρηγορότερη είναι η λήψη.
- **Εύκολη συντήρηση** - Για να κάνετε μια αλλαγή για ολόκληρη τη σελίδα, απλά αλλάζτε το στυλ και όλα τα στοιχεία σε όλες τις ιστοσελίδες και θα ενημερωθούν αυτόματα.
- **Ανώτερα στυλ από την HTML** - η CSS παρέχει ένα ευρύτερο φάσμα ιδιοτήτων από την HTML, που σας επιτρέπει να βελτιστοποιήσετε την εμφάνιση και μιας ιστοσελίδας HTML.
- **Συμβατότητα πολλαπλών συσκευών** - Τα φύλλα ύφους επιτρέπουν τη βελτιστοποίηση του περιεχομένου για περισσότερους από έναν τύπους συσκευών. Μέσω της χρήσης του ίδιου εγγράφου HTML, δίνεται η δυνατότητα

παρουσίασης διαφορετικών εκδοχών ενός ιστότοπου για φορητές συσκευές όπως είναι οι PDA (Προσωπικοί Ψηφιακοί Οδηγοί) και τα κινητά τηλέφωνα ή για σκοπούς εκτύπωσης.

- **Παγκόσμια πρότυπα ιστού** – Τώρα τα χαρακτηριστικά της HTML καταργούνται και συνιστάται η χρήση της CSS. Έτσι λοιπόν, θα ήταν καλύτερα να αρχίσετε να χρησιμοποιείτε την CSS σε όλες τις σελίδες HTML, για να τις καταστήσετε συμβατές με μελλοντικά προγράμματα περιήγησης.

1.4 Ποιος Δημιουργεί και Διατηρεί την CSS;

Η CSS δημιουργείται και συντηρείται μέσω μιας ομάδας ατόμων εντός του W3C, που ονομάζεται Ομάδα Εργασίας της CSS. Η ομάδα εργασίας της CSS δημιουργεί έγγραφα που ονομάζονται τεχνικά χαρακτηριστικά ή τεχνικές προδιαγραφές. Όταν ένα τεχνικό χαρακτηριστικό/τεχνική προδιαγραφή έχει συζητηθεί και επικυρωθεί επίσημα από τα μέλη της W3C, γίνεται σύσταση.

Τα επικυρωμένα τεχνικά χαρακτηριστικά ονομάζονται συστάσεις επειδή το W3C δεν έχει κανέναν ουσιαστικό έλεγχο επί της εφαρμογής της γλώσσας. Ανεξάρτητες εταιρείες και οργανισμοί είναι υπεύθυνοι για τη δημιουργία του λογισμικού.

ΣΗΜΕΙΩΣΗ: Η Κοινοπραξία Παγκόσμιου Ιστού (World Wide Web Consortium ή W3C) είναι μια ομάδα που κάνει συστάσεις σχετικά με το πώς λειτουργεί το Διαδίκτυο και πώς θα πρέπει να εξελιχθεί.

1.5 Εκδόσεις της CSS

Η CSS κυκλοφόρησε αρχικά το 1996 και παρείχε τις δυνατότητες για τη διαμόρφωση χαρακτηριστικών κειμένου όπως η στοίχιση, συμπεριλαμβανομένων των επιλογών γραμματοσειράς και το χρώματος έμφασης του κειμένου, το φόντο και άλλα χαρακτηριστικά. Το CSS2 κυκλοφόρησε το 1998 με επιπρόσθετες επιλογές στυλ για άλλους τύπους μέσων δικτύωσης, έτσι ώστε να μπορεί να χρησιμοποιηθεί για το σχεδιασμό της διάταξης μιας ιστοσελίδας. Το CSS3 κυκλοφόρησε το 1999 και σ' αυτό

προστέθηκαν ιδιότητες τύπου παρουσίασης που επιτρέπουν τη δημιουργία μιας παρουσίασης από έγγραφα.

2 – Σύνταξη κανόνων CSS

Μια CSS αποτελείται από ένα σύνολο κανόνων αναφορικά με το στυλ/ύφος που ερμηνεύονται από το πρόγραμμα περιήγησης και στη συνέχεια εφαρμόζονται στα αντίστοιχα στοιχεία μιας ιστοσελίδας. Ένας κανόνας αποτελείται από τρία μέρη:

- **Επιλογέας:** Ο επιλογέας είναι μια ετικέτα HTML στην οποία θα εφαρμοστεί ένα στυλ. Αυτό θα μπορούσε να είναι οποιαδήποτε `<h1>` ετικέτα `<table>`.?
- **Ιδιότητα:** Η ιδιότητα είναι ένας τύπος χαρακτηριστικού ετικέτας της HTML. Με απλά λόγια, όλα τα χαρακτηριστικά της HTML μετατρέπονται σε ιδιότητες CSS, δηλαδή χρώμα, περίγραμμα κ.λπ.
- **Τιμή:** Οι τιμές εκχωρούνται στις ιδιότητες. Για παράδειγμα, η ιδιότητα χρώματος μπορεί να έχει την τιμή κόκκινου χρώματος είτε `# F1F1F1` κλπ.

2.1 Επιλογείς CSS

Οι επιλογείς CSS χρησιμοποιούνται για την «εύρεση» (ή επιλογή) των στοιχείων HTML που θέλετε να διαμορφώσετε.

Μπορούμε να διαιρέσουμε τους επιλογείς CSS σε πέντε κατηγορίες:

- **Απλοί επιλογείς** (επιλογή στοιχείων με βάση το όνομα, το αναγνωριστικό, την κλάση)
- **Επιλογείς συνδυασμού** (επιλογή στοιχείων με βάση μια συγκεκριμένη σχέση μεταξύ τους)
- **Επιλογείς ψευδοκατηγορίας** (επιλογή στοιχείων με βάση μια συγκεκριμένη κατάσταση)
- **Επιλογείς ψευδοστοιχείων** (επιλογή και στυλιστική ανάπτυξη ενός μέρους ενός στοιχείου)

- **Επιλογείς ιδιοτήτων** (επιλογή στοιχείων με βάση μια ιδιότητα ή τιμή ιδιότητας)

2.1.1 Ο επιλογέας στοιχείου CSS

Ο επιλογέας στοιχείων επιλέγει ουσιαστικά στοιχεία HTML με βάση το όνομα του στοιχείου.

```
p {  
  text-align: center;  
  color: red;  
}
```

2.1.2 Ο επιλογέας ταυτότητας CSS

Ο επιλογέας ταυτότητας χρησιμοποιεί ένα χαρακτηριστικό αναγνώρισης ενός στοιχείου HTML για να επιλέξει ένα συγκεκριμένο στοιχείο.

Το αναγνωριστικό ενός στοιχείου είναι μοναδικό σε μια σελίδα, οπότε ο επιλογέας ταυτότητας χρησιμοποιείται για την επιλογή ενός μοναδικού στοιχείου!

Για να επιλέξετε ένα στοιχείο με ένα συγκεκριμένο αναγνωριστικό, γράψτε έναν χαρακτήρα hash (#), ακολουθούμενο από το αναγνωριστικό του στοιχείου.

```
#para1 {  
  text-align: center;  
  color: red;  
}
```

2.1.3 Ο επιλογέας κλάσης CSS

Ο επιλογέας κατηγορίας επιλέγει στοιχεία HTML με μια συγκεκριμένη ιδιότητα κατηγορίας.

Για να επιλέξετε στοιχεία με μια συγκεκριμένη κατηγορία, γράψτε έναν περιοδικό χαρακτήρα (.), ακολουθούμενο από το όνομα της κατηγορίας.

```
h1, h2, p {  
  text-align: center;  
  color: red;  
}
```

```
.center {  
  text-align: center;  
  color: red;  
}
```

2.1.4 Ο καθολικός επιλογέας CSS

Ο καθολικός επιλογέας (*) επιλέγει όλα τα στοιχεία HTML στη σελίδα.

```
* {  
  text-align: center;  
  color: blue;  
}
```

2.1.5 Ομαδοποίηση Επιλογέων

Μπορείτε να εφαρμόσετε στυλ σε πολλούς επιλογείς, αν θέλετε. Απλά διαχωρίστε τους επιλογείς με ένα κόμμα, όπως δίνεται στο ακόλουθο παράδειγμα:

```
h1, h2, p {  
  text-align: center;  
  color: red;  
}
```

3 - Σχόλια CSS

- Τα σχόλια χρησιμοποιούνται για να επεξηγούν τον κώδικα και μπορεί να σας βοηθήσουν να επεξεργαστείτε τον πηγαίο κώδικα σε μεταγενέστερη ημερομηνία.
- Τα σχόλια αγνοούνται από τα προγράμματα περιήγησης.

Ένα σχόλιο CSS ξεκινά με /* και τελειώνει με */

```
/* This is a single-line comment */  
p {  
  color: red;  
}
```

4 - Χρώματα CSS

Τα χρώματα στην CSS μπορούν να καθοριστούν με τους ακόλουθους τύπους σημειογραφίας:

- Δεκαεξαδικά χρώματα
- Δεκαεξαδικά χρώματα με διαφάνεια
- Χρώματα RGB
- Χρώματα RGBA
- Χρώματα HSL
- Χρώματα HSLA
- Καθιερωμένα ονόματα χρωμάτων φυλλομετρητή/Αποκλειστικά ανά φυλλομετρητή ονόματα χρωμάτων

- # - # - # - # - # [gnome-commander.master.el.po](#) (el)

4.1 Δεκαεξαδικά χρώματα

Ένα δεκαεξαδικό χρώμα ορίζεται με: #RRGGBB, όπου οι δεκαεξαδικοί ακέραιοι αριθμοί RR (κόκκινο), GG (πράσινο) και BB (μπλε) καθορίζουν τα συστατικά στοιχεία του χρώματος. Όλες οι τιμές πρέπει να είναι μεταξύ 00 και FF.

Για παράδειγμα, η τιμή # 0000ff αποδίδεται ως μπλε, επειδή το μπλε συστατικό στοιχείο έχει οριστεί στην υψηλότερη τιμή του (ff) και τα άλλα έχουν οριστεί στο 00.

Example

Define different HEX colors:

```
#p1 {background-color: #ff0000;} /* red */
#p2 {background-color: #00ff00;} /* green */
#p3 {background-color: #0000ff;} /* blue */
```

4.2 Δεκαεξαδικά χρώματα με διαφάνεια

Ένα δεκαεξαδικό χρώμα ορίζεται με: #RRGGBB. Για να προσθέσετε διαφάνεια, προσθέστε δύο επιπλέον ψηφία μεταξύ 00 και FF.

Example

Define different HEX colors with transparency:

```
#p1a {background-color: #ff000080;} /* red transparency */  
#p2a {background-color: #00ff0080;} /* green transparency */  
#p3a {background-color: #0000ff80;} /* blue transparency */
```

4.3 Χρώματα RGB

Μια τιμή χρώματος RGB καθορίζεται με τη [συνάρτηση rgb\(\)](#), η οποία έχει την ακόλουθη σύνταξη:

rgb(κόκκινο, πράσινο, μπλε)

Κάθε παράμετρος (κόκκινη, πράσινη και μπλε) ορίζει την ένταση του χρώματος και μπορεί να είναι ένας ακέραιος αριθμός μεταξύ 0 και 255 ή μια ποσοστιαία τιμή (από 0% έως 100%).

Για παράδειγμα, η τιμή `rgb(0,0,255)` αποδίδεται ως μπλε, επειδή η μπλε παράμετρος ρυθμίζεται στην υψηλότερη τιμή της (255) και οι άλλες ρυθμίζονται στο 0.

Επίσης, οι ακόλουθες τιμές ορίζουν ισόποσα χρώματα: `rgb(0,0,255)` και `rgb(0%,0%,100%)`.

Example

Define different RGB colors:

```
#p1 {background-color: rgb(255, 0, 0);} /* red */
#p2 {background-color: rgb(0, 255, 0);} /* green */
#p3 {background-color: rgb(0, 0, 255);} /* blue */
```

4.4 Χρώματα RGBA

Οι τιμές χρώματος RGBA είναι μια επέκταση των τιμών χρώματος RGB με ένα κανάλι άλφα - το οποίο καθορίζει την αδιαφάνεια του αντικειμένου.

Μια τιμή χρώματος RGB καθορίζεται με τη [συνάρτηση rgb\(\)](#), η οποία έχει την ακόλουθη σύνταξη:

rgba(κόκκινο, πράσινο, μπλε, άλφα)

Η παράμετρος άλφα είναι ένας αριθμός μεταξύ 0,0 (πλήρως διαφανής) και 1,0 (πλήρως αδιαφανής).

Example

Define different RGB colors with opacity:

```
#p1 {background-color: rgba(255, 0, 0, 0.3);} /* red with opacity */
#p2 {background-color: rgba(0, 255, 0, 0.3);} /* green with opacity */
#p3 {background-color: rgba(0, 0, 255, 0.3);} /* blue with opacity */
```

4.5 Χρώματα HSL

Το HSL αντιπροσωπεύει την απόχρωση, τον κορεσμό και την φωτεινότητα - και αντιπροσωπεύει μια κυλινδρική συντεταγμένη αναπαράσταση χρωμάτων.

Μια τιμή χρώματος RGB καθορίζεται με τη [συνάρτηση rgb\(\)](#), η οποία έχει την ακόλουθη σύνταξη:

hsl(απόχρωση, κορεσμός, φωτεινότητα)

Η απόχρωση είναι ένας βαθμός στον τροχό των χρωμάτων (από 0 έως 360) - το 0 (ή 360) είναι κόκκινο, το 120 είναι πράσινο, το 240 είναι μπλε. Ο κορεσμός είναι μια ποσοστιαία τιμή. Το 0% σημαίνει μια απόχρωση του γκρι και 100% είναι το πλήρες χρώμα. Η φωτεινότητα είναι επίσης μια ποσοστιαία τιμή. Το 0% είναι μαύρο, το 100% είναι λευκό.

Example

Define different HSL colors:

```
#p1 {background-color: hsl(120, 100%, 50%);} /* green */
#p2 {background-color: hsl(120, 100%, 75%);} /* light green */
#p3 {background-color: hsl(120, 100%, 25%);} /* dark green */
#p4 {background-color: hsl(120, 60%, 70%);} /* pastel green */
```

4.6 Χρώματα HSLA

Οι τιμές χρώματος RGBA είναι μια επέκταση των τιμών χρώματος RGB με ένα κανάλι άλφα - το οποίο καθορίζει την αδιαφάνεια του αντικειμένου.

Μια τιμή χρώματος RGB καθορίζεται με τη [συνάρτηση rgb\(\)](#), η οποία έχει την ακόλουθη σύνταξη:

hsla(απόχρωση, κορεσμός, φωτεινότητα, άλφα)

Η παράμετρος άλφα είναι ένας αριθμός μεταξύ 0,0 (εντελώς διαφανής) και 1,0 (εντελώς αδιαφανής).

Example

Define different HSL colors with opacity:

```
#p1 {background-color: hsla(120, 100%, 50%, 0.3);} /* green with opacity */  
#p2 {background-color: hsla(120, 100%, 75%, 0.3);} /* light green with opacity */  
#p3 {background-color: hsla(120, 100%, 25%, 0.3);} /* dark green with opacity */  
#p4 {background-color: hsla(120, 60%, 70%, 0.3);} /* pastel green with opacity */
```

4.7 Καθιερωμένα ονόματα χρωμάτων

φυλλομετρητή/Αποκλειστικά ονόματα χρωμάτων ανά φυλλομετρητή

Υπάρχουν 140 καθιερωμένα ονόματα χρωμάτων στις προδιαγραφές χρώματος HTML και CSS.

