



#### ΠΕΡΙΕΧΟΜΕΝΑ:

1. Ιστορική Αναδρομή
2. Κανόνες CSS και συσχέτιση με HTML
3. Συνήθειες Επιλογείς
4. Κληρονόμηση
5. Μεταβλητές
6. Disclaimer σειράς CSS και μια ιδέα...

#### ΠΡΟΑΠΑΙΤΟΥΜΕΝΑ:

1. HTML

Ανδρέας Γ.

Σμαραγδένιος Χορηγός Μαθήματος

Χρήστος Σ.

Χρυσός Χορηγός Μαθήματος

### Τι είναι η CSS (Cascading Style Sheets):

- Μία HTML σελίδα έχει περιεχόμενο με σημασιολογία.
- Με τη **CSS μπορούμε να εφαρμόσουμε στυλ** (μεταξύ άλλων: διεύθυνση στοιχείων στη σελίδα, χρώμα, γραμματοσειρά, animation) στα στοιχεία της HTML σελίδας
- Οι κανόνες της CSS δουλεύουν ως εξής: «Επίλεξε το τάδε στοιχείο της σελίδας και έφαρμοσε τους εξής κανόνες στυλ»
- **Οι κανόνες γράφονται σε φύλλα στυλ (style sheets)** - συνήθως ξεχωριστά αρχεία.
- Μπορούμε να έχουμε πολλά φύλλα στυλ για την ίδια σελίδα και πολλούς κανόνες για το ίδιο στοιχείο (ωστόσο, μόνο ένας από τους κανόνες θα επικρατήσει).
- Γι' αυτό και ονομάζονται **Cascading Style Sheets (αλληλοεπικαλυπτόμενα φύλλα στυλ)** - «Ο συνδυασμός πολλών φύλλων στυλ και η επίλυση των συγκρούσεων που προκαλούνται από τους κανόνες τους» Lie, PhD thesis)

### Μικρή Ιστορική Αναδρομή:

- **1993: HTML 1.0:** Υποστηρίζει κείμενο και εικόνες.
- **1994: Håkon Wium Lie** (συνεργάτης του Tim Berners-Lee στο CERN). Προτείνει (και γράφει το πρώτο draft) της CSS.  
**Bert Bos** (κατασκεύαζε τον browser Argo) ενσωματώνει σε αυτόν τη CSS.  
[επίμαχη ιδέα => το στυλ θα καθορίζεται (με συγκρούσεις) και από το χρήστη και από το σχεδιαστή της σελίδας]
- **1995:** Θέσπιση W3C (WWW Consortium). Οι Lie και Bos εργάζονται εκεί.
- **1996: CSS1.0:** Το W3C "συστήνει" την ενσωμάτωση της στους browsers. Περιέχει κανόνες για βασικά στοιχεία στυλ (1998-CSS2.0: Πρόσθετα στοιχεία στυλ)
- **1999: CSS3.0:** Δεν βγαίνουν πια εκδόσεις, η CSS έχει σπάσει σε ξεχωριστά πακέτα που το καθένα έχει δικό του κύκλο ανάπτυξης, με προς τα πίσω συμβατότητα και πλέον υπάρχουν ετήσια snapshots - βλ. και <https://www.w3.org/Style/CSS/current-work>



Håkon Wium Lie



Bert Bos

### Παράδειγμα 1: hello (βλ. βίντεο)

hello.html

Hello World!

Welcome to my page

με style1.css

Hello World!

Welcome to my page

με style2.css

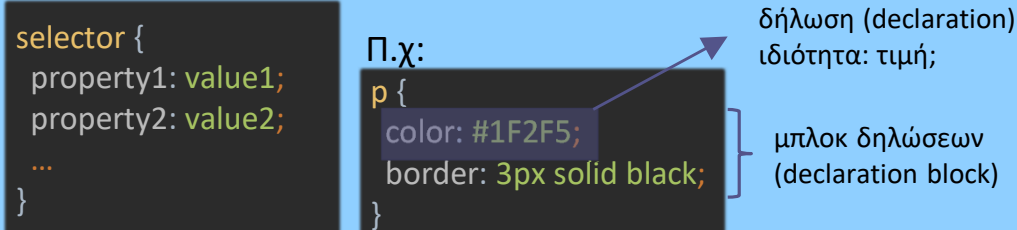
HELLO WORLD!

