



# CSS

Cascading Style Sheets

OpenSource UoM



# CSS...what?

- Η CSS (Cascading Style Sheets) μας επιτρέπει να δημιουργούμε όμορφες ιστοσελίδες
- Είναι μία γλώσσα μορφοποίησης και όχι μία γλώσσα προγραμματισμού
- Είναι μία γλώσσα για τον καθορισμό του τρόπου παρουσίασης των εγγράφων (.html) στους χρήστες – πώς είναι διαμορφωμένα, τοποθετημένα κλπ.
- Η CSS μπορεί να χρησιμοποιηθεί για πολύ βασική διαμόρφωση κειμένου εγγράφου (π.χ. να αλλάξει το χρώμα ή το μέγεθος της γραμματοσειράς), μπορεί να χρησιμοποιηθεί για τη δημιουργία μίας διάταξης-layout (π.χ. μετατροπή μίας μονής στήλης κειμένου σε κάποια διάταξη με κύριο περιεχόμενο και sidebar), όπως επίσης και για effect ή animations.

# Πώς είναι δομημένη μία σελίδα;

- Είναι σημαντικό να καταλάβουμε ότι μία σελίδα και τα περιεχόμενά της δεν είναι τίποτα παρά μόνο **κουτάκια** στον χώρο της οθόνης μας.

**Block elements** → Τα αντικείμενα της σελίδας HTML που καλύπτουν ολόκληρη την γραμμή

**Inline/Inline-Block elements** → Τα αντικείμενα της σελίδας HTML που πιάνουν τόσο χώρο όσο και το περιεχόμενό τους

## Browser defaults

The browser will style HTML documents using an internal stylesheet. This ensures that headings are larger than normal text, links are highlighted and structures such as lists and tables are understandable.

Paragraphs are spaced out. List items get a bullet or number, [Links are highlighted and underlined.](#)

