

Τεχνολογία Διαδικτύου

3. CSS

Γρηγόρης Τζιάλλας

Καθηγητής

Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Σχολή Θετικών Επιστημών

Πανεπιστήμιο Θεσσαλίας

CSS – Μορφοποίηση ιστοσελίδων

- CSS : Cascading Style Sheets
 - Χρησιμοποιείται για την μορφοποίηση ιστοσελίδων
 - Αποτελείται από κανόνες οι οποίοι επιλέγουν στοιχεία html και καθορίζουν τις ιδιότητες τους (πχ. χρώμα, γραμματοσειρά, μέγεθος, στοίχιση κ.λπ.)
 - Συνήθως ορίζονται σε ένα ή περισσότερα ξεχωριστά αρχεία με την κατάληξη css.
 - Η χρήση αρχείων css:
 - διαχωρίζει το περιεχόμενο των ιστοσελίδων από την μορφή τους
 - δίνει την δυνατότητα επαναχρησιμοποίησης της μορφής από άλλες ιστοσελίδες και εφαρμογές
 - Μια ιστοσελίδα ή ένα στοιχείο html μπορεί να καθορίζει μορφή, αν και καλό είναι να αποφεύγεται.

Κανόνες CSS

Γενική μορφή κανόνων:

```
selector{  
    property1:value;  
    property2:value;  
    ..  
}
```

Παράδειγμα καθορισμού χρώματος για όλες τις παραγράφους

```
p{  
    color:blue;  
}
```

Μορφοποίηση με αρχείο css

- Αρχείο css (/css/style1.css)

```
p{  
    color:blue;  
}
```

- Σύνδεση αρχείου με ιστοσελίδα:
 - Δήλωση στο τμήμα head μιας ιστοσελίδας ότι χρησιμοποιεί για την μορφοποίηση της το αρχείο css

```
<link rel="stylesheet" href="/css/style1.css">
```

Μορφοποίηση με ετικέτα style στο τμήμα head ή body μιας ιστοσελίδας

```
<head>
  <title>CSS</title>
  <meta charset="UTF-8">
  <link rel="stylesheet" href="/css/style1.css">
  <style>
    h1{
      color:red;
      text-align: center;
    }
  </style>
</head>
```

Μορφοποίηση ενός στοιχείου με χρήση της ιδιότητας style

```
<div style="color:green;">Παράδειγμα μορφοποίησης</div>
```

Παράδειγμα μορφοποίησης

```
<!DOCTYPE html>
<html lang="el">
<head>
  <title>CSS</title>
  <meta charset="UTF-8">
  <link rel="stylesheet" href="/css/style1.css">
  <style>
    h1{
      color:red;
      text-align: center;
    }
  </style>
</head>
<body>
  <h1>CSS</h1>
  <div style="color:green;">Παράδειγμα μορφοποίησης</div>
  <p>Παράγραφος 1</p>
  <p>Παράγραφος 2</p>
</body>
</html>
```

/css/style1.css

```
p {
  color:blue;
}
```

CSS

Παράδειγμα μορφοποίησης

Παράγραφος 1

Παράγραφος 2

Επιλογή στοιχείων html (Selectors)

Συνδυασμός	Παράδειγμα	Περιγραφή
*	*	Επιλογή όλων των στοιχείων
tag	div	Επιλέγει όλα τα στοιχεία με ετικέτα div
#id	#Title	Επιλέγει το στοιχείο id Title
.class	.mainTitle	Επιλέγει όλα τα στοιχεία με κλάση mainTitle

Συνδυασμοί επιλογέων

Συνδυασμός	Παράδειγμα	Περιγραφή
element.class	p.abc	Επιλογή στοιχείων p με κλάση abc
element element	div p	Επιλογή στοιχείων p τα οποία περιέχονται σε στοιχείο div
selector , selector	div, p	Επιλογή στοιχείων div και p

Επιλογή με βάση χαρακτηριστικό

- Με την χρήση αγκυλών μπορούμε να επιλέξουμε στοιχεία με συγκεκριμένες ιδιότητες και τιμές ιδιοτήτων:

tag[attr]	Επιλογή όλων των στοιχείων τα οποία έχουν την ιδιότητα attr
tag[attr="value"]	Επιλογή όλων των στοιχείων τα οποία έχουν την ιδιότητα attr ίση με την τιμή value
tag[attr^="value"]	Επιλογή όλων των στοιχείων των οποίων η τιμή της ιδιότητα attr αρχίζει με value
tag[attr\$="value"]	Επιλογή όλων των στοιχείων των οποίων η τιμή της ιδιότητα attr τελειώνει με value
tag[attr*="value"]	Επιλογή όλων των στοιχείων των οποίων η τιμή της ιδιότητα attr περιέχει το value

Επιλογή με βάση ψευδοστοιχεία

- Ψευδοστοιχεία

::first-letter	Επιλέγει το πρώτο γράμμα
::first-line	Επιλέγει την πρώτη γραμμή
::marker	Επιλέγει τα bullets σε μία λίστα
::selection	Επιλέγει το μέρος ενός στοιχείου που έχει επιλέξει ο χρήστης
::before	Επιλέγει το στοιχείο πριν από κάποιο στοιχείο

Επιλογή με δομικές ψευδοκλάσεις

Ψευδοκλάσεις με βάση το περιεχόμενο (παιδιά / children) στοιχείων

:nth-child(n)	Επιλέγει στοιχεία που είναι n-οστά παιδιά Όπου n αριθμός, λέξη-κλειδί (odd ή even), ή συνάρτηση (π.χ. 2n=>τα πολλαπλάσια του 2)
:nth-last-child(n)	Επιλέγει στοιχεία που είναι n-οστά παιδιά (αντιστρόφως) Όπου n αριθμός, λέξη-κλειδί (odd ή even), ή συνάρτηση (π.χ. 2n=>τα πολλαπλάσια του 2)
::marker	Επιλέγει τα bullets σε μία λίστα
:first-child	Επιλέγει στοιχεία που είναι πρώτα παιδιά
:last-child	Επιλέγει στοιχεία που είναι τελευταία παιδιά
:only-child	Επιλέγει τα μοναχοπαίδια
:root	Επιλέγει τη ρίζα (στην HTML είναι το <html>)
:empty	Επιλέγει τα στοιχεία που είτε δεν έχουν άλλα στοιχεία ως παιδιά, είτε περιέχουν μόνο whitespace (κενά, tabs, κοκ)

Επιλογή με ψευδοκλάσεις συμπεριφοράς

Ψευδοκλάσεις με βάση την συμπεριφορά / τρέχουσα κατάσταση των στοιχείων

:hover	Επιλέγει τα στοιχεία που δείχνει ο δείκτης του ποντικιού
:visited	Επιλέγει τους συνδέσμους που έχει επισκεφθεί ο χρήστης
:active	Επιλέγει τα ενεργά στοιχεία
:focus	Επιλέγει το στοιχείο που έχει τον έλεγχο (focus)
:required	Επιλέγει τα απαιτούμενα πεδία μιας φόρμας
:read-only	Επιλέγει τα πεδία που είναι μόνο για ανάγνωση

Παραδείγματα επιλογέων και συνδυασμών τους

h1

a:link

.manythings

#onething

*

.box p

.box p:first-child

h1, h2, .intro

Προτεραιότητα κανόνων (Specificity)

- Είναι δυνατόν να καθορίζονται περισσότεροι από ένας κανόνες για την μορφοποίηση ενός στοιχείου. Για αυτό άλλωστε δόθηκε και το όνομα Cascading Style Sheets.
- Οι κανόνες στοιχείων html που ορίζονται με την ιδιότητα style (inline) επικρατούν κανόνων που ορίζονται σε ξεχωριστό τμήμα style (internal) ή σε ξεχωριστά αρχεία css (external).
- Για τον ίδιο τύπο κανόνων (inline, internal ή external), επικρατεί ο πιο εξειδικευμένος κανόνας
(π.χ. `p.red{color:red}` επικρατεί του `p{color:black}`)
- Αν οι κανόνες έχουν τον ίδιο τύπο και ίδια εξειδίκευση, επικρατεί αυτός που ορίζεται τελευταίος.

Υπολογισμός προτεραιότητας κανόνων

(from https://www.w3schools.com/css/css_specificity.asp)

- Inline style gets a specificity value of 1000, and is always given the highest priority!
- Start at 0, add 100 for each ID value, add 10 for each class value (or pseudo-class or attribute selector), add 1 for each element selector or pseudo-element.
- There is one exception to this rule: if you use the !important rule, it will even override inline styles!

```
p {  
  background-color red !important  
}
```

Inline στυλ μετράει για 1000

Επιλογέας ID μετράει για 100

Επιλογέας κλάσης, ψευδο-κλάσης και χαρακτηριστικού μετράει για 10

Επιλογέας τύπου και ψευδο-στοιχεία μετράνε για 1

Ο καθολικός επιλογέας μετράει για 0

Παράδειγμα υπολογισμού προτεραιότητας μορφοποίησης

Επιλογέας	Προτεραιότητα	Υπολογισμός
p	1	1
p.test	11	1 + 10
p#demo	101	1 + 100
<p style="color: pink;">	1000	1000
#demo	100	100
.test	10	10
p.test1.test2	21	1 + 10 + 10
#navbar p#demo	201	100 + 1 + 100
*	0	0 (the universal selector is ignored)

Κληρονομικότητα CSS

- Στην HTML, κάθε στοιχείο θεωρείται απόγονος (descendant) κάποιου άλλου στοιχείου, επιτρέποντάς το να κληρονομεί ιδιότητες από το στοιχείο-γονέα
- Ο κάθε απόγονος μπορεί να έχει και δικούς του απογόνους, εκτός και αν πρόκειται για στοιχεία με κενές ετικέτες, όπως το `img`
- Η ιεραρχική δομή των απογόνων μιας ιστοσελίδας καλείται DOM Tree (Document Object Model tree)
- Ο απόγονος ενός στοιχείου κληρονομεί τις ιδιότητες του γονικού στοιχείου του
- Ο απόγονος μπορεί να επανακαθορίσει κάποια ιδιότητα που κληρονομεί.

