



Εκπαιδευτικά Περιβάλλοντα Διαδικτύου

Ενότητα 6: CSS

Θρασύβουλος-Κωνσταντίνος Τσιάτσος
Τμήμα Πληροφορικής



Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



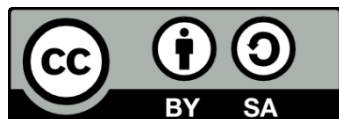
Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.





CSS



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΣΠΑ
2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Περιεχόμενα ενότητας

1. Διαδοχικά Φύλλα Στυλ (CSS)
2. Τι είναι το CSS
3. Ιστορικά στοιχεία
4. Πλεονεκτήματα χρήσης CSS
5. Χρήση των Style Sheets
6. Κανόνες Σύνταξης CSS
7. Ιδιότητες CSS



Σκοποί ενότητας

- Εξοικείωση με τις βασικές έννοιες και δυνατότητες των CSS (Διαδοχικά Φύλλα Στυλ)





ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

Διαδοχικά Φύλλα Στυλ (CSS)

Τι είναι Διαδοχικό Φύλλο Στυλ 1/4

- CSS σημαίνει Cascading Style Sheets (Διαδοχικά Φύλλα Στυλ)
- Κάθε HTML στοιχείο έχει ένα προκαθορισμένο στυλ
- Η αλλαγή του προκαθορισμένου στυλ ενός HTML στοιχείου γίνεται μέσω του style attribute
- Αυτός ο τρόπος αλλαγής του στυλ - δηλαδή μέσω της χρήσης του style attribute των στοιχείων HTML- ονομάζεται inline
- Η αλλαγή στυλ μπορεί να γίνει με ακόμα δύο τρόπους:
 - τον internal τρόπο - χρησιμοποιώντας, δηλαδή, ένα <style> element στο κομμάτι <head> του HTML εγγράφου, όπου δηλώνονται τα επιθυμητά στυλ των στοιχείων του HTML εγγράφου
 - ή με τον external τρόπο - χρησιμοποιώντας, δηλαδή, ένα ή περισσότερα εξωτερικά (external) Style Sheets τα οποία αποθηκεύονται σε αρχεία CSS



Τι είναι Διαδοχικό Φύλλο Στυλ 2/4

- Πιο συνηθισμένος και ο προτεινόμενος τρόπος για αλλαγή του στυλ μιας ιστοσελίδας
- Απλή style sheet γλώσσα η οποία περιγράφει την εμφάνιση και τη μορφοποίηση ενός εγγράφου το οποίο έχει γραφεί με χρήση γλώσσας σήμανσης
- Μπορεί να χρησιμοποιηθεί σε κάθε είδους αρχείου XML, όπως plain XML και SVG
- Η πιο συχνή χρήση του είναι για την αλλαγή του στυλ ιστοσελίδων γραμμένων σε HTML
- Μαζί με την HTML και την JavaScript, το CSS αποτελεί βασική τεχνολογία που χρησιμοποιείται κατά κόρον σήμερα για τη δημιουργία ιστοσελίδων και διεπαφών χρήστη για web και mobile applications



Τι είναι Διαδοχικό Φύλλο Στυλ 3/4

- Σχεδιάστηκε κυρίως για να διευκολύνει τον διαχωρισμό του περιεχομένου του εγγράφου από την παρουσίασή του, για λόγους προσβασιμότητας, ευελιξίας, ευκολότερης διαμοίρασης και μικρότερης πολυπλοκότητας λόγω της ελαχιστοποίησης της επανάληψης κώδικα στη δόμηση του περιεχομένου
- Δίνει στο σχεδιαστή/ προγραμματιστή τη δυνατότητα να διαχωρίσει τις οδηγίες σχετικά με το πώς θα εμφανίζονται τα διάφορα στοιχεία στην ιστοσελίδα σε ένα ξεχωριστό αρχείο ή σε μια ενότητα του εγγράφου HTML
- Έτσι, διατηρώντας τις οδηγίες στυλ σε εξωτερικά .css αρχεία, μπορεί να αλλάξει με πολύ εύκολο τρόπο την εμφάνιση πολλών ιστοσελίδων, γλιτώνοντας κόπο και χρόνο



Τι είναι Διαδοχικό Φύλλο Στυλ 4/4

- Για τον λόγο αυτό, το World Wide Web Consortium (W3C) δημιούργησε το CSS, με στόχο το διαχωρισμό όλης της μορφοποίησης από το HTML έγγραφο, και την αποθήκευσή του σε ένα ξεχωριστό αρχείο CSS. Συνεπώς, το CSS:
 - Καθορίζει το επίπεδο παρουσίασης (presentation layer) της διεπαφής χρήστη
 - Ορίζει στον περιηγητή το πώς πρέπει να παρουσιαστεί το κάθε στοιχείο στην οθόνη του χρήστη



Ιστορικά στοιχεία 1/3

- Το CSS προτάθηκε για πρώτη φορά το 1994 από τον Håkon Wium Lie, συνεργάτη του Tim Berners-Lee στο CERN
- Μετά από συζητήσεις, προέκυψε το 1996 η πρώτη W3C Recommendation για το CSS (CSS1).
- Το CSS επιτρέπει να επηρεάζεται το στυλ ενός εγγράφου από πολλαπλά style sheets (“cascading”). Κάποιες από τις δυνατότητες που παρείχε η CSS1 προδιαγραφή ήταν:
 - Υποστήριξη ιδιοτήτων γραμματοσειράς
 - Προσδιορισμός χρώματος γραμματοσειράς, φόντου και άλλων στοιχείων
 - Διευθέτηση στοίχισης γραμματοσειράς, εικόνων, πινάκων, κλπ.



