

### Source

```
<source src="/media/cc0-videos/flower.webm" type="video/webm">
```

### Track

```
<video controls src="/media/cc0-videos/friday.mp4">  
  <track default kind="captions" srclang="en" src="/media/examples/friday.vtt" />  
  Sorry, your browser doesn't support embedded videos.  
</video>
```

### Link

```
<link href="/media/examples/link-element-example.css" rel="stylesheet">
```

### Style (CSS)

```
<style>  
p {  
  color: #26b72b;  
}  
</style>
```

section: a generic standalone section of a document

```
<section>  
  <h2>Introduction</h2>  
  <p>This document provides a guide to help with the important task of choosing the correct  
Apple.</p>  
</section>
```

nav: a section of a page whose purpose is to provide navigation links

aside: represents a portion of a document whose content is only indirectly related to the document's main content. Asides are frequently presented as sidebars or call-out boxes.

DL: description list. The element encloses a list of groups of terms (specified using the <dt> element) and descriptions (provided by <dd> elements).

### Πίνακας

```
<table>  
  <caption>Ένας μικρός πίνακας</caption>  
  <thead></thead>  
  <tbody>  
    <tr>  
      <th>Έτος</th>  
    </tr>  
  </tbody>  
  <tfoot></tfoot>  
</table>
```

### FORMS

#### datalist

```
<datalist id="ice-cream-flavors">  
  <option value="Chocolate">  
  <option value="Coconut">  
  <option value="Mint">  
  <option value="Strawberry">  
  <option value="Vanilla">  
</datalist>
```

[ <fieldset>, <legend> ], <label>, <input>

<select>, <optgroup>, <option>

```
<select id="dino-select">
  <optgroup label="Theropods">
    <option>Tyrannosaurus</option>
    <option>Velociraptor</option>
    <option>Deinonychus</option>
  </optgroup>
  <optgroup label="Sauropods">
    <option>Diplodocus</option>
    <option>Saltasaurus</option>
    <option>Apatosaurus</option>
  </optgroup>
</select>
```

```
<progress id="file" max="100" value="70"> 70% </progress>
```

```
<meter id="fuel"
  min="0" max="100"
  low="33" high="66" optimum="80"
  value="50">
  at 50/100
</meter>
```

Non-braking space -> κενό σε κελί. Μεταξύ δυο λέξεων για να μην τις χωρίζει όταν πχ μικραίνει το παράθυρο.

Ειδικοί χαρακτήρες να δω.  
Html special characters mdn

Id -> χρήση ενός element για να το ονοματίσω το unique element  
Class -> ομαδοποιεί σε ένα σύνολο τα elements για χρήση στην css  
Title -> μου δίνει ένα βοηθητικό κείμενο για πχ links

MOBILE

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

Content (comma separated):

-width=

- ακέραιος αριθμός (π.χ. 320) ή device-width

-initial-scale=

- αριθμός (π.χ. 1.2)

-minimum-scale=

- αριθμός (αν είναι 1.0 αποτρέπεται το zoom-out)

-maximum-scale=

- αριθμός (αν είναι 1.0 αποτρέπεται το zoom-in)

-user-scalable=

- no

Πολλά properties

```
<h1 style="font-style: italic; font-weight: normal;">Μερικές Λίστες</h1>
```

\*\*\*\*\*

Δημιουργία κλάσης

```
.myBold {  
font-weight: bold;  
}
```

```
<li class="myBold">Γάλα</li>
```

\*\*\*\*\*

Κατοχή ιδιοτήτων πολλών κλάσεων

```
<li class="myBold myItalic">Τυρί</li>
```

\*\*\*\*\*

Attribute: id

Μοναδική αναφορά σε ένα element!

```
#myID {  
font-style: italic;  
font-weight: normal;  
}
```

```
<h1 id="myID">Μερικές Λίστες</h1>
```

\*\*\*\*\*

External CSS

```
<link rel="stylesheet" href="myStyles.css">
```

\*\*\*\*\*

Selectors

- X > Y (child selector)

- #container > ul {} Selects all ul elements where the parent is a container id (άμεσα παιδιά)

- X Y (descendant selector)

- li a {} Selects all a elements inside li elements (απόγονοι, είτε παιδιά, είτε εγγόνια)

- Attribute selector

- a[title], a[href="https://www.uom.gr"] {}

- Pseudo-classes

- a:visited, section:not(#references) {}

- p:lang(el), input[type="checkbox"]:checked {}

\*\*\*\*\*

CSS Specificity

- Inline Styles

- Specificity

- ID selectors

- Class selectors, Attributes selectors, Pseudo-classes

- Elements, Pseudo-elements

- Order of Specification

\*\*\*\*\*

```
html {  
    font-size: 50px  
}
```

```
h1 {  
    font-size: 0.7rem;  
}
```

Να βάζουμε το html element (κρίσιμο σημείο, σημείο αναφοράς) και μετά όλα τα άλλα να τα χρησιμοποιούμε με relative em (δουλεύει καλά και με εμφωλεμένα, θυγατρικά κλπ).