Για παράδειγμα : μπλε, κόκκινο, κοραλί, καφέ, κλπ:

Example

Define different color names:

```
#p1 {background-color: blue;}  
#p2 {background-color: red;}  
#p3 {background-color: coral;}  
#p4 {background-color: brown;}
```

4.8 Το currentcolor Keyword

Το **currentcolor** Keyword αναφέρεται στην τιμή της ιδιότητας χρώματος ενός στοιχείου.

Example

The border color of the following <div> element will be blue, because the text color of the <div> element is blue:

```
#myDIV {  
  color: blue; /* Blue text color */  
  border: 10px solid currentcolor; /* Blue border color */  
}
```

5 - Φόντο CSS (background)

- Οι ιδιότητες φόντου CSS χρησιμοποιούνται για την προσθήκη εφέ φόντου σε ένα στοιχείο.
- Ιδιότητες φόντου

- **background-color**
- **background-image**
- **background-repeat**
- **background-attachment**
- **background-position**
- **background** (shorthand property)

5.1 Χρώμα φόντου CSS (background-color)

Η ιδιότητα **χρώμα φόντου** καθορίζει το χρώμα φόντου ενός στοιχείου.

Example

The background color of a page is set like this:

```
body {  
  background-color: lightblue;  
}
```

5.2 Εικόνα φόντου CSS

Η ιδιότητα μιας **εικόνας φόντου** καθορίζει μια εικόνα για χρήση ως φόντο ενός στοιχείου.

Από προεπιλογή, η εικόνα επαναλαμβάνεται έτσι ώστε να καλύπτει ολόκληρο το στοιχείο.

Example

Set the background image for a page:

```
body {  
  background-image: url("paper.gif");  
}
```

5.3 Επανάληψη φόντου CSS (background-repeat)

Από προεπιλογή, η ιδιότητα **φόντου-εικόνας** επαναλαμβάνει μια εικόνα τόσο οριζόντια όσο και κάθετα.

Ορισμένες εικόνες θα πρέπει να επαναλαμβάνονται μόνο οριζόντια ή κάθετα, διαφορετικά θα φαίνονται παράξενες.

5.4 Εισαγωγή φόντου CSS (background attachment)

Η ιδιότητα **εισαγωγής φόντου** καθορίζει εάν μια εικόνα φόντου πρέπει να μετακινηθεί ή να διορθωθεί.

Example

Specify that the background image should be fixed:

```
body {  
  background-image: url("img_tree.png");  
  background-repeat: no-repeat;  
  background-position: right top;  
  background-attachment: fixed;  
}
```

Example

Specify that the background image should scroll with the rest of the page:

```
body {  
  background-image: url("img_tree.png");  
  background-repeat: no-repeat;  
  background-position: right top;  
  background-attachment: scroll;  
}
```

5.5 Φόντο CSS - Σύντομη ιδιότητα/Στενογραφία (shorthand)

Για να συντομεύσετε τον κώδικα, είναι επίσης δυνατό να καθορίσετε όλες τις ιδιότητες φόντου σε μια μόνο ιδιότητα. Αυτό ονομάζεται στενογραφία.

Αντί να γράψετε:

```
body {  
    background-color: #ffffff;  
    background-image: url("img_tree.png");  
    background-repeat: no-repeat;  
    background-position: right top;  
}
```

Μπορείτε να χρησιμοποιήσετε την ιδιότητα **φόντου** με στενογραφία:

Example

Use the shorthand property to set the background properties in one declaration:

```
body {  
    background: #ffffff url("img_tree.png") no-repeat right top;  
}
```

6 – Όρια CSS (borders)

Οι ιδιότητες περιγράμματος/συνόρων CSS σας επιτρέπουν να καθορίσετε το στυλ, το πλάτος και το χρώμα του περιγράμματος ενός στοιχείου.

I have borders on all sides.

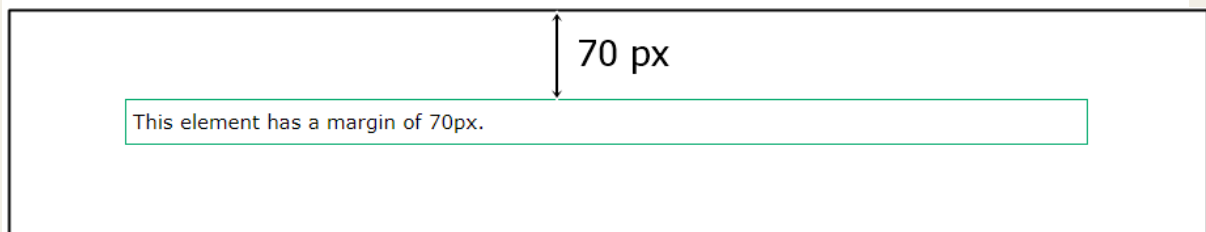
I have a red bottom border.

I have rounded borders.

I have a blue left border.

7 - Περιθώρια CSS (margins)

Τα περιθώρια χρησιμοποιούνται για τη δημιουργία χώρου γύρω από στοιχεία, εκτός των καθορισμένων ορίων.



Με την CSS, έχεις τον πλήρη έλεγχο των περιθωρίων. Υπάρχουν ιδιότητες για τον καθορισμό του περιθωρίου για κάθε πλευρά ενός στοιχείου (πάνω, δεξιά, κάτω και αριστερά).

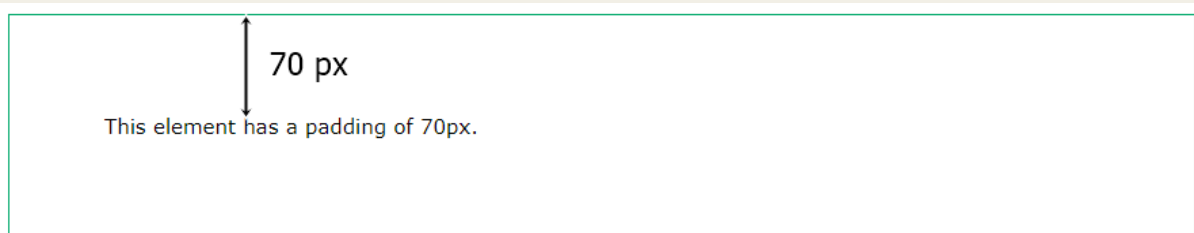
Example

Set different margins for all four sides of a <p> element:

```
p {
  margin-top: 100px;
  margin-bottom: 100px;
  margin-right: 150px;
  margin-left: 80px;
}
```

8 - Επένδυση CSS (padding)

Η ιδιότητα **επένδυσης** (padding) χρησιμοποιείται για τη δημιουργία χώρου γύρω από το περιεχόμενο ενός στοιχείου, μέσα σε οποιοδήποτε καθορισμένο περίγραμμα.



Με την CSS, έχετε τον πλήρη έλεγχο μιας επένδυσης. Υπάρχουν ιδιότητες για τον καθορισμό του περιθωρίου για κάθε πλευρά ενός στοιχείου (πάνω, δεξιά, κάτω και αριστερά).

Example

Set different padding for all four sides of a <div> element:

```
div {
  padding-top: 50px;
  padding-right: 30px;
  padding-bottom: 50px;
  padding-left: 80px;
}
```

9 - Ύψος και πλάτος CSS

Οι ιδιότητες **ύψους** και **πλάτους** χρησιμοποιούνται για τον καθορισμό του ύψους και του πλάτους ενός στοιχείου.

Οι ιδιότητες ύψους και πλάτους δεν περιλαμβάνουν επένδυση, περιγράμματα ή περιθώρια. Ορίζει το ύψος/πλάτος της περιοχής μέσα στην επένδυση (padding), το περίγραμμα και το περιθώριο του στοιχείου.

Οι ιδιότητες **ύψους** και **πλάτους** μπορεί να έχουν τις ακόλουθες τιμές:

- **auto** - Αυτό είναι προεπιλογή. Το πρόγραμμα περιήγησης υπολογίζει το ύψος και το πλάτος
- **μήκος** - Ορίζει το ύψος/πλάτος σε px, cm κ.λπ.
- **%** - Ορίζει το ύψος/πλάτος σε ποσοστιαία τιμή του μπλοκ που περιέχει
- **αρχικό** - Ορίζει το ύψος/πλάτος στην προεπιλεγμένη τιμή
- **CSS** «Κληρονομικότητα» - Το ύψος/πλάτος θα κληρονομείται από τη γονική του τιμή

Example

Set the height and width of a <div> element:

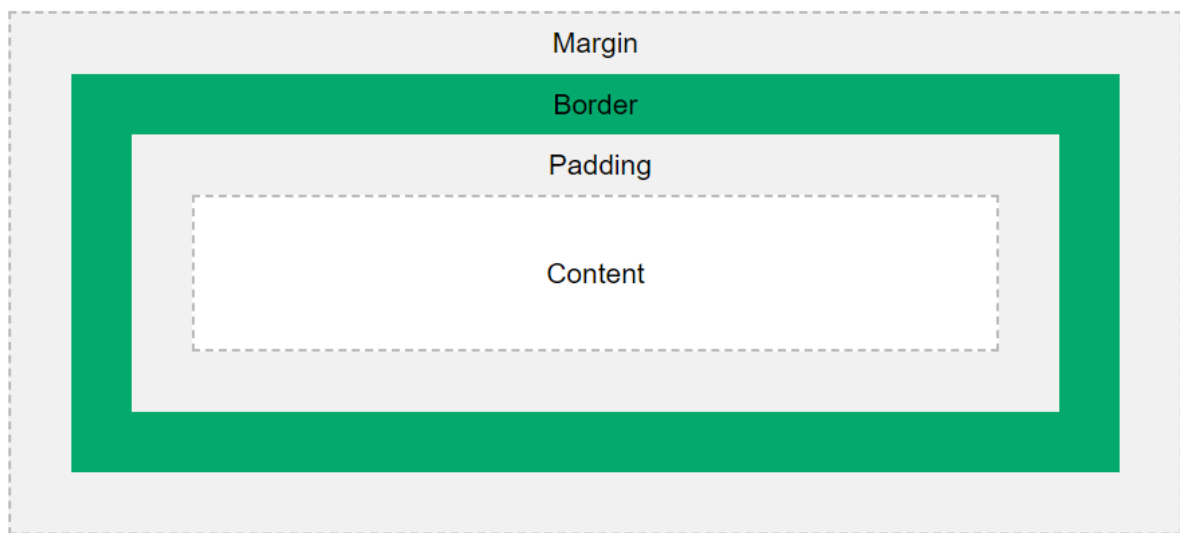
```
div {
  height: 200px;
  width: 50%;
  background-color: powderblue;
}
```

10 - Το μοντέλο κουτιού CSS (Box Model)

Στην CSS, ο όρος «μοντέλο κουτιού» (Box Model) χρησιμοποιείται όταν μιλάμε για το σχεδιασμό και τη διάταξη.

Το μοντέλο κουτιού CSS είναι ουσιαστικά ένα πλαίσιο που καλύπτει κάθε στοιχείο HTML. Αποτελείται από: περιθώρια, περιγράμματα, επένδυση, και το περιεχόμενο.

Η παρακάτω εικόνα απεικονίζει το μοντέλο πλαισίου:



Επεξήγηση των διάφορων μερών:

- Περιεχόμενο - Το περιεχόμενο του πλαισίου, όπου εμφανίζονται το κείμενο και οι εικόνες
- Επένδυση - Διασαφηνίζει μια περιοχή γύρω από το περιεχόμενο. Η επένδυση είναι διαφανής
- Περίγραμμα - Ένα περίγραμμα που περιτριγυρίζει την επένδυση και το περιεχόμενο
- Περιθώριο - Διασαφηνίζει μια περιοχή έξω από τα σύνορα. Το περιθώριο είναι διαφανές

Το μοντέλο κουτιού CSS μας επιτρέπει να προσθέσουμε ένα περίγραμμα γύρω από τα στοιχεία και να ορίσουμε το διάστημα μεταξύ των στοιχείων.

11 - Περίγραμμα CSS (outline)

Ένα περίγραμμα είναι μια γραμμή που σχεδιάζεται γύρω από τα στοιχεία, ΕΚΤΟΣ των ορίων, για να κάνει το στοιχείο «να ξεχωρίζει».

This element has a black border and a green outline with a width of 10px.

12 - Κείμενο CSS

Η CSS διαθέτει πολλές ιδιότητες για τη μορφοποίηση κειμένου.

TEXT FORMATTING

This text is styled with some of the text formatting properties. The heading uses the text-align, text-transform, and color properties. The paragraph is indented, aligned, and the space between characters is specified. The underline is removed from this colored "Try it Yourself" link.

- Χρώμα Κειμένου
- Στοίχιση Κειμένου
- Διακόσμηση κειμένου
- Μετασχηματισμός κειμένου
- Διάστιχο Κειμένου:
- Σκίαση κειμένου

13 - Γραμματοσειρές CSS (Fonts)

- Η επιλογή της κατάλληλης γραμματοσειράς μπορεί να ενισχύσει την ταυτότητα της επωνυμίας σας.
- Η επιλογή μιας γραμματοσειράς που είναι ευανάγνωστη είναι σημαντική. Η γραμματοσειρά προσθέτει αξία στο κείμενό σας. Είναι επίσης σημαντικό να επιλέξετε το σωστό χρώμα και το σωστό μέγεθος κειμένου για τη γραμματοσειρά.

Generic Font Family	Examples of Font Names
Serif	Times New Roman Georgia Garamond
Sans-serif	Arial Verdana Helvetica
Monospace	Courier New Lucida Console Monaco
Cursive	<i>Brush Script MT</i> <i>Lucida Handwriting</i>
Fantasy	Copperplate Papyrus

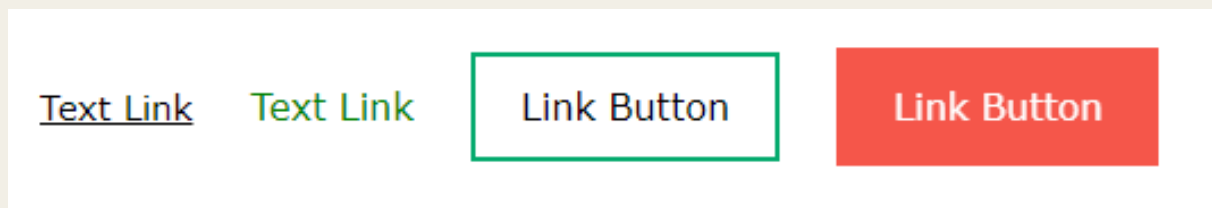
14 - Εικονίδια CSS (icons)

- Ο απλούστερος τρόπος για να προσθέσετε ένα εικονίδιο σε μια σελίδα HTML, είναι με μια βιβλιοθήκη εικονιδίων, όπως το Font Awesome.
- Προσθέστε το όνομα της καθορισμένης κατηγορίας εικονιδίων σε οποιοδήποτε ενσωματωμένο στοιχείο HTML (όπως `<i>` ή ``).
- Όλα τα εικονίδια στις παρακάτω βιβλιοθήκες εικονιδίων, είναι κλιμακούμενα διανυσματικά γραφικά που μπορούν να προσαρμοστούν με την CSS



15 - Συνδεσμοί CSS (links)

Οι σύνδεσμοι μπορούν να διαμορφωθούν με οποιαδήποτε ιδιότητα CSS (π.χ. χρώμα, γραμματοσειρά-οικογένεια, φόντο κ.λπ.)



