

LUCRAREA 4. FOI DE STILURI (CSS)

CSS (*Cascading Style Sheets*) este unul din limbajele de stiluri utilizate pentru prezentarea documentelor scrise într-un limbaj de marcare precum HTML, XHTML (pentru pagini Web), XML (*Extensible Markup Language*), SVG (*Scalable Vector Graphics*) sau XUL (*XML User Interface Language*). Tehnologia CSS este folosită pentru crearea paginilor Web moderne (alături de HTML5 și Javascript), a interfețelor pentru utilizatori sau în aplicații pe dispozitive mobile.

O foaie de stiluri (*style sheet*) este o colecție de reguli stilistice folosite pentru prezentarea unui document (de exemplu, pentru a descrie fonturile, culorile și macheta prezentării vizuale a unui document electronic). Marcajele și atributele pentru formatare apărute în versiunea HTML 3.2 îngreunau crearea site-urilor complexe, în care trebuia ca fiecare pagină să fie individual editată și proprietățile dorite să fie modificate. CSS a devenit soluția necesară pentru formatarea layout-ului paginii și a elementelor componente.

De-a lungul timpului au apărut mai multe versiuni CSS și profiluri; versiunile aduc noi facilități și se bazează pe cele anterioare, în timp ce profilurile sunt subseturi concepute pentru un anumit dispozitiv sau o interfață particulară (de exemplu în televiziune sau tipografii, pentru dispozitive mobile sau imprimante).

Într-o pagină Web se pot insera următoarele tipuri de stiluri:

- **stil local** (pentru un singur element HTML): este o metodă recomandată doar dacă se dorește modificarea proprietăților pentru o singură apariție a unui element (stilul se aplică doar acelei instanțe a elementului, iar pentru fiecare trebuie specificat încă o dată). Pentru a defini un stil local (*inline*) se folosește atributul global *style* în interiorul elementului care va fi formatat:

```
<p style="color: red; margin-left: 20px;">Paragraf care foloseste un stil</p>
```

- **set de stiluri interne** (în interiorul antetului - secțiunea `<head>`): metoda este recomandată când o singură pagină folosește acel set de stiluri, deoarece acestea sunt valabile doar în documentul respectiv. Se folosește elementul de marcare `<style>`, iar în interiorul acestuia se introduc proprietățile și valorile respective (vezi punctul 8 din Capitolul 3):

```
<head>
<style type="text/css">
hr { color: blue; }
p { margin-left: 20px; }
body { background-color: green; }
</style>
</head>
```

- **foaie de stiluri externe** (într-un fișier extern cu extensia tipică `.css`): este în general metoda recomandabilă de utilizare a stilurilor deoarece aspectul întregului site, nu doar al unei pagini, se poate controla unitar prin editarea unui singur fișier. O modificare în fișierul de stiluri se va propaga în toate paginile site-ului care au o referință spre acest fișier. Pentru a defini o legătură (referință) către un fișier extern de stiluri se folosește marcajul `<link>` în interiorul secțiunii `<head>` (vezi punctul 7.4. din Capitolul 3):

```
<head>
<link rel="stylesheet" type="text/css" href="stil.css">
</head>
```

Fișierul `.css` poate fi creat cu orice editor de texte și nu conține marcaje HTML, ci doar expresii pentru definirea stilurilor. Începând cu HTML 4.0 (1997), se recomandă mutarea tuturor formătărilor într-un fișier extern de stiluri.

4.1. Sintaxa

O foaie de stiluri constă într-o listă de reguli, fiecare regulă fiind formată la rândul său din unul sau mai mulți *selectori* (elementele pe care dorim să le stilizăm) și un *bloc de declarații* (între acolade) care conține *proprietățile* selectorilor (atributele pe care dorim să le specificăm) și *valorile* asociate acestora.

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

4.1.1. Selector

Este folosit pentru a identifica (selecta) elementele care se vor stiliza.

Selector de element

Pentru a selecta un anumit element de marcare, în numele selectorului se folosește acel nume al elementului.

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

Exemplu: "toate elementele de marcare de tipul <p> vor fi înclinate și de culoare galbenă".

