

Utilizarea stilurilor CSS

(Cascade Style Sheets)

- **CSS** - etichete folosite pentru formatarea paginilor web (formatare text, background, aranjare in pagină, etc.).

Beneficiile sintaxei CSS sunt:

- formatarea este introdusă într-un singur loc pentru tot documentul
- editarea rapidă a etichetelor
- datorită introducerii într-un singur loc a etichetelor se obține o micșorare a codului paginii, implicit încărcarea mai rapidă a acesteia

Specificarea stilurilor

Sintaxa CSS este structurată pe trei nivele:

1. *nivelul 1* – proprietățile etichetelor din documentul HTML, tip **inline**
2. *nivelul 2* – informația introdusă în blocul `<head>`, tip **embedded (intern)**
3. *nivelul 3* - comenzile aflate în pagini separate, tip **extern**

- Comenzile CSS de nivel 1 sau **inline** suprascriu orice alte comenzi CSS. Sunt amplasate în interiorul etichetelor HTML aflate în zona BODY și au sintaxa:

```
<eticheta style="codul CSS dorit">  
...textul sau obiectul asupra căruia este aplicat  
codul CSS...  
</eticheta>
```

- Comenzile de nivel 2 sau **embedded** sunt cele amplasate oriunde între perechea de etichete `<head>` și `</head>` conform sintaxei:

```
<style type="text/css">  
... comenzi CSS ...  
</style>
```

style -specifică unde începe și unde se termină blocul CSS, iar *type* este folosit pentru a ascunde de browserele vechi, care nu cunosc sintaxa CSS, conținutul blocului *style*.

- Folosirea unui fișier **extern** sau nivel 3 care conține comenzi CSS poate fi utilizat dacă mai multe fișiere HTML vor accesa același fișier extern CSS, eliminând timpul necesar introducerii codului corespunzător în fiecare pagină și totodată editarea comenzilor într-un singur loc pentru mai multe fișiere.
- Extensia acestor fișiere este .css.
- Legătura paginilor HTML cu fișierele externe CSS se face prin

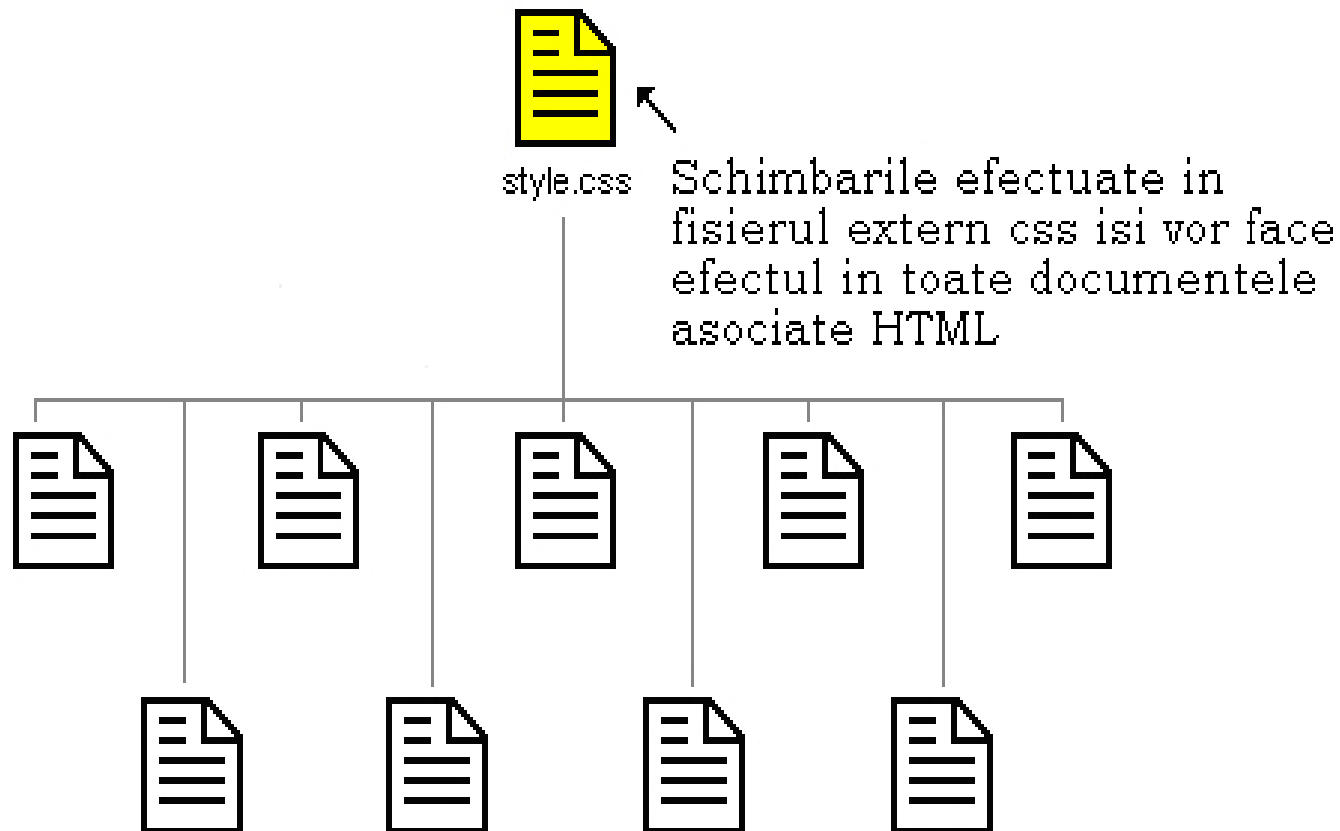
<link rel="stylesheet" type="text/css" href="fisier_css.css">

