

## Tema: Pseudo-clase – stilizarea referințelor. Pseudo-elemente. Elementele DIV și SPAN

După această lecție:



- vei afla ce sunt pseudo-clasele și pseudo-elementele și cum pot fi utilizate acestea în selectorii CSS;
- vei cunoaște elementul HTML, DIV, utilizat pentru gruparea altor elemente HTML și vei învăța cum poate fi el stilizat;
- vei afla cum pot fi stilizate conținuturi *inline* ale elementelor HTML, cu SPAN.

La lecțiile trecute am reușit să cunoaștem mai multe proprietăți CSS, utilizate pentru stilizarea diferitor elemente HTML. La această lecție vom continua să cunoaștem noi posibilități ale limbajului CSS.

### Pseudo-clase în CSS

O **pseudo-clasă** este utilizată pentru a defini o stare specială a unui anumit element HTML. De obicei, pseudo-clasele sunt utilizate pentru: definirea stilurilor unui element atunci când utilizatorul plasează mouse-ul deasupra lui sau definirea diferită a stilurilor pentru link-urile vizitate sau nevizitate etc.

Sintaxa generală de utilizare a unei pseudo-clase este:

***selector:pseudo-class***

```
{  
  proprietate:valoare;...  
}
```

Pseudoclaselor *:link*, *:visited*, *:hover* și *:active* sunt utilizate pentru stilizarea referințelor dintr-un document web.

Exemplu de utilizare a pseudo-claselor pentru stilizarea referințelor:



```
/* link nevizitat */  
a:link { color: #00ff00; }  
/* link vizitat */  
a:visited { color: #00aa00; }  
/* mouse-ul deasupra link-ului */  
a:hover { color: #aa00ff; }  
/* link activ, selectat */  
a:active { color: #0000ff; }
```

**Observație:** Mai sus am utilizat sintaxa `/* link vizitat */`, astfel am introdus un comentariu într-un fișier `.css`. Reține aceasta!

Atunci când se definesc toate evenimentele, utilizând pseudo-clasele, pentru referințe, uneori, contează ordinea definirii lor: 1) `a:link { ... }`, 2) `a:visited { ... }`, 3) `a:hover { ... }`, 4) `a:active { ... }`. În unele cazuri, la schimbarea ordinii definirii stilurilor, unele proprietăți CSS pot să nu se aplice corect.

### Unele exemple de stilizare a referințelor

Referințele pot fi stilizate în diferite moduri, utilizând pseudo-clasele - `a:link`, `a:visited`, `a:hover`, `a:active` – și utilizând orice proprietăți CSS, precum: `color`, `background`, `font-family` etc., decorând, într-un fel sau altul, referințele.

Eu voi realiza un exemplu, dar trebuie să știi că poți stiliza și altfel referințele.

Adaug în documentul web câteva referințe:



```
<a href="#unu">Despre noi</a>
<a href="#doi">Produse</a>
<a href="#trei">Contacte</a>
```

Iar în fișierul de stiluri – stilizez referințele astfel:



```
/* link nevizitat */
a:link {
    color: rgb(199, 240, 247);
    text-decoration: none;
    padding: 10px;
    margin: 5px;
    font-size: 1.5em;
}
/* link vizitat */
a:visited { color: rgb(241, 151, 213);}
/* mouse-ul de-asupra link-ului */
a:hover {
    color: whitesmoke;
    text-decoration: underline;
    padding: 10px;
    margin: 5px;
}
/* link activ, selectat */
a:active { color: #cacadf;}
```

Rezultatul este prezentat în următoarea imagine:



Observă aplicarea stilurilor: pe prima referință am plasat cursorul de-asupra ei – s-a schimbat culoarea textului și a apărut linia de subliniere, cea de-a două încă nu a fost vizitată și a rămas culoarea setată în **:link**, iar cea de-a treia referință a fost deja accesată, și textul a devenit roz. Imaginile-referințe tot pot fi stilizate frumos prin adăugare de borduri sau umbre, schimbare filtre etc.

Atunci când vom învăța tranzițiile – vei vedea cum putem realiza schimbarea lentă a stilurilor. Acum trecerea de la un stil la altul se face rapid.

### Pseudo-clasa „:first-child” și „:last-child”

Pseudo-clasa **:first-child** permite selectarea primului element-urmas al unui element-părinte. Iar pseudo-clasa **:last-child** — a ultimului element-urmas.

De exemplu: **li:last-child { ... }** – un astfel de selector va alege ultimului element dintr-o listă.

### Pseudo-clasa „:nth-child”

Pseudo-clasele **:first-child** și **:last-child** fac parte din familia pseudo-claselor, ce permit selectarea elementelor în funcție de poziția acestora. Pseudo-clasa **:nth-child** permite selectarea elementelor-urmasi, conform numărului lor de ordine.