Ιστορικά στοιχεία 2/3

- Για τον λόγο αυτό, το W3C ξεκίνησε την αναθεώρηση της προδιαγραφής CSS2 και τη δημιουργία της προδιαγραφής CSS 2.1
- Η δημιουργία της CSS3 προδιαγραφής άρχισε το 1998 και είναι υπό ανάπτυξη
- Ενώ η CSS2 είναι μια μεγάλη προδιαγραφή η οποία ορίζει διάφορα χαρακτηριστικά, η CSS3 χωρίζεται σε πολλά ξεχωριστά αρχεία (“modules”)
- Κάθε ένα από αυτά τα modules προσθέτει ή επεκτείνει χαρακτηριστικά ήδη ορισμένα στην CSS2, διατηρώντας έτσι συμβατότητα “προς τα πίσω” (backwards compatibility)



Ιστορικά στοιχεία 3/3

Έκδοση	Χαρακτηριστικά
CSS 1	<p>Έγινε επίσημη σύσταση (recommendation) του W3C τον Δεκέμβριο του 1996. Αρχισε να χρησιμοποιείται στους περισσότερους περιηγητές από το 2000 και μετά.</p> <p>Ανάμεσα σε άλλα υποστήριξε:</p> <ul style="list-style-type: none">• Στοιχισι κειμένου, εικόνων, πινάκων, κλπ.• Ιδιότητες γραμματοσειράς• Μοναδικά αναγνωριστικά (ids) για στοιχεία• Κλάσεις για ομάδες ιδιοτήτων
CSS 2	<p>Δημοσιεύτηκε ως W3C Recommendation τον Μάιο του 1998, αλλά δεν υποστηρίχθηκε ποτέ πλήρως.</p> <p>Κάποιες νέες δυνατότητες που εισήγαγε:</p> <ul style="list-style-type: none">• Τύπους μέσων (media types)• z-index• Bidirectional text• Νέες ιδιότητες για γραμματοσειρές (πχ. σκιά - shadow)
CSS 2.1	<p>Δημοσιεύτηκε τον Ιούνιο του 2005.</p> <ul style="list-style-type: none">• Διορθώνει λάθη της προδιαγραφής CSS2• Αφαιρεί στοιχεία που δεν υποστηρίζονται πλήρως από τους περιηγητές• Επεκτείνει τα χαρακτηριστικά που υποστηρίζονται πλήρως
CSS 3.0	<p>Υπό ανάπτυξη. Βασικά χαρακτηριστικά:</p> <ul style="list-style-type: none">• Χωρίζεται σε modules.• Αποτελείται από πολλές διαφορετικές προδιαγραφές• Εισάγει νέες δυνατότητες και εφέ, όπως: animations, transitions, gradients, multiple backgrounds, 3D transforms, κλπ.• Παρέχει τη δυνατότητα για advanced selectors



Πλεονεκτήματα χρήσης CSS 1/2

- Μεγαλύτερη ευελιξία, καλύτερα αποτελέσματα: με χρήση του CSS είναι πλέον δυνατές μορφοποιήσεις δύσκολες ή αδύνατες με χρήση μόνο HTML
- Γρηγορότερες ιστοσελίδες: χρησιμοποιώντας εξωτερικό αρχείο HTML, ο περιηγητής το αποθηκεύει στην μνήμη cache την πρώτη φορά που φορτώνει κάποια ιστοσελίδα ενός ιστοτόπου, οπότε δε χρειάζεται να το κατεβάσει κάθε φορά που ο χρήστης επισκέπτεται μια ιστοσελίδα του ιστοτόπου
- Η συντήρηση αλλά και η διαμοίραση των ιστοσελίδων είναι πλέον πολύ ευκολότερη, καθώς η εμφάνιση μιας ολόκληρης ιστοσελίδας μπορεί να ελέγχεται από ένα ή περισσότερα εξωτερικά αρχεία CSS
 - Σε περίπτωση που ο σχεδιαστής της ιστοσελίδας θέλει να αλλάξει κάτι στην εμφάνιση κάποιου στοιχείου, αρκεί να αλλάξει το αρχείο CSS. Αυτό είναι ιδιαίτερα χρήσιμο στην περίπτωση που πρέπει να γίνει αλλαγή ενός στοιχείου που υπάρχει σε διάφορα σημεία στην ιστοσελίδα, καθώς χρειάζεται πλέον μια μόνο αλλαγή στο CSS.



Πλεονεκτήματα χρήσης CSS 2/2

- Ο κώδικας HTML είναι πλέον λιγότερο πολύπλοκος, πιο ευανάγνωστος και συντομότερος, δεδομένου ότι οι κανόνες μορφοποίησης γράφονται μόνο μια φορά και εφαρμόζονται με ιεραρχικό τρόπο (“precedence rules”)
- Ο σχεδιαστής της ιστοσελίδας έχει μεγαλύτερο έλεγχο όσον αφορά στο στυλ της ιστοσελίδας, ενώ μπορεί να χρησιμοποιεί πολλαπλά CSS αρχεία για μια ιστοσελίδα
- Καλύτερο Search Engine Optimization (SEO): το περιεχόμενο είναι ανεξάρτητο από τη μορφοποίησή του, οπότε είναι ευκολότερο να γίνει καταγραφή και αρχειοθέτηση (indexing) από τις μηχανές αναζήτησης



