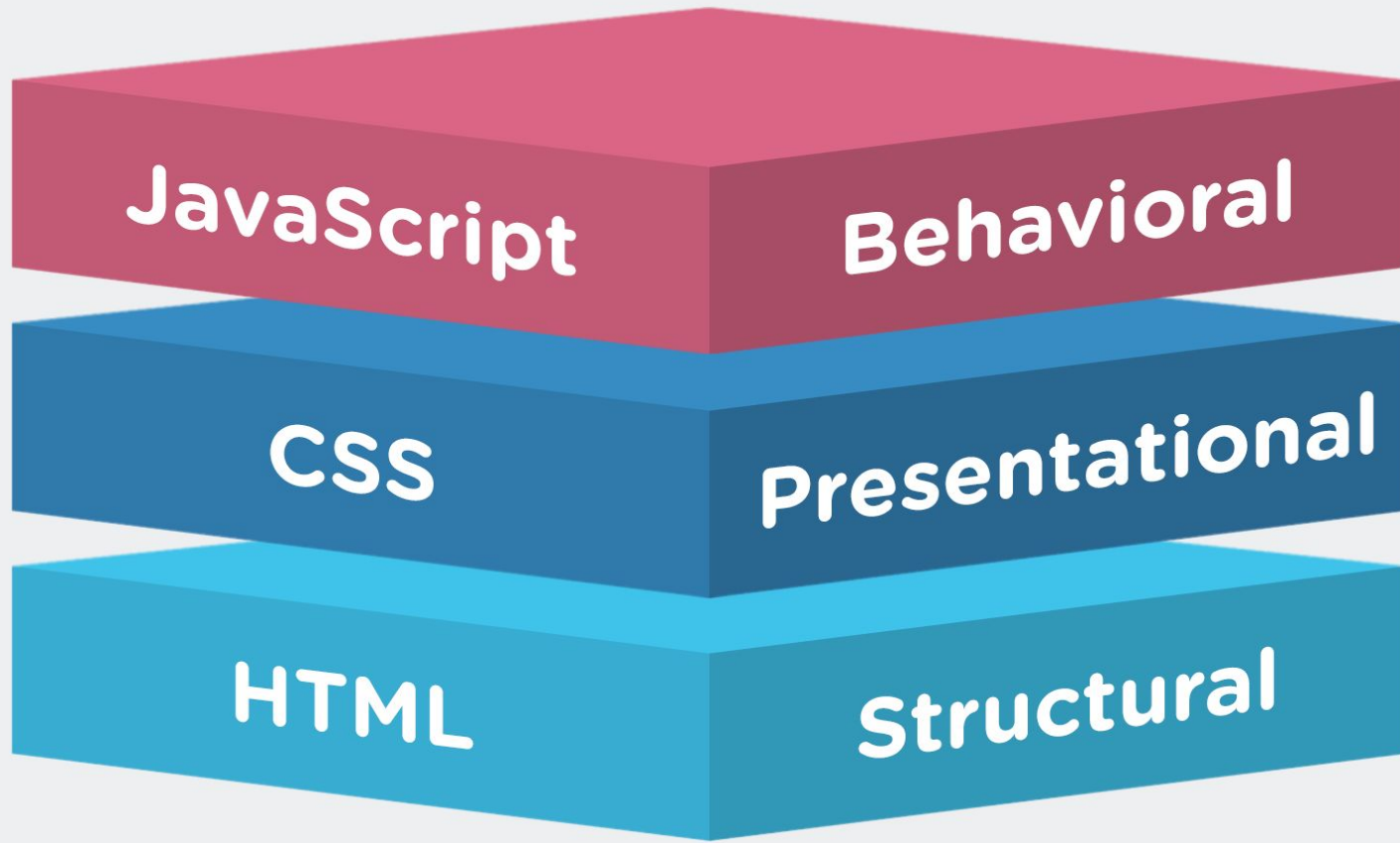




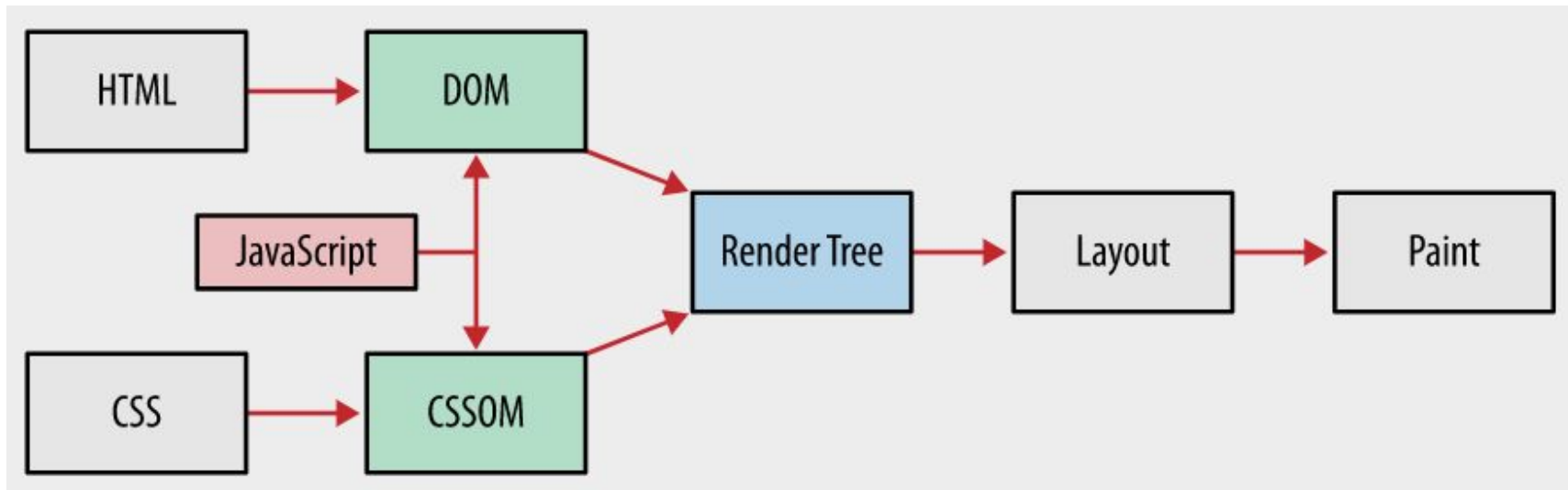
Saptamana 3

Partea 1

Programare Front-End



Ciclul de viata al unei pagini web (*lifecycle*)



CSS



1. **Cascading Style Sheets and CSS Object Model**

CSS

CSS sau **Cascading Style Sheets** este un limbaj folosit pentru a crea foi de stiluri - stylesheets care descriu felul in care se prezinta vizual un document scris in HTML (poate fi folosit si pentru alte tipuri de structuri) - de exemplu, pentru a schimba font-ul, culoarea sau marimea textului, pentru a imparti continutul in mai multe coloane sau pentru a adauga animatii ori alte particularitati ce tin de cromatica -. Versiunea actuala a acestui limbaj este **CSS3**.

Foile de stiluri contin mai multe elemente de urmatoarele tipuri:

- **selectori**: folositi pentru a selecta un anumit element dintr-un document HTML pe care vor fi aplicate anumite:
- **proprietati**: folosite pentru a defini modul in care elementul(ele) selectat(e) sunt afisate pe ecran

Exemplu: Vreau ca pe **toate paragrafele din documentul HTML** pe care l-am creat sa aiba **marimea font-ului 16px** si **culoarea textului rosu**.