Επιπλέον, οι σύνδεσμοι μπορούν να διαμορφωθούν με διαφορετικούς τρόπους ανάλογα με την κατάσταση στην οποία βρίσκονται.

Οι τέσσερις σύνδεσμοι είναι:

- **a:link** - ένας κανονικός, μη επισκέψιμος σύνδεσμος
- **a:visited** - ένας σύνδεσμος που έχει επισκεφθεί ο χρήστης
- **a:hover** - ένας σύνδεσμος όταν ο χρήστης μετακινεί το ποντίκι από πάνω του
- **a:active** - ένας σύνδεσμος τη στιγμή που πατιέται

16 - Λίστες CSS (lists)

Οι ιδιότητες λιστών CSS σας επιτρέπουν:

- Να ορίσετε διαφορετικούς δείκτες αντικειμένων λίστας για ταξινομημένες λίστες
- Να ορίσετε διαφορετικούς δείκτες στοιχείων λίστας για μη ταξινομημένες λίστες
- Να ορίσετε μια εικόνα ως δείκτη στοιχείου λίστας
- Να προσθέσετε χρώματα φόντου σε λίστες και φτιάξετε λίστες αντικειμένων.

The list-style-type Property

Example of unordered lists:

- Coffee
 - Tea
 - Coca Cola
- Coffee
 - Tea
 - Coca Cola

Example of ordered lists:

- I. Coffee
 - II. Tea
 - III. Coca Cola
- a. Coffee
 - b. Tea
 - c. Coca Cola

17 - Προβολή CSS (display)

- Η ιδιότητα Display (Προβολή) καθορίζει εάν/πώς εμφανίζεται ένα στοιχείο.
- Κάθε στοιχείο HTML έχει μια προεπιλεγμένη τιμή εμφάνισης ανάλογα με τον τύπο του στοιχείου που είναι. Η προεπιλεγμένη τιμή εμφάνισης για τα περισσότερα στοιχεία είναι επιπέδου μπλοκ ή επιπέδου inline.

The `<div>` element is a block-level element.

Examples of block-level elements:

- `<div>`
- `<h1>` - `<h6>`
- `<p>`
- `<form>`
- `<header>`
- `<footer>`
- `<section>`

18 - Πίνακες CSS (tables)

Η εμφάνιση ενός πίνακα HTML μπορεί να βελτιωθεί σημαντικά με την CSS:

Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany
Berglunds snabbköp	Christina Berglund	Sweden
Centro comercial Moctezuma	Francisco Chang	Mexico
Ernst Handel	Roland Mendel	Austria
Island Trading	Helen Bennett	UK
Königlich Essen	Philip Cramer	Germany
Laughing Bacchus Winecellars	Yoshi Tannamuri	Canada
Magazzini Alimentari Riuniti	Giovanni Rovelli	Italy

19 - Μέγιστο πλάτος CSS (max-width)

- Το βασικότερο πρόβλημα που προκύπτει συνήθως σχετικά με την ετικέτα `<div>` πιο πάνω είναι ότι το παράθυρο του προγράμματος περιήγησης είναι μικρότερο

από το πλάτος του στοιχείου. Στην περίπτωση αυτή, το πρόγραμμα περιήγησης προσθέτει μια οριζόντια γραμμή κύλισης στη σελίδα.

- Η χρήση, αντίθετα, ενός μέγιστου πλάτους, στην περίπτωση αυτή, θα βελτιώσει τον χειρισμό μικρών παραθύρων από το πρόγραμμα περιήγησης. Αυτό είναι σημαντικό όταν αναπτύσσετε έναν ιστότοπο που θα χρησιμοποιηθεί σε μικρές συσκευές.

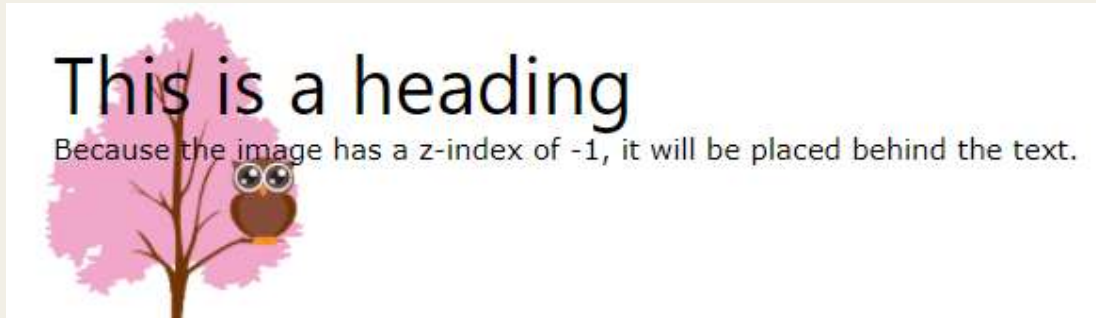


20 – Τοποθέτηση στοιχείων CSS (position)

- Η ιδιότητα τοποθέτησης στοιχείων CSS μας επιτρέπει να ορίσουμε ακριβείς συντεταγμένες τοποθέτησης για οποιοδήποτε στοιχείο σε μια σελίδα.
- Υπάρχουν πέντε διαφορετικές τιμές τοποθέτησης:
 - static
 - relative
 - fixed
 - absolute
 - sticky

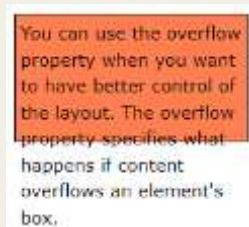
21 - CSS Z-index

- Η ιδιότητα z-index καθορίζει τη σειρά στοίβαξης ενός στοιχείου (ποιο στοιχείο πρέπει να τοποθετηθεί μπροστά ή πίσω από τα άλλα).
- Ένα στοιχείο μπορεί να έχει θετική ή αρνητική σειρά στοίβαξης:

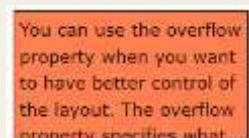


22 - Ιδιότητα Υπερχείλισης CSS (overflow)

- Η ιδιότητα overflow (υπερχείλιση) ορίζει αν θα γίνεται απόσπαση του περιεχομένου ή προσθήκη γραμμών κύλισης όταν το περιεχόμενο ενός στοιχείου είναι πολύ μεγάλο για να χωρέσει σε μια οριοθετημένη περιοχή.
- Η ιδιότητα υπερχειλίσης έχει τις ακόλουθες τιμές:
- visible



- hidden



- scroll

overflow property specifies what happens if content overflows an element's box.

- auto

You can use the overflow property when you want to have better control of the layout. The

23 - CSS Float

Η ιδιότητα float χρησιμοποιείται για την τοποθέτηση και τη μορφοποίηση του περιεχομένου

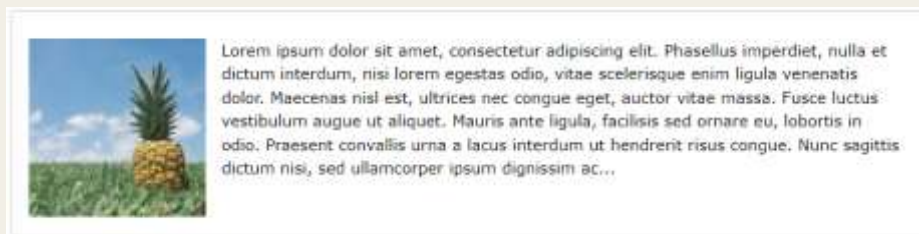
Παράδειγμα - **float: δεξιά**

Το ακόλουθο παράδειγμα ορίζει ότι μια εικόνα πρέπει να επιπλέει προς τα **δεξιά** σε ένα κείμενο:



Παράδειγμα - **float: αριστερά**

Το ακόλουθο παράδειγμα ορίζει ότι μια εικόνα πρέπει να επιπλέει προς τα **αριστερά** σε ένα κείμενο:



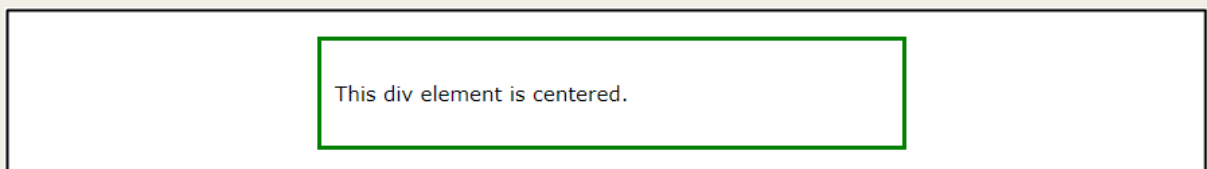
24 - Ενσωματωμένο μπλοκ CSS (inline-block)

- Η εμφάνιση: το ενσωματωμένο μπλοκ επιτρέπει τον ορισμό του πλάτους και του ύψους σε ένα στοιχείο.
- Με την εμφάνιση: inline-block τα περιθώρια (margins) και οι επενδύσεις (paddings) που βρίσκονται πάνω και κάτω διατηρούνται.
- Η εμφάνιση: inline-block δεν προσθέτει μια αλλαγή γραμμής μετά το στοιχείο, έτσι ώστε το στοιχείο να μπορεί να περιστοιχιστεί δίπλα σε άλλα στοιχεία.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum consequat scelerisque elit sit amet consequat. Aliquam erat volutpat. Aliquam venenatis gravida nisl sit amet facilisis. Nullam cursus fermentum velit sed laoreet.

25 - Ευθυγράμμιση CSS (align)

- Για να κεντράρετε οριζόντια ένα στοιχείο μπλοκ (όπως <div>), χρησιμοποιήστε το περιθώριο : auto
- Η ρύθμιση του πλάτους του στοιχείου θα αποτρέψει την επέκτασή του στις άκρες του περιεχομένου του.
- Στη συνέχεια, το στοιχείο θα λάβει το καθορισμένο πλάτος και ο υπόλοιπος χώρος θα διαχωριστεί εξίσου μεταξύ των περιθωρίων:



26 - Συνδυαστές CSS (combinators)

- Ένας επιλογέας CSS μπορεί να περιέχει περισσότερους από έναν απλούς επιλογείς. Μεταξύ των απλών επιλογέων, μπορούμε να συμπεριλάβουμε έναν συνδυαστή.
- Υπάρχουν τέσσερις διαφορετικοί συνδυαστές στην CSS :
- επιλογέας «απόγονος»/ descendant selector (διάστημα) -> π.χ. `div p { φόντο-χρώμα: κίτρινο }`
- παιδικός επιλογέας (>) -> π.χ. `div > p { φόντο-χρώμα: κίτρινο }`
- παρακείμενος επιλογέας αδελφών/ adjacent sibling selector (+) -> π.χ. `DIV + p {χρώμα φόντου: κίτρινο }`
- επιλογέας γενικών συνδυασμών αδελφών/ general sibling selector (~) -> π.χ. `div ~ p { φόντο-χρώμα: κίτρινο }`

27 - Ψευδό-κλάση CSS (pseudo-class)

- Μια ψευδό-κλάση χρησιμοποιείται για να καθορίσει μια ειδική κατάσταση ενός στοιχείου.
- Για παράδειγμα, μπορεί να χρησιμοποιηθεί για:
 - Το στυλ ενός στοιχείου όταν ένας χρήστης μετακινεί το ποντίκι από πάνω του
 - Το στυλ επισκέψιμων και μη επισκέψιμων συνδέσμων με διαφορετικό τρόπο
 - Τη στιλιστική ανάπτυξη ενός στοιχείου όταν εστιάζεται

```
/* unvisited link */
a:link {
  color: #FF0000;
}

/* visited link */
a:visited {
  color: #00FF00;
}

/* mouse over link */
a:hover {
  color: #FF00FF;
}

/* selected link */
a:active {
  color: #0000FF;
}
```

28 - CSS Ψευδο-στοιχείο (pseudo-element)

- Ένα ψευδο-στοιχείο CSS χρησιμοποιείται για το στυλ συγκεκριμένων τμημάτων ενός στοιχείου.
- Για παράδειγμα, μπορεί να χρησιμοποιηθεί για:
 - Το στυλ του πρώτου γράμματος ή γραμμής ενός στοιχείου
 - Την εισαγωγή περιεχομένου πριν ή μετά το περιεχόμενο ενός στοιχείου

```
p::
```

Το ψευδο-στοιχείο `::first-line` (πρώτη γραμμή) χρησιμοποιείται για να προσθέσει ένα ειδικό στυλ στην πρώτη γραμμή ενός κειμένου.

29 - Αδιαφάνεια CSS (opacity)

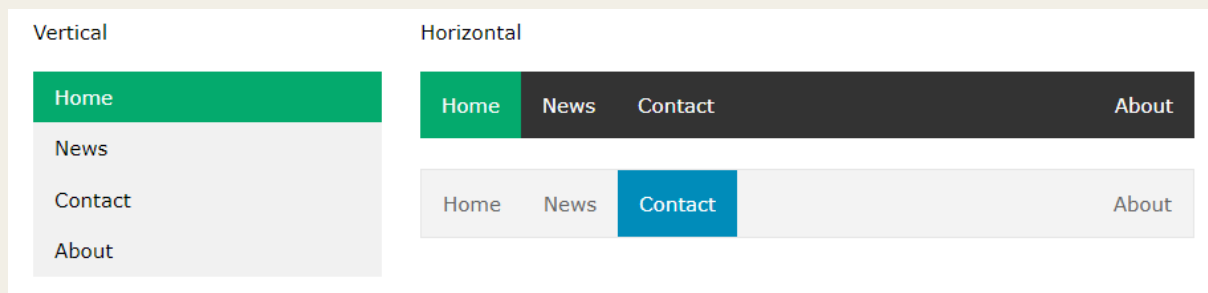
- Η ιδιότητα αδιαφάνειας καθορίζει τον βαθμό αδιαφάνειας/διαφάνειας ενός στοιχείου.

- Η ιδιότητα αδιαφάνειας μπορεί να λάβει τιμές από το 0,0 - 1,0. Η χαμηλότερη τιμή είναι και η πιο διαφανής.



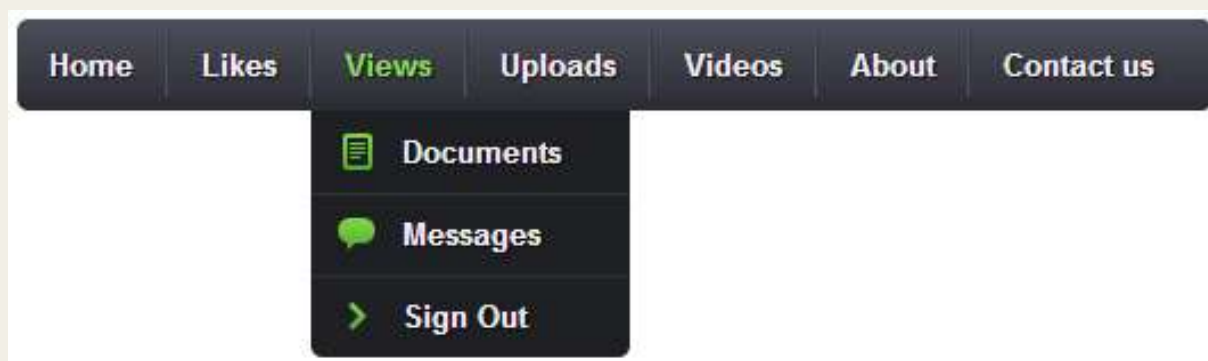
30 - Γραμμή πλοήγησης CSS (navigation bar)

Με την CSS μπορείτε να μετατρέψετε βαρετά μενού επιλογών HTML σε ευπαρουσίαστες γραμμές πλοήγησης.