Παράδειγμα κληρονομικότητας

```
body {  
    color: whitesmoke;  
    background-color: gray;  
}  
h1 {  
    color: lightskyblue;  
    text-align: center;  
}  
div {  
    color: black;  
    background-color: lightgray;  
}
```

```
<body>  
  <h1>Κληρονομικότητα CSS</h1>  
  <div>Άρθρο 1  
    <p>Παράγραφος 1</p>  
    <p>Παράγραφος 2</p>  
  </div>  
  <p>Παράγραφος 3</p>  
</body>
```

Κληρονομικότητα CSS

Άρθρο 1

Παράγραφος 1

Παράγραφος 2

Παράγραφος 3

Μορφοποίηση κειμένου (1)

- color: καθορίζει το χρώμα της γραμματοσειράς.
- background-color: καθορίζει το χρώμα υπόβαθρου
- font-size: καθορίζει το μέγεθος της γραμματοσειράς.
- font-family: καθορίζει την επιθυμητή γραμματοσειρά για το κείμενο, καθώς και εναλλακτικές επιλογές εάν η γραμματοσειρά που ορίστηκε δεν υπάρχει

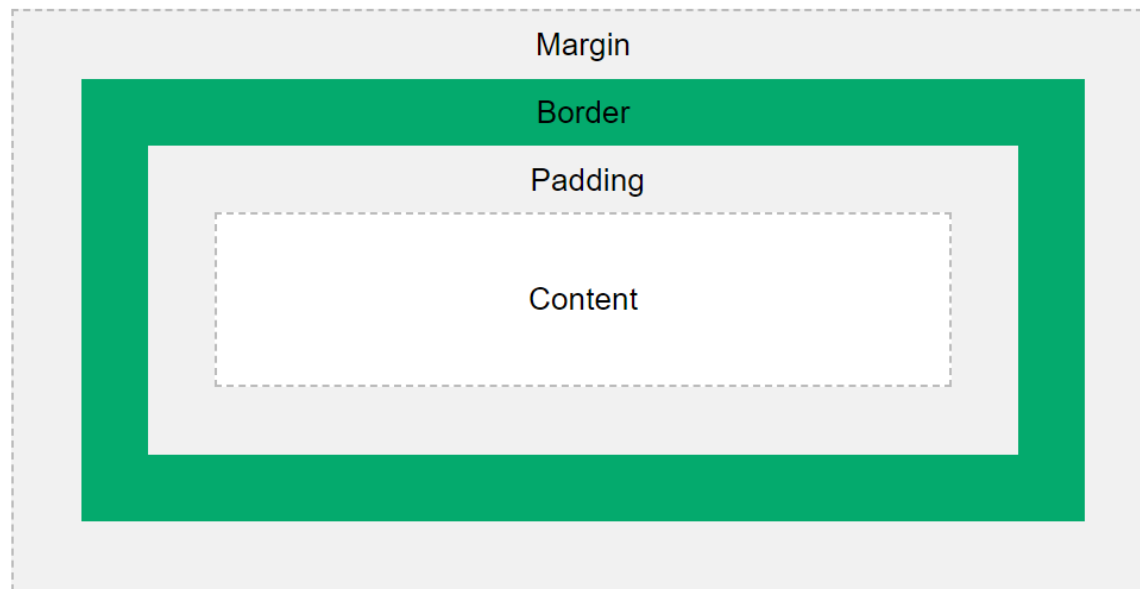
```
body {  
    font-family: 'Times New Roman', Times, serif;  
}  
h1, h2, h3 {  
    font-family: 'Franklin Gothic Medium', 'Arial Narrow', Arial, sans-serif;  
}  
div, p {  
    font-family: Georgia, 'Times New Roman', Times, serif;  
}
```

Μορφοποίηση κειμένου (2)

- font-style: normal, italic, oblique
- text-align: left, right, center, justified
- text-decoration: π.χ. underline red double
- text-transform: uppercase, lowercase, capitalize
- text-indent: Ορίζει την εσοχή της πρώτης γραμμής μιας παραγράφου
- text-shadow: π.χ. 4px 4px 5px red (horizontal offset, horizontal offset, blur radius, color)
- letter-spacing: π.χ. 5px;
- line-height:
- word-spacing:
- white-space: nowrap απενεργοποιεί την αναδίπλωση κειμένου

CSS Box Model

- Κάθε στοιχείο html, είτε είναι block είτε είναι inline, χρησιμοποιεί τις παρακάτω περιοχές:
 - content: Το περιεχόμενο του στοιχείου
 - padding: Περιοχή από το περιεχόμενο έως το περίγραμμα
 - border: Το περίγραμμα του στοιχείου
 - margin: Το περιθώριο (απόσταση από άλλα στοιχεία)

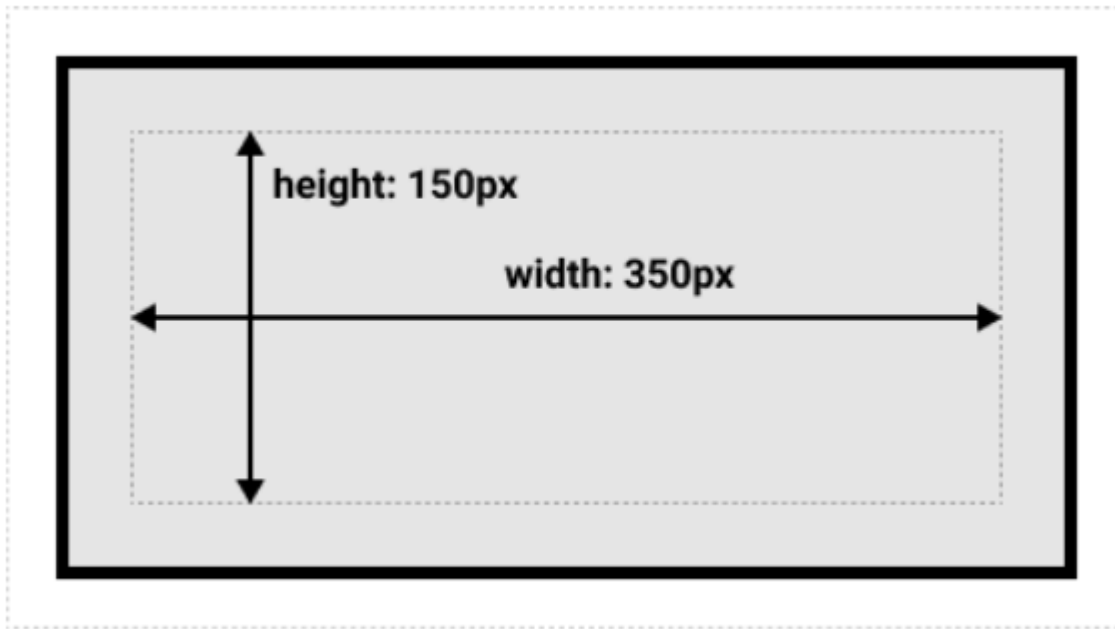


Πλάτος και ύψος στοιχείων html

- width, height: καθορίζουν το πλάτος και ύψος του περιεχομένου (content) ενός στοιχείου
- Στο τελικό ύψος και πλάτος πρέπει να προστεθούν και οι άλλες περιοχές (padding, border και margin)
- max-width, min-width, max-height, min-height: καθορίζουν το μέγιστο και ελάχιστο πλάτος και ύψος του περιεχομένου (content) ενός στοιχείου
- box-sizing: border-box
 - καθορίζει ότι το πλάτος και το ύψος ενός στοιχείου συμπεριλάβουν το περίγραμμα και το περιθώριο από το περίγραμμα

Παράδειγμα ιδιότητας box-sizing

Για ορισμό πλάτους 350 px και ύψους 150px αναλόγως την τιμή της ιδιότητας box-sizing προκύπτουν τα παρακάτω τελικά μεγέθη στοιχείων html



box-sizing: content-box



box-sizing: border-box

Απόλυτες τιμές μήκους

https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/Values_and_units

Unit	Name	Equivalent to
cm	Centimeters	1cm = 37.8px = 25.2/64in
mm	Millimeters	1mm = 1/10th of 1cm
Q	Quarter-millimeters	1Q = 1/40th of 1cm
in	Inches	1in = 2.54cm = 96px
pc	Picas	1pc = 1/6th of 1in
pt	Points	1pt = 1/72nd of 1in
px	Pixels	1px = 1/96th of 1in

Σχετικές τιμές μήκους (1)

https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/Values_and_units

Unit	Relative to
em	Font size of the parent, in the case of typographical properties like font-size , and font size of the element itself, in the case of other properties like width .
ex	x-height of the element's font.
ch	The advance measure (width) of the glyph "0" of the element's font.
rem	Font size of the root element.
lh	Line height of the element.
rlh	Line height of the root element. When used on the font-size or line-height properties of the root element, it refers to the properties' initial value.
vw	1% of the viewport's width.
vh	1% of the viewport's height.

Σχετικές τιμές μήκους (2)

https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/Values_and_units

Unit	Relative to
vmin	1% of the viewport's smaller dimension.
vmax	1% of the viewport's larger dimension.
vb	1% of the size of the initial containing block in the direction of the root element's block axis .
vi	1% of the size of the initial containing block in the direction of the root element's inline axis .
svw, svh	1% of the small viewport 's width and height, respectively.
lvw, lvh	1% of the large viewport 's width and height, respectively.
dvw, dvh	1% of the dynamic viewport 's width and height, respectively.

Καθορισμός τιμών με ποσοστά

- Στην πλειοψηφία των περιπτώσεων ένα ποσοστό αντιμετωπίζεται ως μία σχετική τιμή μήκους.
- Αν ένα ποσοστό χρησιμοποιείται για τον καθορισμό του πλάτους μιας παραγράφου, το ποσοστό αναφέρεται στο συνολικό πλάτος της σελίδας.
- Αν το ποσοστό καθορίζει το μέγεθος μιας γραμματοσειράς, το ποσοστό αναφέρεται στο μέγεθος της γονικής γραμματοσειράς.