Atributele indică:

rel - fișierul este tip stylesheet

type - tip text ce conține comenzi CSS

href - fișierul sau adresa fișierului CSS



Documente HTML cu legatura catre acelasi style css

- **Etichetele CSS** sunt cunoscute sub numele de **elemente sau selectori**

eticheta HTML

`<eticheta proprietate="valoare">`

eticheta CSS

`element {proprietate : valoare;}`

Obs. *element* poate fi o **eticheta HTML**, un **id** sau un **class**, iar proprietatea și valoarea sunt aceleași ca în HTML.

- Sintaxa generală a unei declarații de stil este:

```
listă_selectori{ proprietatea_1:listă_valori1;  
                proprietatea_2:listă_valori2; ...  
                proprietatea_n:listă_valorin; }
```

Selectorii sunt utilizați pentru determinarea elementelor HTML asupra cărora vor fi aplicate stilurile.

Ex. `h1 { font-family:"Arial"; font-size:15px; }`

Obs. Proprietățile și valorile constituie definiția regulii CSS. Acestea se regăsesc sub forma de perechi, despărțite de caracterul două puncte ":", fiecare pereche se termină cu un caracter punct și virgulă ";".

- **Selectorul HTML** — redefinește modul de afișare a conținutului unei etichete HTML

- **Clasa** — o regulă care poate fi aplicată oricărei etichete HTML

<selector class="nume_clasa"> Text </selector>

**.nume_clasa {proprietate:valoare; proprietate:valoare;
...}**

- **Identificator** - stilizează elementele dintr-o pagină care apar o singură dată

< selector id="nume_id"> Text </ selector >

#nume_id {proprietate:valoare; proprietate:valoare;... }

OBS: **class** și **id** sunt comenzi care dau unei formatări CSS un nume. Se folosesc atunci când dorim să aplicăm un style de formatare unei anumite zone.

- **Elementul class** – poate fi aplicat în mai multe locuri în cadrul aceleiași pagini sau în pagini diferite.
- necesită existența comezilor CSS în zona **HEAD** sau într-un **fișier extern**.

Exemplu:

```
<html>
<head> <meta charset="utf-8">
<title>Exemplul 1 </title>
<style type="text/css">
  .rosu{color: #FF0000;}
</style>
</head>
<body>
<p class="rosu">Primul text roșu introdus prin class "rosu"</p>
Text negru
<p class="rosu">Al doilea text roșu introdus prin class "rosu"</p>
</body> </html>
```

Primul text roșu introdus prin class "rosu"

Text negru

Al doilea text roșu introdus prin class "rosu"

- **Elementul id** se aplică unui style de format o singură dată sau la o singură etichetă HTML, dând un *nume* acelui style.
- Acest element necesită existența comezilor CSS în **zona HEAD** sau într-un **fișier extern**.