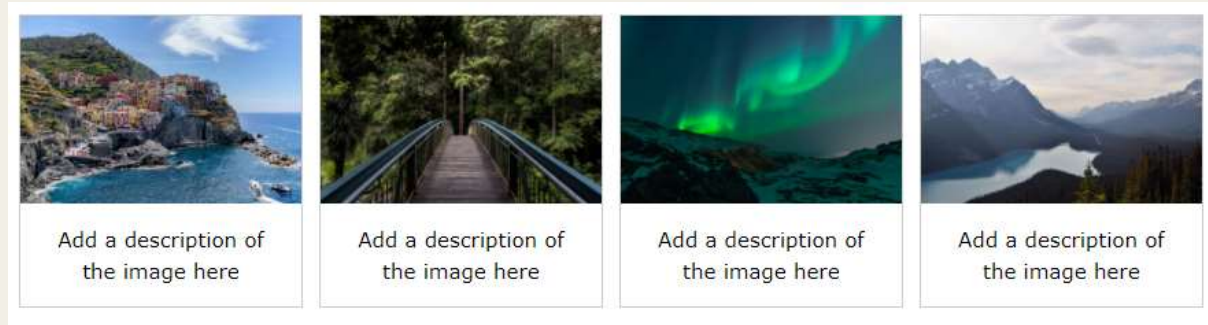
31 - Αναπτυσσόμενο μενού CSS (dropdowns)

Με την CSS μπορείτε να μετατρέψετε βαρετά μενού επιλογών HTML σε ευπαρουσίαστες γραμμές πλοήγησης.



32 - Συλλογή εικόνων CSS (image gallery)

Η CSS μπορεί να χρησιμοποιηθεί για τη δημιουργία μιας συλλογής εικόνων.



33 – Εικόνες Sprintes CSS

- Μια εικόνα sprints είναι μια συλλογή από εικόνες που τοποθετούνται σε μια ενιαία εικόνα.
- Μια ιστοσελίδα με πολλές εικόνες μπορεί να πάρει πολύ χρόνο να φορτωθεί και δημιουργεί πολλαπλά αιτήματα διακομιστή.
- Η χρήση των εικόνων sprints θα μειώσει τον αριθμό των αιτημάτων διακομιστή και θα αποθηκεύσει το εύρος ζώνης.



34 - Επιλογείς Attr CSS

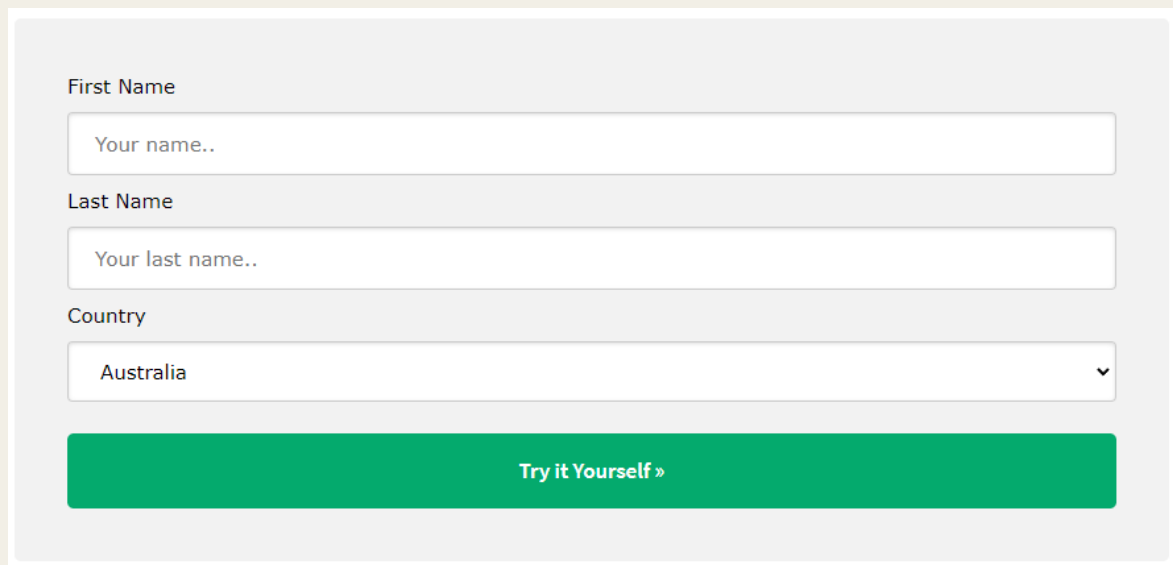
- Ο επιλογέας [attribute] χρησιμοποιείται για την επιλογή στοιχείων με μια προκαθορισμένη ιδιότητα.

- Το ακόλουθο παράδειγμα επιλέγει όλα τα <a> στοιχεία με μια ιδιότητα - στόχου:

```
a[target] {  
  background-color: yellow;  
}
```

35 - Φόρμες CSS (forms)

Η εμφάνιση μιας φόρμας HTML μπορεί να βελτιωθεί σημαντικά με την CSS:



First Name

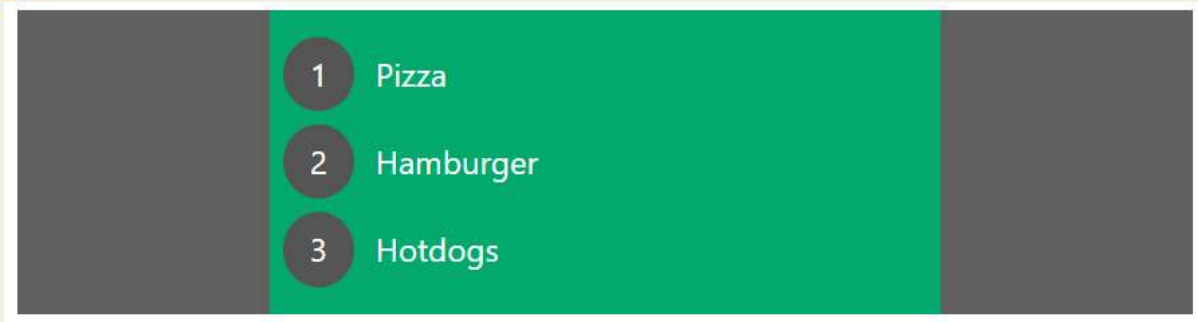
Last Name

Country

Try it Yourself »

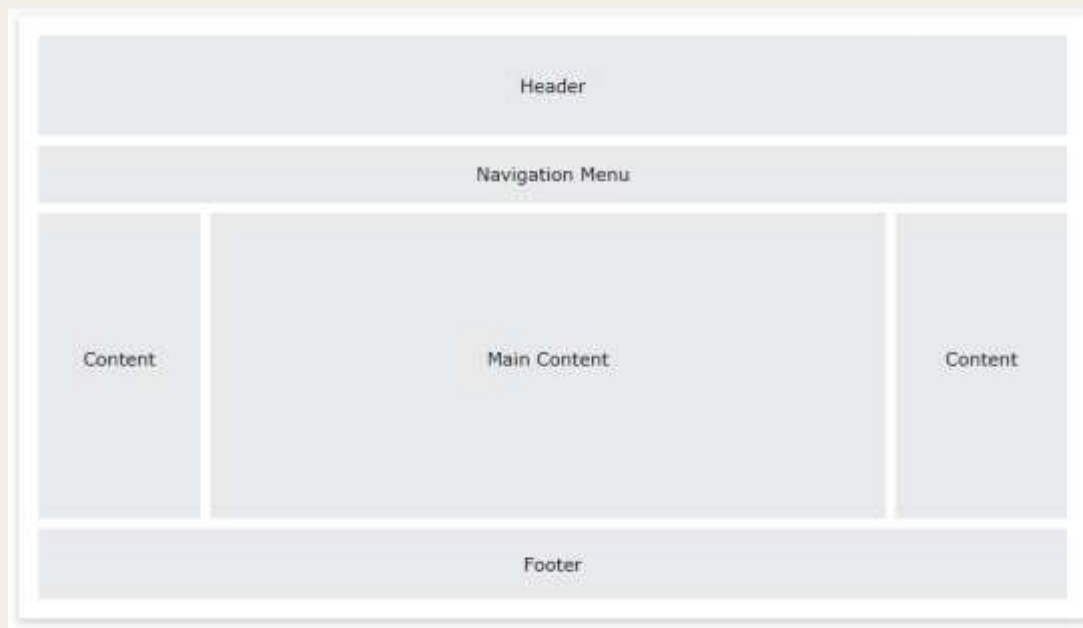
36 – Μετρητές CSS (Counters)

Οι μετρητές CSS είναι «μεταβλητές» που διατηρούνται από την CSS, οι τιμές των οποίων μπορούν να αυξηθούν από τους κανόνες CSS (για να παρακολουθείτε πόσες φορές χρησιμοποιούνται). Οι μετρητές CSS σας επιτρέπουν να προσαρμόζετε την εμφάνιση του περιεχομένου με βάση την τοποθέτησή του στο έγγραφο.



37 - Διάταξη Ιστοσελίδας CSS (website layout)

Ένας ιστότοπος χωρίζεται συχνά σε κεφαλίδες, μενού, περιεχόμενο και υποσέλιδο



38 – Μονάδες μέτρησης CSS (units)

- Η CSS έχει πολλές διαφορετικές μονάδες για τη διαμόρφωση ενός μήκους.

- Πολλές ιδιότητες CSS λαμβάνουν τιμές «μήκους», όπως πλάτος, περιθώριο, επένδυση (padding), μέγεθος γραμματοσειράς κ.λπ.,
- Το μήκος είναι ένας αριθμός που ακολουθείται από μια μονάδα μήκους, όπως 10px, 2em, κ.λπ.,

Unit	Description
cm	centimeters
mm	millimeters
in	inches (1in = 96px = 2.54cm)
px *	pixels (1px = 1/96th of 1in)
pt	points (1pt = 1/72 of 1in)
pc	picas (1pc = 12 pt)

39 - Ειδικότητα CSS (specificity)

- Εάν υπάρχουν δύο ή περισσότεροι κανόνες CSS που επισημαίνουν το ίδιο στοιχείο, ο επιλογέας με την υψηλότερη τιμή ειδικότητας θα «κερδίσει» και η δήλωση στυλ του θα εφαρμοστεί στο στοιχείο HTML.
- Θεωρήστε την ειδικότητα ως σκορ/βαθμό που καθορίζει ποια δήλωση στυλ θα εφαρμοστεί τελικά σε ένα στοιχείο.

```
<html>
<head>
  <style>
    p {color: red;}
  </style>
</head>
<body>

<p>Hello World!</p>

</body>
</html>
```

Στο πιο πάνω παράδειγμα, χρησιμοποιήσαμε το στοιχείο «p» ως επιλογή και ορίσαμε ένα κόκκινο χρώμα για το στοιχείο. Έτσι λοιπόν, το κείμενο θα είναι κόκκινο.

40 - CSS !important

- The !important είναι ένας κανόνας της CSS που χρησιμοποιείται για να προσθέσει μεγαλύτερη σημασία σε μια ιδιότητα/τιμή από το κανονικό.
- Στην πραγματικότητα, αν χρησιμοποιήσεις τον κανόνα **!important**, θα παρακάμψει ΟΛΟΥΣ τους προηγούμενους κανόνες στυλ για τη συγκεκριμένη ιδιότητα στο στοιχείο!

```
#myid {  
  background-color: blue;  
}  
  
.myclass {  
  background-color: gray;  
}  
  
p {  
  background-color: red !important;  
}
```

41 – Μαθηματικές συναρτήσεις CSS (math function)

- Οι μαθηματικές συναρτήσεις CSS επιτρέπουν τη χρήση μαθηματικών εκφράσεων ως τιμές ιδιοτήτων. Μερικές μαθηματικές συναρτήσεις είναι: calc(), max() και min() συναρτήσεις.
- Παράδειγμα: η χρήση του calc() για τον υπολογισμό του πλάτους ενός στοιχείου <div>:

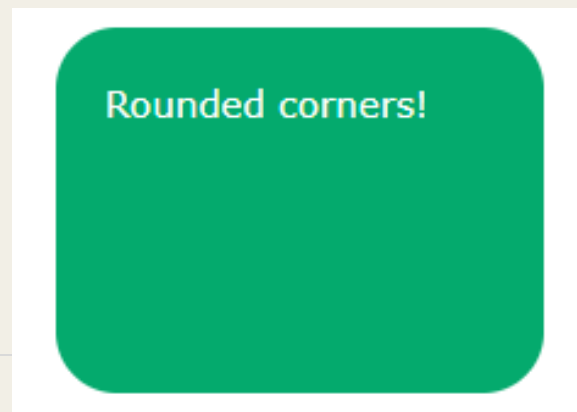
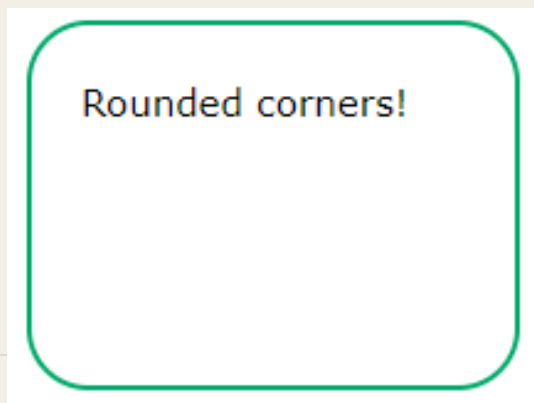
```
#div1 {  
  position: absolute;  
  left: 50px;  
  width: calc(100% - 100px);  
  border: 1px solid black;  
  background-color: yellow;  
  padding: 5px;  
}
```

Κεφάλαιο 2: Προηγμένοι κανόνες σύνταξης CSS

1. Στρογγυλεμένες γωνίες CSS (rounded corners)

Η ιδιότητα **ορίων ακτίνας (rounded corners)** καθορίζει την ακτίνα των γωνιών ενός στοιχείου.

Συμβουλή: Αυτή η ιδιότητα σας επιτρέπει να προσθέσετε στρογγυλεμένες γωνίες σε στοιχεία!