Παράδειγμα καθορισμού τιμών με ποσοστά

```
.div1 {  
    border: 2px solid black;  
}  
.div2 {  
    border: 2px solid black;  
    width: 50%;  
    font-size: 80%;  
}
```

```
<body>  
  <h2>Percentages</h2>  
  <div class="div1">Παράγραφος 1</div>  
  <p></p>  
  <div class="div2">Παράγραφος 2</div>  
</body>
```

Percentages

Παράγραφος 1

Παράγραφος 2

Μορφοποίηση περιγράμματος

- border-width: π.χ. 5px
- border-style: solid, dashed, dotted, double, groove, inset, outset, none..
- border-color: π.χ. black
- border-radius: χρησιμοποιείται για στρογγυλοποίηση των γωνιών του περιγράμματος (rounded borders)

Τύποι περιγραμμάτων

```
p.dotted {border-style: dotted;}  
p.dashed {border-style: dashed;}  
p.solid {border-style: solid;}  
p.double {border-style: double;}  
p.groove {border-style: groove;}  
p.ridge {border-style: ridge;}  
p.inset {border-style: inset;}  
p.outset {border-style: outset;}  
p.none {border-style: none;}  
p.hidden {border-style: hidden;}  
p.mix {border-style: dotted dashed solid double;}
```

```
<p class="dotted">A dotted border.</p>  
<p class="dashed">A dashed border.</p>  
<p class="solid">A solid border.</p>  
<p class="double">A double border.</p>  
<p class="groove">A groove border.</p>  
<p class="ridge">A ridge border.</p>  
<p class="inset">An inset border.</p>  
<p class="outset">An outset border.</p>  
<p class="none">No border.</p>  
<p class="hidden">A hidden border.</p>  
<p class="mix">A mixed border.</p>
```

A dotted border.

A dashed border.

A solid border.

A double border.

A groove border.

A ridge border.

An inset border.

An outset border.

No border.

A hidden border.

A mixed border.

Ιδιότητες για τον καθορισμό της μορφής των πλευρών περιγράμματος

- border-top-width
- border-top-style
- border-top-color
- border-right-width
- border-right-style
- border-right-color
- border-bottom-width
- border-bottom-style
- border-bottom-color
- border-left-width
- border-left-style
- border-left-color

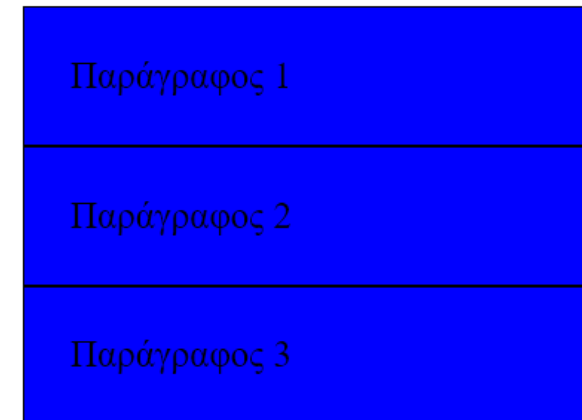
Padding – Καθορισμός απόστασης από περίγραμμα

- padding-top, padding-bottom, padding-left, padding-right

ή εναλλακτικά σε μία γραμμή:

- padding: 25px 50px 75px 100px;
 - top padding is 25px
 - right padding is 50px
 - bottom padding is 75px
 - left padding is 100px
- padding: 20px ή padding: 10px 20px

Padding



```
div {  
  width: fit-content;  
  border: 1px solid black;  
  padding: 20px 20px 20px 20px;  
  background-color: blue;  
}
```

```
<body>  
  <h2>Padding</h2>  
  <div>Παράγραφος 1</div>  
  <div>Παράγραφος 2</div>  
  <div>Παράγραφος 3</div>  
</body>
```

Margins – Καθορισμός περιθωρίων

- margin-top, margin-bottom, margin-left, margin-right

ή εναλλακτικά σε μία γραμμή:

- margin: 25px 50px 75px 100px;
 - top margin is 25px
 - right margin is 50px
 - bottom margin is 75px
 - left margin is 100px

```
div {  
    border: 1px solid black;  
    margin: 25px 50px 75px 100px;  
    background-color: blue;  
}
```

- margin: auto
 - Τοποθετεί αυτόματα τα κατάλληλα περιθώρια ώστε να στοιχίσει οριζόντια στο κέντρο το στοιχείο

Καθορισμός ιδιοτήτων Outline

- outline-width, outline-style, outline-color,
ή εναλλακτικά:
- outline: 5px solid red
- outline-width values:
 - thin (1px)
 - medium (3px)
 - thick (5px)
 - Συγκεκριμένη τιμή (σε px, pt, cm, em, κ.λπ.)

Μορφοποίηση συνδέσμων

Παράδειγμα μορφοποίησης συνδέσμων με χρήση ψευδοκλάσεων συμπεριφοράς

```
a:link {  
    color: blue;  
}  
a:visited {  
    color: green;  
}  
  
a:hover {  
    color: red;  
    font-size: 110%;  
}
```

Ιδιότητες για την μορφοποίηση λιστών

- list-style-type: circle, square, upper-roman, lower-alpha, ..
- list-style-image: url('sqpurple.gif'); (χρήση εικόνας)
- list-style-position: outside, inside

```
ol{
  list-style-type: lower-greek;
  list-style-position: inside;
}
ul{
  list-style-type: circle;
  list-style-position: inside;
}
```

```
<body>
  <h1>Lists</h1>
  <div>
    <ol>
      <li>
        item one
      </li>
      <li>
        item two
      </li>
    </ol>
    <ul>
      <li>
        item one
      </li>
      <li>
        item two
      </li>
    </ul>
  </div>
</body>
```

Lists

α. item one
β. item two

○ item one
○ item two

Ιδιότητες για την μορφοποίηση πινάκων

- `border-spacing`: καθορίζει την απόσταση μεταξύ κελιών
- `border-collapse: collapse` (καθορίζει ότι τα κελιά έχουν κοινό περίγραμμα)
- `caption-side`: `top`, `bottom`, `block-start`, `block-end`, `inline-start`, `inline-end` (καθορίζει την πλευρά που εμφανίζεται η λεζάντα του πίνακα)
- `vertical-align`: καθορίζει την κάθετη στοίχιση των κελιών.
Εφαρμόζεται σε στοιχεία `tr`, `th`, `td`

Παράδειγμα μορφοποίησης πίνακα

```
table, tr, td, th {
  border: 1px solid black;
  border-collapse: collapse;
  padding: 5px;
}
table{
  width: 100%;
}
th {
  height: 30px;
  background-color: gray;
  color: white;
  text-align: center;
}
tr{
  text-align: right;
  height: 50px;
}
tr:nth-child(even) {
  background-color: lightgray;
}
tr:hover {
  background-color: lightblue;
}
caption {
  caption-side: bottom;
}
.tableContainer {
  overflow-x:auto;
  align-content: center;
}
#Column1{
  width:30%;
}
```

```
<div class="tableContainer">
  <table>
    <caption>Table 1</caption>
    <tr>
      <th id="Column1">AA</th>
      <th>Description</th>
    </tr>
    <tr>
      <td>1</td>
      <td>Item 1</td>
    </tr>
    <tr>
      <td>2</td>
      <td>Item 2</td>
    </tr>
    <tr>
      <td>3</td>
      <td>Item 3</td>
    </tr>
    <tr>
      <td>4</td>
      <td>Item 4</td>
    </tr>
  </table>
</div>
```

AA	Description
1	Item 1
2	Item 2
3	Item 3
4	Item 4

Table 1

overflow-x: καθορίζει αν θα αποκοπεί σε περίπτωση υπερχείλισης το περιεχόμενο, αν θα εμφανισθεί ή θα προστεθεί μπάρα κύλισης (scroll bar).

Ιδιότητες για την διάταξη στοιχείων

- **display:** block, inline, inline-block, none ..
 - Καθορίζει πως θα εμφανισθεί ένα στοιχείο html. Με την ιδιότητα αυτή μπορούμε να ορίσουμε εάν ένα στοιχείο εμφανίζεται ως block, inline, inline-block ή καθόλου
- **visibility:** hidden
 - Αποκρύπτει ένα στοιχείο html, όπως και το display: none, αλλά δεν ελευθερώνει το χώρο που καταλαμβάνεται από το στοιχείο
- **position:** static, relative, fixed, absolute, sticky
 - static: Προκαθορισμένη θέση η οποία ακολουθεί την φυσική ροή
 - relative: σε σχετική θέση από την κανονική του θέση του η οποία καθορίζεται με τις ιδιότητες top, bottom, left και right
 - fixed: σε απόλυτη θέση σε σχέση με το παράθυρο του φυλλομετρητή (viewport)
 - absolute: σε απόλυτη θέση σε σχέση με το γονικό του στοιχείο
 - sticky: παραμένει στην ίδια θέση σε περίπτωση χρήσης της μπάρας κύλισης
- **z-index:** Καθορίζει την σειρά εμφάνισης σε επικαλυπτόμενα στοιχεία. Το στοιχείο με μεγαλύτερο z-index υπερκαλύπτει τα άλλα στοιχεία

Ιδιότητα display

- `display: block`
 - Εμφανίζει ένα στοιχείο ως block (παράγραφος που καταλαμβάνει όλο το πλάτος)
- `display: inline`
 - Εμφανίζει ένα στοιχείο ως inline. Το στοιχείο καταλαμβάνει μέρος του πλάτους της οθόνης και οι ιδιότητες που αφορούν το πλάτος και το ύψος του αγνοούνται
- `display: inline-block`
 - Εμφανίζει ένα στοιχείο ως inline με την διαφορά από το `display: inline` ότι οι ιδιότητες που αφορούν το πλάτος και το ύψος μπορούν να χρησιμοποιηθούν

Παράδειγμα display inline και inline-block

```
.inline{
  display: inline;
  width: 100px;
  height: 60px;
  border: 1px solid black;
}
.inlineBlock{
  display: inline-block;
  width: 100px;
  height: 60px;
  border: 1px solid red;
}
</style>
```