Exemplu:

```
<html>
<head> <meta charset="utf-8">
<title>Exemplul 2</title>
  <style type="text/css">
    #albastru{color: #0000FF;}
  </style>
</head>
<body>
  <p id="albastru">Text albastru introdus prin id "albastru"</p>
  Text negru
</body> </html>
```

Text albastru introdus prin id "albastru"

Text negru

Crearea etichetelor HTML personalizate

- etichetele **** (creează o etichetă în linie) și **<div>** (creează o zonă proprie în pagină)

` ... `

` ... `

.nume_clasa {proprietate1:valoare1; proprietate2:valoare2;}

#nume_id{proprietate1:valoare1; proprietate2:valoare2; ... }

Exemplu:

`propoziție scrisă cu verde`

`.verdeboldtext {`

`font-size: normal;`

`color: green;`

`font-weight: bold;`

`}`

propoziție scrisă cu verde

- Pentru configurarea unui bloc separat de restul conținutului unui document HTML se folosește eticheta **<div>**. Aceasta creează o zonă proprie în pagină, cu linie nouă atât deasupra sa cât și dedesubtul său.

- etichete <div>

<div class="nume_clasa"> ... </div>

<div id="div1"> ... </div>

- Definirea acestor etichete <div> într-un cod CSS se face astfel:

div { proprietate1:valoare1; proprietate2:valoare2; ... }

.nume_clasa { proprietate1:valoare1; proprietate2:valoare2;
... }

#div1 { proprietate1:valoare1; proprietate2:valoare2; ... }

```
<!DOCTYPE html>
<html> <head> <meta charset="utf-8"/>
<title>HTML div Tag</title> </head>
<body>
<!-- primul grup -->
  <div style="color:red">
    <h4>Acesta este primul grup</h4>
    <p>Lista de legume</p>
    <ul> <li>Roșii</li>
        <li>Ardei</li>
        <li>Cartofi</li>
        <li>Salată</li>
    </ul>
  </div>
<!-- al doilea grup -->
  <div style="color:green">
    <h4>Acesta este al doilea grup</h4>
    <p>Lista de fructe</p>
    <ul>   <li>Măr</li> <li>Pară</li>
          <li>Prună</li> <li>Cireșe</li>
    </ul>
  </div> </body> </html>
```

Acesta este primul grup

Lista de legume

- Roșii
- Ardei
- Cartofi
- Salată

Acesta este al doilea grup

Lista de fructe

- Măr
- Pară
- Prună
- Cireșe

- Definirea de reguli similare

**selector1, selector2, ... { proprietate1: valoare1;
proprietate2: valoare2; ... }**

- Definirea etichetelor imbricate

**selector1 selector2 ... { proprietate1: valoare1;
proprietate2: valoare2; ... }**

- stilurile etichetelor din interior preiau stilurile etichetelor părinte = moștenirea stilurilor

<http://learn.shayhowe.com/advanced-html-css/complex-selectors/>


```
<!DOCTYPE html>
<html> <head> <style type="text/css">
li
{ color:red;
}
li li
{ color:green;
}
h1,h2
{ color: blue;
}
</style> </head>
<body>
```

Prima mea lista:

```
<ul> <li> primul element din prima lista
<li> al doilea element din prima lista
</ul>
```

A doua lista:

```
<ul><li> primul element din a doua lista
```

```
<ul> <li> primul element din sublista
<li> al doilea element din sublista
</ul>
<li> al doilea element din a doua lista
</ul>
<h1> Primul text
<h2> Textul al doilea
</body> </html>
```

Prima mea lista:

- primul element din prima lista
- al doilea element din prima lista

A doua lista:

- primul element din a doua lista
 - primul element din sublista
 - al doilea element din sublista
- al doilea element din a doua lista

Primul text

Textul al doilea

Determinarea ordinii execuției

1. Regulile CSS scrise în interiorul paginii HTML, în cadrul etichetei "style" din secțiunea "head" au o **prioritate mai mare** decât cele scrise într-un **fișier extern**, iar regulile scrise în **interiorul etichetelor** au o **prioritate mai mare** decât cele din cadrul etichetei "style" din secțiunea "head".
2. Momentul apariției – cu cât un stil apare mai târziu, cu atât importanța lui este mai mare. Astfel, definițiile unei etichete copil au prioritatea mai mare și anulează toate stilurile precedente cu care intră în conflict.