Definire	Utilizare	Observații
p { font-style: italic; color: yellow; }	<p id="par1">Paragraf 1</p> <p>Paragraf 2</p>	Ambele paragrafe vor avea aceleași proprietăți (text înclinat, culoare galben)

Selector de identificator (#id)

Pentru a selecta elementul cu un anumit identificator *id* (unic într-o pagină Web, specificat prin atributul global *id*), numele selectorului conține id-ul precedat de semnul # (*number sign* sau *hash character*).

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

Exemplu: "elementul cu id-ul head1 va fi aliniat la dreapta și de culoare verde".

Definire	Utilizare	Observații
#head1 { text-align: right; color: green; }	<h1 id="head1">Titlu</h1>	Elementul <h1> (cu id-ul head1) va avea proprietățile specificate (aliniere la dreapta, culoare verde)

Selector de clasă (.class)

Pentru a selecta toate elementele dintr-o anumită clasă (specificată prin atributul global *class*), numele selectorului conține numele clasei precedat de semnul . (punct).

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

Exemplu: "toate elementele din clasa clasa1 vor fi scrise cu font Arial de 12 pt".

Definire	Utilizare	Observații
.clasa1 { font-family: Arial; font-size: 12pt; }	<h1 class="clasa1"> Titlu</h1> <p class="clasa1"> Paragraf1</p> <p>Paragraf2</p>	Primele 2 elemente (cele cu clasa <i>clasa1</i>) vor avea proprietățile specificate (font Arial, dimensiune 12 pt), iar ultimul element <p> va avea proprietățile implicite

Selectorul unui anumit element dintr-o clasă

Pentru a selecta doar anumite elemente dintr-o clasă, în numele selectorului se folosesc numele elementului și numele clasei, separate prin semnul punct.

element.class { proprietate1: valoare1; proprietate2: valoare2; ... }

Exemplu: "toate elementele <p> din clasa clasa2 vor fi scrise cu font Arial de 12 pt".

Definire	Utilizare	Observații
p.clasa2 { font-family: Arial; font-size: 12pt; }	<h1 class="clasa2">Titlu</h1> <p class="clasa2">Paragraf1</p> <p>Paragraf2</p>	Doar elementul <p> cu clasa <i>clasa2</i> (al doilea element) va avea proprietățile specificate (font Arial, dimensiune 12 pt)

Selector pentru un grup de elemente

Pentru minimizarea codului, selectorii cu proprietăți comune se pot grupa prin enumerare (separați prin virgulă):

element1, element2 { proprietate1: valoare1; proprietate2: valoare2; ... }

Exemplu: "toate elementele de tip Heading nivel 1 și 2 vor fi aliniate la centru și subliniate".

h1, h2 { text-align: center; text-decoration: underline; }

Selector pentru toate elementele

Pentru a specifica toate elementele din pagina Web se folosește selectorul * (*asterisk*):

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

Exemplu: "toate elementele vor fi scrise cu galben și vor avea fundal verde".

* { color: yellow; background-color: green; }

Combinarea selectorilor

Pentru a preciza cu exactitate și mai mare elementele care se stilizează, există următoarele metode de combinare a selectorilor:

- element1 element2 { ... } - se aplică pentru toate elementele *element2* descendente ale elementului *element1* (nu neapărat urmași direcți - copii):

Definire	Utilizare	Observații
div p { background-color: green; }	<div> <p>Paragraf 1</p> <p>Paragraf 2</p> </div>	Se aplică în acest caz pentru Paragraf 1 și Paragraf 2

- element1 > element2 { ... } - se aplică pentru toate elementele *element2* care sunt descendente imediate ale elementului *element1* (urmași direcți - copii, nu descendenți indirecti):

Definire	Utilizare	Observații
div > p { background-color: green; }	<div> <p>Paragraf 1</p> <p>Paragraf2</p> </div>	Se aplică în acest caz doar pentru Paragraf 1

- element1 + element2 { ... } - se aplică pentru elementul *element2* situat imediat după *element1*, pe același nivel (*element1* și *element2* sunt frați adiacenți):

Definire	Utilizare	Observații
div + p { background-color: green; }	<div><p>Paragraf 1</p></div> <p>Paragraf 2</p> <p>Paragraf 3</p>	Se aplică în acest caz doar pentru Paragraf 2