2. Εικόνες περιγράμματος CSS (border images)

- Η ιδιότητα εικόνας περιγράμματος CSS σας επιτρέπει να καθορίσετε μια εικόνα που θα χρησιμοποιηθεί αντί για το κανονικό περίγραμμα γύρω από ένα στοιχείο.
- Η ιδιότητα αυτή αποτελείται από τρία μέρη:
 - Η εικόνα που θα χρησιμοποιηθεί ως περίγραμμα
 - Ο καθορισμός των ορίων όπου θα τεμαχιστεί η εικόνα
 - Ο καθορισμός των μεσαίων τμημάτων αν πρόκειται να επαναληφθούν ή να επεκταθούν



3. Φόντο CSS (background)

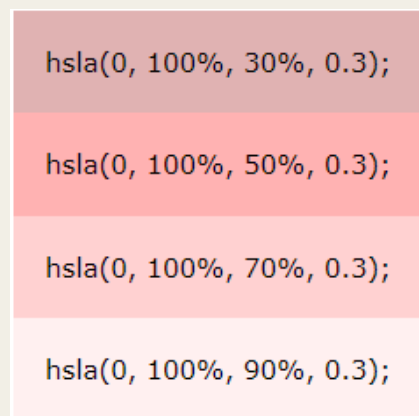
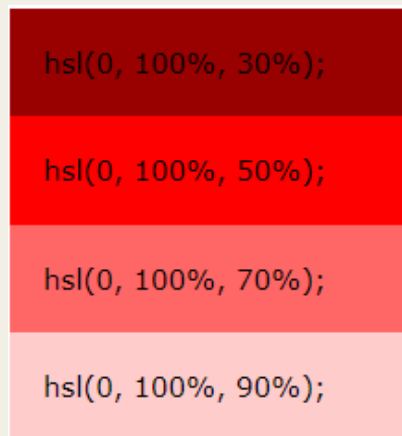
- Η CSS σας επιτρέπει να προσθέσετε πολλαπλές εικόνες φόντου για ένα στοιχείο, μέσω της ιδιότητας φόντου-εικόνας
- Το ακόλουθο παράδειγμα έχει δύο εικόνες φόντου, η πρώτη εικόνα είναι ένα λουλούδι (ευθυγραμμισμένη με την κάτω δεξιά γωνία) και η δεύτερη εικόνα είναι ένα φόντο χαρτιού (ευθυγραμμισμένη με την πάνω αριστερή γωνία)

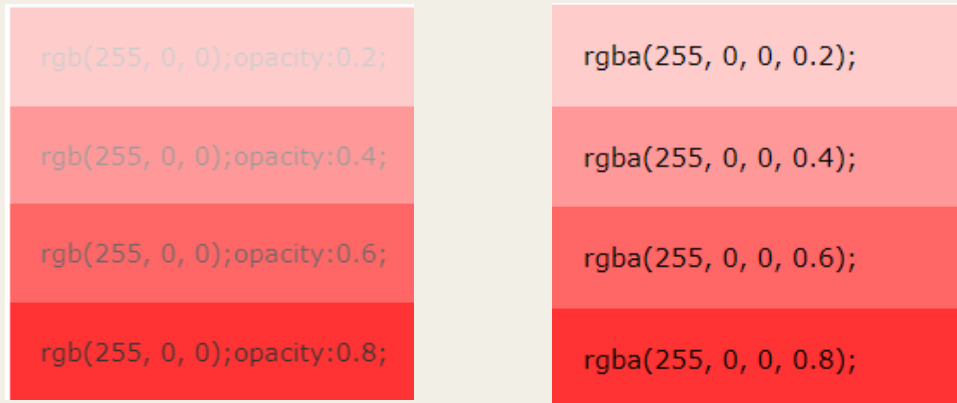
```
#example1 {
  background-image: url(img_flwr.gif), url(paper.gif);
  background-position: right bottom, left top;
  background-repeat: no-repeat, repeat;
}
```



4. Χρώματα CSS

- Η CSS υποστηρίζει +140 ονόματα χρωμάτων, δεκαεξαδικές τιμές, τιμές RGB, τιμές RGBA, τιμές HSL, τιμές HSLA και αδιαφάνεια.





5. Λέξεις-κλειδιά για διάφορες αποχρώσεις χρωμάτων CSS (Color Keywords)

Στο σημείο αυτό θα αναλύσουμε τις λέξεις κλειδιά των ιδιοτήτων transparent, currentcolor και inherit:

- Η ιδιότητα transparent keyword χρησιμοποιείται για να μετατρέψει ένα χρώμα διαφανές. Αυτή η ιδιότητα χρησιμοποιείται συχνά για να μετατρέψει ένα διαφανές χρώμα φόντου σε ένα στοιχείο.
- Η ιδιότητα currentcolor keyword είναι σαν μια μεταβλητή που διατηρεί την τρέχουσα τιμή της χρωματικής ιδιότητας ενός στοιχείου.

Αυτή μπορεί να φανεί χρήσιμη αν θέλετε ένα συγκεκριμένο χρώμα να είναι συμβατό με ένα στοιχείο ή μια σελίδα.

- Η ιδιότητα inherit keyword διευκρινίζει ότι μια ιδιότητα θα πρέπει να κληρονομήσει την τιμή της από το γονικό της στοιχείο.

Η ιδιότητα inherit keyword μπορεί να χρησιμοποιηθεί για οποιαδήποτε ιδιότητα CSS και σε οποιοδήποτε στοιχείο HTML.

6. Διαβαθμίσεις CSS (gradient)

Οι διαβαθμίσεις CSS σάς επιτρέπουν να εμφανίζετε ομαλές μεταβάσεις μεταξύ δύο ή περισσότερων καθορισμένων χρωμάτων.

Η CSS ορίζει τρεις τύπους διαβαθμίσεων:

- Γραμμικές διαβαθμίσεις (μετακινώντας προς τα κάτω/πάνω/αριστερά/δεξιά/διαγώνια)
- Ακτινικές διαβαθμίσεις (οι οποίες ορίζονται από το κέντρο τους)
- Κωνικές διαβαθμίσεις (οι οποίες περιστρέφονται γύρω από ένα κεντρικό σημείο)



7. Σκίαση CSS (shadows)

Με την CSS μπορείτε να προσθέσετε σκιάσεις στο κείμενο και στα στοιχεία.

Στο κεφάλαιο αυτό θα μάθετε για τις ακόλουθες ιδιότητες :

- σκίαση κειμένου (text-shadow)
- Σκίαση κουτιού (box-shadow)

With CSS you can create shadow effects!



Shadows

8. Εφέ κειμένου CSS (text effects)

Στο κεφάλαιο αυτό θα μάθετε για το **Text-overflow**, **word-wrap**, **word-break**, **writing-mode** :

- Η ιδιότητα υπερχείλισης κειμένου (**text overflow**) καθορίζει τον τρόπο με τον οποίο το περιεχόμενο υπερχείλισης κειμένου που δεν εμφανίζεται θα πρέπει να επισημαίνεται στον χρήστη.
- Η ιδιότητα **CSS word-wrap** επιτρέπει το συλλαβικό σπάσιμο των μεγάλων λέξεων και τη μεταφορά μέρους τους στην επόμενη γραμμή.
- -Η ιδιότητα περιτύλιξης λέξεων CSS ορίζει τους κανόνες αλλαγής γραμμής για μια λέξη.

- Η ιδιότητα **writing mode CSS** καθορίζει εάν οι γραμμές του κειμένου είναι οριζοντίως ή καθέτως.

9. Γραμματοσειρές Ιστού CSS (Web Fonts)

- Οι γραμματοσειρές ιστού επιτρέπουν στους σχεδιαστές ιστοσελίδων να χρησιμοποιούν γραμματοσειρές που δεν είναι εγκατεστημένες στον υπολογιστή ενός χρήστη.
- Όταν βρείτε/αγοράσετε τη γραμματοσειρά που θέλετε να χρησιμοποιήσετε, απλά συμπεριλάβετε το αρχείο γραμματοσειράς στον διακομιστή Web και θα μεταφορτωθεί αυτόματα στον χρήστη όταν χρειαστεί.
- Οι δικές σας γραμματοσειρές ορίζονται στον κανόνα CSS @ font-face

The @font-face Rule

With CSS, websites can use **fonts other than the pre-selected "web-safe" fonts**.

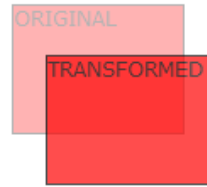
10. Μετασχηματισμοί CSS 2D (Transforms)

Οι μετασχηματισμοί CSS 2D σας επιτρέπουν να μετακινείτε, να περιστρέφετε, να κλιμακώνετε και να κυρτώνετε στοιχεία.

The rotate() Method



The translate() Method



The scale() Method



11. Μετασχηματισμοί CSS 3D

Με την ιδιότητα μετασχηματισμού CSS 3D μπορείτε να χρησιμοποιήσετε τις ακόλουθες μεθόδους μετασχηματισμού 3D:

- RotateX
- RotateY
- RotateZ

The rotateY() Method



The rotateX() Method



12. Μεταβάσεις CSS 3D (transitions)

- Οι μεταβάσεις CSS σας επιτρέπουν να αλλάξετε τις τιμές μιας ιδιότητας ομαλά για μια δεδομένη διάρκεια.
- Για να δημιουργήσετε ένα εφέ μετάβασης, πρέπει να καθορίσετε δύο πράγματα:
 - την ιδιότητα CSS στην οποία θέλετε να προσθέσετε ένα εφέ
 - τη διάρκεια του εφέ
- Το παρακάτω παράδειγμα δείχνει ένα κόκκινο <div> στοιχείο 100px * 100px. Το <div> στοιχείο έχει επίσης καθορίσει ένα εφέ μετάβασης για την ιδιότητα πλάτους, με διάρκεια 2 δευτερόλεπτα:

```
div {  
  width: 100px;  
  height: 100px;  
  background: red;  
  transition: width 2s;  
}
```

13. Κινούμενα Σχέδια CSS (Animations)

- Ένα κινούμενο σχέδιο επιτρέπει σε ένα στοιχείο να μεταλλάσσει σταδιακά από το ένα στυλ στο άλλο.
- Μπορείτε να αλλάξετε όσες ιδιότητες CSS θέλετε, όσες φορές θέλετε.
- Για να χρησιμοποιήσετε ένα κινούμενο σχέδιο CSS, πρέπει πρώτα να καθορίσετε μερικά βασικά πλαίσια για το κινούμενο σχέδιο.
- Τα πλαίσια-κλειδιά διατηρούν τα στυλ που θα έχει το στοιχείο σε συγκεκριμένες χρονικές στιγμές.
- Αυτές είναι οι κύριες ιδιότητες του κινούμενου σχεδίου:

- `@keyframes`
- `animation-name`
- `animation-duration`
- `animation-delay`
- `animation-iteration-count`
- `animation-direction`
- `animation-timing-function`
- `animation-fill-mode`
- `animation`

14. Συμβουλές εργαλείων CSS (Tooltips)

- Μια επεξήγηση εργαλείου χρησιμοποιείται συχνά για να διασαφηνίσει πρόσθετες πληροφορίες σχετικά με κάτι όταν ο χρήστης μετακινεί τον δείκτη του ποντικιού πάνω από ένα στοιχείο:

default top	blue top	green top	orange top	red top
tooltip top	tooltip top blue	tooltip top green	tooltip top orange	tooltip top red
tooltip bottom	tooltip bottom blue	tooltip bottom green	tooltip bottom orange	tooltip bottom red
default bottom	blue bottom	green bottom	orange bottom	red bottom

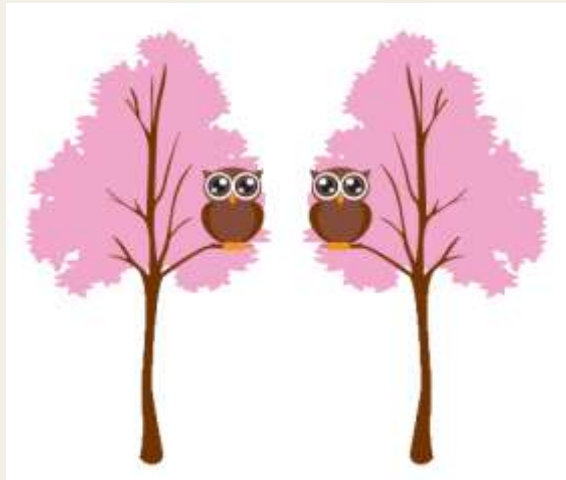
15. Εικόνες στυλ CSS (Style images)

- Η χρήση της CSS για την στιλιστική ανάπτυξη των εικόνων σας επιτρέπει να καθορίσετε ομοιόμορφα τον τρόπο εμφάνισης των εικόνων στον ιστότοπό σας με ένα περιορισμένο σύνολο κανόνων :
- προσθέστε ένα περίγραμμα και αλλάξτε το σχήμα και το μέγεθος της εικόνας, ...



16. Εικόνες με εφέ αντανάκλασης CSS (image reflection)

- Η ιδιότητα αντανάκλασης πλαισίου χρησιμοποιείται για τη δημιουργία εφέ αντανάκλασης μιας εικόνας.
- Η τιμή της ιδιότητας αντανάκλασης πλαισίου μπορεί να είναι : κάτω, πάνω, αριστερά ή δεξιά



17. CSS Object-fit

- Η ιδιότητα Object-fit CSS χρησιμοποιείται για τον καθορισμό του τρόπου με τον οποίο ένα ή <video> θα πρέπει να αλλάξει το μέγεθός του, ούτως ώστε να χωράει στον περιεχόμενο ενός στοιχείου.
- Αυτή η ιδιότητα δίνει εντολή στο περιεχόμενο να γεμίσει τον περιέκτη με διάφορους τρόπους, όπως πχ. «διατήρησε αυτή την αναλογία διαστάσεων» ή «τέντωσε και χρησιμοποίησε όσο το δυνατόν περισσότερο χώρο».

- Κοιτάξτε την παρακάτω εικόνα από το Παρίσι. Αυτή η εικόνα έχει πλάτος 400 εικονοστοιχεία (pixels) και ύψος 300 εικονοστοιχεία:



18. Θέση αντικειμένου CSS (Object-position)

- Ας πούμε ότι το μέρος της εικόνας που εμφανίζεται, δεν είναι τοποθετημένο όπως το θέλουμε. Για να τοποθετήσετε την εικόνα με τον επιθυμητό τρόπο, θα χρησιμοποιήσουμε την ιδιότητα **θέσης αντικειμένου CSS**
- Εδώ θα χρησιμοποιήσουμε την ιδιότητα **θέσης αντικειμένου CSS** για να τοποθετήσουμε την εικόνα έτσι ώστε το μεγάλο παλαιό κτίριο να βρίσκεται στο κέντρο:



19. Ιδιότητα Μάσκας-Στρώσης CSS (Masking)

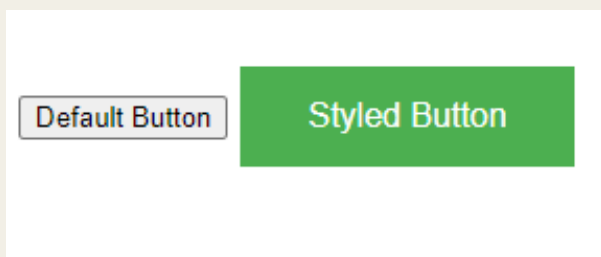
- Με την ιδιότητα **CSS Masking** σας δίνεται η δυνατότητα να δημιουργήσετε μια μάσκα-στρώσης την οποία θα τοποθετήσετε πάνω από ένα στοιχείο για να αποκρύψετε μερικώς ή πλήρως τμήματά του.
- Η ιδιότητα μάσκας-στρώσης CSS καθορίζει ποια μέρη της στρώσης μιας εικόνας είναι αδιαφανή, ημιδιαφανή ή διαφανή.

Μια εικόνα με μάσκα-στρώσης μπορεί να είναι μια εικόνα PNG, μια εικόνα SVG, μια διαβάθμιση CSS (gradient) ή ένα <mask>στοιχείο SVG.