Αριθμός % σε σχέση με το μέγεθος της οθόνης (viewport όπως για κινητά)

```
h1 {  
    font-size: 5vw;  
}
```

vw και vh

Για να στοιχίσει στο κέντρο της σελίδας το element  
margin: auto;

float property

Ροή κειμένου γύρω από εικόνες.

```
img {  
    float: left;  
    margin: 2px 2px 0 2px;  
}
```

\*\*\*\*\*

# display property (SOS)

▷ none: Απόκρυψη

▷ inline: Μέρος μιας συνεχούς γραμμής περιεχομένου, με αναδίπλωση λέξεων (width και height αγνοούνται, top και bottom padding, border και margin απλά επικαλύπτουν τα υπόλοιπα στοιχεία της ροής)

▷ block: Αλλαγή γραμμής πριν και μετά, πλήρες box model

▷ inline-block: Συμπεριφέρονται σαν ένα ενιαίο αντικείμενο εντός μιας συνεχούς γραμμής περιεχομένου, πλήρες box model

▷ flex / inline-flex: Συμπεριφορά ως block/inline με βάση το Flexible Box Layout Module

▷ grid / inline-grid: Συμπεριφορά ως block/inline με βάση το Grid Layout Module

flex = μικρότερα σημεία, λιγότερων στοιχείων (μενού)-->μέρη σελίδας (layouts 1 διάστασης μικρά)

grid = μεγάλη επιφάνεια (όλη η σελίδα)-->όλων των παραθύρων (layouts 2 διαστάσεων μεγάλα)

εμφώλευση flex μέσα σε grid

\*\*\*\*\*

# position property

fixed / sticky -> αν θέλω να καρφώσω σε ένα σημείο το στοιχείο (πχ το navigation bar)

absolute -> ένα αντικείμενο να ακολουθεί κάποιο άλλο.

\*\*\*\*\*

At-rules

media -> διαφορετικές αποφάσεις. Κάνει την σελίδα responsive

max-width: 600px

αν έχει μέγεθος μέχρι 600px (0-600) τότε να ισχύει ο κανόνας να μην εμφανίζεται το aside

min-width: 700px

αν έχει μέγεθος από 700px και πάνω (700-ως τέρμα)

\*\*\*\*\*

@media all and (max-width: 800px) {

.navigation {

justify-content: space-around; # με κενά γύρω τους και ίσα σε μέγεθος

}

}

\*\*\*\*\*

flexbox

<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

grid

<https://css-tricks.com/snippets/css/complete-guide-grid/>

\*\*\*\*\*

Transform

.translate {

transform: translate(50px, -20px);

}

```
.rotate {
    transform-origin: top left;
    transform: rotate(-15deg);
}
```

```
.scale {
    transform-origin: top left;
    transform: scale(1.5, 0.8);
}
```

```
.skew {
    transform: skew(10deg, 5deg);
    transform-origin: top left;
}
```

```
*****
Transition
```

```
p.expand {
    border: 2px solid rgb(172, 172, 172);
    background-color: rgb(130, 253, 149);
    width: 150px;
    height: 20px;
    overflow: hidden;
    padding: 5px;
    margin: 5px;
    transition: width 180ms, height 0.4s;
}
```

```
p.expand::after {
    content: " ...";
}
```

```
p.expand:hover, p.expand:focus {
    width: 300px;
    height: 100px;
}
```

```
*****
Grid
```

```
/* Colorize and assign grid names */
header { background-color: yellow;          grid-area: head; }
nav     { background-color: yellowgreen;    grid-area: navi; }
main    { background-color: orange;         grid-area: main; text-align: left; }
article { background-color: skyblue;        grid-area: news; }
aside   { background-color: violet;         grid-area: info; }
footer  { background-color: lightgray;      grid-area: foot; }
```

```
/* Grid container - small screens */
body {
    display: grid;
    grid-gap: 5px;
    grid-template-columns: 1fr;
    grid-template-areas:
        "head"
        "navi"
        "main"
        "news"
        "info"
        "foot";
}
```

```
/* Medium screens */
@media all and (min-width: 600px) {
```

```

    body {
      grid-template-columns: 1fr 2fr;
      grid-template-areas:
        "head head"
        "navi navi"
        "news main"
        "info foot";
    }
  }

  /* Large screens */
  @media all and (min-width: 800px) {
    nav {
      justify-content: flex-start;
      flex-direction: column;
    }
    a { flex: 0; }

    body {
      grid-template-columns: 1fr 4fr 1fr;
      grid-template-areas:
        "head head head"
        "navi main news"
        "navi main info"
        "foot foot foot";
    }
  }

```

Imperative: διαδικαστικό. Σαν την C  
Object-oriented: αντικειμενοστρεφής όπως η Java

Javascript είναι και τα δυο. Είναι συναρτησιακή γλώσσα επίσης.