CSS

```
selector1 {  
    proprietate-a: valoare-a;  
    proprietate-b: valoare-b;  
    ...  
    proprietate-p: valoare-p;  
}  
  
selector2 {  
    proprietate-l: valoare-l;  
    proprietate-o: valoare-o;  
}  
  
...  
  
selectorN {  
    proprietate-k: valoare-k;  
}
```

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>WANTSOME</title>
5     <meta charset="UTF-8" />
6   </head>
7
8   <body>
9     <h1 style="color: blue">We love Wantsome</h1>
10    <p style="font-size: 25px">Internal CSS</p>
11    <ol>
12      <li style="font-weight: bold">HTML</li>
13      <li>CSS</li>
14      <li>JavaScript</li>
15    </ol>
16  </body>
17 </html>
18
```

<https://q7r35mor29.cod>

We love Wantsome

Internal CSS

1. HTML
2. CSS
3. JavaScript


```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <title>Wantsome</title>
6     <style>
7       h1 {
8         color: red;
9       }
10
11       p {
12         font-size: 20px;
13       }
14
15       li {
16         font-weight: bold;
17       }
18     </style>
19   </head>
20   <body>
21     <h1>We love wantsome</h1>
22     <p>Stim deja putin front-end ! Pana acum am invatat notiuni pre
23   <ol>
24     <li>Web Apps Architecture</li>
25     <li>UI/UX</li>
26     <li>Waterfall, Agile, SCRUM, Kanban</li>
27     <li>HTML</li>
28   </ol>
29 </body>
30 </html>
```

< > ↺ <https://ly979rlqll.codesandbox.io/> 🖨️ 📱 🌙 🌞

We love wantsome

Stim deja putin front-end ! Pana acum am invatat notiuni precum:

1. Web Apps Architecture
2. UI/UX
3. Waterfall, Agile, SCRUM, Kanban
4. HTML

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>WANTSOME</title>
5     <meta charset="UTF-8" />
6     <link rel="stylesheet" href="src/styles.css" />
7   </head>
8
9   <body>
10    <h1>We love Wantsome</h1>
11    <p>External CSS</p>
12    <ol>
13      <li>HTML</li>
14      <li>CSS</li>
15      <li>JavaScript</li>
16    </ol>
17  </body>
18 </html>
19
```

<https://q7r35mor29.codes>

We love Wantsome

External CSS

1. HTML
2. CSS
3. JavaScript

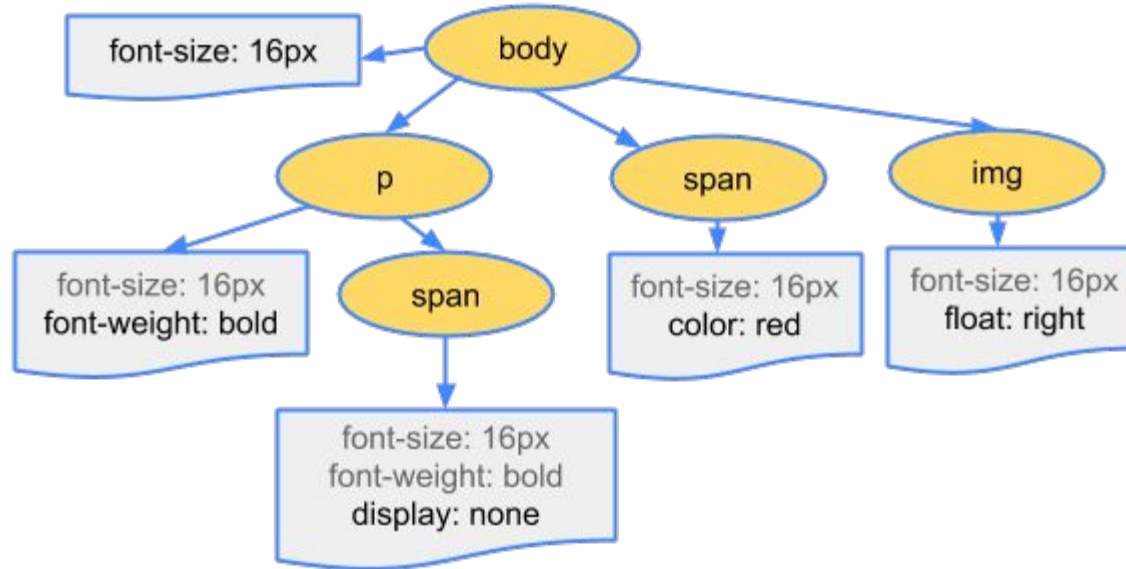
Exemplu foaie de stiluri: external, path

CSSOM

CSSOM sau **Cascading Style Sheets Object Model** reprezinta o structura arborescenta alcatuita din obiecte care definesc obiectele elementele ce structureaza un anumit document HTML, alaturi de proprietatile CSS corespunzatoare definite in cadrul uneia sau a mai multe foi de stiluri.