20. Κουμπιά CSS (Buttons)

- Υπάρχουν πολλές ιδιότητες για την στυλιστική ανάπτυξη ενός κουμπιού στην CSS
- Πιο κάτω παρατίθεται ένα παράδειγμα:



```
.button {
  background-color: #4CAF50;
  border: none;
  color: white;
  padding: 15px 32px;
  text-align: center;
  text-decoration: none;
  font-size: 16px;
  margin: 4px 2px;
}
```

21. Σελιδοποίηση CSS (Pagination)

Αν έχετε μια ιστοσελίδα με πολλές σελίδες, μπορεί να θέλετε να προσθέσετε κάποιο είδος σελιδοποίησης σε κάθε σελίδα και πιο κάτω παρατίθενται μερικά παραδείγματα :



22. Πολλαπλές στήλες CSS (Multiple Columns)

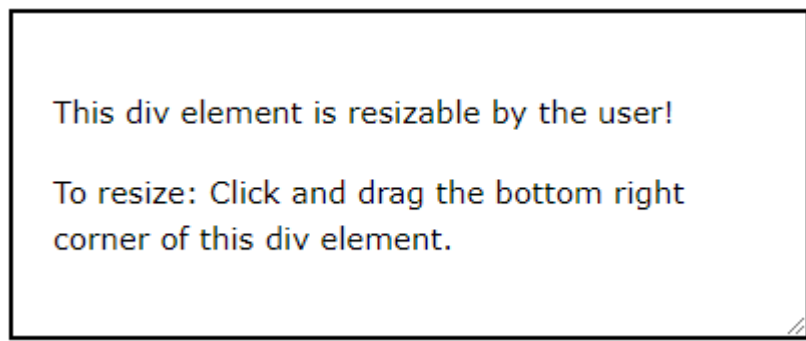
Η διάταξη πολλαπλών στηλών CSS επιτρέπει τον εύκολο ορισμό πολλαπλών στηλών κειμένου - ακριβώς όπως στις εφημερίδες:



23. Διεπαφή Χρήστη CSS (User Interface)

Στο κεφάλαιο αυτό θα μάθετε για τις ακόλουθες ιδιότητες διεπαφής χρήστη CSS:

- Η ιδιότητα αυξομείωσης καθορίζει εάν (και πώς) ένα στοιχείο πρέπει ή μπορεί να αλλάξει το μέγεθός του από το χρήστη.



- Η ιδιότητα **outline-offset** προσθέτει χώρο μεταξύ ενός περιγράμματος ή της άκρης ή του περιγράμματος ενός στοιχείου.



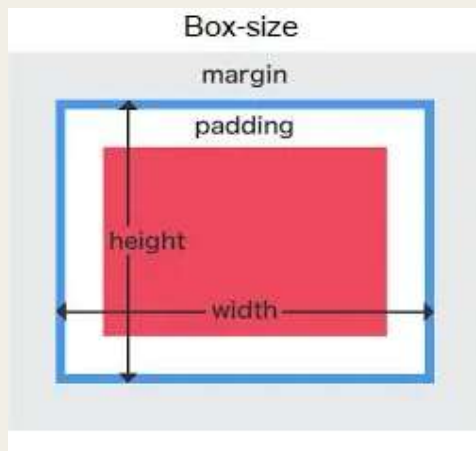
24. Μεταβλητές CSS (Variables)

- Η συνάρτηση var() χρησιμοποιείται για την εισαγωγή της τιμής μιας μεταβλητής CSS.
- Οι μεταβλητές CSS έχουν πρόσβαση στο Dom, πράγμα που σημαίνει ότι μπορείτε να δημιουργήσετε μεταβλητές με τοπική ή παγκόσμια εμβέλεια, να αλλάξετε τις μεταβλητές με JavaScript ή ακόμη και να αλλάξετε τις μεταβλητές με βάση τα ερωτήματα πολυμέσων.

- Μια καλή χρήση των μεταβλητών CSS μπορεί να γίνει για την επιλογή των χρωμάτων κατά τη φάση του σχεδιασμού. Αντί να αντιγράψετε και να επικολλήσετε τα ίδια χρώματα ξανά και ξανά, μπορείτε να τα τοποθετήσετε σε μεταβλητές.

25. Ιδιότητα ορισμού μεγέθους κιβωτίου CSS (Box sizing)

- Η ιδιότητα μεγέθους κουτιού στο CSS καθορίζει τον τρόπο με τον οποίο ο χρήστης θα πρέπει να υπολογίζει το συνολικό πλάτος και ύψος ενός στοιχείου, δηλαδή της επένδυσης (padding) ενός περιγράμματος, που πρέπει να περιλαμβάνονται ή όχι.



26. Ερωτήματα μέσω CSS (Media Queries)

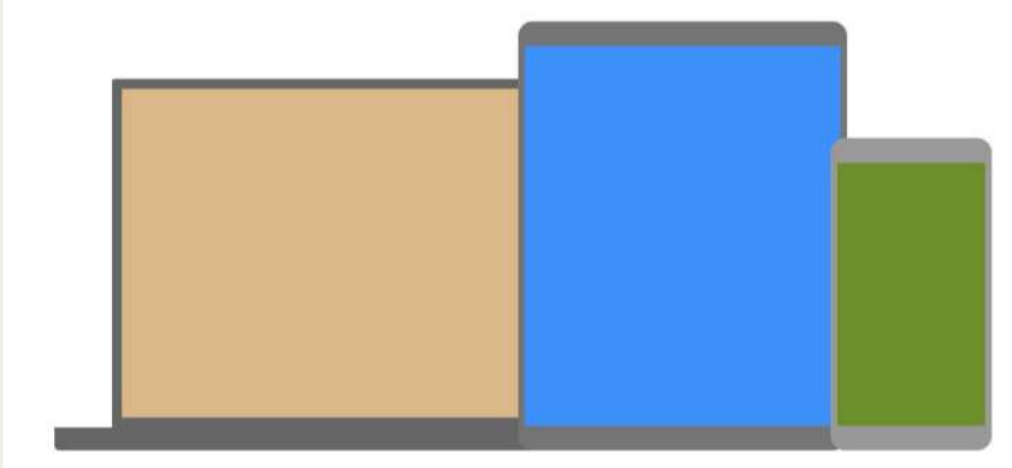
Τα ρωτήματα πολυμέσων στο CSS3 επέκτειναν την ιδέα των τύπων πολυμέσων CSS2: Αντί να αναζητήσουν έναν τύπο συσκευής, εξετάζουν τις δυνατότητες μιας συσκευής.

Τα ερωτήματα πολυμέσων μπορούν να χρησιμοποιηθούν για να ελέγξουν πολλούς παραμέτρους, όπως:

- Το πλάτος και ύψος της θύρας προβολής
- Το πλάτος και το ύψος της διάταξης
- Τον προσανατολισμό (εάν δηλαδή το tablet/τηλέφωνο βρίσκεται σε λειτουργία οριζόντιας ή κατακόρυφης κλίσης (πορτρέτο) και σύνθεσης μιας εικόνας)
- Η ανάλυση/ ευκρίνεια μιας εικόνας (resolution)

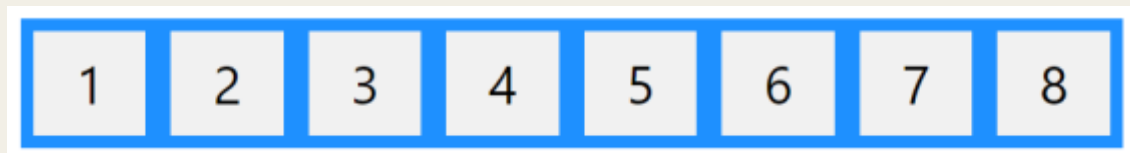
Η χρήση ερωτημάτων πολυμέσων είναι μια δημοφιλής τεχνική για την εφαρμογή ενός ειδικά σχεδιασμένου φύλλου ύφους σε επιτραπέζιους υπολογιστές, φορητούς υπολογιστές, tablet και κινητά τηλέφωνα (όπως τηλέφωνα iPhone και Android).

27. Παραδείγματα MQ CSS (MQ Examples)



28. Ιδιότητα Ευέλικτης Ενότητας Διάταξης Κουτιού CSS (Flexbox)

Η Ενότητα Ευέλικτης Διάταξης Κουτιού Διάταξης της CSS, διευκολύνει το σχεδιασμό μιας ευέλικτης δομής διάταξης που ανταποκρίνεται χωρίς τη χρήση πλωτήρα (float) ή τοποθέτησης (position).



Κεφάλαιο 3 : Κατασκευή Ιστοσελίδων Άμεσης Ανταπόκρισης/Φόρτωσης (RWD)

1. Εισαγωγή στο RWD:

Ο Σχεδιασμός Ιστοσελίδας Άμεσης Ανταπόκρισης/Φόρτωσης (Responsive Web Design) δίνει τη δυνατότητα σχεδιασμού και κατασκευής μιας ευπαρουσίαστης ιστοσελίδας σε όλους τους υφιστάμενους τύπους συσκευών.

Το RWD χρησιμοποιεί μόνο HTML και CSS.

Μέσω του RWD, καθίσταται εφικτή η προβολή μιας ιστοσελίδας σε πολλούς διαφορετικούς τύπους συσκευών όπως οι επιτραπέζιοι υπολογιστές, τα tablet και τα κινητά τηλέφωνα.

Οι ιστοσελίδες δεν θα πρέπει να παραλείπουν πληροφορίες που είναι καταλληλότερες σε μικρότερες συσκευές, αλλά να αναδιαμορφώνουν το περιεχόμενό τους ανάλογα με το τύπο συσκευής που χρησιμοποιεί ο χρήστης ώστε να φέρουν το ίδιο άρτιο αποτέλεσμα και να προσφέρουν την ίδια εύκολη και γρήγορη εμπειρία πλοήγησης:



Η ιδιότητα αποκριτικής ιστοσχεδίασης (Responsive Web Design/RWD) χρησιμοποιείται στους CSS και HTML για να αλλάξετε το μέγεθος, να αποκρύψετε, να συρρικνώσετε, να μεγεθύνετε ή να μετακινήσετε το περιεχόμενο ανάλογα με τις διαφορετικές ανάγκες προβολής που απορρέουν από τους διάφορους τύπους συσκευών.

2. Θύρα προβολής RWD (Viewpoint)

Η θύρα προβολής είναι ο ορατός χώρος μιας ιστοσελίδας από τον χρήστη.

Η θύρα προβολής ποικίλλει ανάλογα με τη συσκευή και θα είναι μικρότερη σε ένα κινητό τηλέφωνο από ό, τι σε μια οθόνη υπολογιστή.

Πριν κυκλοφορήσουν τα tablets και τα κινητά τηλέφωνα, οι ιστοσελίδες σχεδιάζονταν μόνο για οθόνες υπολογιστών και γι' αυτό συνήθως ο σχεδιασμός τους ήταν στατικός και σε σταθερό μέγεθος.

Αργότερα όμως, όταν η πλοήγηση στο διαδίκτυο επεκτάθηκε στις συσκευές tablet και στα κινητά τηλέφωνα, οι ιστοσελίδες σταθερού μεγέθους ήταν πολύ μεγάλες για να χωρέσουν στη θύρα προβολής. Για την επίλυση του εν λόγω προβλήματος, τα προγράμματα περιήγησης στις μικρότερου μεγέθους συσκευές μειώθηκαν για να μπορέσουν να χωρέσουν ολόκληρο το περιεχόμενο μιας ιστοσελίδας στην οθόνη τους.

Το αποτέλεσμα όμως δεν ήταν ακόμη άρτιο, αλλά μάλλον μια γρήγορη επίλυση του προβλήματος!!

Ρύθμιση της θύρας προβολής

Η HTML5 εισήγαγε μια μέθοδο που επιτρέπει στους σχεδιαστές ιστοσελίδων να ελέγχουν τη θύρα προβολής, μέσω της ετικέτας `<meta>`.

Θα πρέπει να συμπεριλάβετε την ακόλουθη ετικέτα θύρα προβολής `<meta>` σε όλες τις ιστοσελίδες σας:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Αυτό δίνει στο πρόγραμμα περιήγησης οδηγίες σχετικά με τον τρόπο ελέγχου των διαστάσεων και αυξομείωσης του μεγέθους μιας σελίδας.

Το τμήμα `width=device-width` ορίζει το πλάτος της ιστοσελίδας ώστε να ακολουθεί το πλάτος της οθόνης μιας συσκευής (το οποίο θα διαφέρει ανάλογα με το μέγεθος της εκάστοτε οθόνης προβολής).

Η αρχική κλίμακα = τμήμα `1.0` ορίζει το αρχικό πεδίο εστίασης όταν μια ιστοσελίδα φορτώνεται σε ένα πρόγραμμα περιήγησης στον ιστό. Πιο κάτω παρατίθενται δυο παραδείγματα ιστοσελίδων, εκ των οποίων το πρώτο αποτελεί παράδειγμα ιστοσελίδας χωρίς κανόνα μετα-ετικέτας θύρας προβολής ενώ το δεύτερο αποτελεί παράδειγμα με κανόνα μετα-ετικέτας θύρας προβολής :



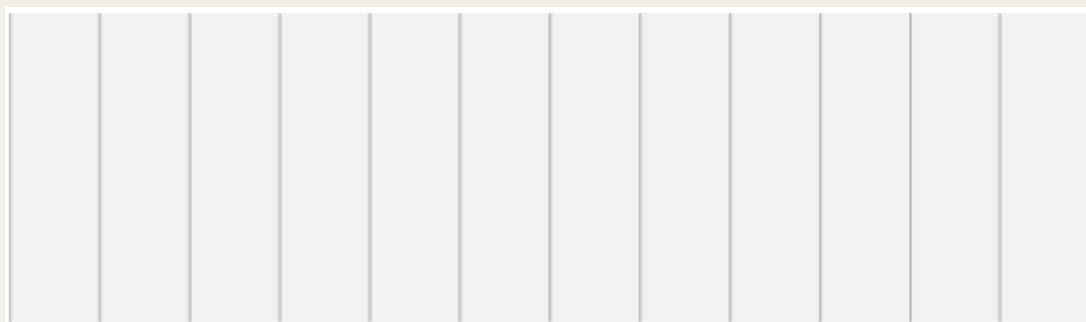
Without the viewport meta tag



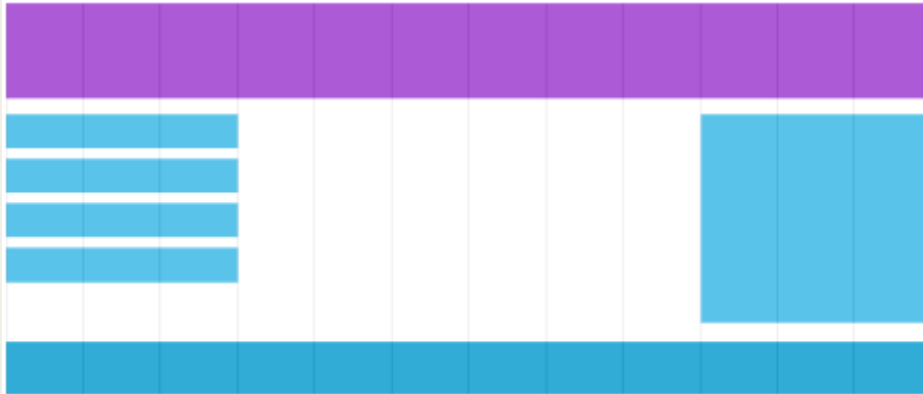
With the viewport meta tag

3. Προβολή πλέγματος RWD (Grid-View)

Πολλές ιστοσελίδες βασίζονται σε μια προβολή πλέγματος, πράγμα που σημαίνει ότι η σελίδα χωρίζεται σε στήλες:



Πολλές ιστοσελίδες βασίζονται σε μια προβολή πλέγματος, πράγμα που σημαίνει ότι η σελίδα χωρίζεται σε στήλες:



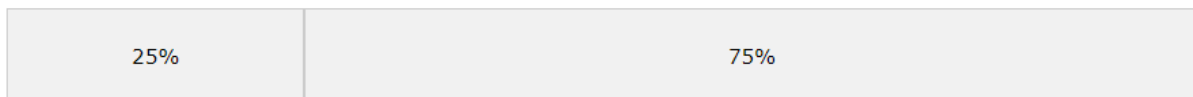
Κατασκευή ενός αποκριτικού πλέγματος-προβολής

Πρώτα βεβαιωθείτε ότι όλα τα στοιχεία HTML έχουν την ιδιότητα μεγέθους πλαισίου που έχει οριστεί στο πλαίσιο περιγράμματος (border-box). Αυτό εξασφαλίζει ότι η επένδυση και το περίγραμμα περιλαμβάνονται στο συνολικό πλάτος και ύψος των στοιχείων.