Sintaxa: **selector:nth-child(expresie)**, unde *expresie* poate fi un număr sau chiar o formulă. De exemplu: **li:nth-child(2) { ... }**, **li:nth-child(4) { ... }**, **li:nth-child(2n) { ... }** etc.

Primul selector exemplificat, va selecta cel de-al doilea element din listă, cel de-al doilea selector – pe al patrulea element din listă, iar ultimul exemplu va selecta toate elementele ce stau în pozițiile pare. Voi realiza un exemplu pentru claritate.

Fie următorul tabel HTML:



```
<table>
  <caption>Preturile produselor noastre</caption>
  <tr>
```

```

        <th>Denumire produs</th>
        <th>Pret (lei)</th>
    </tr>
    <tr>
        <td>Cappuccino</td>
        <td>30</td>
    </tr>
    <tr>
        <td>Espresso</td>
        <td>15</td>
    </tr>
    <tr>
        <td>Latte</td>
        <td>25</td>
    </tr>
</table>

```

Pentru el și elementele lui voi defini următoarele stiluri:



```

table {
    border: 2px solid rgb(214, 209, 214);
    border-collapse: collapse;
    width: 30%;
    margin: 20px auto;
}
th, td {
    border: 2px solid rgb(214, 209, 214);
    border-collapse: collapse;
    height: 25px;
    padding: 5px;
}
tr:nth-child(2n-1) {
    background-color: #002633;
    color: #e6f9ff;
}
tr:nth-child(2n) {
    background-color: #e6f9ff;
    color: #002633;
}

```

Rezultatul prezentării va fi mai deosebit decât până acum:

Preturile produselor noastre	
Denumire produs	Pret (lei)
Cappuccino	30
Espesso	15
Latte	25

Observă cum au fost aplicate stilurile pentru liniile tabelului – culorile alternează de la linie la linie, iar acest fapt se datorează pseudo-clasei **:nth-child**.

### Pseudo-clasa „:focus”

Pseudo-clasa **:focus** permite selectarea și definirea stilurilor pentru acel element, care la moment este în focus. Elemente „evidențiate” sau focusate, pot fi nu doar câmpurile din formular.

Exemplu:



```
...
<input type="text" name="nume" placeholder="Introdu aici numele tau..."
autocomplete="off" />
...
```

Cu stilul: `input:focus {border: 3px solid #e73c08e7;}`

Rezultatul:

Poți accesa acest link [https://www.w3schools.com/css/css\\_pseudo\\_classes.asp](https://www.w3schools.com/css/css_pseudo_classes.asp), pentru a vedea și aplica și alte pseudo-clase.

### Pseudo-elemente CSS

Un pseudo-element CSS este folosit pentru definirea stilului pentru o anumită parte a unui element. Pseudo-elementul poate fi utilizat pentru:

- definirea stilului primei litere sau primei linii ale unui element;
- Inserarea conținutului înainte sau după, conținutul unui element etc.

Forma generală la utilizarea unui pseudo-element:

**selector::pseudo-element**

```
{
  proprietate:valoare; ...
}
```

**Pseudo-elementul „::first-line”**

Se utilizează pentru a defini stilul primei linii dintr-un text. Poate fi utilizat și pentru un bloc de elemente. Proprietățile CSS care pot fi aplicate pseudo-elementului **::first-line** sunt: *font, color, background, word-spacing, letter-spacing, text-decoration, vertical-align, text-transform, line-height* etc.

**Pseudo-elementul „::first-letter”**

Se utilizează pentru a defini stilul primului caracter dintr-un text. Poate fi utilizat și pentru un bloc de elemente. Proprietățile CSS care pot fi aplicate pseudo-elementului **::first-letter** sunt: *font, color, background, margin, padding, border, text-decoration, text-transform, line-height* etc.

Voi scrie un exemplu, în care voi defini stilul primei litere, dar și a primului caracter din toate paragrafele de text, ale documentului web:



```
p::first-letter{
  font-size: 1.7em;
}
p::first-line{
  color: rgb(199, 240, 200);
}
```

Rezultatul va fi:

Deoarece cafeaua are un conținut ridicat de antioxidanți - fenol, substanțe volatile - dacă este consumată în cantități moderate, poate avea efecte benefice asupra sănătății. Un studiu făcut în 2013 arată că numai 9% din orășeni nu consumă cafea - sub o anumită formă cunoscută: cafea solubilă, espresso, cappuccino, latte etc.

Și dacă micșorez lățimea ferestrei browserului – observă cum este stilizată diferit prima literă și prima linie de text:

Deoarece cafeaua are un conținut ridicat de antioxidanți - fenol, substanțe volatile - dacă este consumată în cantități moderate, poate avea efecte benefice asupra sănătății. Un studiu făcut în 2013 arată că numai 9% din orașeni nu consumă cafea - sub o anumită formă cunoscută: cafea solubilă, espresso, cappuccino, late etc.