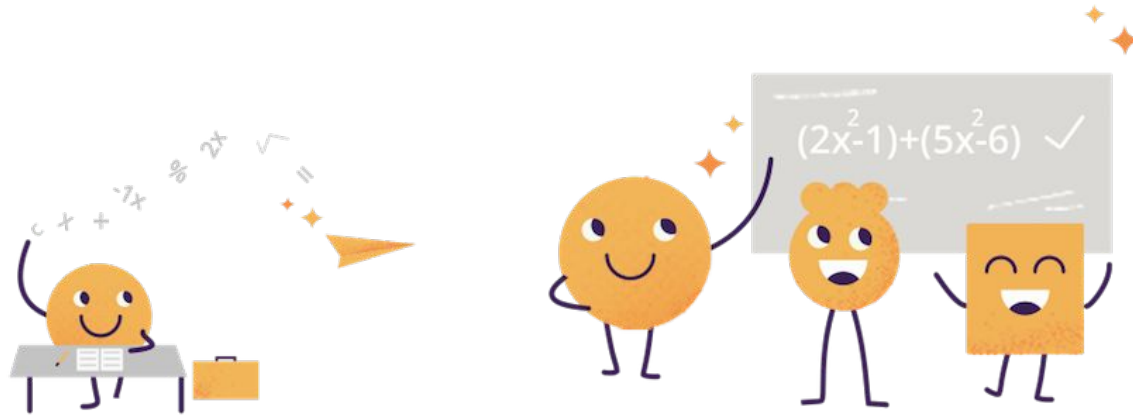
Aceasta structura este creata de catre browser in momentul in care o pagina este incarcata, asociind fiecarui element HTML specificat cate un obiect si serveste ca o **interfata** pentru accesarea, modificarea si stergerea proprietatilor CSS asociate unui element. (vom vedea mai tarziu cine foloseste aceasta interfata)

Exemplu de structura CSSOM



PRACTICE: CSS

<http://bit.do/CSSEX>



2. CSS Selectors and Pseudo-selectors


CSS Selectors – Selectorii de baza

- **element** - selecteaza toate elemente de tipul ... - **p { color: blue }**


- **.class** - toate elementele care au o anumita clasa

- **#id** - selecteaza elementul care are id-ul specificat


- ***** - selecteaza toate elementele din DOM



```
1 .title-color {  
2   color: blue  
3 }  
4 /* <div class="title-color"> </div> */
```



```
1 #title-color {  
2   color: red  
3 }  
4 /* <div id="title-color"> </div> */
```



```
1 * {  
2   color: green;  
3 }  
4 /* tot textul de pe pagina va avea culoarea verde */  
5
```

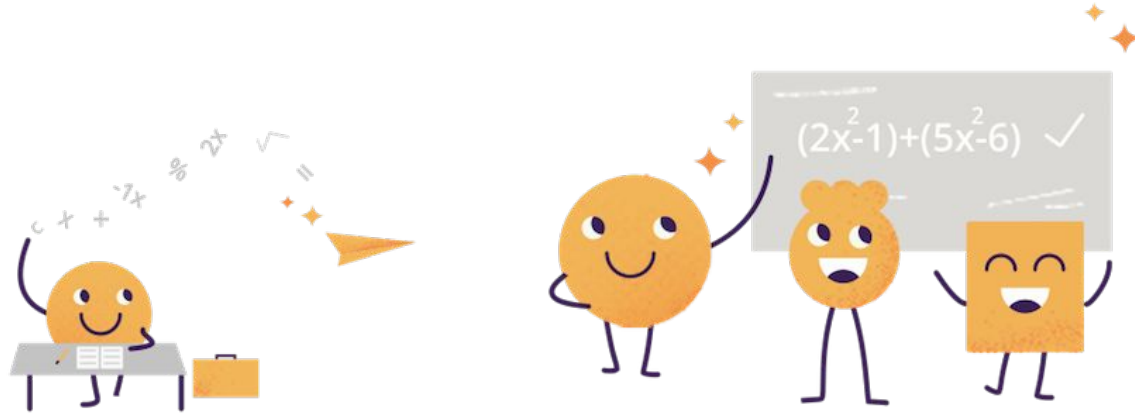
CSS Selectors – Selectie bazata pe atribute