- $element1 \sim element2 \{ \dots \}$ - se aplică pentru toate elementele *element2* de pe același nivel cu *element1* (*element1* și *element2* sunt frați):

Definire	Utilizare	Observații
<code>div ~ p</code> <code>{ background-color: green; }</code>	<code><div></code> <code><p>Paragraf 1</p></code> <code></div></code> <code><p>Paragraf 2</p></code> <code><p>Paragraf 3</p></code>	Se aplică în acest caz pentru Paragraf 2 și Paragraf 3

4.1.2. Valori

În funcție de proprietatea folosită, valorile acesteia pot fi:

- numere întregi (0...9) sau reale (punctul este separator zecimal), pozitive sau negative. Valorile numerice pot fi urmate de o unitate de măsură a lungimii.
- procente, calculate relativ la o altă valoare, de obicei o unitate de lungime;
- șiruri de caractere - dacă există ghilimele simple sau duble sau alte caractere speciale, acestea trebuie precedate de câte un \ (*backslash*);
- nume de culori, nume de fonturi;
- cuvinte cheie (*auto*, *inherit*, *initial*, *unset*) și cuvinte rezervate, definite în specificațiile CSS (*normal*, *bold*, *center*, *solid*, *absolute* etc.);
- URL (*Uniform Resource Locator*, adresa unei imagini pentru fundalul unui element sau pentru o listă).

4.1.3. Unități de măsură

CSS pune la dispoziția utilizatorilor mai multe tipuri de unități pentru a exprima lungimea (mărimea fontului, dimensiunile obiectelor, distanțele față de margini, spațiile libere sau chenarele). Unele mărimi au o istorie veche în tipografie (*point*, *pica*), altele sunt uzual folosite (*inch*, *cm*), iar odată cu dezvoltarea tehnologiei informației a apărut o nouă unitate de măsură: *pixel*. Proprietățile pot avea oricare dintre tipurile de unități de măsură, însă trebuie să se țină cont unde va fi folosit acel stil: la documente tipărite sau pentru afișarea pe ecran.

- **unități relative**, folosite în general atunci când materialul va fi afișat pe un ecran, având în vedere varietatea de dispozitive și rezoluțiile acestora. Unitățile sunt relative la dimensiunea fontului sau la dimensiunea *viewport*-ului (fereastra browser-ului):
 - **em** - este dimensiunea calculată a fontului curent, diferă pentru fiecare element și depinde de setările dpi (*Dots per inch*, densitatea punctelor pentru elementele tipărite) / *ppi* (*Pixels per inch*, densitatea pixelilor la un monitor, TV, cameră video sau scanner), tipul de font și dimensiunea lui. Este folosită pentru a crea machete scalabile care păstrează aspectul paginii când se modifică dimensiunea fontului. Exemplu: `.clasa1 {font-size: 10px; padding: 2em; }` - valoarea proprietății *padding* va fi de $2 \times 10\text{px} = 20\text{px}$.
 - **rem** (*root em*) - relativ la mărimea fontului elementului rădăcină, `<html>`; spre deosebire de *em*, care poate varia pentru fiecare element, *rem* este constantă în document.
 - **ex** (*x-height*) - înălțimea literei "x", este distanța dintre linia de bază și linia de mijloc a literei minuscule x din fontul curent; comparativ cu *em*, este o mărime mai puțin folosită și mai instabilă în browsere. În multe fonturi $1\text{ex} \approx 0,5\text{em}$.
 - **ch** (*character unit*, lățimea caracterului "0" - zero);
 - **vw** (relativ la 1% din lățimea - *width* - *viewport*-ului);
 - **vh** (relativ la 1% din înălțimea - *height* - *viewport*-ului);
 - **vmin** (relativ la 1% din dimensiunea cea mai mică a *viewport*-ului);
 - **vmax** (relativ la 1% din dimensiunea cea mai mare a *viewport*-ului);
 - **%** (unități procentuale scrise sub forma: număr%, +număr% sau -număr%).