## Pseudo-elementele „::before” și „::after”

Pseudo-elementul **::before** este folosit pentru a insera un anumit conținut înaintea unui element, iar **::after** este utilizat pentru a insera un anumit conținut după un element HTML.

Înainte de a exemplifica utilizarea acestor pseudo-elemente vreau să descriu și proprietatea CSS, **content**, care este utilizată, de obicei, împreună cu pseudo-elementele **::before** și **::after**, pentru a insera un content. Proprietatea este suportată de toate browserele web. Unele valori ale proprietății **content**: *none*, *counter*, *string*, *open-quote*, *close-quote*, *no-open-quote*, *no-close-quote*, *url*.

Să vedem și un exemplu de utilizare a proprietății **content**, împreună cu pseudo-elementul **::before**. Pentru aceasta voi crea o listă:



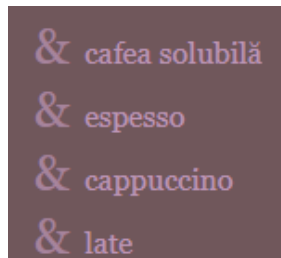
```
<ul>
    <li>cafea solubilă</li>
    <li>espresso</li>
    <li>cappuccino</li>
    <li>late</li>
</ul>
```

Și voi defini stilurile, specificând ca lista să nu posede marcate standard, dar înaintea fiecărui element al listei să se insereze simbolul “&”:



```
ul {
    list-style: none;
    padding: 0;
    margin: 0;}
li {
    padding-left: 16px;
    color: rgba(221, 177, 221, 0.7);}
li::before {
    content: "&";
    font-size: 1.7em;
    padding-right: 8px;
    color: rgba(221, 177, 221, 0.5); }
```

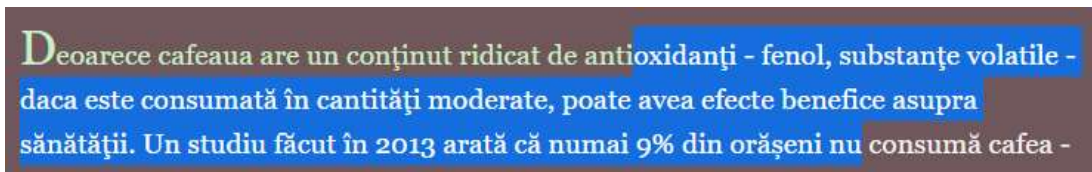
Rezultatul interpretării va fi:



### Pseudo-elementul "::selection"

Pseudo-elementul **::selection** evidențiază o porțiune a elementului HTML, care este marcată de utilizator. Pseudo-elementul **::selection** poate fi aplicat următoarelor proprietăți CSS: *color*, *background*, *cursor*, și *outline*.

Valorile implicite, în cazul în care utilizatorul marchează o porțiune de text dintr-o pagină web, sunt: text alb pe fundal albastru – sper că ai observat.

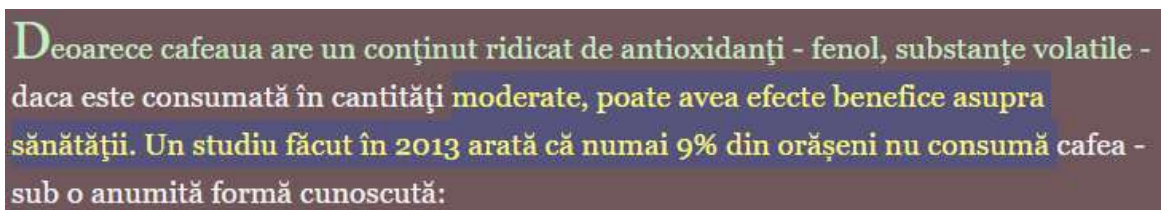


Dar, aceste valori implicite pot fi schimbate... De exemplu:



```
::selection{  
  color:rgb(253, 251, 121);  
  background-color:#525283da;  
}
```

Rezultatul aplicării acestui stil la marcare, va fi:



Alte pseudo-elemente poți găsi la acest link:

[http://www.w3schools.com/css/css\\_pseudo\\_elements.asp](http://www.w3schools.com/css/css_pseudo_elements.asp)



## Elementele HTML **DIV** și **SPAN**

Elementele **DIV** și **SPAN**, de obicei, nu sunt folosite singure, având efecte deosebite dacă se utilizează în combinație cu alte elemente HTML și împreună cu stiluri CSS.