Inline Block

Παράγραφος με στοιχείο Inline

Παράγραφος με στοιχείο Inline Block

```
<h1>Inline Block</h1>
<div>Παράγραφος με στοιχείο
  <div class="inline">Inline</div>
</div>
<div>Παράγραφος με στοιχείο
  <div class="inlineBlock">Inline Block</div>
</div>
```


Παράδειγμα positioning

```
.posStatic, .posRelative, .posFixed,  
.posSticky, .posAbsolute{  
  border: 1px solid black;  
  background-color: darkgray;  
  width: 300px;  
}  
.posRelative{  
  position: relative;  
  left: 30px;  
}  
.posFixed{  
  position: fixed;  
  top: 0px;  
  right: 5px;  
}  
.posSticky{  
  position: sticky;  
  top: 30px;  
}  
.posAbsolute{  
  position: absolute;  
  top: 50px;  
  left: 20px;  
}  
.posContainer{  
  position: relative;  
  border: 1px solid red;  
  background-color: lightgray;  
  width: 400px;  
  height: 600px;  
  z-index: -100;  
}
```

```
<h2>Positioning</h2>  
<div class="posStatic">Static Position (Default)</div>  
<div class="posRelative">Relative Position</div>  
<div class="posFixed">Fixed Position</div>  
<div class="posSticky">Sticky Position</div>  
<div class="posContainer">  
  <div class="posAbsolute">Absolute Position</div>  
</div>
```

Positioning

Static Position (Default)
Relative Position
Sticky Position

Absolute Position

Overflow

- overflow: visible, hidden, scroll, auto
 - Καθορίζει τι γίνεται σε περίπτωση που το μέγεθος ενός στοιχείου δεν αρκεί για την εμφάνιση του περιεχομένου του
 - visible – ορατό, hidden – αποκρύπτεται, clip – αποκόπτεται, scroll – εμφανίζεται μπάρα κύλισης – auto – εμφανίζεται μπάρα κύλισης όταν χρειάζεται
- overflow-x: ιδιότητα overflow για το πλάτος (οριζόντια)
- overflow-y: ιδιότητα overflow για το ύψος (κάθετα)
- overflow-wrap: normal, break-word, anywhere
 - καθορίζει την αναδίπλωση σε περίπτωση υπερχείλισης

Παράδειγμα overflow

```
.overVisible, .overClip, .overHidden,  
.overScroll, .overAuto{  
  border: 1px solid black;  
  background-color: darkgray;  
  width: 60px;  
  height: 50px;  
  margin: 30px;  
}  
.overVisible{ overflow: visible;}  
.overHidden{ overflow: hidden;}  
.overClip{ overflow: clip;}  
.overScroll{ overflow: scroll;}  
.overAuto{ overflow: auto;}
```

```
<h2>Overflow</h2>  
<div class="overVisible">Visible One Two Three Four</div>  
<div class="overHidden">Hidden One Two Three Four</div>  
<div class="overClip">Clip One Two Three Four</div>  
<div class="overScroll">Scroll One Two Three Four</div>  
<div class="overAuto">Auto One Two Three Four</div>
```

Overflow

Visible
One Two
Three

Four

Hidden
One Two
Three

Clip One
Two
Three

Scroll
One

Auto
One
Two

Παράδειγμα Top Menu με Display inline

```
.menu {  
    background-color: lightgray;  
    list-style-type: none;  
    text-align: center;  
    margin: 0;  
    padding: 0;  
}  
.menu li {  
    display: inline-block;  
    font-size: 16px;  
    padding: 8px;  
}  
.menu li a {  
    display: block;  
    color: #000;  
    padding: 4px;  
    text-decoration: none;  
}  
.menu li a:hover {  
    background-color: #555;  
    color: white;  
}
```

```
<h2>Inline menu</h2>  
<ul class="menu">  
    <li><a href="#home">Αρχή</a></li>  
    <li><a href="#about">Σχετικά</a></li>  
    <li><a href="#project">Εφαρμογές</a></li>  
    <li><a href="#contact">Επικοινωνία</a></li>  
</ul>
```

Top menu

Αρχή Σχετικά Εφαρμογές Επικοινωνία

Παράδειγμα πλευρικού μενού

```
ul.menu {  
  background-color: lightgrey;  
  list-style-type: none;  
  position: fixed;  
  width: 150px;  
  height: 100%;  
  margin: 0;  
  padding: 0;  
}  
ul.menu li {  
  display: block;  
  color: black;  
  padding: 4px;  
}  
li a {  
  display: block;  
  color: #000;  
  padding: 8px 16px;  
  text-decoration: none;  
}  
li a:hover {  
  background-color: #555;  
  color: white;  
}  
.main {  
  margin-left: 150px;  
  padding: 10px;  
  height: 1000px;  
}
```

```
<ul class="menu">  
  <li><a href="#home">  
    Αρχή</a></li>  
  <li><a href="#about">  
    Σχετικά</a></li>  
  <li><a href="#projects">  
    Εφαρμογές</a></li>  
  <li><a href="#contact">  
    Επικοινωνία</a></li>  
</ul>  
  
<div id="home" class="main">  
  <h2>Αρχή</h2>  
  <p>..</p>  
</div>  
<div id="about" class="main">  
  <h2>Σχετικά</h2>  
  <p>..</p>  
</div>  
<div id="projects" class="main">  
  <h2>Εφαρμογές</h2>  
  <p>..</p>  
</div>  
<div id="contact" class="main">  
  <h2>Επικοινωνία</h2>  
  <p>..</p>  
</div>
```

Αρχή

Σχετικά

Εφαρμογές

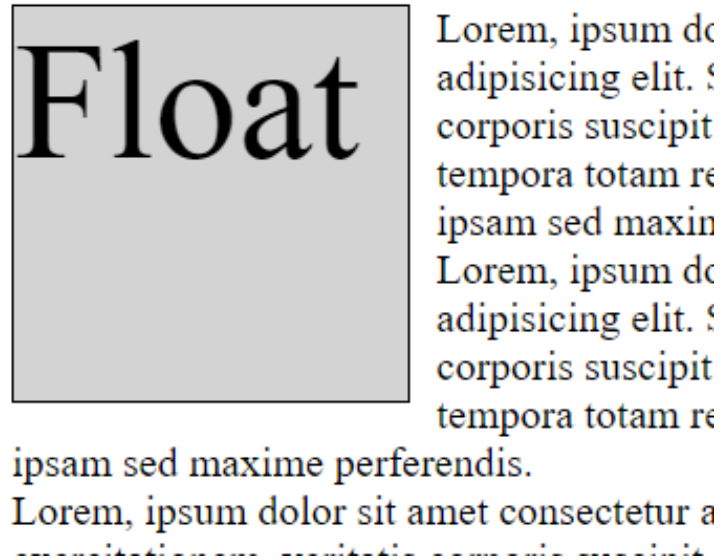
Επικοινωνία

Αρχή

..

Ιδιότητες float clear

- float: left, right, ..
 - Αλλάζει την θέση που εμφανίζεται ένα στοιχείο. Το στοιχείο εμφανίζεται αριστερά ή δεξιά της γραμμής ενώ το υπόλοιπα στοιχεία «ρέουν» γύρω του σύμφωνα με την φυσική ροή εμφάνισης (όπως μια εικόνα και το κείμενο δίπλα της)
 - Παράδειγμα float: left;

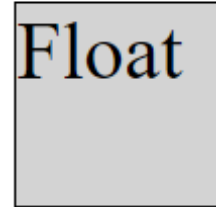


- clear: both, left, right, ..
 - Δίνει την δυνατότητα σε ένα στοιχείο να μην επιτρέπει άλλα στοιχεία με ιδιότητα float να εμφανίζονται δίπλα του ή αριστερά ή δεξιά του

Παράδειγμα float

```
.float {  
  float: left;  
  width: 80px;  
  height: 80px;  
  border: 1px solid black;  
  margin-right: 10px;  
  margin-bottom: 5px;  
  font-size: 32px;  
  background-color: lightgray;  
}
```

Float



Lorem, ipsum dolor sit amet consectetur adipisicing elit. Sit, eius exercitationem, veritatis corporis suscipit deleniti animi placeat nam alias, tempora totam rerum culpa quam distinctio! Iusto ipsam sed maxime perferendis.

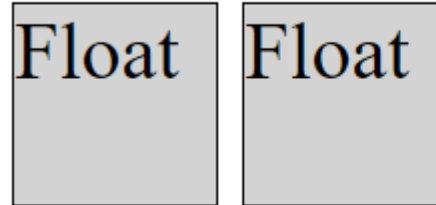
Lorem, ipsum dolor sit amet consectetur adipisicing elit. Sit, eius exercitationem, veritatis corporis suscipit deleniti animi placeat nam alias, tempora totam rerum culpa quam distinctio! Iusto ipsam sed maxime perferendis.

```
<h2>Float</h2>  
<div class="float">Float</div>  
<div>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Sit, eius  
  exercitationem, veritatis corporis suscipit deleniti animi placeat nam alias,  
  tempora totam rerum culpa quam distinctio! Iusto ipsam sed maxime perferendis.  
</div>  
<div>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Sit, eius  
  exercitationem, veritatis corporis suscipit deleniti animi placeat nam alias,  
  tempora totam rerum culpa quam distinctio! Iusto ipsam sed maxime perferendis.  
</div>
```

Παράδειγμα float με πολλαπλά στοιχεία

Float

```
.float {  
  float: left;  
  width: 80px;  
  height: 80px;  
  border: 1px solid black;  
  margin-right: 10px;  
  margin-bottom: 5px;  
  font-size: 32px;  
  background-color: lightgray;  
}
```



Lorem, ipsum dolor sit amet consectetur adipisicing elit. Sit, eius exercitationem, veritatis corporis suscipit deleniti animi placeat nam alias, tempora totam rerum culpa quam distinctio! Iusto ipsam sed maxime

preferendis.