- Item One
- Item Two

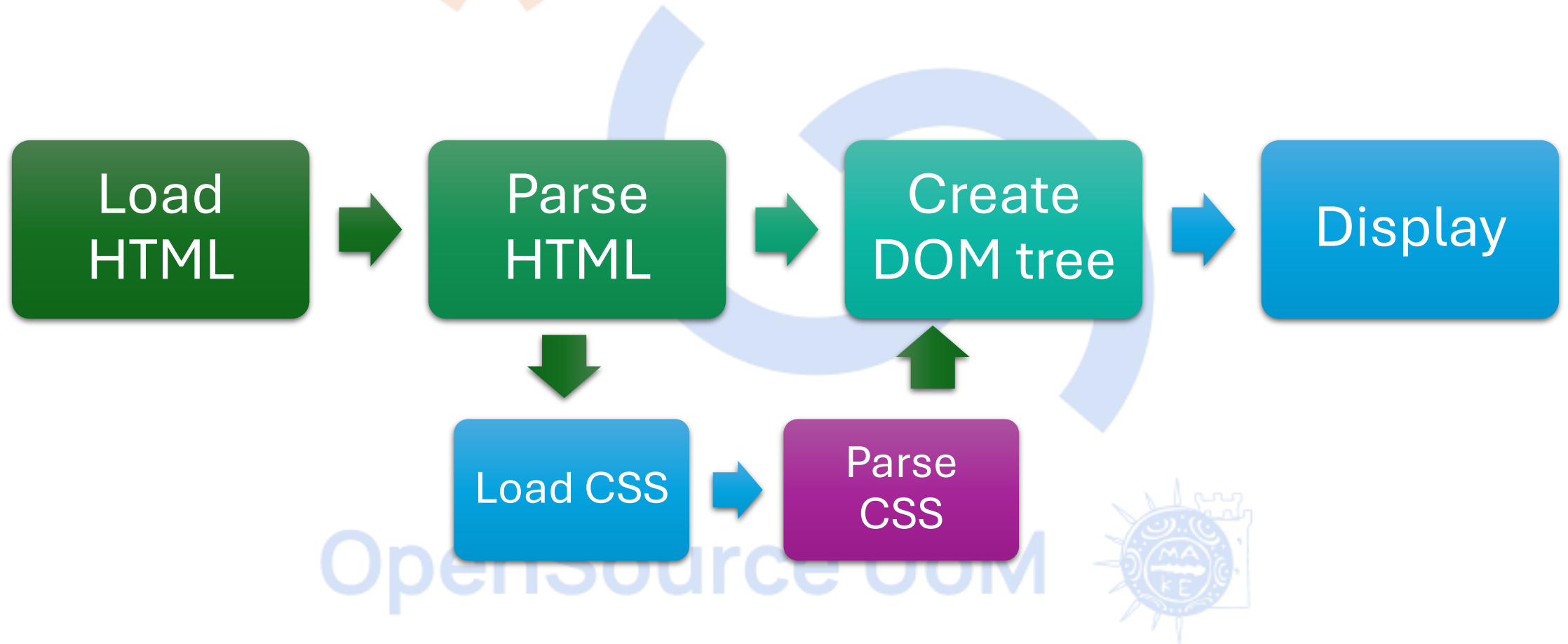
## A level 2 heading

You can change all of this with CSS.

# How is CSS structured?



# How CSS works?



# How CSS works? (explained)

1. Ο browser φορτώνει το αρχείο HTML
2. Ο κώδικας HTML μετατρέπεται στο DOM (Document Object Model)
3. Ο browser αντλεί τους περισσότερους από τους πόρους που συνδέονται με το έγγραφο HTML, όπως εικόνες και γενικότερα media καθώς και συνδεδεμένα αρχεία CSS.
4. Ο browser αναλύει το CSS που αντλείται και ταξινομεί τους διαφορετικούς κανόνες ανάλογα με τους selectors σε “buckets”, π.χ. Element, class, id, etc. Με βάση τους selectors υπολογίζει ποιοι κανόνες πρέπει να εφαρμοστούν σε ποιους κόμβους του DOM και τους αποδίδει στυλ όπως απαιτείται (δέντρο απόδοσης)
5. Το δέντρο απόδοσης είναι τοποθετημένο στη δομή που θα πρέπει να εμφανίζεται μετά την εφαρμογή των κανόνων σε αυτό
6. Η οπτική απεικόνιση της σελίδας εμφανίζεται στην οθόνη (painting)

# DOM?!?



- Το DOM (Document Object Model) είναι ένα API που αναπαριστά και αλληλεπιδράει με οποιοδήποτε έγγραφο HTML ή XML. Το DOM είναι ένα μοντέλο εγγράφου που φορτώνεται στον browser και αναπαριστά το έγγραφο ως δέντρο κόμβων ή δέντρο DOM, όπου κάθε κόμβος αναπαριστά μέρος του εγγράφου (π.χ. ένα στοιχείο)
- Επιτρέπει στον κώδικα που εκτελείται σε έναν browser να έχει πρόσβαση και να αλληλεπιδρά με κάθε κόμβο του εγγράφου.
- Οι κόμβοι μπορούν να μετακινηθούν και να αλλάξουν.
- Θα το δούμε καλύτερα με την χρήση της JavaScript 😊

# DOM representation

```
<p>  
  Let's use:  
  <span>Cascading</span>  
  <span>Style</span>  
  <span>Sheets</span>  
</p>
```

```
P  
├ "Let's use:"  
├ SPAN  
│ └ "Cascading"  
├ SPAN  
│ └ "Style"  
└ SPAN  
  └ "Sheets"
```

OpenSource UoM





# CSS Selectors

- Ορίζει τα μοτίβα για την επιλογή στοιχείων στα οποία στην συνέχεια εφαρμόζεται ένα σύνολο κανόνων CSS μαζί με την ιδιαιτερότητά τους.
  - **Global selector** → επιλέγει όλα τα αντικείμενα της HTML  
`* {}`
  - **Type selector** → επιλέγει τα αντικείμενα βάσει του HTML tag  
`p {}, h1 {}`
  - **Class selector** → επιλέγει τα αντικείμενα βάσει της τιμής που έχουν στο class attribute  
`.class-name {}`
  - **Id selector** → επιλέγει τα αντικείμενα βάσει της τιμής που έχουν στο id attribute  
`#id-name {}`