Τι ονομάζεται “Cascade” στο CSS

- Αναφέρεται στο συνδυασμό:
 - των προκαθορισμένων στυλ του περιηγητή
 - των εξωτερικών style sheets
 - και των inline, internal (ή embedded) και user-defined styles
- ...σε ένα style sheet
- Στο CSS επιτρέπεται τα στοιχεία να κληρονομούν στυλ από στοιχεία - γονείς. Ένα στοιχείο - παιδί κληρονομεί το στυλ του από το στοιχείο - γονέα, εκτός κι αν οριστεί ένα συγκεκριμένο στυλ για το στοιχείο - παιδί.
- Το “Cascade” ορίζει ότι κάθε ξεχωριστό στυλ έχει διαφορετική προτεραιότητα η οποία επηρεάζει την κληρονομικότητα του CSS.
- Τα στυλ μπορούν να ταξινομηθούν ως προς την προτεραιότητά τους σε φθίνουσα σειρά, ως εξής:
 - Inline styles
 - Internal styles
 - External style sheet
 - Browser default



Χρήση των Style Sheets 1/5

- **Internal Styles:**

- Χρησιμοποιείται στην περίπτωση που μία μόνο ιστοσελίδα ενός ιστοτόπου έχει διαφορετικό στυλ από τις υπόλοιπες σελίδες
- Μπορεί, έτσι, ο σχεδιαστής της ιστοσελίδας να εφαρμόσει κάποιους κανόνες CSS μόνο στη συγκεκριμένη ιστοσελίδα ή να τροποποιήσει τους υπάρχοντες κανόνες CSS του ιστοτόπου
- Ορίζεται στο κομμάτι “head” της ιστοσελίδας, μέσα στην ετικέτα <style>
- ```
<style type="text/css">
body {}
</style>
```



# Χρήση των Style Sheets 2/5

- **Inline Styles:**

- Το inline style ενός μόνο στοιχείου ορίζεται μέσω της ιδιότητας “style” της σχετικής ετικέτας
- Χρησιμοποιείται όταν στόχος είναι η μορφοποίηση ενός μόνο στοιχείου της ιστοσελίδας και η εν λόγω μορφοποίηση δε θα χρειαστεί για κάτι άλλο στην ιστοσελίδα
- Για παράδειγμα, στον ακόλουθο κώδικα ορίζεται ότι το μέγεθος της γραμματοσειράς στη συγκεκριμένη παράγραφο είναι 12px:
- `<p style="font size: 12px"> MyParagraph!</p>`



# Χρήση των Style Sheets 3/5

- **External Style Sheet:**

- Ιδανικό για εφαρμογή του στυλ σε πολλές σελίδες
- Κάθε ιστοσελίδα θα πρέπει να συμπεριλαμβάνει έναν σύνδεσμο προς το style sheet
- Αυτό γίνεται με χρήση της ετικέτας <link>, η οποία πρέπει να είναι στο κομμάτι “head” της ιστοσελίδας
- Αποτελεί τη σωστότερη χρήση CSS
- Παράδειγμα χρήσης:
- <head>  
    <link rel="stylesheet" type="text/css" href="style.css">  
</head>



# Χρήση των Style Sheets 4/5

- Το αρχείο style.css μπορεί να γραφεί σε οποιονδήποτε text editor
- Πρέπει να μην περιέχει ετικέτες HTML και να αποθηκευθεί με κατάληξη “.css”
- Παραδείγματος χάρη, μπορεί να είναι το παρακάτω:

```
body {
 background-color: blue;
}
h1 {
 color: black;
 margin-left: 25px;
}
```



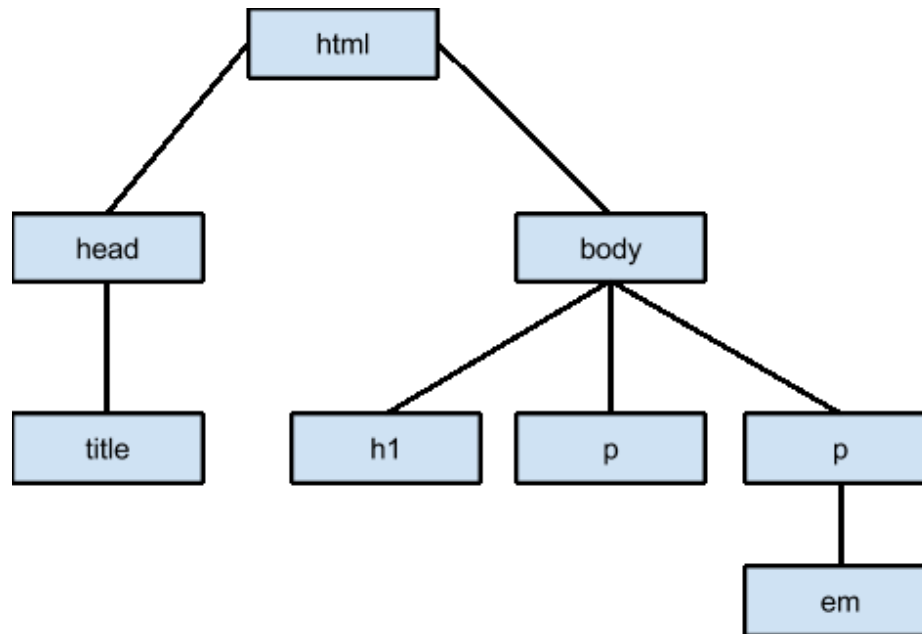
# Χρήση των Style Sheets 5/5

- Μπορεί, επίσης, να γίνεται αναφορά σε πολλαπλά external style sheets στο ίδιο HTML έγγραφο.
- Μεγαλύτερη προτεραιότητα έχει αυτό που έχει γραφεί τελευταίο.
- **Ποιό στυλ θα χρησιμοποιηθεί εάν ορίζονται περισσότερα από ένα στυλ για κάποιο HTML στοιχείο;**
- Μεγαλύτερη προτεραιότητα έχουν τα inline styles (μέσα σε ετικέτα στοιχείου HTML), οπότε θα παρακάμψει (override) το style που ορίζεται στην ετικέτα <head> (internal style)
- Έπειτα, θα παρακάμψει το style που ορίζεται σε external style sheets, αλλά και στο προκαθορισμένο στυλ του περιηγητή (default)