WELCOME TO MY PAGE

### Κανόνες CSS (CSS rules):

- Κάθε κανόνας έχει τη λειτουργία: «Επίλεξε αυτά τα στοιχεία HTML και θέσε μία ιδιότητά τους (π.χ το χρώμα, τη γραμματοσειρά κ.ο.κ.) ίση με μια τιμή»

### Συντακτικό κανόνα:



- selector (επιλογή):** Καθορίζει ποια HTML στοιχεία επιλέγονται (στο παράδειγμα όλες οι παράγραφοι)
- property (ιδιότητα):** Υπάρχουν εκατοντάδες ιδιότητες που καθορίζουν την εμφάνιση του στοιχείου  
Προσοχή: Στην άνω κάτω τελεία (μετά το property) και το ερωτηματικό (μετά το πέρας της τιμής)

### Συντακτικές Παρατηρήσεις:

- Οι αλλαγές γραμμής δεν είναι υποχρεωτικές, π.χ. είναι συντακτικά έγκυρο και το εξής:

```
selector { property1: value1; property2: value2; ... }
```

(αλλά θα προτιμούμε τη σύνταξη με αλλαγές γραμμής)

- Το ερωτηματικό της τελευταίας δήλωσης μπορεί να παραληφθεί (ωστόσο και πάλι, καλό θα είναι να το γράφουμε)

### Σύνδεση με HTML:

- Inline:** Μέσα στα στοιχεία HTML (βλ. inline.html)

```
<body>  
<h1>Hello World!</h1>  
<p style="color: red; font-size: 34px;">Welcome to my page</p>  
</body>
```

(πολύ κακός τρόπος: Μπερδένει περιεχόμενο με στυλ -> Να αποφεύγεται δια ροπάλου)

- Στο head της HTML:** Χρησιμοποιώντας το στοιχείο style (βλ. headstyle.html):

```
<head>  
...  
<style>  
body {  
...  
}  
</style>  
</head>
```

(ΟΚ τρόπος: Καλός όταν πειραματιζόμαστε και για μικρά projects)

- Σε εξωτερικό (external) αρχείο (.css):** Με το στοιχείο link του head ως εξής (βλ. project external). Αναφέρεται ως stylesheet:

```
<!-- hello.html -->  
<head>  
...  
<link rel="stylesheet" href="styles/style.css">  
</head>
```

(Ο καλύτερος και πιο συνηθισμένος τρόπος: Όταν δουλεύουμε σε πραγματικό project. Το ίδιο stylesheet χρησιμοποιείται από πολλές σελίδες)

### Επιλογείς (Selectors)

- Παρέχεται μια ποικιλία επιλογέων, με τους οποίους μπορούμε να «στοχεύσουμε», με απόλυτη λεπτομέρεια, οποιοδήποτε στοιχείο HTML.
- **Συνηθισμένοι επιλογείς είναι:**
  - **\* {...} (universal (καθολικός) επιλογέας):** Επιλέγει όλα τα στοιχεία.
  - **tag {...} (type ή element selector (επιλογέας τύπου στοιχείων)):** Επιλέγει όλα τα στοιχεία HTML που είναι του συγκεκριμένου τύπου (π.χ. h1, p, κοκ).
  - **#id {...} (επιλογέας μοναδικού στοιχείου):**
    - Βάζουμε σε ένα μοναδικό στοιχείο HTML το attribute **id="name"** (name είναι όνομα της επιλογής μας, case-sensitive, αρχίζει με γράμμα και έπειτα επιτρέπονται χαρακτήρες, αριθμοί, - και \_)
    - Ο επιλογέας **#name {...}** θα στοχεύει αυτό το μοναδικό στοιχείο

### Παράδειγμα 2: selectors1.html

```

<body>
<h1>Hello World!</h1>
<p id="first-paragraph">
  Welcome to my page</p>
<p>This is a paragraph</p>
</body>

```

```

<style>
* {
  color: red;
}
#first-paragraph {
  background-color: black;
}
</style>

```



### • **.class {...} (επιλογέας πολλών στοιχείων):**

- Βάζουμε σε όσα στοιχεία HTML θέλουμε το attribute **class="name"** (name είναι όνομα της επιλογής μας)
- Ο επιλογέας **.name {...}** θα στοχεύει αυτά τα στοιχεία.

### Παράδειγμα 4: selectors2.html

```

<body>
<h1 class="black">Hello
  World!</h1>
<p>First Paragraph</p>
<p class="black">Second
  Paragraph</p>
<p>Third Paragraph</p>
</body>

```

```

<style>
* {
  color: red;
}
.blue {
  background-color: black;
}
</style>

```



- Εφόσον σε ένα στοιχείο καθοριστεί το ίδιο property μέσω διαφορετικών επιλογέων, υπάρχουν κανόνες που ο browser ακολουθεί για να επιλέξει τη δήλωση που «νικάει»
- Για τους επιλογείς που είδαμε, ισχύει: universal < type < class < id < inline
- Υπάρχουν πιο περίπλοκοι επιλογείς που θα δούμε στο επόμενο μάθημα.

### Παράδειγμα 5: cascade.html

```

<div
  id="id"
  class="class"
  style="background-
    color: blue;">
</div>

```

```

<style>
* { background-color: turquoise; }
div { background-color: yellow; }
#id { background-color: black; }
.class { background-color: red; }
</style>

```



### Παρατήρηση:

- Αν χρειαζόταν να πηγαίνουμε σε κάθε στοιχείο (ή ομάδα στοιχείων) και αλλάζαμε τις τιμές, τα .css αρχεία θα ήταν μη διαχειρίσιμα.