# Pseudo-elements

- Ένα **CSS Pseudo-element** είναι μία λέξη-κλειδί που προστίθεται σε έναν selector και μας επιτρέπει να διαμορφώσουμε ένα συγκεκριμένο μέρος του επιλεγμένου στοιχείου (ή των επιλεγμένων στοιχείων).
- Τα συναντάμε ΠΑΝΤΑ μετά από την χρήση των δύο (2) άνω-κάτω τελειών:  
**selector::pseudo-element**
- Για παράδειγμα, το pseudo-element **::first-line**

```
p::first-line {  
  color: blue;  
  text-transform: uppercase;  
}
```

LOREM IPSUM IS SIMPLY DUMMY TEXT OF THE PRINTING AND TYPESETTING INDUSTRY.

Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

# Pseudo-classes

- Μία CSS Pseudo-class είναι μία λέξη-κλειδί που προστίθεται σε έναν selector και καθορίζει μια ειδική κατάσταση του επιλεγμένου στοιχείου (ή των επιλεγμένων στοιχείων)

```
button:hover {  
  color: blue;  
}
```

Hover me! Single

Hover me! On hover

OpenSource UoM



# :hover



- Το pseudo-class **:hover** ταιριάζει όταν ο χρήστης αλληλεπιδρά με ένα στοιχείο υπόδειξης, αλλά δεν το ενεργοποιεί απαραίτητα. Γενικά, ενεργοποιείται όταν ο χρήστης περνάει πάνω από ένα στοιχείο με τον κέρσορα (δείκτη του ποντικιού)

HTML	CSS	OUTPUT
<pre>1 .joinBtn { 2   width: 10em; 3   height: 5ex; 4   background-color: gold; 5   border: 2px solid firebrick; 6   border-radius: 10px; 7   font-weight: bold; 8   color: black; 9   cursor: pointer; 10 } 11 12 .joinBtn:hover { 13   background-color: bisque; 14 } 15</pre>		<p>Would you like to join our quest?</p> <div><div>Confirm</div><div>Confirm</div> On hover</div>

# :active



- Το pseudo-class **:active** αναπαριστά ένα στοιχείο (όπως ένα κουμπί) που ενεργοποιείται από το χρήστη. Κατά τη χρήση ποντικιού, η «ενεργοποίηση» συνήθως ξεκινά όταν ο χρήστης πατήσει το κύριο κουμπί του ποντικιού.

HTML	CSS	OUTPUT
	<pre>1 .joinBtn { 2   width: 10em; 3   height: 5ex; 4   background-image: linear-gradient(135deg, #f34079 5   border: none; 6   border-radius: 5px; 7   font-weight: bold; 8   color: white; 9   cursor: pointer; 10 } 11 12 .joinBtn:active { 13   box-shadow: 2px 2px 5px #fc894d; 14 }</pre>	<div>Would you like to subscribe to our channel?</div> <div>Subscribe</div> <div>Subscribe On active</div>

# :focus



- Το pseudo-class :focus αντιπροσωπεύει ένα στοιχείο (όπως μια είσοδο φόρμας) που έχει λάβει εστίαση (focus). Γενικά, ενεργοποιείται όταν ο χρήστης κάνει κλικ ή πατάει πάνω σε ένα στοιχείο ή το επιλέγει με το πλήκτρο Tab από το πληκτρολόγιο.

HTML

CSS

OUTPUT

```
1 label {
2   display: block;
3   margin-top: 1em;
4 }
5
6 input:focus {
7   background-color: lightblue;
8 }
9
10 select:focus {
11   background-color: ivory;
12 }
13
```

Which flavor would you like to order?