# Δένδρο εγγράφου (Document Tree)

- Κάθε HTML και XHTML έγγραφο είναι ένα Document Tree.
- Όταν ο περιηγητής φορτώνει μια ιστοσελίδα, δημιουργεί το λεγόμενο “Document Object Model” της ιστοσελίδας, το οποίο κατασκευάζεται ως δέντρο από αντικείμενα.
- Στο ακόλουθο παράδειγμα αναλύεται η δομή ενός Document Tree:



Εικόνα 1



# Κληρονομικότητα 1/3

- 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`
- Έτσι διαμορφώνεται μια ιεραρχία, η οποία αποτελεί το DOM Tree (Document Object Model tree)
- Τα στοιχείο - απόγονος, λοιπόν, μπορεί να κληρονομήσει μια ιδιότητα από τον γονέα του
- Αυτό συμβαίνει στην περίπτωση που δεν έχει οριστεί κάποιο συγκεκριμένο στυλ για το στοιχείο - παιδί





# Κληρονομικότητα 2/3

- Κάποιες ιδιότητες CSS, οι οποίες μπορούν να κληρονομηθούν και σε στοιχεία απογόνους είναι οι εξής:
  - Color
  - font-size
  - font-family
  - font-style
  - font-weight
  - text-align
  - text-decoration



# Κληρονομικότητα 3/3

- Ο σχεδιαστής της ιστοσελίδας μπορεί να ορίσει μια διαφορετική τιμή μέσω κανόνα CSS για μια συγκεκριμένη ιδιότητα ενός στοιχείου
- Στην περίπτωση αυτή, θα εφαρμοστεί αυτή η τιμή και όχι η κληρονομημένη (inherited) τιμή του στοιχείου
- Εάν θέλουμε ένα στοιχείο να αποκτήσει για μια ιδιότητα την τιμή της ιδιότητας αυτής του γονέα του, ενώ αυτή κανονικά δεν κληρονομείται, τότε υπάρχει η δυνατότητα να οριστεί ως τιμή της ιδιότητας για το στοιχείο αυτό η τιμή “inherit”



# Κανόνες Σύνταξης CSS 1/3

- Ένας κανόνας (rule) CSS αποτελείται από δύο μέρη:
  - τον επιλογέα (CSS selector)
  - και το μπλοκ δήλωσης (declaration block)
- Ο επιλογέας δείχνει στο στοιχείο που θα μορφοποιηθεί
- Το μπλοκ δήλωσης αφορά στο πώς αυτό θα μορφοποιηθεί. Περιέχει όνομα ιδιότητας (CSS property) και τιμή
- Οι ιδιότητες διαχωρίζονται με “;”. Η σύνταξη είναι η ακόλουθη:
  - selector { property: value;}



# Κανόνες Σύνταξης CSS 2/3

- Ο επιλογέας μπορεί να περιέχει και πολλαπλές ιδιότητες:
  - `selector { property1: value1; property2: value2; }`
- Κάθε μπλοκ δήλωσης περιέχεται σε αγκύλες και κάθε δήλωση πρέπει να τερματίζεται με “;”
- Για παράδειγμα, ο κώδικας για να οριστεί ότι όλα τα στοιχεία `<p>` του συγκεκριμένου εγγράφου θα έχουν στοίχιση στο κέντρο και κείμενο χρώματος μαύρο, είναι ο ακόλουθος:
  - `p {color:black;text align:center;}`



# Κανόνες Σύνταξης CSS 3/3

- Για λόγους αναγνωσιμότητας, μπορεί να γραφεί μία μόνο δήλωση ανά γραμμή. Το προηγούμενο, δηλαδή, παράδειγμα μπορεί να ξαναγραφεί ως εξής:

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

- Τέλος, μπορεί να γίνει δήλωση ιδιοτήτων για πολλαπλούς επιλογείς. Αντί, δηλαδή, για τον ακόλουθο κώδικα:

```
– h1 h2 h3 { color: yellow; }
```



# Σχόλια CSS

- Τα σχόλια αποτελούν τμήματα κώδικα που “παραβλέπονται” από τον περιηγητή και χρησιμεύουν στην επεξήγηση του κώδικα
- Στο CSS η ανάγκη για σχόλια δεν είναι τόσο μεγάλη, αλλά μπορούν να χρησιμοποιηθούν για να διαχωρίζουν νοηματικά το αρχείο και για να παρέχουν χρήσιμες πληροφορίες
- Τα σχόλια εισάγονται μέσα σε `/* ... */`
- Μπορούν να εκτείνονται σε περισσότερες από μία γραμμές



# Επιλογείς CSS

- Οι επιλογείς CSS (CSS selectors) επιτρέπουν την επιλογή των HTML στοιχείων στα οποία θα εφαρμοστεί κάποια μορφοποίηση
- Χρησιμοποιούνται για να επιλεγούν στοιχεία βάσει του τύπου τους, της κλάσης τους, του id τους, κ.α.
- Ουσιαστικά, αποτελούν *patterns* για να αντιστοιχίζονται (match) στοιχεία στο document tree, το οποίο παρουσιάστηκε σε προηγούμενη υποενότητα
- Η προδιαγραφή του CSS ορίζει διάφορα είδη επιλογέων
- Συνοπτικά, οι βασικότεροι επιλογείς είναι οι ακόλουθοι:



# Επιλογέας “στοιχείο”

- Επιλογέας “στοιχείο” (element selector ή type selector):
  - Αποτελείται μόνο από το όνομα μιας ετικέτας HTML
  - Επιλέγει το όνομα του στοιχείου βάσει του τύπου του (element type)
  - Για παράδειγμα, έστω ότι πρέπει να εφαρμοστεί το ίδιο στυλ σε όλα τα στοιχεία `<p>` μιας ιστοσελίδας
  - Θα επιλεγούν με ακόλουθο τον ακόλουθο τρόπο:

```
p {

 text-align: center;
 color: black;
}
```





# Επιλογέας “\*”

- Επιλογέας “\*” (universal selector):
  - ο επιλογέας αυτός “ταιριάζει” το όνομα κάθε στοιχείου
  - Οι ιδιότητες που θα συμπεριληφθούν σε αυτόν τον κανόνα CSS θα εφαρμοστούν σε κάθε στοιχείο της ιστοσελίδας
  - Χρησιμοποιείται, συνήθως, σε συνδυασμό με άλλους επιλογείς
  - Στο ακόλουθο παράδειγμα, όλοι οι απόγονοι (descendants) του h1 θα έχουν κόκκινο χρώμα κειμένου:
    - `h1 * { color : red }`



# Επιλογέας “id”

- Επιλογέας “id” (id selector):
  - Χρησιμοποιεί την ιδιότητα “id” ενός HTML στοιχείου για να επιλέξει ένα συγκεκριμένο στοιχείο
  - Το id θα πρέπει να είναι μοναδικό, δηλαδή δε θα πρέπει να υπάρχουν δύο ή περισσότερα στοιχεία στην ιστοσελίδα μας με το ίδιο id
  - Η χρήση των ids είναι παρόμοια με τη χρήση κλάσεων
  - Είναι προτιμότερη η χρήση ids όταν το στοιχείο στο οποίο θα εφαρμοστεί η μορφοποίηση είναι μοναδικό
  - Επίσης, αξίζει να σημειωθεί ότι το id έχει υψηλότερη προτεραιότητα σε σχέση με τις κλάσεις
  - Για την επιλογή του επιθυμητού στοιχείου, γράφουμε τον χαρακτήρα “#”, ακολουθούμενο από το id του στοιχείου
  - Το id δε μπορεί να ξεκινάει με αριθμό. Για παράδειγμα:
    - `#paragraph1 {`  
    `text-align: center;`  
    `color: blue;`  
    `}`



# Επιλογέας “κλάση” 1/2

- Επιλογέας “κλάση” (class selector):
  - Ο επιλογέας αυτός ξεκινάει με μια τελεία “.”, ακολουθούμενη από το όνομα της επιθυμητής κλάσης
  - Ο περιηγητής “ταιριάζει” όσα στοιχεία στην ιστοσελίδα έχουν αυτήν την κλάση στην ιδιότητά τους “class”, και εφαρμόζει μορφοποίηση βάσει των ιδιοτήτων που θα γράψουμε στον κανόνα CSS
  - Ένα στοιχείο μπορεί να ανήκει σε περισσότερες από μια κλάσεις, δηλωμένες στην ιδιότητά του class με κενά ανάμεσά τους
  - Τα ονόματα των κλάσεων δε μπορούν να ξεκινούν από αριθμούς, όπως και τα ids
  - Με τη χρήση κλάσεων, γίνεται εφικτή η ομαδοποίηση στοιχείων HTML, ώστε να τους αποδοθεί μετά συγκεκριμένη μορφοποίηση μέσω CSS
  - Δεν υπάρχει περιορισμός για το πόσο συχνά θα χρησιμοποιηθεί μια κλάση σε μία ιστοσελίδα
  - Στο παρακάτω παράδειγμα, όλα τα στοιχεία με class="center" θα έχουν στοίχιση στο κέντρο:

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



# Επιλογέας “κλάση” 2/2

- Είναι, επίσης, εφικτό να δηλωθεί ότι μόνο συγκεκριμένα στοιχεία κάποιας κλάσης θα πρέπει να επηρεάζονται από τον CSS κανόνα
- Στο επόμενο παράδειγμα θα εφαρμόζεται στοίχιση στο κέντρο στα στοιχεία με ετικέτα <p> και ιδιότητα class=“center”

```
p.center {
 text-align: center;
}
```



# Επιλογέας “απόγονος”

- Επιλογέας “απόγονος” (descendant selector): επιλέγει στοιχεία που είναι απόγονοι άλλων στοιχείων στο document tree.
- Παράδειγμα:  

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



# Επιλογείς

- Μπορούμε να συνδυάσουμε περισσότερους από έναν επιλογείς, δημιουργώντας σύνθετους επιλογείς CSS.
- Υπάρχουν διάφοροι τρόποι συνδυασμού επιλογέων. Για παράδειγμα:
  - `selector1, selector2, selector3, ...` : εφαρμόζεται σε κάθε στοιχείο που πληροί τις προϋποθέσεις επιλογής για κάποιον από τους επιλογείς που χρησιμοποιήθηκαν
  - `selector1, selector2, selector3, ... , selectorn` : εφαρμόζεται σε κάθε στοιχείο που πληροί τις προϋποθέσεις επιλογής του `selectorn`, το οποίο βρίσκεται σε στοιχείο που πληροί τις προϋποθέσεις του `selectorn-1`, κ.ο.κ



# Ψευδο-κλάσεις & ψευδο-στοιχεία

## 1/4

- Με χρήση των ψευδο-κλάσεων (pseudo-classes) είναι δυνατή η επιλογή στοιχείων βάσει χαρακτηριστικών άλλων πέρα από το όνομά τους, τις ιδιότητές τους ή το περιεχόμενό τους
- Για παράδειγμα, μπορούν να χρησιμοποιηθούν για να ορίσουμε διαφορετική μορφοποίηση για έναν υπερσύνδεσμο που έχουμε επισκεφθεί (visited) και για έναν υπερσύνδεσμο που δεν έχουμε επισκεφθεί (unvisited)
- Ουσιαστικά, παρέχουν τη δυνατότητα για μορφοποίηση βασισμένη σε πληροφορία που βρίσκεται εκτός του Document Tree
- Επιτρέπονται οπουδήποτε σε επιλογείς, και τα ονόματά τους είναι case-insensitive