Προσθέστε τον ακόλουθο κώδικα στην CSS:

```
* {  
  box-sizing: border-box;  
}
```

Το ακόλουθο παράδειγμα δείχνει μια απλή αποκριτική ιστοσχεδίαση, με δύο στήλες:



```
.menu {  
  width: 25%;  
  float: left;  
}  
.main {  
  width: 75%;  
  float: left;  
}
```

4. Ερωτήματα πολυμέσων RWD (Media Queries)

- Ένα ερώτημα πολυμέσων είναι μια τεχνική CSS που εισάχθηκε στο CSS3.
- Χρησιμοποιεί τον κανόνα @media για να συμπεριλάβει ένα μπλοκ ιδιοτήτων CSS μόνο αν μια συγκεκριμένη συνθήκη είναι αληθής.
- Εάν το παράθυρο του προγράμματος περιήγησης είναι 600px ή μικρότερο, το χρώμα φόντου θα είναι γαλάζιο:

```
@media only screen and (max-width: 600px) {  
  body {  
    background-color: lightblue;  
  }  
}
```

Προσθήκη σημείου διακοπής (breakpoint)

Νωρίτερα στον παρόντα οδηγό, κατασκευάσαμε μια ιστοσελίδα με σειρές και στήλες, η οποία ανταποκρινόταν μεν, αλλά δεν φαινόταν ευπαρουσίαστη δε, σε μια μικρή οθόνη προβολής.

Τα ερωτήματα πολυμέσων μπορούν να μας βοηθήσουν σχετικά με αυτό. Μπορούμε να προσθέσουμε ένα σημείο διακοπής όπου ορισμένα μέρη του σχεδιασμού θα συμπεριφέρονται διαφορετικά σε κάθε πλευρά του σημείου διακοπής.



Χρησιμοποιήστε ένα ερώτημα πολυμέσων για να προσθέσετε ένα σημείο διακοπής στο 768px:

Παράδειγμα: Όταν η οθόνη (ένα παράθυρο προγράμματος περιήγησης) γίνεται μικρότερη από 768px, κάθε στήλη πρέπει να έχει πλάτος 100%:

```
/* For desktop: */  
.col-1 {width: 8.33%;}  
.col-2 {width: 16.66%;}  
.col-3 {width: 25%;}  
.col-4 {width: 33.33%;}  
.col-5 {width: 41.66%;}  
.col-6 {width: 50%;}  
.col-7 {width: 58.33%;}  
.col-8 {width: 66.66%;}  
.col-9 {width: 75%;}  
.col-10 {width: 83.33%;}  
.col-11 {width: 91.66%;}  
.col-12 {width: 100%;}  
  
@media only screen and (max-width: 768px) {  
  /* For mobile phones: */  
  [class*="col-"] {  
    width: 100%;  
  }  
}
```

5. Εικόνες RWD

- Εάν η ιδιότητα πλάτους έχει οριστεί σε ένα συγκεκριμένο ποσοστό και η ιδιότητα ύψους έχει οριστεί ως “auto”, η εικόνα θα ανταποκρίνεται και θα κλιμακώνεται προς τα πάνω και προς τα κάτω:

```
img {  
  width: 100%;  
  height: auto;  
}
```


- Εάν η ιδιότητα μέγιστου πλάτους έχει οριστεί στο 100%, η εικόνα θα μειωθεί αν χρειαστεί, αλλά ποτέ δεν θα αυξηθεί για να είναι μεγαλύτερη από το αρχικό της μέγεθος:

```
img {  
  max-width: 100%;  
  height: auto;  
}
```

Σημειώστε ότι στο παραπάνω παράδειγμα, η εικόνα μπορεί να αυξηθεί ώστε να είναι μεγαλύτερη από το αρχικό της μέγεθος. Μια καλύτερη λύση, σε πολλές περιπτώσεις, θα ήταν η χρήση της ιδιότητας μέγιστου πλάτους (max-width).

Χρήση της ιδιότητας μέγιστου πλάτους:

Η ιδιότητα μέγιστου πλάτους έχει οριστεί στο 100%, η εικόνα θα μειωθεί αν χρειαστεί, αλλά ποτέ δεν θα αυξηθεί για να είναι μεγαλύτερη από το αρχικό της μέγεθος:

```
img {  
  max-width: 100%;  
  height: auto;  
}
```

6. Βίντεο RWD

- Εάν η ιδιότητα πλάτους έχει οριστεί στο 100%, το πρόγραμμα αναπαραγωγής βίντεο θα ανταποκρίνεται και θα αυξομειώνεται προς τα πάνω και προς τα κάτω:

```
video {  
  width: 100%;  
  height: auto;  
}
```

- Εάν η ιδιότητα μέγιστου πλάτους έχει οριστεί στο 100%, το πρόγραμμα αναπαραγωγής βίντεο θα μειωθεί αν χρειαστεί, αλλά δεν θα αυξηθεί ποτέ ώστε να είναι μεγαλύτερο από το αρχικό του μέγεθος:

```
video {  
  max-width: 100%;  
  height: auto;  
}
```

Σημειώστε ότι στο παραπάνω παράδειγμα, το πρόγραμμα αναπαραγωγής βίντεο μπορεί να αυξηθεί ώστε να είναι μεγαλύτερο από το αρχικό του μέγεθος. Μια καλύτερη λύση, σε πολλές περιπτώσεις, θα ήταν η χρήση της ιδιότητας μέγιστου πλάτους.

Χρήση της ιδιότητας μέγιστου πλάτους (max-width):

Εάν η ιδιότητα μέγιστου πλάτους έχει οριστεί στο 100%, το πρόγραμμα αναπαραγωγής βίντεο θα μειωθεί αν χρειαστεί, αλλά δεν θα αυξηθεί ποτέ ώστε να είναι μεγαλύτερο από το αρχικό του μέγεθος:

```
video {  
  max-width: 100%;  
  height: auto;  
}
```

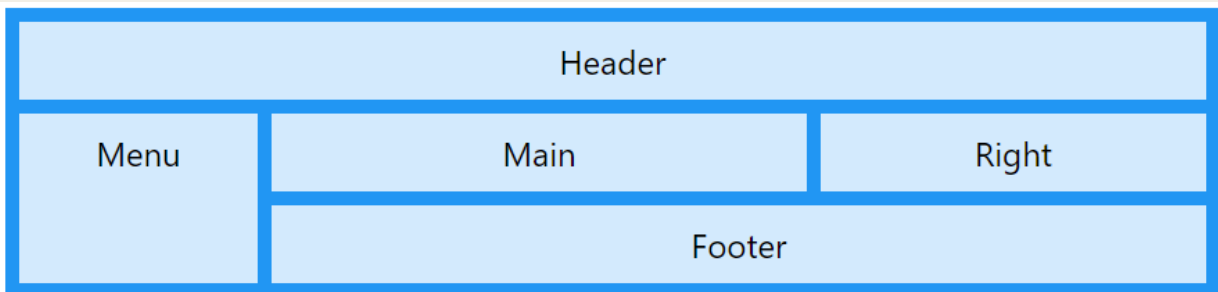
7. Πλαίσια RWD (Frameworks)

- Υπάρχουν πολλά δωρεάν CSS Frameworks που προσφέρουν αποκριτική ιστοσεδίαση (Responsive Design).
- Ένας πολύ καλός τρόπος για να δημιουργήσετε ένα σχεδιασμό ιστοσελίδας που να ανταποκρίνεται, είναι να χρησιμοποιήσετε ένα φύλλο ύφους που επίσης ανταποκρίνεται, όπως το W3.CSS
- Το W3.CSS διευκολύνει την ανάπτυξη ιστότοπων που το περιεχόμενό τους είναι ευπαρουσίαστο σε οποιοδήποτε μέγεθος!
- Ένα άλλο δημοφιλές πλαίσιο είναι το Bootstrap. Χρησιμοποιεί την HTML και την CSS για να αναπτύξει και να σχεδιάσει ιστοσελίδες που ανταποκρίνονται

Κεφάλαιο 4: Πλέγμα CSS (Grid)

1. Εισαγωγή στο CSS Grid (Grid Layout)

- Η ιδιότητα διάταξης πλέγματος CSS προσφέρει ένα σύστημα διάταξης ενός ιστοτόπου με βάση το πλέγμα, τις σειρές και τις στήλες, καθιστώντας έτσι ευκολότερο τον σχεδιασμό και την ανάπτυξη ιστοσελίδων χωρίς να χρειάζεται η χρήση πλωτών και τοποθέτησης στοιχείων.



2. Δοχείο/κοντέινερ Πλέγματος (Grid Container)

- Για να κάνετε ένα στοιχείο HTML να συμπεριφέρεται ως Grid Container, πρέπει να ορίσετε την ιδιότητα προβολής (display) σε μια δομή πλέγματος ή δομή ενσωματωμένου πλέγματος (inline-grid)
- Ένα Grid Container αποτελείται από στοιχεία δομής πλέγματος, τοποθετημένα μέσα σε στήλες και σειρές.

1	2	3	4
5	6	7	8

3. Στοιχείο πλέγματος (grid item)

- Ένα δοχείο πλέγματος περιέχει στοιχεία πλέγματος.
- Από προεπιλογή, ένα δοχείο/κοντέινερ πλέγματος έχει ένα στοιχείο πλέγματος για κάθε στήλη, σε κάθε γραμμή, αλλά μπορείτε να διαμορφώσετε τα στοιχεία πλέγματος έτσι ώστε να καλύπτουν πολλαπλές στήλες ή/και γραμμές.
- Η ιδιότητα στήλης πλέγματος (grid-column) ορίζει σε ποια/ες στήλη(ες) θα τοποθετηθεί/ουν σε ένα στοιχείο.
- Μπορείτε να ορίσετε πού θα ξεκινήσει ένα στοιχείο και πού θα τελειώσει.

1				2	3
4	5	6	7	8	9
10	11	12	13	14	15

Κεφάλαιο 5: Sass

1. Εισαγωγή στο Sass

Τι είναι το Sass;

- Sass σημαίνει Syntactically Awesome Stylesheet
- Το Sass είναι μια επέκταση του CSS
- Το Sass είναι προ-επεξεργαστής του CSS
- Το Sass είναι πλήρως συμβατό με όλες τις εκδόσεις του CSS
- Το Sass μειώνει τις επαναλήψεις του CSS και ως εκ τούτου εξοικονομεί χρόνο
- Το Sass σχεδιάστηκε από τον Hampton Catlin και αναπτύχθηκε από τη Natalie Weizenbaum το 2006
- Το Sass μπορεί να το μεταφορτώσει και να το χρησιμοποιήσει κανείς ελεύθερα

Γιατί να χρησιμοποιήσετε το SASS;

Τα φύλλα ύφους γίνονται διαρκώς μεγαλύτερα, πιο περίπλοκα και πιο δύσκολα μπορούν να διατηρηθούν. Εδώ είναι το σημείο στο οποίο ένας προ-επεξεργαστής CSS μπορεί να βοηθήσει.

Το Sass σας επιτρέπει να χρησιμοποιείτε λειτουργίες που δεν υπάρχουν στο CSS, όπως μεταβλητές, ένθετους κανόνες, mixins, εισαγωγές, inherits, ενσωματωμένες λειτουργίες κτλ.

Ένα απλό παράδειγμα γιατί το Sass είναι χρήσιμο

Ας πούμε ότι έχουμε μια ιστοσελίδα με τρία κύρια χρώματα όπως φαίνονται πιο κάτω:

```
#a2b9bc
```

```
#b2ad7f
```

```
#878f99
```

Πόσες φορές πρέπει να πληκτρολογήσουμε αυτές τις δεκαεξαδικές τιμές; Πολλές φορές. Και πόσες φορές πρέπει να πληκτρολογήσουμε για τις τιμές των αποχρώσεων των ίδιων χρωμάτων;

Αντί λοιπόν να πληκτρολογήσουμε τις προαναφερθέντες τιμές πολλές φορές, μπορούμε να χρησιμοποιήσετε το Sass και να γράψουμε μόνο τον κώδικα πιο κάτω:

```
/* define variables for the primary colors */
$primary_1: #a2b9bc;
$primary_2: #b2ad7f;
$primary_3: #878f99;

/* use the variables */
.main-header {
  background-color: $primary_1;
}

.menu-left {
  background-color: $primary_2;
}

.menu-right {
  background-color: $primary_3;
}
```

Έτσι λοιπόν, όταν χρησιμοποιείτε το Sass, και το κύριο χρώμα αλλάζει, το μόνο που χρειάζεται είναι να το αλλάξετε σε ένα μέρος.

Πώς λειτουργεί το Sass;

Ένα πρόγραμμα περιήγησης δεν καταλαβαίνει τον κώδικα Sass. Ως εκ τούτου, θα χρειαστείτε έναν προ-επεξεργαστή Sass για να μετατρέψετε τον κώδικα Sass σε έναν τυπικό κώδικα τύπου CSS. Αυτή η διαδικασία ονομάζεται μεταγλώττιση. Για τη μεταγλώττιση, θα πρέπει να χρησιμοποιήσετε έναν αναμεταδότη (ένα είδος προγράμματος) στον οποίο θα δώσετε τον κώδικα Sass για να το λάβετε πίσω στη συνέχεια υπό μορφή CSS.

Τύπος αρχείου

Τα αρχεία Sass έχουν την επέκταση αρχείου ".scss".

Σχόλια Sass

Το SASS υποστηρίζει τυπικά σχόλια CSS/* comment */, και επιπλέον υποστηρίζει ενσωματωμένα (inline) σχόλια // comment:

```
/* define primary colors */
$primary_1: #a2b9bc;
$primary_2: #b2ad7f;

/* use the variables */
.main-header {
  background-color: $primary_1; // here you can put an inline comment
}
```

Μεταβλητές Sass

Οι μεταβλητές είναι ένας τρόπος αποθήκευσης πληροφοριών που μπορείτε να χρησιμοποιήσετε αργότερα.