Full Name:

Flavor:

Full Name:

On focus

# CSS combinators

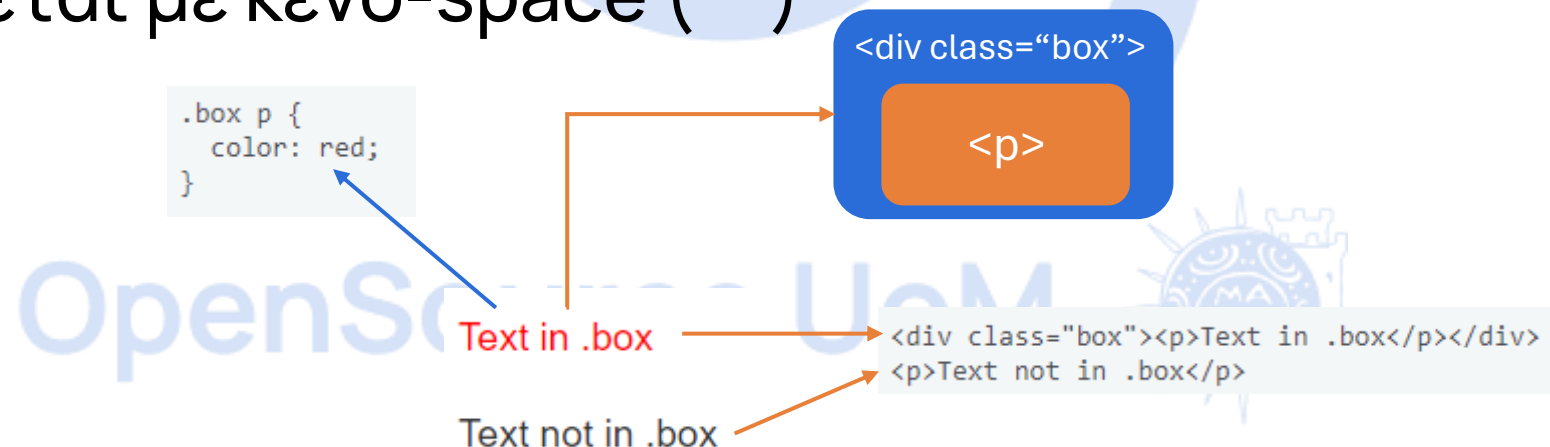
- Οι συγκεκριμένοι selectors ονομάζονται combinators, επειδή συνδυάζουν άλλους selectors με τρόπο που τους δίνει μια χρήσιμη σχέση μεταξύ τους και τη θέση του περιεχομένου στο έγγραφο.
- Οι combinators χωρίζονται στις εξής κατηγορίες:
  - Descendant combinator
  - Child combinator
  - Next-sibling combinator
  - Subsequent-sibling combinator

OpenSource UoM



# Descendant combinator

- Ο Descendant combinator **συνδυάζει δύο (ή περισσότερους) selectors που ταιριάζουν με έτσι ώστε τα στοιχεία που ταιριάζουν με τον δεύτερο selector να επιλέγονται εάν έχουν ένα προγονικό στοιχείο που ταιριάζει με τον πρώτο selector.**
- Συμβολίζεται με κενό-space (" ")