# Ψευδο-κλάσεις & ψευδο-στοιχεία

## 2/4

- Σύνταξη:

```
selector:pseudo-class {
 property:value;
}
```

```
/* unvisited link */
```

```
a:link {
 color: #FFCCCC;
}
```

```
/* visited link */
```

```
/* unvisited link */
```

```
a:link {
 color: #FFCCCC;
}
```

```
/* visited link */
```

```
a:visited {
 color: #1A1414;
}
```





# Ψευδο-κλάσεις & ψευδο-στοιχεία

## 3/4

- Οι συνηθέστερα χρησιμοποιούμενες ψευδο-κλάσεις είναι οι εξής:
  - a:link (χρησιμοποιείται για στοιχεία τύπου a)
  - a:visited (για στοιχεία τύπου a που έχει ήδη επισκεφθεί ο χρήστης)
  - a:active (για στοιχεία τύπου a, τη στιγμή που ο χρήστης έχει πατημένο το ποντίκι πάνω σε αυτά)
  - στοιχείο E:hover (χρησιμοποιείται για τα στοιχεία E τη στιγμή που ο χρήστης κάνει “hover”, δηλαδή έχει το δείκτη του ποντικιού πάνω σε κάποιο από αυτά )
  - στοιχείο E:focus (χρησιμοποιείται κυρίως για στοιχεία φορμών και εφαρμόζεται στα στοιχεία E που εκείνη τη στιγμή έχουν “focus”)
  - στοιχείο E: empty (επιλέγει κάθε στοιχείο E που δεν έχει παιδιά)



# Ψευδο-κλάσεις & ψευδο-στοιχεία

## 4/4

- Μπορεί να γίνει χρήση pseudo-element για να μορφοποιηθούν συγκεκριμένα μέρη ενός στοιχείου, όπως για παράδειγμα μόνο το πρώτο γράμμα ενός στοιχείου
- Η σύνταξη των ψευδο-στοιχείων είναι παρόμοια με αυτή των ψευδο-κλάσεων, δηλαδή:

```
selector::pseudo-element {
 property:value;
}
```

- Παράδειγμα:

```
p::first-letter {
 color: #ff0000;
 font-size: large;
}
```



# Ιδιότητες CSS για φόντο 1/4

- **background-color:** καθορίζει το χρώμα φόντου ενός στοιχείου. Παράδειγμα χρήσης

```
body {
 background-color: #000000;
}
```

- **background-image:** ορίζει μια εικόνα φόντου. Δέχεται ως τιμή τη διεύθυνση της εικόνας σε παρενθέσεις, με “url” μπροστά από αυτή
  - Από προεπιλογή, η εικόνα επαναλαμβάνεται για να καλύπτει όλο το στοιχείο στο οποίο θα είναι εικόνα φόντου

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



# Ιδιότητες CSS για φόντο 2/4

- **background-repeat**: από προεπιλογή, η εικόνα θα επαναλαμβάνεται τόσο στον οριζόντιο, όσο και στον κάθετο άξονα (αυτό δεν είναι πάντα επιθυμητό). Οι τιμές που παίρνει είναι:
  - no-repeat: καμία επανάληψη
  - repeat-x: η εικόνα επαναλαμβάνεται στον οριζόντιο άξονα
  - repeat-y: η εικόνα επαναλαμβάνεται στον κάθετο άξονα
  - repeat: η εικόνα επαναλαμβάνεται και οριζόντια και κάθετα (προεπιλογή)

```
body {
 background-image: url("bg.png");
 background-repeat: repeat-x;
}
```



# Ιδιότητες CSS για φόντο 3/4

- **background-position:** η ιδιότητα αυτή χρησιμοποιείται για να οριστεί το που θα τοποθετηθεί η εικόνα φόντου που ορίστηκε προηγουμένως για εικόνα φόντου, τόσο στον κάθετο όσο και στον οριζόντιο άξονα. Οι τιμές που παίρνει συνήθως είναι:

- left, right, center: οριζόντια θέση εικόνας
- top, bottom, center: κάθετη θέση εικόνας

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



# Ιδιότητες CSS για φόντο 4/4

- Είναι, επίσης, δυνατό να οριστεί και αριθμητικά η θέση της εικόνας φόντου, με ποσοστό επί τις εκατό (%) ή με κάποια μονάδα μεγέθους (pixels):
  - background position: 20% 35%;
  - background position: 100px 30px;
- Μπορεί να χρησιμοποιηθεί η ιδιότητα **background**, για τον ορισμό όλων των ιδιοτήτων που είδαμε παραπάνω σε μία δήλωση. Για παράδειγμα:

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



# Ιδιότητες CSS για κείμενο 1/7

- **color:** η ιδιότητα αυτή μας επιτρέπει να αλλάξουμε το χρώμα του κειμένου. Το προεπιλεγμένο χρώμα κειμένου ορίζεται στον body selector:

```
body {
 color: blue; }
```

```
h1 {
 color: #000000 }
```

- **font-size:** αφορά στο μέγεθος της γραμματοσειράς. Εάν το μέγεθος της γραμματοσειράς δεν καθοριστεί, τότε παίρνει την προεπιλεγμένη τιμή, η οποία είναι 16px. Προτείνεται η χρήση των κατάλληλων ετικετών της HTML, όπως <h1> ... <h6> για κεφαλίδες και <p> για παραγράφους.

```
p {
 font-size: 20px;
}
```