Exemplu de stil CSS în fișier extern

1. Se creează un fișier care să conțină numai descrierea stilurilor și se salvează cu extensia ".css". Conținutul acestui fișier coincide cu conținutul unui bloc `<style>...</style>`, fără ca acești delimitatori să fie incluși.
2. În fișierul HTML care utilizează stilurile definite în fișierul creat la punctul 1, se include în blocul `<head>...</head>` o etichetă **<link>**, având trei atribute:
 - atributul **"rel"** cu valoarea "stylesheet";
 - atributul **"type"** cu valoarea "text/css".
 - atributul **"href"** având ca valoare adresa URL a fișierului creat la punctul 1;

```
<!DOCTYPE html>
<html>
<head>
<LINK REL= "stylesheet" TYPE=
"text/css" HREF= "fis_ext.css">
<title>Pagina de text stilizat</title>
</head>
<body>
<h2>Acesta este text formatat ca
Header 1</h2>
<p>Acesta este un paragraf</p>
<h3>Acesta este text formatat ca
Header 2</h3>
</body>
</html>
```

Fișierul fis_ext.css

```
h2,h3
{ color:#483d8b;
  font-family: "arial";
}
p
{ font-family: "arial";
  margin-left: 10pt;
}
body
{ background-color:yellow;
}
```

Acesta este text formatat ca Header 1

Acesta este un paragraf

Acesta este text formatat ca Header 2

Definirea stilurilor

color:culoare – precizează culoarea textului.

background:culoare URL(imagine) - fundalul folosit.

background-position sunt acceptate două valori:

- în prima poziție poate fi:
top, center, bottom, percentage sau pixel
- în a doua poziție poate fi:
right, center, left, percentage sau pixel

font-family –fontului utilizat pentru text

**selector { font-family:"nume_font1", "nume_font2", ...,
nume_generic }**

*nume generice de fonturi: "serif", "sans-serif", "cursive",
"monospace" sau "fantasy" (opțional)*

- *FONT-STYLE: stil – permite stabilirea stilului folosit*

Ex. P {font-style: *italic* sau *normal* }

- *FONT-WEIGHT – permite stabilirea grosimii pentru text*

Ex. P {font-weight: *bold* sau *bolder* sau *lighter* sau *normal* }

- *FONT-SIZE – permite stabilirea mărimii utilizate pentru text*

- *FONT-DECORATION – permite folosirea unor efecte speciale pentru text*

Ex. P {font-decoration: *underline* sau *overline* sau *line-through* sau *none*}

- *FONT-HEIGHT – permite stabilirea distanței dintre liniile paragrafului.*

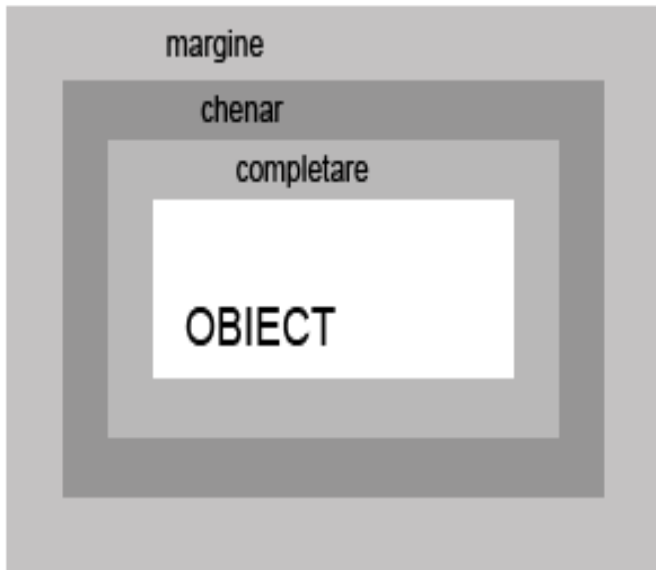
Ex. P {font-height: 8pt sau 150% }

Configurarea textului

- Spațiul între litere:
`selector { letter-spacing:valoare }`
- Spațiul dintre cuvinte:
`selector { word-spacing: valoare }`
- Spațiul între linii: `selector { line-height:valoare }`
- Mărimea (tipul) literelor:
`selector { text-transform: valoare }`
- Alinierea textului: `selector { text-align:valoare }`
- Alinierea pe verticală a textului:
`selector { vertical-align:valoare }`
- Formarea paragrafelor: `selector {text-indent:valoare}`