Lorem, ipsum dolor sit amet consectetur adipisicing elit. Sit, eius exercitationem, veritatis corporis suscipit deleniti animi placeat nam alias, tempora totam rerum culpa quam distinctio! Iusto ipsam sed maxime preferendis.

```
<h2>Float</h2>  
<div class="float">Float</div>  
<div class="float">Float</div>  
<div>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Sit, eius  
  exercitationem, veritatis corporis suscipit deleniti animi placeat nam alias,  
  tempora totam rerum culpa quam distinctio! Iusto ipsam sed maxime preferendis.  
</div>  
<div>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Sit, eius  
  exercitationem, veritatis corporis suscipit deleniti animi placeat nam alias,  
  tempora totam rerum culpa quam distinctio! Iusto ipsam sed maxime preferendis.  
</div>
```


Παράδειγμα float και clear

```
.float {  
  float: left;  
  width: 80px;  
  height: 80px;  
  border: 1px solid black;  
  margin-right: 10px;  
  margin-bottom: 5px;  
  font-size: 32px;  
  background-color: lightgray;  
  clear:left;  
}
```

Float

Float

Lorem, ipsum dolor sit amet consectetur adipisicing elit. Sit, eius exercitationem, veritatis corporis suscipit deleniti animi placeat nam alias, tempora totam rerum culpa quam distinctio! Iusto ipsam sed maxime preferendis.

Float

Lorem, ipsum dolor sit amet consectetur adipisicing elit. Sit, eius exercitationem, veritatis corporis suscipit deleniti animi placeat nam alias, tempora totam rerum culpa quam distinctio! Iusto ipsam sed maxime preferendis.

```
<h2>Float</h2>  
<div class="float">Float</div>  
<div class="float">Float</div>  
<div>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Sit, eius  
  exercitationem, veritatis corporis suscipit deleniti animi placeat nam alias,  
  tempora totam rerum culpa quam distinctio! Iusto ipsam sed maxime preferendis.  
</div>  
<div>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Sit, eius  
  exercitationem, veritatis corporis suscipit deleniti animi placeat nam alias,  
  tempora totam rerum culpa quam distinctio! Iusto ipsam sed maxime preferendis.  
</div>
```

CSS FlexBox

- Πριν την εισαγωγή του FlexBox, η τοποθέτηση στοιχείων χρησιμοποιούσε blocks, inlines για κείμενο, tables και την ιδιότητα position
- Ένα στοιχείο html ορίζεται ως FlexBox με την ιδιότητα display: flex
- Τα περιεχόμενα του FlexBox εμφανίζονται ως flex-items και η τοποθέτηση τους γίνεται με την λογική του flexbox και όχι της φυσικής ροής ή οποία είναι η προκαθορισμένη.

Παράδειγμα FlexBox

```
.flex {  
  display: flex;  
  background-color: grey;  
  padding: 5px;  
}  
.flex > * {  
  background-color: lightgrey;  
  border: 1px solid black;  
  padding: 20px;  
  font-size: 2rem;  
}
```

```
<div class="flex">  
  <div>1</div>  
  <div>2</div>  
  <div>3</div>  
  <div>4</div>  
  <div>5</div>  
  <div>6</div>  
</div>
```

Basic Flex



Ιδιότητες FlexBox

- flex-direction: row (default), column, row-reverse, column-reverse
 - Ιδιότητα που καθορίζει αν τα στοιχεία διατάσσονται οριζόντια (row) ή κατακόρυφα (column)
- flex-wrap: wrap, wrap-reverse, nowrap
 - ιδιότητα η οποία καθορίζει αν τα στοιχεία αναδιπλώνονται σε νέα γραμμή ή στήλη όταν το πλάτος ή το ύψος για την προβολή των στοιχείων δεν επαρκεί
- flex-flow: συνδυάζει τις ιδιότητες flex-direction και flex-wrap
 - π.χ. flex-flow: row wrap;
- justify-content: (ορίζει στοίχιση και αποστάσεις)
 - left, right, center, space-evenly, space-around, space-between, ..
- align-items: (ορίζει στοίχιση και αποστάσεις)
 - stretch (default), flex-start, flex-end, center, space-around, space-between, ..

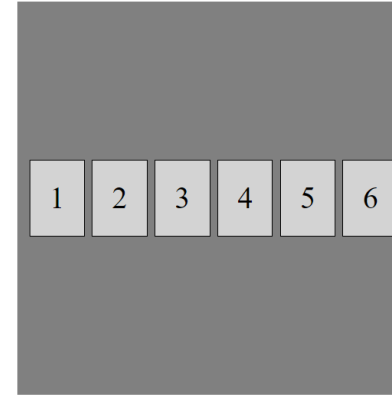
Για flex-flow: row η ιδιότητα justify-content ορίζει την οριζόντια στοίχιση και η ιδιότητα align-items την κατακόρυφη στοίχιση

Για flex-flow: column η ιδιότητα justify-content ορίζει την κάθετη στοίχιση και η ιδιότητα align-items την οριζόντια στοίχιση

Παράδειγμα ιδιοτήτων FlexBox

```
.flex {  
  display: flex;  
  flex-direction: row;  
  width: 400px;;  
  height: 400px;  
  justify-content: space-evenly;  
  align-items: center;;  
  background-color: grey;  
  padding: 5px;  
}  
.flex > * {  
  background-color: lightgrey;  
  border: 1px solid black;  
  padding: 20px;  
  font-size: 2rem;  
}
```

FlexBox Properties



```
<h1>FlexBox Properties</h1>  
<div class="flex">  
  <div>1</div>  
  <div>2</div>  
  <div>3</div>  
  <div>4</div>  
  <div>5</div>  
  <div>6</div>  
</div>
```

Ιδιότητες στοιχείων FlexBox

- **flex-grow**: καθορίζει το σχετικό μέγεθος μεγέθυνσης σε σύγκριση με τα υπόλοιπα στοιχεία του FlexBox
 - Αριθμός ο οποίος έχει αρχική τιμή 0 που δεν επιτρέπει την μεγένθυση.
- **flex-shrink**: καθορίζει το σχετικό μέγεθος σμίκρυνσης σε σύγκριση με τα υπόλοιπα στοιχεία του FlexBox
 - Δεν επιτρέπει την σμίκρυνση του μεγέθους του στοιχείου όταν δοθεί η τιμή 0. Η αρχική τιμή είναι 1 (όλα τα στοιχεία με ίδιο σχετικό μέγεθος)
- **flex-basis**: Καθορίζει το αρχικό πλάτος ή ύψος του στοιχείου.
 - Ανάλογα με τις τιμές flex-grow ή flex-shrink το στοιχείο μπορεί να αλλάζει μέγεθος
- **flex**: Συνδυασμός των ιδιοτήτων flex-grow, flex-shrink και flex-basis
 - π.χ. flex: 0 0 200px;
- **align-self**: Υπερκαλύπτει την ιδιότητα align-items του FlexBox και επανακαθορίζει την στοίχιση του στοιχείου.
- **order**: ορίζει (προαιρετικά) την σειρά εμφάνισης ενός στοιχείου

<https://css-tricks.com/understanding-flex-grow-flex-shrink-and-flex-basis/>

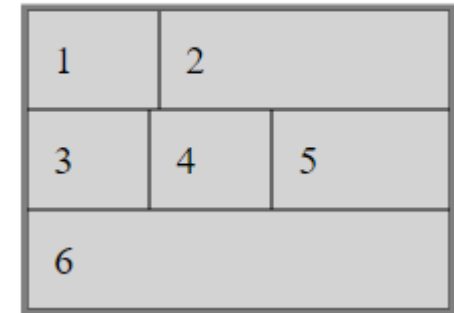
Παράδειγμα ιδιοτήτων στοιχείων FlexBox

```
.flex {  
  display: flex;  
  flex-direction: row;  
  flex-wrap: wrap;  
  background-color: grey;  
  padding: 5px;  
}  
.flex > * {  
  background-color: lightgrey;  
  border: 1px solid black;  
  padding: 20px;  
  font-size: 2rem;  
  flex-grow: 1;  
  flex-basis: 50px;  
}  
.flexSize3{  
  flex-grow: 3;  
  flex-shrink: 0;  
  flex-basis: 150px;  
}  
.flexFixed{  
  flex-grow: 0;  
  flex-shrink: 0;  
  flex-basis: 100px;  
}
```

FlexBox Item Properties

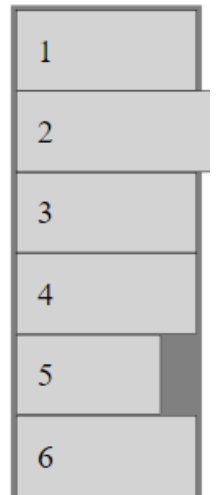


FlexBox Item Properties



Με την ιδιότητα flex-wrap το περιεχόμενο του FlexBox αναδιπλώνεται όταν το πλάτος δεν επαρκεί.