Internal και external

- Inline μπορώ να εκτελέσω κώδικα μέσα σε κάποιο global variables με events στα element.  
OXI
- `<script>` `</script>` μέσα στο body. ΜΟΝΟ αν χρειαζομαι κώδικα στην συγκεκριμένη σελίδα

Εξωτερικά (ΠΙΟ ΑΓΑΠΗΤΟ)

`<script src="myFile.js"></script>`

`alert(' ');` εμφανίζει ένα μικρό παραθυράκι με μήνυμα.  
συμπεριφορά μπλοκινγκ. Περιμένει να πατήσουμε OK στο παράθυρο.

`console.log(..)` τα μηνύματα θα τα βλέπω στην κονσόλα.

Στην κονσόλα θα βλέπω συντακτικά σφάλματα.

Από τους χρωματισμούς του editor θα καταλάβω τα σφάλματα.

`if true` ? `κανε αυτό` : `else κανε αυτο`

EXPRESSION είναι οτιδήποτε παράγει τιμές

- Σκέτος αριθμός ή σκέτο string
- `2 + 3` επιστρέφει τιμή 5
- `x = y = 3;`
- `Math.sqrt(9)`

STATEMENT ΔΕΝ επιστρέφει τιμή (ένα πρόγραμμα είναι μια σειρά από statements)

- `var x;`
- `if (x>2) {};` δεν επιστρέφει τιμή οπότε είναι statement

ES5

θα δηλώνουμε `var` κάθε μεταβλητή.

Οποτεδήποτε ξεκινάμε κάθε αρχείο, να γράφουμε  
`'use strict';`

προστατεύω τον εαυτό μου. Να σταματάει κάθε φορά που βλέπει μια μεταβλητή που δεν είναι δηλωμένη `var`.

ES6

Δηλώνουμε αντί για `var`

`const x = 6` πρέπει να δηλώσω αμέσως την τιμή της μεταβλητής και αυτή δεν αλλάζει.  
Χρησιμοποιείται για δήλωση ΣΤΑΘΕΡΩΝ.

`let x = 6` επιτρέπεται να αλλάξουν τιμή. Η εμβέλεια.

`var` έχει εμβέλεια την συνάρτηση ενώ `const` και `let` έχουν εμβέλεια τα άγκιστρα τα οποία περιέχονται.

Οι μεταβλητές αρχίζουν από `_` και `$`

ΤΥΠΟΙ ΔΕΔΟΜΕΝΩΝ

\* Primitive τιμές: αριθμοί, string, boolean, null, undefined

Όταν δεν δηλώνω την τιμή μιας μεταβλητής, τότε ο τύπος της είναι undefined  
null είναι κενά objects.

\* Complex values: array, plain object, function και ότι άλλο δεν είναι primitive.

Infinity διαίρεση με 0 και ο τύπος του είναι αριθμός.



NaN (not a number): 'test'\*2 θεωρείται ότι είναι NaN

ΠΡΟΣΟΧΗ στους δεκαδικούς στην ακρίβεια τους

#### \* Strings

- Όταν σε ένα άθροισμα εμφανιστεί κάποιο string, έχουμε μετατροπή σε string και concatenation
- test: 1+2+3+'4' -> 1+2 =3 +3 =6 και μετά μετατρέπει το 6 σε string και βλέπουμε 64

- Template strings:

```
var s = 'Hello';  
console.log(`Program: ${s} world`);
```

Δηλαδή βάζω το σημάδάκι δίπλα στο 1 (το ``) και με \$ εισάγω την τιμή που έχει εκχωρηθεί στην μεταβλητή μου.

#### \* Boolean

“Falsy”: false, 0, -0, NaN, '', null, undefined

== : Ισότητα τιμών μετά από ενδεχόμενη μετατροπή (test: 0==false)

=== : Έλεγχος ταύτισης τιμών και τύπων (test: 0===false)