### Κληρονόμηση (inheritance):

- Αν η ιδιότητα ενός στοιχείου δεν καθορίζεται από κάποιον επιλογέα, τότε η τιμή της κληρονομείται από το γονικό του στοιχείο (στον κώδικα HTML)

### Παράδειγμα 6: inheritance.html



```
<body>
<h1>Hello World!</h1>
<p>Welcome to my page</p>
</body>
```

```
body {
  font-family: "Critic Std", cursive;
  ...
}
h1 {
  border: 3px solid #1F2F16;
  ...
  text-align: center;
}
p {
  ...
  text-align: center;
}
```

### Σημειώσεις:

- Γενικά, κληρονομούνται τα περισσότερα properties.
- Π.χ. το font-family που καθορίζει τον τύπο της γραμματοσειράς κληρονομείται από το body στα h1, p
- Ωστόσο υπάρχουν properties που δεν κληρονομούνται (όπως το border που καθορίζει το είδος του πλαισίου του κουτιού).
- Σε κάθε property που θα μελετήσουμε θα αναφέρουμε αν αυτό κληρονομείται ή όχι.

### Άσκηση 1:

- Διαπιστώστε ότι το text-align (στοίχιση κειμένου στο κουτί) κληρονομείται, θέτοντας το στο body και αφαιρώντας το από τα h1, p.
- Διαπιστώστε ότι το border δεν κληρονομείται, αφαιρώντας το από το h1 και θέτοντας το στο body.

### Σημείωση:

- Οι browsers έχουν ένα default stylesheet που χρησιμοποιούν (πολλές φορές θα μας μπερδέψει, γι' αυτό χρήσιμο είναι το **reset.css** (βλ. βίντεο και project: **reset**))
- Σημαντικό: Εφόσον γράψουμε δύο φορές τιμή σε property, επικρατεί αυτή που γράψαμε πιο πρόσφατα.
- Έτσι ο browser θέτει τιμή σε κάποιο property, το reset.css την θέτει σε κάποια προβλέψιμη τιμή και έπειτα εμείς την επανακαθορίζουμε κατά τις ανάγκες μας.
- Βλέπε και βίντεο για χρήση των Developer Tools του Chrome



**Εξαιρετικά χρήσιμο:**

- Μία τιμή μπορεί να δηλωθεί πολλές φορές σε μία σελίδα μας.
- (π.χ. μπορεί σε πολλά στοιχεία, να έχουμε δηλώσει ότι το χρώμα είναι «μπλε»)
- Αν θέλουμε να αλλάξουμε το «μπλε» σε «κόκκινο», πρέπει να πάμε σε όλα τα σημεία δηλώσης και να κάνουμε την αλλαγή.
- Αντίθετα, αν **δηλώσουμε μια μεταβλητή**, με τιμή «μπλε» και θέσουμε όλα τα σημεία δήλωσης ίσα με την τιμή της μεταβλητής, τότε **αρκεί να κάνουμε την αλλαγή μόνο στην μεταβλητή** και θα αλλάξουν αυτόματα όλα τα σημεία δήλωσης.

- Ορίζουμε μια μεταβλητή με το συντακτικό:
  - **--varname: value;**
    - πρέπει υποχρεωτικά να ξεκινάει με δύο παύλες
    - το όνομα (varname) είναι της αρεσκείας μας
    - η τιμή (value) είναι μια έγκυρη τιμή ενός property (π.χ. χρώμα, μήκος κ.λπ.)
    - Τη δηλώνουμε σε μια περιοχή δήλωσης κανόνων.

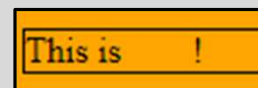
**Παρατήρηση:**

- Συνήθως τις μεταβλητές τις δηλώνουμε στον ψευδοεπιλογέα :root (βλ. επόμ. μάθημα) ώστε να είναι ορατές σε όλο το έγγραφο.
- Δηλώνουμε ότι ένα property έχει την τιμή της μεταβλητής, χρησιμοποιώντας τη συνάρτηση var():
  - **property: var(--varname);**

**Παράδειγμα 7: variables.html**

```
<body>
  <div class="class">
    This is <span id="id">blue</span>!
  </div>
</body>
```

```
:root {
  --myvar: orange;
}
body {
  background-color: var(--myvar);
}
#id {
  color: var(--myvar);
}
.class {
  border: 1px solid black;
  background-color: var(--myvar);
}
```

**Σημαντικό:**

- Η μεταβλητή κληρονομείται στα στοιχεία.
- Αν ένα στοιχείο την επαναορίσει με μία νέα τιμή, τότε αυτό και οι απόγονοί του θα βλέπουν τη νέα τιμή.

**Παράδειγμα 8: variables2.html**

```
.class {
  --myvar: blue;
  border: 1px solid black;
  background-color: var(--myvar);
}
```



# How IT people see each other



codepen.io