FlexBox Item Properties



```
<h1>FlexBox Item Properties</h1>  
<div class="flex">  
  <div>1</div>  
  <div class="flexSize3">2</div>  
  <div>3</div>  
  <div>4</div>  
  <div class="flexFixed">5</div>  
  <div>6</div>  
</div>
```

CSS Grid

- Με το Grid, κάνοντας χρήση γραμμών και στηλών, μπορούμε να ορίσουμε την διάταξη μιας ιστοσελίδας πολύ πιο απλούστερα σε σχέση με τον ορισμό της με την χρήση των ιδιοτήτων float και positioning.
- Δεν είναι συμβατό με παλιούς browsers
- Για την χρήση του, ορίζουμε ένα στοιχείο ως grid δίνοντας στην ιδιότητα display την τιμή grid ή inline-grid
- Ένα grid αποτελείται από rows (γραμμές) και columns (στήλες)
- Με τις ιδιότητες gap, column-gap, row-gap ορίζονται τα κενά μεταξύ των στηλών και των γραμμών

```
.grid-container {  
  display: grid;  
  column-gap: 50px;  
}
```


Grid Items

- Τα στοιχεία ενός grid τοποθετούνται στο grid με τις ιδιότητες:

- grid-row-start
- grid-row-end
- grid-column-start
- grid-column-end

οι οποίες καθορίζουν τις αρχικές ή και τελικές στήλες και γραμμές του στοιχείου. Για την τοποθέτηση μπορούν να χρησιμοποιηθούν και οι «σύντομες» ιδιότητες:

- grid-row: αρχή ή αρχή / τέλος ή αρχή / span πλήθος
- grid-column: αρχή ή αρχή / τέλος ή αρχή / span πλήθος
- grid-area: γραμμή αρχής / στήλη αρχής ή και / γραμμή τέλους / στήλη τέλους

- Η ιδιότητα αυτή μπορεί να χρησιμοποιηθεί και για την χρήση ονομάτων περιοχών σε συνδυασμό με την ιδιότητα grid-template-areas του grid η οποία ορίζει την διάταξη του grid με την χρήση ονομάτων

π.χ. grid-template-areas:

```
" area1 area2 area3"
```

```
" area4 area4 area5"
```

```
" area6 area6 area6";
```

Παράδειγμα τοποθέτησης στο grid

```
.grid {  
  display: grid;  
  background-color: grey;  
  padding: 5px;  
}  
.grid div {  
  background-color: lightgrey;  
  border: 1px solid black;  
  padding: 20px;  
  font-size: 2rem;  
  text-align: center;  
}  
.gridItem1 {  
  grid-column: 1;  
  grid-row: 1; }  
.gridItem2 {  
  grid-column: 2/3;  
  grid-row: 1/2; }  
.gridItem3 {grid-area: 1/3/2/4;}  
.gridItem4 {grid-area: 2/1/2/3;}  
.gridItem5 {grid-area: 2/3;}  
.gridItem6 {grid-area: 3/1/3/4;}
```

Basic Grid

1	2	3
4		5
6		

```
<h1>Basic Grid</h1>  
<div class="grid">  
  <div class="gridItem1">1</div>  
  <div class="gridItem2">2</div>  
  <div class="gridItem3">3</div>  
  <div class="gridItem4">4</div>  
  <div class="gridItem5">5</div>  
  <div class="gridItem6">6</div>  
</div>
```

Παράδειγμα τοποθέτησης στο grid με ονόματα περιοχών

```
.grid {
  display: grid;
  background-color: grey;
  padding: 5px;
  grid-template-areas:
    " area1 area2 area3"
    " area4 area4 area5"
    " area6 area6 area6";
}

.grid > * {
  background-color: lightgrey;
  border: 1px solid black;
  padding: 20px;
  font-size: 2rem;
  text-align: center;
}

.gridItem1 {grid-area: area1;}
.gridItem2 {grid-area: area2;}
.gridItem3 {grid-area: area3;}
.gridItem4 {grid-area: area4;}
.gridItem5 {grid-area: area5;}
.gridItem6 {grid-area: area6;}
```

Grid with names

2	1	3
4		5
6		

```
<h1>Basic Grid</h1>
<div class="grid">
  <div class="gridItem1">1</div>
  <div class="gridItem2">2</div>
  <div class="gridItem3">3</div>
  <div class="gridItem4">4</div>
  <div class="gridItem5">5</div>
  <div class="gridItem6">6</div>
</div>
```

Ιδιότητες του grid

- `grid-template-rows` και `grid-template-columns`
 - Με τις ιδιότητες αυτές ορίζεται το πλήθος των γραμμών και στηλών και προαιρετικά το ύψος ή πλάτος τους. Οι γραμμές προστίθενται αυτόματα αν δεν ορισθούν με την ιδιότητα `grid-template-rows`. Παραδείγματα:
 - `grid-template-columns: auto auto auto auto;` (4 στήλες με αυτόματη πλάτος)
 - `grid-template-columns: 80px 200px auto 40px;` (4 στήλες με ορισμό πλάτους)
 - `grid-template-rows: 80px 200px;` (Δύο γραμμές με ορισμό ύψους)
- `justify-content`: (ορίζει την οριζόντια στοίχιση και αποστάσεις)
 - `stretch` (default), `left`, `right`, `center`, `space-evenly`, `space-around`, `space-between`, ..
- `align-items`: (ορίζει την κατακόρυφη στοίχιση και αποστάσεις)
 - `stretch` (default), `flex-start`, `flex-end`, `center`, `space-around`, `space-between`, ..

Παράδειγμα αυτόματης τοποθέτησης στο grid

Auto Grid

1	2	3
4	5	6
7	8	9

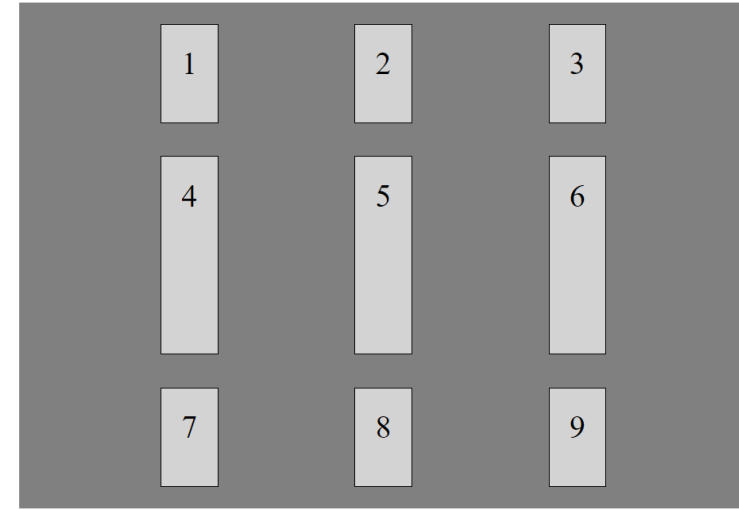
```
.grid {  
  display: grid;  
  grid-template-columns: auto auto auto;  
  background-color: grey;  
  padding: 5px;  
}  
.gridItem {  
  background-color: lightgrey;  
  border: 1px solid black;  
  padding: 20px;  
  font-size: 2rem;  
  text-align: center;  
}
```

```
<h1>Basic Grid</h1>  
<div class="grid">  
  <div class="gridItem">1</div>  
  <div class="gridItem">2</div>  
  <div class="gridItem">3</div>  
  <div class="gridItem">4</div>  
  <div class="gridItem">5</div>  
  <div class="gridItem">6</div>  
  <div class="gridItem">7</div>  
  <div class="gridItem">8</div>  
  <div class="gridItem">9</div>  
</div>
```

Παράδειγμα ορισμού στοίχισης σε grid

Justify / Align Grid

```
.grid {  
  display: grid;  
  height: 500px;  
  grid-template-columns: auto auto auto;  
  grid-template-rows: 100px 200px 100px;  
  background-color: grey;  
  padding: 5px;  
  justify-content: space-evenly;  
  align-content: space-around;  
}  
.gridItem {  
  background-color: lightgrey;  
  border: 1px solid black;  
  padding: 20px;  
  font-size: 2rem;  
}
```



```
<h1>Justify / Align Grid</h1>  
<div class="grid">  
  <div class="gridItem">1</div>  
  <div class="gridItem">2</div>  
  <div class="gridItem">3</div>  
  <div class="gridItem">4</div>  
  <div class="gridItem">5</div>  
  <div class="gridItem">6</div>  
  <div class="gridItem">7</div>  
  <div class="gridItem">8</div>  
  <div class="gridItem">9</div>  
</div>
```

Responsive Web Design

- Μία ιστοσελίδα προβάλλεται σε διάφορες οθόνες Η/Υ και συσκευών:
 - Η/Υ με 27" desktop monitor
 - Laptop
 - Tablet
 - Mobile
- Η ενημέρωση του φυλλομετρητή για την προβολή μιας ιστοσελίδας στο μέγεθος της συσκευής γίνεται με την δήλωση στο τμήμα head:
`<meta name="viewport" content="width=device-width, initial-scale=1.0">`

Για την προσαρμογή της ιστοσελίδας σε διάφορους τύπους συσκευών χρησιμοποιούνται media queries

Media Queries

- Με τον ορισμό ενός CSS media query καθορίζονται οι ιδιότητες στοιχείων html ανάλογα με τις διαστάσεις του viewport
- Παράδειγμα:

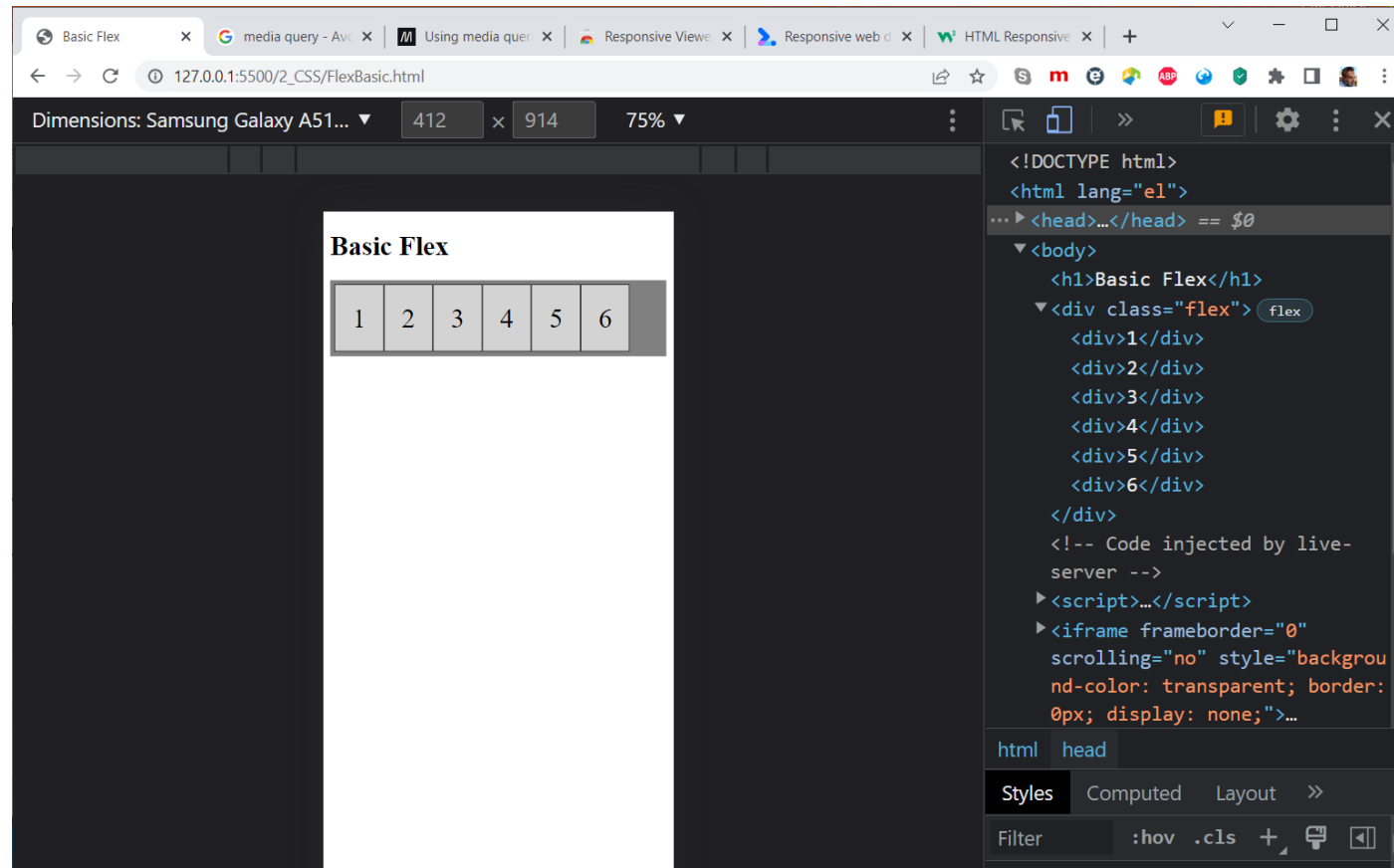
```
@media (max-width: 500px) {  
  article {  
    margin: 0 2px;  
  }  
}
```