- **[atribut]** - elementele ce contin atributul specificat - **a[href] { color: blue }**
- **[atribut="val"]** - elementele ce contin atributul specificat cu o valoare egala cu "val"
 - **a[href="/wantsome/"] { color: red }**
- **[atribut~="val"]** - elementele ce contin atributul specificat, avand o valoare care include "val" separat de spatiu
 - **abbr[title~="Sunt Frontender"] { color: red }**
 - *se ia in considerare spatiul*
 - (ex: va selecta `<p title="Sunt Frontender la Levi"> </p>` ; nu va selecta `<p title="Sunt FrontenderLaLevi"> </p>`)
- **[atribut|="val"]** - elementele ce contin atributul specificat, avand o valoare care include "val" separat de "-"
 - **html[lang|"en"] { color: red }**

CSS Selectors – Selectie bazata pe attribute

- **[atribut^="val"]** - elementele ce contin atributul specificat, avand o valoare care incepe cu "val"
 - **a[href^="http://"] { color: red }**
- **[atribut\$="val"]** - elementele ce contin atributul specificat, avand o valoare care se termina cu "val"
 - **a[href\$=".com"] { color: red }**
- **[atribut*="val"]** - elementele ce contin atributul specificat, avand o valoare care contine "val"
 - **abbr[title*="Sunt Frontender"] { color: red }**
 - nu se ia in considerare spatiul
 - (ex: va selecta `<p title="Sunt Frontender la Levi"> </p>` si `<p title="Sunt FrontenderLaLevi"> </p>`)

PRACTICE: CSS Basic and Attribute Selectors



CSS Selectors: Pseudo-clase

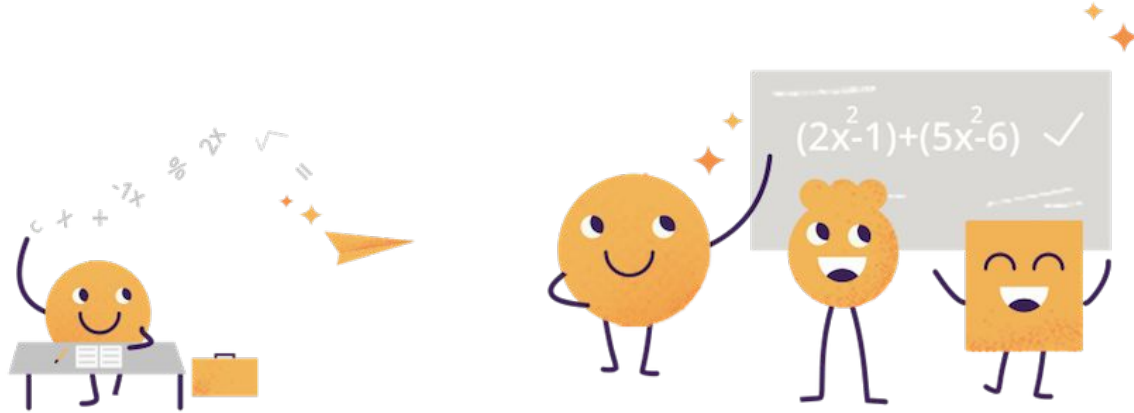
- **:link** - se aplica linkului care n-a fost vizitat - `a:link { color: blue }`
- **:visited** - se aplica linkului care a fost vizitat `a:visited { color: red }`
- **:active** - se aplica linkului care are ca "href" locatia curenta `a:active { color: green }`
- **:hover** - se aplica atunci cand are loc evenimentul **hover** peste un link `a:hover { color: red }`
- **:focus** - se aplica atunci cand focusam un link - `a:focus { color: red }`
- **:target** - se aplica unui element care reprezinta o "legatura" (<h3>bla</h3>) - `h3:target { color: red }`