#### \* null και undefined

- undefined σημαίνει no value ή no existence

#### \* Arrays

Μονοδιάστατα, πολυδιάστατα, δυναμική επαύξηση

Αν προσθέσω πχ 25ο στοιχείο σε array των 2 στοιχείων, τα 2-24 είναι undefined

#### \* ΣΥΝΑΡΤΗΣΕΙΣ

- null σημαίνει no object ή emptiness

## CSS Selectors

### ΣΗΜΑΝΤΙΚΑ

SetInterval(): επανάληψη εκτέλεσης κάποιας συνάρτησης

setTimeout(): επανάληψη εκτέλεσης μιας συνάρτησης μια φορά

getComputedStyle(): αναγνώριση ιδιότητας, πχ CSS styles ενός element

<http://us.to/>

element objects όταν τα τσιμπήσει η javascript

document.body.children[0] : είμαι στο πρώτο element που είναι παιδί του body

πρόσβαση σε elements ορίζοντας πχ κάποιο id και το τσιμπάω πχ getElements

### ΣΗΜΑΝΤΙΚΑ ΔΙΑΦΑΝΕΙΑ 17

getElementById(str\_id): το id θα είναι μοναδικό, οπότε επιστρέφει ΜΟΝΟ ένα element object

querySelector(str\_selector): Κάνει την ίδια δουλειά όπως και το querySelectorAll αλλά επιστρέφει ΜΟΝΟ το πρώτο element object από αυτά που βρήκε

querySelectorAll(str\_selector): εκτελεί τον css selector και επιλέγει ΟΛΑ τα elements που έχουν το όνομα κλάσης που έχει δηλωθεί στην παρένθεση. Επιστρέφει array από element objects. Πχ  
querySelectorAll('.myClass')

click = mousedown + mouseup + χωρίς μετακίνηση + σε λιγότερο από 300ms

dblclick = 2 click σε λιγότερο από 500ms η πρώτη από δεύτερη

event listener

συνάρτηση που περιμένει πότε θα εκτελεστεί ο κώδικάς της

όλα τα attributes γίνονται properties

getAttribute(attr), setAttribute(attr, value), removeAttribute(attr):  
Ανάγνωση / εγγραφή / διαγραφή attributes

attributes σε filenames λαμβάνω το όνομα του αρχείου με getAttribute

document.querySelector('img').src βγάζει πλήρες path

document.querySelector('img').getAttribute('src') βγάζει μόνο το όνομα αρχείου

SOS Διαφάνεια 6

κάθε document έχει ένα attribute-property style  
document.querySelector('img').style.border = 'solid black 2px';

Όπου έχει - μεταξύ σε κάποιον CSS style, μπαίνει camelcase.  
document.querySelector('img').style.backgroundColor = 'red';

Για διάβασμα μιας ιδιότητας CSS χρησιμοποιώ  
getComputedStyle(element)

Ανάγνωση width ++++++

getComputedStyle(document.get  
που επιστρέφει object παρόμοιο με το προαναφερθέν style

innerHTML οτιδήποτε περιέχεται από εκεί που ανοίγει έως εκεί που τελειώνει το element

document.body.innerHTML = '<ul><li>ΠΑΟΚ</li></ul>'; σβηνει ότι έχει το body και προσθέτει  
λίστα

document.body.innerHTML += '<ol><li>New</li></ol>'; προσθέτει λίστα με new

document.querySelectorAll('li')[0].insertAdjacentHTML('','');

Adjacent=διπλανός  
beforebegin  
afterbegin

beforeend  
afterend

document.querySelectorAll('li')[0].insertAdjacentHTML('beforebegin','<li>Zero</li>');

ΑΦΑΙΡΩ το element

document.querySelector('ul').removeChild(document.querySelectorAll('li')[0]);

innerHTML, insertAdjacentHTML(position, text), removeChild(child)

cloneNode(deep) να μην έχει id όταν κλωνοποιείται ένα element

Διαφάνεια 15

Χρήσιμες μέθοδοί του:

- preventDefault(): Αποτρέπει την default συμπεριφορά του browser (π.χ. anchors, checkboxes)
- stopPropagation(): Αποτρέπει την περαιτέρω διάδοση του event στις επακόλουθες φάσεις
- stopImmediatePropagation(): Όπως η προηγούμενη αλλά επιπρόσθετα αποτρέπει και την οποιαδήποτε εκτέλεση άλλων event listeners που έχουν

επαυξήσει το συγκεκριμένο event στο συγκεκριμένο object (πιο άμεσο)

#### Διαφάνεια 17

Mouse event properties πρέπει να διαβάσω ποιο κουμπί ποντικίου πιάστηκε

- button
  - Αριθμός: 0: αριστερό πλήκτρο, 1: μεσαίο πλήκτρο, 2: δεξιό πλήκτρο
- clientX, clientY
  - Αριθμός σε pixels της απόστασης του ποντικίου οριζοντίως / κατακορύφως από την επάνω αριστερά γωνία του εγγράφου
- screenX, screenY
  - Αριθμός σε pixels της απόστασης του ποντικίου οριζοντίως / κατακορύφως από την επάνω αριστερά γωνία της οθόνης