Fiecare element este încadrat într-o casetă compusă din următoarele elemente:

- ***marginea (margin)*** este spațiul exterior chenarului până la celelalte elemente
- ***chenarul (border)*** este o bordură care înconjoară elementul
- ***completarea (padding)*** stabilește distanța dintre conținut și chenar
- ***Conținutul (Obiectul)*** include informația utilă (text, tabele, imagini, formulare, etc)



Controale pentru chenare și afișarea elementelor

Lățimea și înălțimea unui element

```
selector { width:valoare; height:valoare }
```

Marginile unui element

```
selector { margin-top:valoare; }  
selector { margin-bottom:valoare; }  
selector { margin-left:valoare; }  
selector { margin-right:valoare; }
```

Tipul de **valoare** folosit poate avea următoarele opțiuni:

- o valoare de tip **lungime (numerică)**;
- o valoare **procentuală (procent)** – creează o margine proporțională cu lățimea elementului părinte;
- valoarea **auto** – lasă controlul marginilor la decizia navigatorului.

Chenarul unui element

selector { border:val_grosime val_stil val_culoare; }

Proprietatea **border** permite stabilirea simultană a atributelor chenarului pentru toate cele patru laturi ale casetei. Aceasta folosește 3 valori distincte, pentru a configura lățimea (grosimea), stilul și culoarea:

val_grosime- reprezintă grosimea chenarului și poate fi una din următoarele tipuri: o valoare de tip lungime (valoarea 0 determină anularea afișării chenarului); un cuvânt cheie (**thin, medium, thick**) care caracterizează o dimensiune relativă.

val_stil - reprezintă numele stilului atribuit chenarului (valoarea "none" anulează afișarea chenarului);

val_culoare - reprezintă culoarea, exprimată în cod hexa sau valoare RGB. (**R**ed **G**reen **B**lue) este un model aditiv de culoare, în care culorile albastru, roșu și verde sunt amestecate în diferite moduri pentru a produce o gamă largă de culori.

Chenarul unui element

selector {border_style: valoare;}

- Aspectul bordurii poate fi stabilit prin atributul **border-style**, iar stilul este aplicat celor patru margini .

selector { border-top:grosime stil culoare; }

selector { border-bottom:grosime stil culoare; }

selector { border-left:grosime stil culoare; }

selector { border-right:grosime stil culoare; }

Adaugarea spatiului in interior, in jurul elementului

selector { padding: valoare/valori; }

selector { padding-top:valoare; }

selector { padding-bottom:valoare; }

selector { padding-left:valoare; }

selector { pading-right:valoare; }

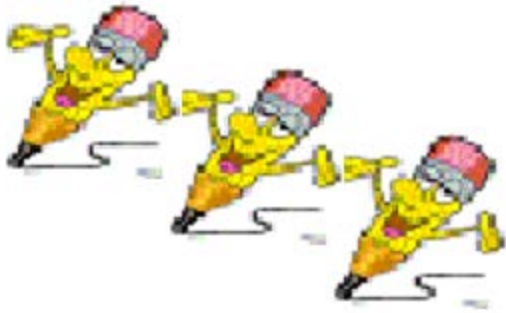
- Poziționarea elementelor folosind CSS este mai precisă decât prin intermediul obiectelor grafice HTML sau a tabelelor, afișarea făcându-se mult mai rapid.
Prin intermediul CSS este permisă poziționarea exactă sau relativă a elementelor într-o fereastră sau în raport cu alte elemente.
Fereastra navigatorului este suprafața în care sunt afișate toate elementele. Ea poate fi redimensionată sau poziționată pe ecran, sau poate fi divizată în alte ferestre prin intermediul cadrelor.
- Toate elementele amplasate în fereastră sunt poziționate relativ la colțul din stânga-sus.
- Tipul de poziție indică navigatorului cum să trateze amplasarea unui element într-o fereastră.