CSS Selectors: Pseudo-clase

- **:checked** - elemente de tip *radio*, *checkbox* sau *option* care sunt bifate sau selectate
 - **.checkbox:checked { background-color: green; }**
- **:first-child** - primul element dintr-un set de elemente care sunt rude (pe acelasi nivel) - **main:first-child { color: red; }**
- **:last-child** - ultimul element dintr-un set de elemente care sunt rude (pe acelasi nivel) - **main:last-child { color: blue; }**
- **:nth-child** - elementele de pe pozitia specificata dintr-un set de elemente care sunt rude (pe acelasi nivel)
 - **main:nth-child(4n)** - fiecare al 4-lea element (din 4 in 4) din sectiunea *main*
- **:not** - elementele pentru care nu se aplica selectorul specificat intre paranteze - **p:not([class='article-p']) { color: black; }**

! Atentie: acestea nu sunt toate pseudo-clasele existente, ci cele mai des utilizate; Verificati referintele !

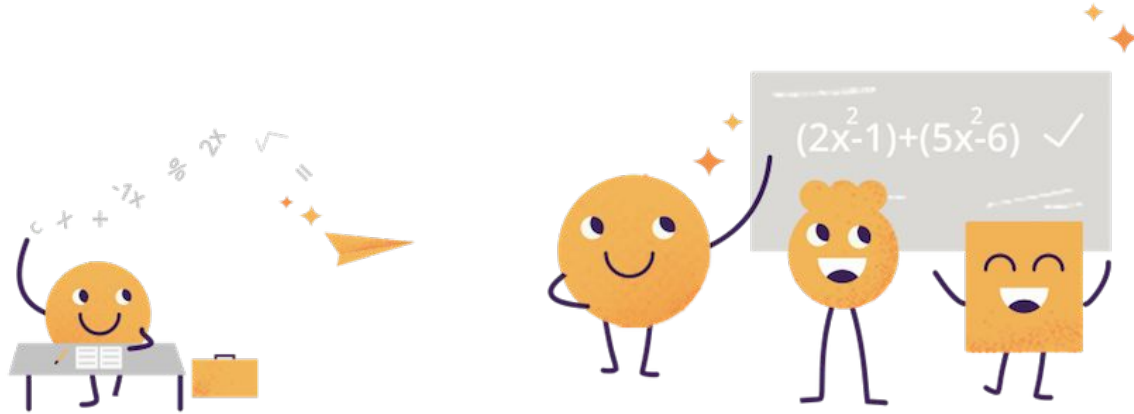
PRACTICE: CSS Pseudo-Classes



CSS Selectors – Pseudo elemente

- **::first-line** , **::first-letter** - prima linie de continut, prima litera a unui element - `p::first-line { font-family: Arial }`
- **::before** , **::after** - inainte si dupa un element - `h1::before {content: "-"}`
- **::selection** - portiunea de continut selectata de utilizator - `h1::before {content: "-"}`