Με το Sass, μπορείτε να αποθηκεύσετε πληροφορίες σε μεταβλητές, όπως:

- ακολουθίες χαρακτήρων
- αριθμοί
- χρώματα
- λογικές τιμές
- λίστες
- μηδενικές τιμές nulls

Το Sass χρησιμοποιεί το σύμβολο \$, ακολουθούμενο από ένα όνομα, για να δηλώσει μεταβλητές:

```
$variablename: value;
```

Στο ακόλουθο παράδειγμα δηλώνονται 4 μεταβλητές που ονομάζονται myFont, myColor, myFontSize και myWidth. Αφού δηλωθούν οι μεταβλητές, μπορείτε να χρησιμοποιήσετε τις μεταβλητές όπου θέλετε:

```
$myFont: Helvetica, sans-serif;
$myColor: red;
$myFontSize: 18px;
$myWidth: 680px;

body {
  font-family: $myFont;
  font-size: $myFontSize;
  color: $myColor;
}

#container {
  width: $myWidth;
}
```

Έτσι, όταν το αρχείο Sass μεταγλωττίζεται, παίρνει τις μεταβλητές (myFont, myColor, κλπ.) και τις εξάγει υπό τη μορφή μεταβλητών τιμών CSS, όπως φαίνονται πιο κάτω:

```
body {  
  font-family: Helvetica, sans-serif;  
  font-size: 18px;  
  color: red;  
}  
  
#container {  
  width: 680px;  
}
```

Πεδίο εφαρμογής μεταβλητών Sass

Οι μεταβλητές Sass είναι διαθέσιμες μόνο στο επίπεδο ένθεσης (nesting) στο οποίο ορίζονται.

Παρατηρήστε το ακόλουθο παράδειγμα:

```
$myColor: red;  
  
h1 {  
  $myColor: green;  
  color: $myColor;  
}  
  
p {  
  color: $myColor;  
}
```

Το χρώμα του κειμένου μέσα σε μια ετικέτα <p> θα είναι κόκκινο ή πράσινο; Θα είναι κόκκινο!

Ο άλλος ορισμός, \$ myColor: green; είναι μέσα στην ετικέτα <h1>, και θα είναι διαθέσιμος μόνο εκεί!

Έτσι, η έξοδος CSS θα είναι:

```
h1 {  
  color: green;  
}  
  
p {  
  color: red;  
}
```

Αυτή είναι λοιπόν η προεπιλεγμένη συμπεριφορά για το πεδίο εφαρμογής μεταβλητών SASS.

Η χρήση της μεταβλητής **Sass !global**

Η προεπιλεγμένη συμπεριφορά για το πεδίο εφαρμογής μεταβλητών μπορεί να παρακαμφθεί χρησιμοποιώντας την μεταβλητή !global.

Η μεταβλητή !global δείχνει ότι μια μεταβλητή είναι καθολική, πράγμα που σημαίνει ότι είναι προσβάσιμη σε όλα τα επίπεδα.

Παρατηρήστε το ακόλουθο παράδειγμα (το ίδιο όπως παραπάνω, αλλά με την προσθήκη μεταβλητής !global):

```
$myColor: red;  
  
h1 {  
  $myColor: green !global;  
  color: $myColor;  
}  
  
p {  
  color: $myColor;  
}
```

Τώρα το χρώμα του κειμένου μέσα σε μια ετικέτα `<p>` θα είναι πράσινο!

Έτσι, η έξοδος CSS θα είναι:

```
h1 {  
  color: green;  
}  
  
p {  
  color: green;  
}
```

Ένθετοι Κανόνες Sass (Nested)

Το Sass σας επιτρέπει να τοποθετήσετε επιλογείς CSS με τον ίδιο τρόπο όπως το HTML.

Παρατηρήστε ένα παράδειγμα κώδικα Sass για την πλοήγηση ενός ιστότοπου:

```
nav {
  ul {
    margin: 0;
    padding: 0;
    list-style: none;
  }
  li {
    display: inline-block;
  }
  a {
    display: block;
    padding: 6px 12px;
    text-decoration: none;
  }
}
```

Παρατηρήστε ότι στο Sass, το ul, το li και οι επιλογείς είναι ένθετα μέσα στον επιλογέα πλοήγησης.

Ενώ στο CSS, οι κανόνες ορίζονται έναν προς έναν (δεν είναι ένθετοι):

```
nav ul {
  margin: 0;
  padding: 0;
  list-style: none;
}
nav li {
  display: inline-block;
}
nav a {
  display: block;
  padding: 6px 12px;
  text-decoration: none;
}
```

Επειδή μπορείτε να μετατρέψετε τις ιδιότητες σε ένθετες στο Sass, είναι πιο εύκολα να διαβαστούν ό,τι οι τυπικές ιδιότητες CSS.

Ένθετες ιδιότητες SASS

Πολλές ιδιότητες CSS έχουν το ίδιο πρόθεμα, όπως font-family, font-size και font-weight ή text-align, text-transform και text-overflow.

Με το Sass μπορείτε να τις γράψετε ως ένθετες ιδιότητες:

```
font: {  
  family: Helvetica, sans-serif;  
  size: 18px;  
  weight: bold;  
}  
  
text: {  
  align: center;  
  transform: lowercase;  
  overflow: hidden;  
}
```

Ο αναμεταδότης Sass θα μετατρέψει τις παραπάνω ιδιότητες σε τυπικές ιδιότητες CSS:

```
font-family: Helvetica, sans-serif;  
font-size: 18px;  
font-weight: bold;  
  
text-align: center;  
text-transform: lowercase;  
text-overflow: hidden;
```

Sass @import and Partials

Το Sass διατηρεί τον κώδικα CSS DRY (ακρώνυμο του «Don't Repeat Yourself/ Να μην επαναληφθεί»). Ένας τρόπος για να γράψετε τον κώδικα DRY είναι να διατηρήσετε τον σχετικό κώδικα σε ξεχωριστά αρχεία.

Μπορείτε να δημιουργήσετε μικρά αρχεία με αποσπάσματα CSS (snippets) για να τα συμπεριλάβετε σε άλλα αρχεία Sass.

Παραδείγματα τέτοιων αρχείων μπορεί να είναι: αρχείο επαναφοράς, μεταβλητές, χρώματα, γραμματοσειρές, μεγέθη γραμματοσειρών κ.λπ.

Εισαγωγή αρχείων SASS

Όπως και το CSS, έτσι και το Sass υποστηρίζει επίσης την οδηγία `@import`.

Η οδηγία `@import` σας επιτρέπει να συμπεριλάβετε το περιεχόμενο ενός αρχείου σε ένα άλλο.

Η οδηγία CSS `@import` παρουσιάζει ένα σημαντικό μειονέκτημα λόγω κυρίως προβλημάτων που αφορούν την αποδοτικότητά τους.

Αυτό γιατί κάθε φορά που την χρησιμοποιείτε δημιουργείται ένα επιπλέον αίτημα HTTP. Ωστόσο, η οδηγία Sass `@import` περιλαμβάνει το αρχείο στο CSS. Επομένως, δεν απαιτείται να χρησιμοποιήσετε ένα αίτημα `http` κατά την εκτέλεση!

```
@import filename;
```

Συμβουλή: Δεν χρειάζεται να καθορίσετε μια επέκταση αρχείου, εφόσον το Sass υποθέτει αυτόματα ότι εννοείτε ένα αρχείο `.sass` ή `.scss`. Μπορείτε επίσης να εισάγετε αρχεία CSS. Η οδηγία `@import` εισάγει το αρχείο και οποιεσδήποτε μεταβλητές ή mixins που ορίζονται στο εισαγόμενο αρχείο και μπορούν στη συνέχεια να χρησιμοποιηθούν στο κύριο αρχείο.

Μπορείτε να εισάγετε όσα αρχεία χρειάζεστε στο κύριο αρχείο:

```
@import "variables";  
@import "colors";  
@import "reset";
```

Ας δούμε ένα παράδειγμα: Ας υποθέσουμε ότι έχουμε ένα αρχείο επαναφοράς (reset file) που ονομάζεται "reset.scss", όπως φαίνεται

πιο κάτω:

```
html,  
body,  
ul,  
ol {  
  margin: 0;  
  padding: 0;  
}
```

και τώρα θέλουμε να εισάγουμε το αρχείο "reset.scss" σε ένα άλλο αρχείο που ονομάζεται "standard.scss".

Να λοιπόν πώς μπορούμε να το κάνουμε: Είναι φυσιολογικό να προσθέτουμε την οδηγία @import στην κορυφή ενός αρχείου. Με αυτόν τον τρόπο, το περιεχόμενό της θα έχει ένα ευρύ πεδίο εφαρμογής:

```
@import "reset";  
  
body {  
  font-family: Helvetica, sans-serif;  
  font-size: 18px;  
  color: red;  
}
```

Έτσι, όταν το αρχείο "standard.css" μεταγλωττίζεται σε αρχείο CSS ο κώδικας θα πρέπει να ακολουθεί την πιο κάτω σύνταξη:

```
html, body, ul, ol {  
  margin: 0;  
  padding: 0;  
}  
  
body {  
  font-family: Helvetica, sans-serif;  
  font-size: 18px;  
  color: red;  
}
```

Sass Partials

Από προεπιλογή, το Sass μεταγλωττίζει όλα τα αρχεία .scss άμεσα. Ωστόσο, όταν θέλετε να εισαγάγετε ένα αρχείο, δεν χρειάζεται το αρχείο να μεταφερθεί απευθείας.

Το Sass έχει ένα μηχανισμό για αυτό: Αν ξεκινήσετε γράφοντας το όνομα ενός αρχείου με μια κάτω παύλα, το Sass δεν θα μπορεί να το μεταγλωττίσει. Τα αρχεία που ονομάζονται με αυτόν τον τρόπο ονομάζονται «αποσπάσματα» (partials) στο Sass.

Έτσι, ένα SASS partial θα πρέπει να υπογραμμίζεται με μια κάτω παύλα:

```
filename;
```

Το ακόλουθο παράδειγμα εμφανίζει ένα απόσπασμα αρχείου Sass που ονομάζεται "_colors.scss". (Αυτό το αρχείο δεν θα μεταγλωττιστεί απευθείας σε "colors.css"):

```
$myPink: #EE82EE;  
$myBlue: #4169E1;  
$myGreen: #8FBC8F;
```

Τώρα, αν εισάγετε το απόσπασμα αρχείου, παραλείψτε τον χαρακτήρα υπογράμμισης της κάτω παύλας. Το Sass κατανοεί ότι θα πρέπει να εισάγει το αρχείο "_colors.scss":

```
@import "colors";  
  
body {  
  font-family: Helvetica, sans-serif;  
  font-size: 18px;  
  color: $myBlue;  
}
```

Οι εντολές @mixin και @include στο Sass

Sass Mixins

Η εντολή @mixin σας επιτρέπει να δημιουργήσετε έναν κώδικα CSS που πρόκειται να επαναχρησιμοποιηθεί σε όλη την ιστοσελίδα.

Η εντολή @include δημιουργήθηκε για να σας επιτρέπει να χρησιμοποιήσετε (include) το mixin.

Ορισμός ενός mixin

Ένα mixin ορίζεται με την εντολή @ mixin.

```
@mixin name {  
  property: value;  
  property: value;  
  ...  
}
```

Στο ακόλουθο παράδειγμα έχουμε τη σύνταξη ενός mixin με το όνομα "important-text":

```
@mixin important-text {  
  color: red;  
  font-size: 25px;  
  font-weight: bold;  
  border: 1px solid blue;  
}
```

Η χρήση ενός mixin

Η οδηγία @include χρησιμοποιείται για να συμπεριλάβει ένα mixin.

```
selector {  
  @include mixin-name;  
}
```

Έτσι λοιπόν, για να συμπεριλάβετε το mixin με το όνομα "important-text" που δημιουργήθηκε παραπάνω καταχωρήστε:

```
.danger {
  @include important-text;
  background-color: green;
}
```

Ο αναμεταδότης Sass θα μεταγλωττίσει των κώδικα SASS σε τυπικό CSS:

```
.danger {
  color: red;
  font-size: 25px;
  font-weight: bold;
  border: 1px solid blue;
  background-color: green;
}
```

Ένα mixin μπορεί επίσης να περιλαμβάνει και άλλα mixins:

```
@mixin special-text {
  @include important-text;
  @include link;
  @include special-border;
}
```

Προσθήκη μεταβλητών σε ένα mixin

Τα mixins μπορούν να λάβουν οδηγίες-εντολές. Με αυτόν τον τρόπο μπορείτε να προσθέσετε μεταβλητές σε ένα mixin.

Παρατηρήστε το πιο κάτω παράδειγμα για να δείτε πώς ορίζεται ένα mixin με εντολές:

```

/* Define mixin with two arguments */
@mixin bordered($color, $width) {
  border: $width solid $color;
}

.myArticle {
  @include bordered(blue, 1px); // Call mixin with two values
}

.myNotes {
  @include bordered(red, 2px); // Call mixin with two values
}

```

Σημειώστε ότι οι εντολές ορίζονται ως μεταβλητές και στη συνέχεια χρησιμοποιούνται ως τιμές (χρώμα και πλάτος) της ιδιότητας των ορίων.

Μετά τη μεταγλώττιση, ο κώδικας CSS θα έχει την εξής μορφή:

```

.myArticle {
  border: 1px solid blue;
}

.myNotes {
  border: 2px solid red;
}

```

Η εντολή @ Extension στο Sass

Η εντολή @extension σας επιτρέπει να μοιράζεστε ένα σύνολο ιδιοτήτων CSS από τον ένα επιλογέα στον άλλο.

Η εντολή @extension είναι χρήσιμη εάν έχετε σχεδόν πανομοιότυπα στοιχεία που διαφέρουν μόνο σε ορισμένες μικρές λεπτομέρειες.

Το ακόλουθο παράδειγμα Sass δημιουργεί πρώτα ένα βασικό στυλ για κουμπιά (αυτό το στυλ θα χρησιμοποιηθεί για τα περισσότερα κουμπιά). Στη συνέχεια, δημιουργούμε ένα στυλ για ένα κουμπί

“Report” και ένα στυλ για ένα κουμπί “Submit”. Τόσο το κουμπί “Report” όσο και το κουμπί “Submit” κληρονομούν όλες τις ιδιότητες CSS από την κλάση .button-basic, μέσω της οδηγίας @extension. Επιπλέον, έχουν τα δικά τους χρώματα που ορίζονται ως εξής:

```
.button-basic {  
  border: none;  
  padding: 15px 30px;  
  text-align: center;  
  font-size: 16px;  
  cursor: pointer;  
}  
  
.button-report {  
  @extend .button-basic;  
  background-color: red;  
}  
  
.button-submit {  
  @extend .button-basic;  
  background-color: green;  
  color: white;  
}
```

Μετά τη μεταγλώττιση, ο κώδικας CSS θα έχει την εξής μορφή:

```
.button-basic, .button-report, .button-submit {
  border: none;
  padding: 15px 30px;
  text-align: center;
  font-size: 16px;
  cursor: pointer;
}

.button-report {
  background-color: red;
}

.button-submit {
  background-color: green;
  color: white;
}
```

Όταν χρησιμοποιείτε την οδηγία @επέκτασης, δεν χρειάζεται να ορίζετε αρκετές κλάσεις για ένα στοιχείο στον κώδικα HTML, όπως την ετικέτα εδώ: `<button class="button-basic button-report">Report this αυτού</button>`. Απλά πρέπει να ορίσετε την ετικέτα `.button-report`. για να λάβετε και τα δύο σύνολα στυλ.

Η οδηγία @ Extension σας επιτρέπει να διατηρείτε τον κώδικά Sass πολύ DRY.