Controale de poziționare

- **Poziționarea statică** **selector { position:static }**
(un element static nu poate fi repositionat în mod explicit)
- **Poziționarea relativă** **selector { position:relative }**
(elementele poziționate relativ pot fi mutate din locația lor folosind proprietățile "top" și "left")
- **Poziționarea absolută** **selector { position:absolute }**
(crează un element neafectat de restul documentului, plasarea lui în fereastră fiind făcută într-o locație precisă definită prin intermediul coordonatelor **x** și **y**, indiferent de pozițiile altor elemente)
- **Poziționarea fixă** **selector { position:fixed }**
(elementul va fi poziționat relativ de fereastră și nu se va mișca chiar dacă vom face **scroll** la pagină)

Poziționare tridimensională

Elementele sunt poziționate pe ecran pe o suprafață bidimensională, dar pot fi așezate și unul deasupra celuilalt, într-o stivă utilizând un indicativ (z-index) începând cu 0, următorul 1 ș.a.m.d. Elementul cu indexul cel mai mare este așezat deasupra.



```
<head><style type="text/css">
<!--
.element1 {position: absolute; left:
30px; top: 30px; z-index: 2}
.element2 {position: absolute; left:
80px; top: 50px; z-index: 1}
.element3 {position: absolute; left:
130px; top: 70px; z-index: 0}
-->
</style> </head>
<body>
<div class="element1"></div>
<div class="element2"></div>
<div class="element3"></div>
</body>
```

Stiluri pentru legături

- In HTML culoarea legăturilor poate fi stabilită prin attributele LINK, ALINK si VLINK declarate in interiorul etichetei BODY.

a definește stilul general pentru legatură

a:link definește stilul legăturii nevizitate

a:visited definește stilul legăturii vizitate

a:active definește stilul legăturii active (nu prea se folosește)

a:hover definește stilul când mouse-ul este deasupra legăturii

```
<html>
<head> <title>Exemplu </title>
<style type="text/css">
<!--
a{font-family: arial;font-size:20px;}
a:link {color: #0000FF;}
a:visited {color: #00FF00;}
a:hover {color: red;}
-->
</style> </head>
<body>
Text normal
<a href="ex2.html">link</a> catre
exemplul 2
</body> </html>
```

[ex1.html](#)

Stiluri pentru liste

În cazul listelor singurul lucru care se poate stiliza este marcatorul listei, fie că este o listă ordonată sau neordonată. Proprietățile CSS pentru stilizarea listelor sunt:

- **list-style-type** – specifică tipul marcatorului listei
- **list-style-image** - specifică imaginea folosită ca marcator
- **list-style-position** - specifică dacă marcatorul lista va apărea în interiorul sau exteriorul conținutului casetei asociate
- **list-style** - proprietatea prescurtată, pune toate proprietățile de stilizare a listelor într-una singură

list-style-type

selector {list-style-type:valoare;}

Valoarea poate fi:

- disc
- circle
- square
- none
- decimal
- lower-roman
- upper-roman
- lower-alpha
- upper-alpha

```

<!DOCTYPE html>
<html> <head> <meta charset="utf-8"/>
<title>Exemplul 3</title>
<style type="text/css">
.ex_1
{list-style-type:square;}
.ex_2
{list-style-type:circle;}
.ex_3
{list-style-type:none;}
</style> </head> <body>
Necesar materiale:
<ol>
    <li>caiete</li>
    <li>creioane</li>
    <li>hârtie</li>
</ol>
<h2> LISTA FRUCTE </h2>
<ul class='ex_1'>
    <li>portocale</li>
    <li>lamai</li>
</ul>

```

```

<ul class='ex_2'>
    <li>mere</li>
    <li>pere</li>
</ul>
<ul class='ex_3'>
    <li>exemplul 1</li>
    <li>exemplul 2</li>
</ul>
</body> </html>

```

Necesar materiale:

1. caiete
2. creioane
3. hârtie

LISTA FRUCTE

- portocale
- lamai
- mere
- pere

exemplul 1
exemplul 2

list-style-image

```
<!DOCTYPE html>
<html> <head> <meta charset="utf-8"/>
<style type="text/css">
<!--
li {list-style-image: url('desktop.png');}
-->
</style> </head> <body>
Listă necesar materiale:
<ol>
    <li>mouse</li>
    <li>hard disk</li>
    <li>memorie</li>
</ol> </body> </html>
```