PRACTICE: CSS Pseudo-Elements



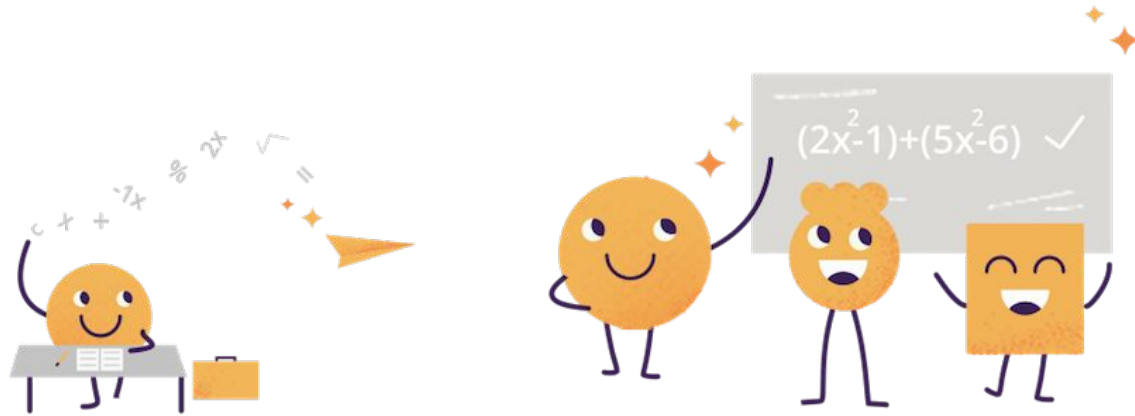
CSS Selectors – Combinators

- **selector1 selector2** - selecteaza elementele de tip **selector2** descendente ale **selector1** - **article p { color: blue }**
- **selector1 > selector2** - elementele de tip **selector2**, copii directi ai **selector1** - **#warning > p { color: red }**
- **selector1 + selector2** - elementul de tip **selector2**, ruda cu **selector1** (au acelasi parinte) - **h1 + p { color: green }**
- **selector ~ selector** - toate elementele de un anumit tip, de dupa cel precizat - **h2 + p {color: blue }**

CSS Selectors – Grupare de selectori

- **selector1, selector2** - selecteaza toate elementele care corespund **selector1** sau **selector2**
 - **header a:hover, .sidebar a:hover, footer a:hover { text-decoration: underline; }**

PRACTICE: CSS Combinators





HTML

HTML + CSS

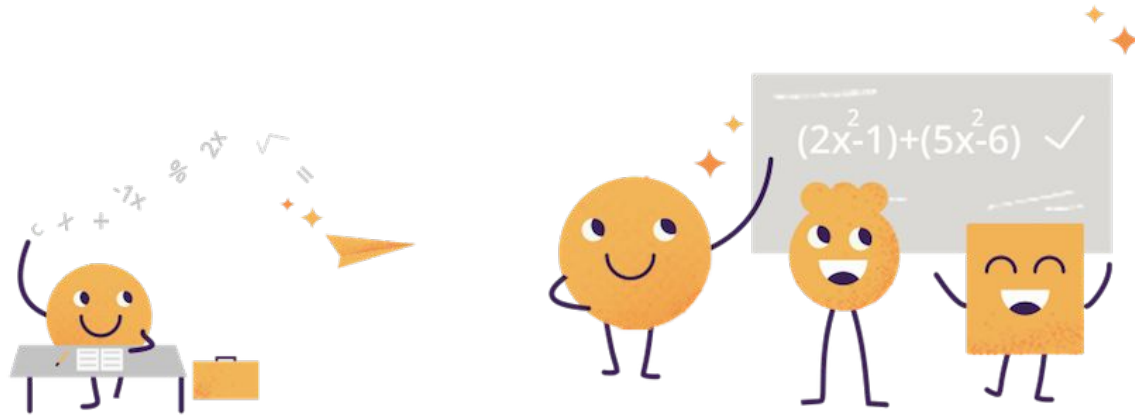
HTML + CSS

4. CSS Reference – Styling Properties

CSS Reference – Proprietati

<https://developer.mozilla.org/en-US/docs/Web/CSS/Reference>

PEER PRACTICE: CSS Properties



5. CSS Specificity

CSS – Specificitate

- Impune un anumit set de reguli ce sunt aplicate de catre browser
- Este de obicei motivul pentru care uneori regulile de CSS nu sunt aplicate, desi v-ati astepta la acest lucru; Insa, un alt selector le suprascrie pentru ca este mai specific, mai prioritar; Concluzie: *Este foarte important sa intelegem cum functioneaza !*
- Fiecare selector are o anumita specificitate
- Regulile care combina mai multi selectori au specificitate mai mare
- Ultima regula scrisa suprascrie prima regula scrisa (daca are aceeasi specificitate) - **CASCADING !**

CSS – Specificitate – Prioritati

1. **Inline styles** `<h1 style="color: red;">`
2. **Ids** `#warning-message`
3. **Classes and attributes** `.class` and `[attribute]`
4. **Elements and pseudo-elements** `:before` and `:after`

PEER PRACTICE: CSS Specificity