# Ιδιότητες CSS για κείμενο 2/7

- **font-family:** για τον ορισμό των γραμματοσειρών που θα χρησιμοποιηθούν για το κείμενο
- Ορίζουμε την επιθυμητή γραμματοσειρά για το κείμενο, καθώς και εναλλακτικές επιλογές, για την περίπτωση που η γραμματοσειρά που ορίστηκε δεν υπάρχει στον υπολογιστή του χρήστη
- Αν δεν υπάρχει καμία από όλες τις γραμματοσειρές που ορίστηκαν, τότε ο περιηγητής επιλέγει κάποια που να ανήκει στη γενική οικογένεια γραμματοσειρών (generic font-family), η οποία ορίζεται τελευταία με την κατάλληλη λέξη-κλειδί
- Για παράδειγμα, μπορεί να οριστεί η γενική οικογένεια “serif”, η οποία αποτελείται από γραμματοσειρές με πατούρες, όπως η Garamond και η Georgia
- Η σύνταξη της ιδιότητας αυτής είναι η ακόλουθη:
  - font family: font|initial|inherit;





# Ιδιότητες CSS για κείμενο 3/7

- **font-weight:** ορίζει το “βάρος” της γραμματοσειράς.
- Συνήθως, οι περισσότερες γραμματοσειρές διατίθενται σε δύο βάρη (κανονικό και έντονο), οπότε ουσιαστικά ορίζει αν το κείμενο θα είναι έντονο ή όχι.
- Η προεπιλεγμένη τιμή είναι “normal”.
- Οι τιμές που μπορεί να πάρει είναι normal, bold, bolder, lighter, initial, inherit, καθώς και 100, 200, 300, 400, 500, 600, 700, 800, 900, όπου το 400 ισοδυναμεί με normal text και το 700 με bold text.
- Παράδειγμα χρήσης:  

```
p.thick {

font-weight: bold; }
```



# Ιδιότητες CSS για κείμενο 4/7

- **font-style:** η ιδιότητα αυτή ορίζει αν η γραμματοσειρά θα είναι κανονική (normal) ή πλάγια (italic)
- Οι τιμές που παίρνει είναι τρεις: normal, italic και oblique
- Ενώ οι δύο τελευταίες έχουν συνήθως το ίδιο αποτέλεσμα, η χρήση της oblique δε συνίσταται λόγω ασυμβατότητας με παλαιότερους περιηγητές
- Παράδειγμα χρήσης:

```
p.italic {
```

```
font-style: italic; }
```



# Ιδιότητες CSS για κείμενο 5/7

- **font:** Όπως η ιδιότητα background που αναλύθηκε παραπάνω, έτσι και η ιδιότητα font δίνει τη δυνατότητα να οριστούν ταυτόχρονα πολλές ιδιότητες που αφορούν σε κείμενο
- **text-align:** ορίζει τη στοίχιση του κειμένου.
  - Οι πιθανές τιμές της είναι left, right, center, justify, initial και inherit

font: italic bold 20px Calibri, Trebuchet MS, Verdana, sans serif;

```
h2 {
text-align: left; }
```



# Ιδιότητες CSS για κείμενο 6/7

- **text-decoration:** δίνει τη δυνατότητα προσθήκης διακόσμησης στο κείμενο. Οι τιμές που παίρνει είναι οι ακόλουθες:
  - none: καμία διακόσμηση
  - underline: υπογράμμιση
  - overline: γραμμή πάνω από το κείμενο
  - line-through: διαγράμμιση
  - initial: προεπιλεγμένη τιμή
  - inherit: τιμή που κληρονομείται από στοιχείο - γονέα
  - Μπορούν να εφαρμοστούν παραπάνω από μία τιμές για το ίδιο κείμενο.



# Ιδιότητες CSS για κείμενο 7/7

- **text-justify:** για τον ορισμό των κενών ανάμεσα στις λέξεις.
- Παίρνει διάφορες τιμές, όπως:
  - auto (ο αλγόριθμος για το justification επιλέγεται από τον περιηγητή)
  - inter-word (αυξάνει/μειώνει το κενό ανάμεσα σε λέξεις)
  - kashida (επιμηκύνει χαρακτήρες), κλπ.

```
div {

 text-align: justify;

 text-justify: inter-word;

}
```



# Ιδιότητες CSS για περίγραμμα 1/7

- **border-style:** ορίζει το στυλ του περιγράμματος. Παίρνει διάφορες τιμές, με πιο συχνές τις ακόλουθες:
  - solid: συμπαγές περίγραμμα (μια ενιαία γραμμή)
  - dashed: διακεκομμένο περίγραμμα
  - dotted: περίγραμμα που αποτελείται από τελείες
  - double: διπλό περίγραμμα
  - groove: 3D grooved περίγραμμα
  - η ιδιότητα αυτή μπορεί να πάρει έως τέσσερις τιμές που αντιστοιχούν στα τέσσερα περιγράμματα ενός στοιχείου, με τη σειρά (top, right, bottom και left).

```
div {
 border-width:5px;
 border-style:groove;
}
```



# Ιδιότητες CSS για περίγραμμα 2/7

- **border-color:** χρησιμεύει για τον ορισμό του χρώματος του περιγράμματος
  - Το χρώμα ορίζεται με έναν από τους τρόπους που περιγράφηκαν παραπάνω
  - Αν δεν οριστεί η ιδιότητα border-color, χρησιμοποιείται το χρώμα που ορίστηκε με την ιδιότητα color
  - Η ιδιότητα border-color πρέπει να οριστεί αφού πρώτα οριστεί το περίγραμμα με την ιδιότητα border-style
  - Μπορεί να πάρει τέσσερις τιμές, που αντιστοιχούν στα τέσσερα borders ενός στοιχείου, με την εξής σειρά: top, right, bottom και left

```
div {
 border-style:solid;
 border-color: #ff0000 #0000ff;
}
```



# Ιδιότητες CSS για περίγραμμα 3/7

- **border-bottom**: Υπάρχουν πάρα πολλές ιδιότητες για τη δήλωση επιμέρους περιγραμμάτων ενός στοιχείου
  - Μια από αυτές είναι η `border-bottom`, η οποία χρησιμεύει για τη δήλωση όλων των ιδιοτήτων του `bottom border` σε μία μόνο δήλωση
  - Οι ιδιότητες που μπορούν να οριστούν, είναι, με τη σειρά:
    - `border-bottom-width`
    - `border-bottom-style`
    - `border-bottom-color`
    - ... συν τις τιμές `initial` και `inherit`.