Περισσότερες πληροφορίες θα βρείτε στην διεύθυνση:

https://developer.mozilla.org/en-US/docs/Web/CSS/Media_Queries/Using_media_queries

Responsive design with Google Chrome

Οι σύγχρονοι φυλλομετρητές διαθέτουν προγραμματιστικά εργαλεία για την προεπισκόπηση ιστοσελίδων σε διάφορες συσκευές



Μετάβαση σε Developer Tools (Ctrl+Shift+I) και στην συνέχεια Toggle Device ToolBar (Ctrl+Shift+M)

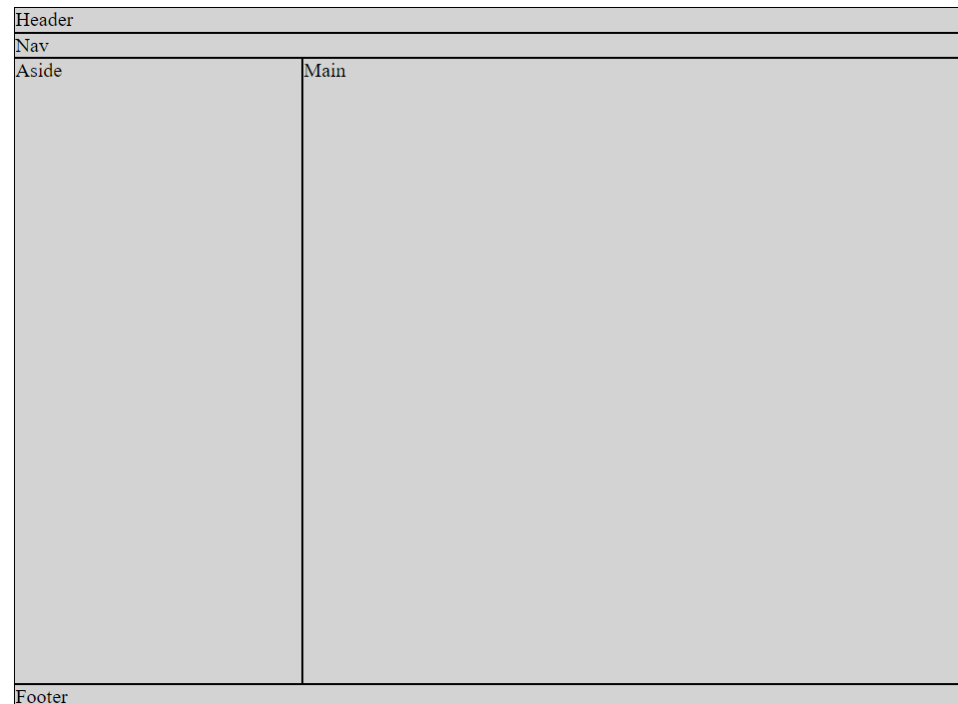
<https://developer.chrome.com/docs/devtools/device-mode/>

Διάταξη ιστοσελίδας - CSS Layout

- Παραδείγματα διάφορων ειδών διατάξεων μπορείτε να βρείτε στην διεύθυνση https://www.w3schools.com/css/css_templates.asp

Σχεδίαση διάταξης σελίδας με χρήση float

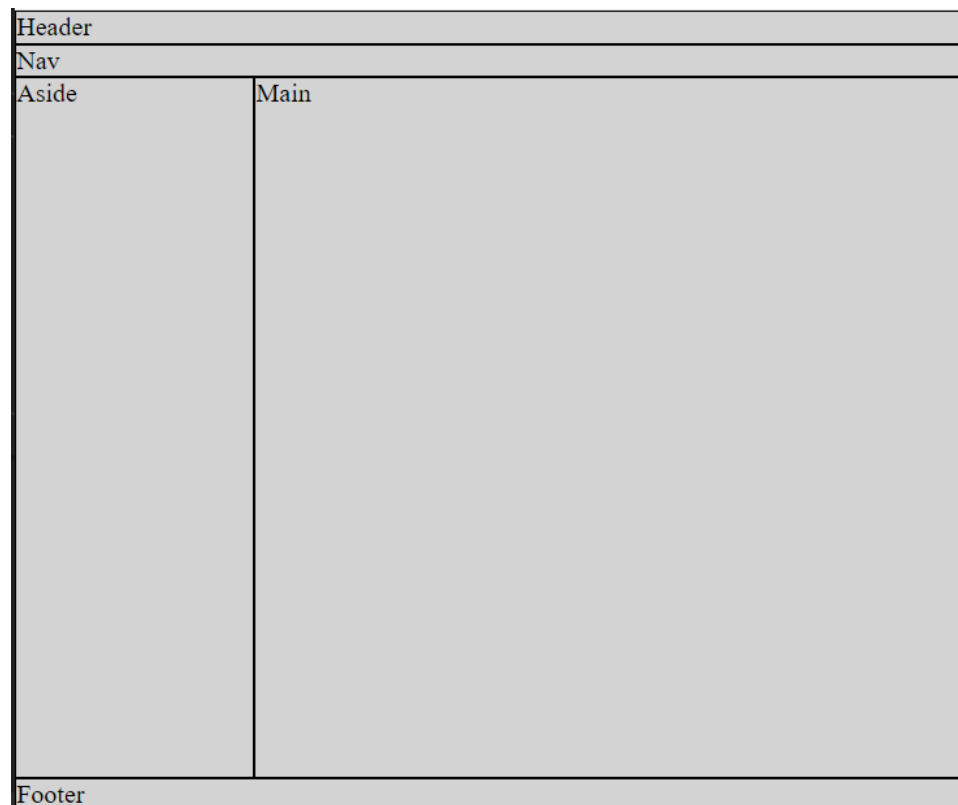
```
*{
    box-sizing: border-box;
    margin: 0;
    padding: 0;
}
header, nav, aside, main, footer {
    border: 1px solid black;
    background-color: lightgrey;
    height: 20px;
}
aside, main {
    float: left;
    height: calc(100vh - 60px);
    box-sizing: border-box;
}
aside{
    width: 30%;
}
main{
    width: 70%;
}
footer {
    clear: both;
}
```



```
<div class="wrapper">
    <header>Header</header>
    <nav>Nav</nav>
    <aside>Aside</aside>
    <main>Main</main>
    <footer>Footer</footer>
</div>
```

Σχεδίαση διάταξης σελίδας με flexbox

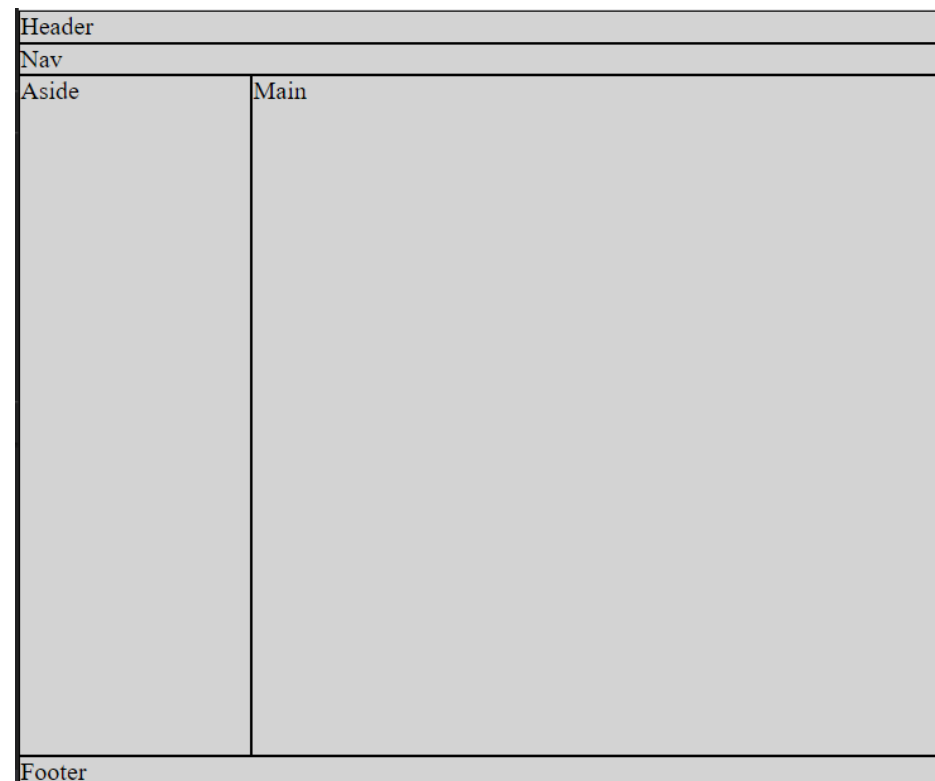
```
*{
  box-sizing: border-box;
  margin: 0;
  padding: 0;
}
body{
  width:100vw;
  height:100vh;
  display:flex;
  flex-direction: column;
}
header, nav, aside, main, footer {
  border: 1px solid black;
  background-color: lightgrey;
}
.mainWrapper{
  display: flex;
  flex-grow:1;
  flex-wrap: wrap;
}
aside {
  flex-grow: 1;
}
main {
  flex-grow: 3;
}
```



```
<header>Header</header>
<nav>Nav</nav>
<aside>Aside</aside>
<main>Main</main>
<footer>Footer</footer>
```