Elementul **DIV** creează secțiuni de blocuri în pagină, forma și grafica cărora pot fi controlate separat, prin stiluri. Elementul **DIV** are atributul HTML **align**, utilizat pentru aliniere pe orizontală, care nu mai este suportat de HTML5. Elementul **SPAN** creează posibilitatea modificării separate a unei porțiuni dintr-un context, putând fi folosit împreună cu o "clasă" și stilizat cu CSS.

Ambele elemente pot avea atributul **style** ori atributul **id** sau **class** pentru stabilirea selectorului necesar și stilizarea lor.

Elementul **DIV** este unul dintre cele mai utilizate în HTML, în ultimul timp. Este suportat de majoritatea browsere-lor web. Sintaxa elementului **DIV** este: `<div>conținut</div>`

În interiorul elementului pot fi incluse orice alte elemente HTML, precum: tabele, formulare, antete, paragrafe de text, imagini sau alte DIV-uri. Cadrul în care sunt adăugate DIV-urile poate avea propriul fundal (background), lungime, înălțime și borduri.

Elementul **SPAN** poate adăuga stiluri grafice unei anumite porțiuni dintr-un context, *in-line*. Cuvintele care se doresc a fi evidențiate trebuie încadrate între tag-urile `<span>` și `</span>`, iar în fișierul cu stiluri se definesc formatările dorite.

Să realizăm un exemplu simplu, în care vom folosi aceste două elemente HTML. Codul HTML este:



```
<body>
  <div class="menu">
    <span>Coffee time</span>
    <a href="#unu">Despre noi</a>
    <a href="#doi">Produse</a>
    <a href="#trei">Contacte</a>
  </div>
  <div class="content">
    <div class="text">
      <p>Deoarece cafeaua are un conținut ridicat de
      antioxidanți - <span>fenol, substanțe volatile</span> - dacă este
      consumată în cantități moderate, poate avea efecte benefice asupra
      sănătății. Un studiu făcut în 2013 arată că numai 9% din orășeni nu
      consumă cafea - sub o anumită formă cunoscută: <span>cafea solubilă,
      espresso, cappuccino, late</span> etc.</p>
    </div>
    
    
  </div>
</div>
```

```

        
        
        
        
    </div>
    <div class="footer"></div>
</body>

```

Stilurile:



```

body{
    font: 16px/30px Georgia, serif;
    color: white;
    background-color: rgba(77, 46, 51, 0.8);
    margin:0;
    padding: 0;
}
*{
    box-sizing: border-box;
}
a:link {
    color: rgb(199, 240, 247);
    text-decoration: none;
    padding:10px;
    margin:5px;
    font-size: 1.3em;
}
a:visited { color: rgb(253, 228, 245);}
a:hover {
    color: whitesmoke;
    text-decoration: underline;
    padding: 10px;
    margin:5px;
}
a:active { color: #cacadf;}
.menu, .content, .footer {
    margin:0 auto;
    padding: 10px 0;
    margin: 0 auto;
    text-align: center;
    width: 100%;
}
.menu, .footer {
    background-color: rgb(77, 46, 70);
    height: 10%;
}
.menu span{
    text-transform: uppercase;
    color: rgba(214, 248, 21, 0.8);
}

```

```
    font-size: 1.5em;
    margin-right: 20px;
}
.content img {
    max-width: 25%;
    height: auto;
    border: 3px dotted rgb(247, 238, 239);
    padding: 3px;
    margin: 5px;
}
.text span {
    color: rgb(243, 250, 140);
}
.footer{
    background-image: url("../images/logo.png");
    background-repeat: repeat-x;
    background-size: 5%;
}
p{
    padding: 10px;
    font-size: 1.1em;
    color: whitesmoke;
}
```

Rezultatul:



Observă cum am utilizat elementele DIV și SPAN. Încearcă și tu să construiești blocuri cu elemente HTML, grupându-le cu DIV, iar apoi – stilizează-le. Succese!

Să repetăm cele învățate în cadrul acestei lecții:



- pseudoclaselor sunt utilizate pentru a stiliza diferite stări ale elementelor HTML. Cel mai frecvent sunt utilizate pentru stilizarea referințelor, atunci când se plasează cursorul deasupra lor sau în momentul când utilizatorul face click pe referință etc.;
- pseudoelementele sunt utilizate pentru a stiliza diferit unele părți ale elementelor HTML;
- elementul DIV este utilizat pentru a defini porțiuni-bloc, formate din grupuri de elemente HTML și pentru aceste blocuri pot fi definite stiluri de prezentare cât mai comodă a acestora;
- elementul SPAN este utilizat pentru a adăuga stiluri *inline*. Adică, la necesitate, pot fi modificate stilurile unor porțiuni de conținuturi web, încadrându-le între elementul SPAN.

Cu alte proprietăți, dar și posibilități ale limbajului CSS – vom continua la lecția următoare.