Listă necesar materiale:



mouse



hard disk



memorie

```
<!DOCTYPE html>
<html> <head>
<style type='text/css'>
.ex_li
{list-style-type:square;
 list-style-image:url('bulina.jpg');}
</style>
</head> <body>
<ul class='ex_li'>
    <li>cafea</li>
    <li>ceai</li>
    <li>suc de portocale</li>
    <li>suc de roșii</li>
</ul>
</body> </html>
```

■ cafea

■ ceai

■ suc de portocale

■ suc de roșii

Afișarea alăturată a articolelor listelor

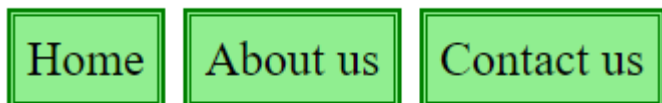
Listele sunt adesea folosite la crearea barelor de navigare orizontale. Pentru aceasta ele sunt așezate una lângă alta. Acest lucru se poate realiza, prin

display:inline;

și

list-style-type:none;

(pentru a nu afișa marcatorii)



```
<head>  <style type='text/css'>
.ex6
    {display:inline;
      list-style-type:none;
      background-color:lightgreen;
      border:3px double green;
      padding:5px 4px;
      margin:10px 1px;}
</style> </head>
<body>
  <ul>
    <li class='ex6'>Home</li>
    <li class='ex6'>About us</li>
    <li class='ex6'>Contact us</li>
  </ul>
</body>
```

Proprietăți CSS specifice pentru stilizarea tabelelor

Proprietatea	Descriere
border-collapse	Determină dacă contururile tabelului se vor uni
border-spacing	Determină spațiul dintre contururile a două căsuțe alăturate
caption-side	Determină locul de amplasare a titlului tabelului
empty-cells	Determină dacă se afișează sau nu, conturul și fundalul pentru căsuțele goale
table-layout	Determină șablonul folosit pentru tabel

**selector{border-collapse
: valoare;}**

collapse- contururile se unesc intr-un singur contur, dacă este posibil (vor fi ignorate proprietățile **border-spacing** și **empty-cells**)

separate- valoare predefinita. Afișează contururi separate pentru fiecare căsuță și tabel

inherit – forțează elementul să moștenească valoarea proprietății elementului părinte.

```
<style type='text/css'>
#ex1_bc
{border-collapse:collapse;}
table, td
{border:2px solid green;}
</style>
<table class='ex1_bc'>
<tr>
<td>Nr.crt.</td>
<td>Nume</td>
<td>Adresa</td>
</tr>
<tr>
<td align=center>1</td>
<td>Popescu Nicolae</td>
<td>Timișoara, str.V.Alecsandri,nr.1</td>
</tr>
</table> ex\_1\_tab.html
```

**selector{border-spacing :
val-num;}**

```
<style type='text/css'>  
.ex2_bs  
  {border-spacing: 10px;}  
table, td  
  {border:2px solid green;}  
</style>
```

```
<table class='ex2_bs'>  
  <tr>  
    <td>&nbsp;</td>  
    <td>Nume</td>  
    <td>Adresa</td>  
  </tr>  
  <tr>  
    <td>1</td>  
    <td>Popescu Nicolae</td>  
    <td>Timișoara, str. V.  
Alecsandri, nr. 1</td>  
  </tr>  
</table><br />
```

[ex_2_tab.html](#)

**selector{caption-side :
valoare;}**

Valorile admise sunt:

- **top** - valoarea predefinită; titlul se așează deasupra tabelului
- **bottom** - titlul se așează sub tabel
- **inherit** – forțează elementul să moștenească valoarea proprietății elementului părinte

```
<style type='text/css'>
```

```
.ex3_cs
```

```
{caption-side:bottom;}
```

```
table, td
```

```
{border:2px solid green;}
```

```
</style>
```

[ex_3_tab.html](#)

selector{empty-cells : valoare;}

Valorile admise sunt:

- **show** - valoarea predefinită; se afișează conturul căsuței goale
- **hide** - conturul căsuțelor goale nu se afișează
- **inherit** – forțează elementul să moștenească valoarea proprietății elementului părinte

```
<style type='text/css'>
```

```
.ex4_ec
```

```
{empty-cells:hide;}
```

```
table, td
```

```
{border:2px solid green;}
```

```
</style>
```

```
<table class='ex4_ec'>  
  <caption>Tabel locatari</caption>  
  <tr>  
    <td></td>  
    <td>Nume</td>  
    <td>Adresa</td>  
  </tr>  
  <tr>  
    <td>1</td>  
    <td>Popescu Nicolae</td>  
    <td>Timișoara, str. V. Alecsandri,  
    nr. 1</td>  
  </tr>  
</table>
```

[ex_4_tab.html](#)

selector{table-layout : valoare;}

Valorile admise sunt:

- **auto** - valoarea predefinită; lăţimea căsuţelor tabel se întind cât să cuprindă conţinutul
- **fixed** – căsuţele dacă sunt dimensionate nu-şi modifică lăţimea
- **inherit** – forţează elementul să moştenească valoarea proprietăţii elementului părinte

[ex5_tab.html](#)

```
<style type='text/css'>  
.ex5_tl  
    {table-layout:fixed;}  
.ex5_d  
    {width:40%;}  
table, td  
    {border:2px solid green;}  
</style>
```

Exemplu formatare tabel folosind
fişier extern css

[ex_6_tab.html](#)

[fis_stil1.css](#)

Animații prin CSS

animation – are rolul de a spune browser-ului că elementul în care se află această proprietate va fi animat

@keyframes – arată locul unde are loc animația

animation-delay –unitatea de timp cu care va întârzia animația

animation-direction –dacă animația va decurge într-un singur sens sau nu

animation-duration- durata în milisecunde sau secunde va dura animația la o rulare

animation-fill-mode – ce stiluri se aplică pe un element când animația este la sfârșit sau are o întârziere

animation-iteration-count – numărul de rulări ale animației

animation-name – numele animației

animation-play-state – animația este în pauză sau rulează

animation-timing-function – viteza de rulare

@keyframes example {

0% {background-color: red; left:0px; top:0px;}

25% {background-color: blue; left:0px; top:200px;}

50% {background-color: green; left:200px; top:200px;}

75% {background-color: yellow; left:200px; top:0px;}

100% {background-color: red; left:0px; top:0px;}

}

[ex_animatie1.html](#)

<https://daneden.github.io/animate.css/>

<https://robots.thoughtbot.com/css-animation-for-beginners>

CSS3 transition

- poate fi utilizat pentru a anima proprietățile CSS, adăugând un efect de animație când se modifică proprietățile CSS ale unui element HTML, schimbând gradual de la un stil la altul

CSS3 transition are 4 componente:

- **transition-property** - Specifica numele proprietatii (sau proprietatilor) la care transition trebuie aplicat (precum: width, color, font-size, etc.)
- **transition-duration** - Specifica durata tranzitiei (animatiei), in secunde (s), milisecunde (ms), (implicit 0).
- **transition-timing-function** - Defineste tipul vitezei efectului in timpul transformarii: **ease** (implicit), **linear**, **ease-in**, **ease-out**, **ease-in-out**
- **transition-delay** - Defineste timpul de asteptare pana cand incepe efectul "transition" (implicit 0)

Stiluri pentru imagini in CSS

- border-radius [ex_imagini_1.html](#)
- border-radius-left
- border-radius-right
- border-radius-top
- border-radius-bottom

- border [ex_imagini_2.html](#)
- opacity - seteaza nivelul de opacitate al unui element HTML, *opacity* permite modificarea transparentei elementului [ex_imagini_3.html](#)

Exemplu de stil CSS în fișier extern

1. Se creează un fișier care să conțină numai descrierea stilurilor și se salvează cu extensia ".css". Conținutul acestui fișier coincide cu conținutul unui bloc `<style>...</style>`, fără ca acești delimitatori să fie incluși.
2. În fișierul HTML care utilizează stilurile definite în fișierul creat la punctul 1, se include în blocul `<head>...</head>` o etichetă **`<link>`**, având trei attribute:
 - atributul **"rel"** cu valoarea "stylesheet";
 - atributul **"type"** cu valoarea "text/css".
 - atributul **"href"** având ca valoare adresa URL a fișierului creat la punctul 1; [pag_css.html](#)