# Ιδιότητες CSS για περίγραμμα 4/7

```
<html>
<head>
<style>

p {

 border-style: solid;
 border-bottom: thick dotted #ff0000;
}

</style>
</head>
<body>

<p>Bottom border example!</p>

</body>
</html>
```



Εικόνα 2



# Ιδιότητες CSS για περίγραμμα 5/7

- **border-width:** η ιδιότητα αυτή ορίζει το πάχος του περιγράμματος, σε κάποια από τις μονάδες μέτρησης που χρησιμοποιούνται στο CSS (πχ. pixels)
  - Επιπλέον, μπορεί να οριστεί με λέξεις-κλειδιά, όπως medium, thin και thick
  - Όπως και άλλες ιδιότητες που επεξηγήθηκαν παραπάνω, έτσι κι αυτή μπορεί να πάρει έως τέσσερις τιμές (για κάθε ένα από τα τέσσερα περιγράμματα του στοιχείου)
- **border:** Η ιδιότητα border αποτελεί συντόμευση για να οριστούν με μία μόνο δήλωση όλες οι παραπάνω ιδιότητες που σχετίζονται με το περίγραμμα
  - Είναι, δηλαδή, αντίστοιχη των ιδιοτήτων font και background



# Ιδιότητες CSS για περίγραμμα 6/7

- **padding**: ορίζει το κενό μεταξύ των ορίων ενός στοιχείου και το περιεχομένου του
- Είναι απαραίτητο να ορίζεται ώστε να μην ενώνεται το περιεχόμενο του στοιχείου με το περίγραμμά του
- Μπορεί να οριστεί διαφορετικό padding ανά πλευρά, όπως στην ιδιότητα border, χρησιμοποιώντας έως τέσσερις τιμές με τη σειρά που παρουσιάστηκε παραπάνω
  - padding: 0px 6px 6px 6px;



# Ιδιότητες CSS για περίγραμμα 7/7

- **margin**: ορίζει τον χώρο μεταξύ των ορίων ενός στοιχείου και όσων το περιβάλλουν
- Είναι, δηλαδή, το αντίθετο από το padding
- Είναι ιδιαίτερα χρήσιμο για εικόνες, ώστε να μην ενώνονται με το κείμενο γύρω από αυτές
- Όπως και προηγουμένως, μπορούν κι εδώ να οριστούν διαφορετικές τιμές margin ανά πλευρά
  - margin: 6px 4px;



# Ιδιότητες CSS για περίγραμμα 7/7

- **float:** επιτρέπει στα στοιχεία που περιβάλλουν το επιθυμητό στοιχείο να “ρέουν” γύρω του, όπως συμβαίνει, για παράδειγμα, στις εικόνες που συνοδεύουν ένα κείμενο
  - Δέχεται τις τιμές right, left και none
- **width, height:** ορίζουν το πλάτος και το ύψος ενός στοιχείου
  - Παίρνουν τιμές είτε επί τις εκατό (%), είτε σε κάποια μονάδα μήκους (πχ. pixels)
  - width: 100px;
  - height: 90%;



# Βιβλιογραφία

- Τσιάτσος Θρασύβουλος. Εκπαιδευτικά Περιβάλλοντα Διαδικτύου. Ηλεκτρονικά Ακαδημαϊκά Συγγράμματα και Βοηθήματα για Επιστήμες Μηχανικών και Πληροφορική, 2015.



# Σημείωμα Χρήσης Έργων Τρίτων

---

- Το Έργο αυτό κάνει χρήση των ακόλουθων έργων:
- Εικονες 1-2, Στιγμιότυπο από υπολογιστή, από το προσωπικό αρχείο του συγγραφέα



# Σημείωμα Αναφοράς

Copyright Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Θρασύβουλος-Κων/νος Τσιάτσος. «Εκπαιδευτικά Περιβάλλοντα Διαδικτύου. Εικονικά μαθησιακά περιβάλλοντα». Έκδοση: 1.0. Θεσσαλονίκη 2014. Διαθέσιμο από τη δικτυακή διεύθυνση: <https://opencourses.auth.gr/courses/OCRS487/>





# Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά - Μη Εμπορική Χρήση - Όχι Παράγωγα Έργα 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

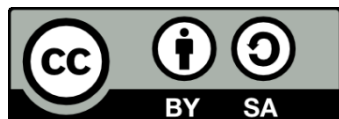
[1] <http://creativecommons.org/licenses/by-nc-nd/4.0/>





# Τέλος ενότητας

Επεξεργασία: <Στέργιος Τέγος>  
Θεσσαλονίκη, <26/05/2015>



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



ΑΡΙΣΤΟΤΕΛΕΙΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΘΕΣΣΑΛΟΝΙΚΗΣ

---

# Σημειώματα

# Σημείωμα Ιστορικού Εκδόσεων Έργου

---

Το παρόν έργο αποτελεί την έκδοση 1.00.



# Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.