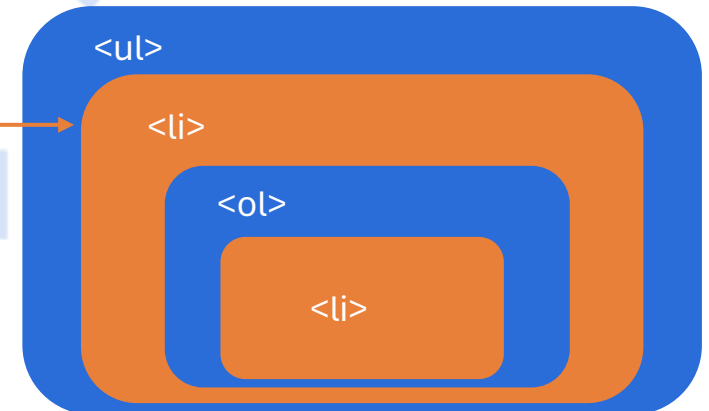
# Child combinator

- Ο Child combinator τοποθετείται μεταξύ δύο CSS selectors. Ταιριάζει μόνο με εκείνα τα στοιχεία που ταιριάζουν με τον δεύτερο selector και είναι τα άμεσα παιδιά των στοιχείων που ταιριάζουν με τον πρώτο. Τα στοιχεία που βρίσκονται πιο κάτω στην ιεραρχία δεν ταιριάζουν.
- Συμβολίζεται με “>”

```
ul > li {  
  border-top: 5px solid red;  
}
```

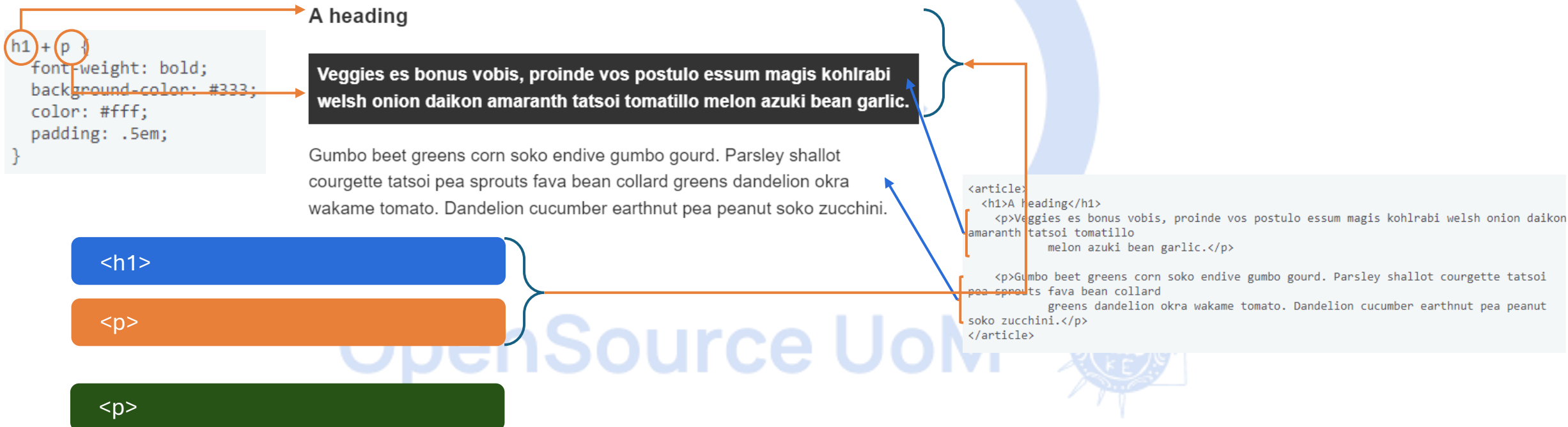
```
<ul>  
  <li>Unordered item</li>  
  <li>Unordered item  
    <ol>  
      <li>Item 1</li>  
      <li>Item 2</li>  
    </ol>  
  </li>  
</ul>
```

- Unordered item
- Unordered item
  - 1. Item 1
  - 2. Item 2



# Next-sibling combinator

- Ο Next-sibling combinator τοποθετείται μεταξύ δύο CSS selectors. Ταιριάζει μόνο σε εκείνα τα στοιχεία που ταιριάζουν με τον δεύτερο selector και είναι το αμέσως επόμενο αδελφό-στοιχείο του πρώτου selector
- Συμβολίζεται με “+”



# Subsequent-sibling combinator

Ο Subsequent-sibling combinator χρησιμοποιείται για να επιλέξουμε τα αδέλφια ενός στοιχείου ακόμα και αν δεν είναι άμεσα γειτονικά

Συμβολίζεται με “~”