Σχεδίαση διάταξης σελίδας με grid

```
*{
  margin: 0;
  padding: 0;
}
.grid{
  width:100vw;
  height:100vh;
  display: grid;
  grid-template-columns: 1fr 3fr;
  grid-template-rows: min-content min-content
                      auto min-content;
  grid-template-areas:
    'header header'
    'nav nav'
    'aside main'
    'footer footer';
}
header, nav, aside, main, footer {
  border: 1px solid black;
  background-color: lightgrey;
}
header {grid-area: header;}
nav {grid-area: nav;}
aside {grid-area: aside;}
main {grid-area: main;}
footer {grid-area: footer;}
```



```
<header>Header</header>
<nav>Nav</nav>
<aside>Aside</aside>
<main>Main</main>
<footer>Footer</footer>
```

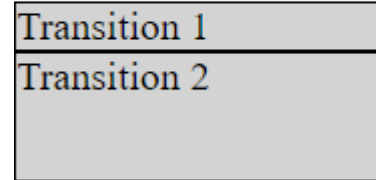
CSS Transitions

- Οι μεταβάσεις CSS επιτρέπουν την σταδιακή μετάβαση τιμών ιδιοτήτων σε προκαθορισμένη χρονική διάρκεια όταν αλλάζει η τιμή μιας ιδιότητας.
- Ιδιότητες μεταβάσεων:
 - transition-property: καθορισμός της ιδιότητας που αλλάζει
 - transition-duration: in s (seconds) or ms (milliseconds) π.χ. 2s
 - transition-timing-function: linear, ease, ease-in, ease-out, ease-in-out
 - transition-delay: καθυστέρηση σε s ή ms
 - transition: συνδυασμός ιδιοτήτων πχ. width 2s linear 1s;

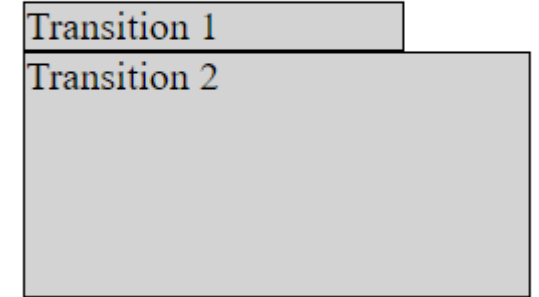
Παράδειγμα μεταβάσεων

```
.div1{
  width:150px;
  border: 1px solid black;
  background-color: lightgray;
  transition: 1200ms;
}
.div1:hover{
  width: 200px;
}
.div2{
  width:150px;
  height:50px;
  border: 1px solid black;
  background-color: lightgray;
  transition: width 1200ms, height 5s;
}
.div2:hover{
  width: 200px;
  height: 100px;
}
```

Transition



Transition



```
<h1>Transition</h1>
<div class="div1">Transition 1</div>
<div class="div2">Transition 2</div>
```

CSS 2DTransforms

- Οι μετασχηματισμοί (transforms) εφαρμόζουν γεωμετρικές αλλοιώσεις όπως π.χ. περιστροφή, μετατόπιση, λοξότητα και κλιμάκωση κ.λπ.
- 2D transformation methods:
 - `translate()`
 - `rotate()`
 - `scaleX()`
 - `scaleY()`
 - `scale()`
 - `skewX()`
 - `skewY()`
 - `skew()`
 - `matrix()`

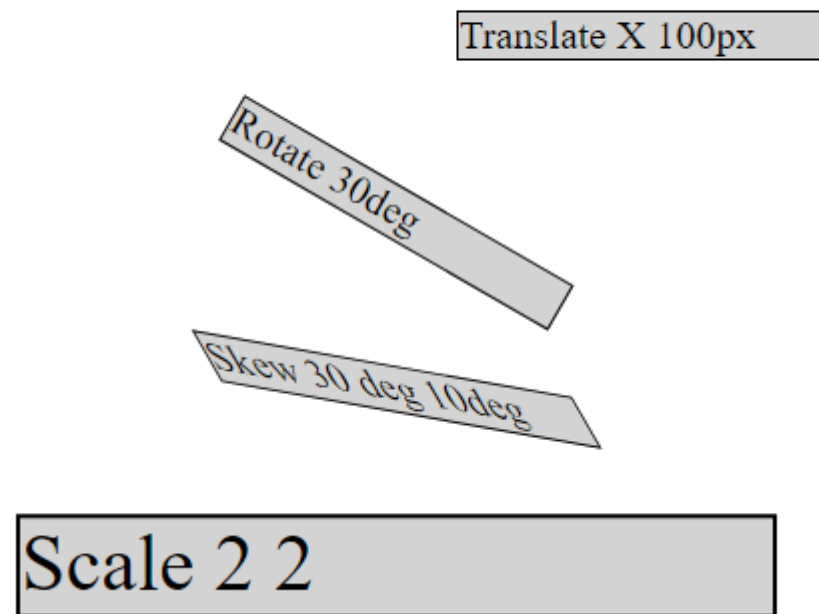
CSS 3DTransforms

Function	Description
<code>matrix3d</code> <code>(n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n)</code>	Defines a 3D transformation, using a 4x4 matrix of 16 values
<code>translate3d(x,y,z)</code>	Defines a 3D translation
<code>translateX(x)</code>	Defines a 3D translation, using only the value for the X-axis
<code>translateY(y)</code>	Defines a 3D translation, using only the value for the Y-axis
<code>translateZ(z)</code>	Defines a 3D translation, using only the value for the Z-axis
<code>scale3d(x,y,z)</code>	Defines a 3D scale transformation
<code>scaleX(x)</code>	Defines a 3D scale transformation by giving a value for the X-axis
<code>scaleY(y)</code>	Defines a 3D scale transformation by giving a value for the Y-axis
<code>scaleZ(z)</code>	Defines a 3D scale transformation by giving a value for the Z-axis
<code>rotate3d(x,y,z,angle)</code>	Defines a 3D rotation
<code>rotateX(angle)</code>	Defines a 3D rotation along the X-axis
<code>rotateY(angle)</code>	Defines a 3D rotation along the Y-axis
<code>rotateZ(angle)</code>	Defines a 3D rotation along the Z-axis
<code>perspective(n)</code>	Defines a perspective view for a 3D transformed element

CSS Transforms παράδειγμα

```
div{
  border: 1px solid black;
  background-color: lightgray;
  width: 150px;
  margin-left: 100px;
  margin-bottom: 50px;
}
.translate{
  transform: translate(100px);
}
.rotate{
  transform: rotate(30deg);
}
.skew{
  transform: skew(30deg, 10deg);
}
.scale{
  transform: scale(2,2);
}
```

2D Transforms



```
<h1>2D Transforms</h1>
<div class="translate">Translate X 100px</div>
<div class="rotate">Rotate 30deg</div>
<div class="skew"> Skew 30 deg 10deg</div>
<div class="scale"> Scale 2 2</div>
```

CSS Animation

- Με την δήλωση `@keyframes` καθορίζεται κίνηση (αλλαγές τιμών ιδιοτήτων)
- Παράδειγμα:

```
@keyframes example {  
  from {background-color: red;}  
  to {background-color: yellow;}  
}
```
- Ένα στοιχείο html μπορεί να καθορίζει τις παρακάτω ιδιότητες για την κίνηση του:
 - `animation-name`
 - `animation-duration`
 - `animation-delay`
 - `animation-iteration-count`
 - `animation-direction`: normal, reverse, alternate, alternate-reverse
 - `animation-timing-function`: : linear, ease, ease-in, ease-out, ease-in-out, cubic-bezier(n,n,n,n), ..
 - `animation-fill-mode`: none, forwards, backwards, both
 - Καθορίζει ποιες ιδιότητες παραμένουν μετά την κίνηση
 - `animation` (συνδυάζει τις παραπάνω ιδιότητες)

Παράδειγμα animation

```
.div1{
  width:150px;
  border: 1px solid black;
  background-color: lightgray;
  animation: animation1 3s;
}
.div2{
  width:150px;
  border: 1px solid black;
  background-color: lightgray;
  animation: animation1 3s linear 2s infinite alternate;
}
@keyframes animation1 {
  from {margin-left: 0px;}
  to {margin-left: 150px;}
}
```

```
<h1>Animations</h1>
<div class="div1">Animation 1</div>
<p></p>
<p></p>
<div class="div2">Animation 2</div>
```

Animations

Animation 1

Animation 2

Αναφορές

- <https://developer.mozilla.org/en-US/docs/Learn/CSS>
- <https://www.w3schools.com/css/default.asp>
- <https://www.w3.org/Style/CSS/>
- <https://www.freecodecamp.org/news/tag/css/>
- Video Tutorial for CSS
 - https://www.youtube.com/watch?v=OXGznpKZ_sA&ab_channel=freeCodeCamp.org
 - https://www.youtube.com/watch?v=1Rs2ND1ryYc&ab_channel=freeCodeCamp.org
- CSS How to / Examples
 - https://www.w3schools.com/css/css_examples.asp

Ο κώδικας είναι διαθέσιμος στο github repository
<https://github.com/uthcst/uthcst.github.io>

url παραδειγμάτων <https://uthcst.github.io/>