- **unități absolute**, măsurabile fizic, folosite acolo unde proprietățile mediului de ieșire sunt cunoscute, de exemplu pentru materialele tipărite:
 - **in** (*inch*; 1 in = 2,54 cm);
 - **cm** (*centimetru*; 1 cm = 10 mm);
 - **mm** (*milimetru*);
 - **pt** (*point*; 1 pt = 1/72 in);
 - **pc** (*pica*; 1 pc = 12 pt);
 - **px** (*pixel*) - nu depinde de fontul curent și nici de unități de măsură precum cm sau inch, ci de echipament.

Relațiile de transformare sunt:

$$1 \text{ in} = 2,54 \text{ cm} = 25,4 \text{ mm} = 72 \text{ pt} = 6 \text{ pc} = 96 \text{ px}$$

Observații:

- între valoarea numerică a proprietății și unitatea de lungime nu trebuie să existe spațiu, altfel pot apărea probleme în anumite browsere. Exemplu: width: 20px, nu ~~width: 20 px~~ (între 20 și px nu trebuie să existe spațiu).
- unele proprietăți acceptă mărimi negative fără a genera o eroare de sintaxă.

4.1.4. Culori. Modelele RGB și CMYK

Un **sistem de culoare aditiv** este folosit în media care utilizează lumina emisă pentru a crea senzația unor culori diverse (de exemplu, pentru televizoare, monitoare PC, ecranele dispozitivelor mobile sau camere digitale foto/video). Sinteza aditivă constă în obținerea luminii de o anumită culoare prin combinarea (adunarea) unor surse de lumină având culori fixate, denumite culori primare, cu diferite intensități.

În **modelul de culoare RGB** se utilizează culorile primare roșu (**Red**), verde (**Green**) și albastru (**Blue**). Dacă se adună cele trei lumini în culorile primare roșu, verde și albastru se obține alb (**White**).

Culori secundare (se obțin prin amestecul culorilor primare):

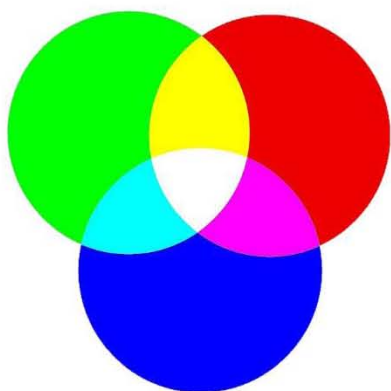
- **Cyan** = verde (Green) + albastru (Blue)
- **Magenta** = albastru (Blue) + roșu (Red)
- **Yellow** = verde (Green) + roșu (Red)

Culori terțiare (se obțin prin amestecul unei culori primare cu una secundară):

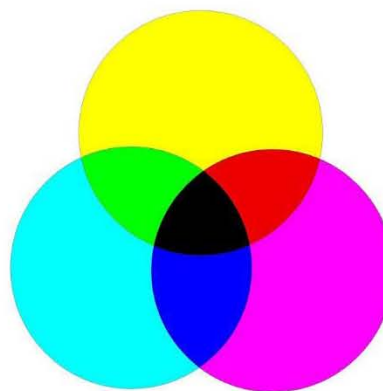
- **Azure** = Cyan + Blue
- **Violet** = Blue + Magenta
- **Rose** = Magenta + Red
- **Orange** = Red + Yellow
- **Chartreuse Green** = Yellow + Green
- **Spring Green** = Green + Cyan

Media care se bazează pe lumina reflectată și coloranți (vopsele, cerneluri, pigmenți și alte substanțe) pentru a produce culori folosește un **sistem de culoare substractiv**. În sinteza substractivă, culoarea se obține prin filtrări succesive ale unei surse albe prin filtre de culori și "tării" diferite.

În tipărirea color se folosesc pigmenți în culorile primare turcoaz (**Cyan**), purpuriu (**Magenta**) și galben (**Yellow**) (se observă că sunt exact culorile secundare de la modelul RGB). Uzual, pentru a acoperi anumite zone întunecate sau de detaliu ale imaginii tipărite se folosește cerneală neagră, componenta K (**Key/Black**) din **modelul de culoare CMYK**. Cyan este complementul lui Red (poate fi folosit ca un filtru care absoarbe roșu) și complet transparent pentru Green și Blue. Similar, Magenta este complementul lui Green, iar Yellow complementul lui Blue. Dacă se adună cele trei culori primare se obține (teoretic) culoarea negru (**Black**).



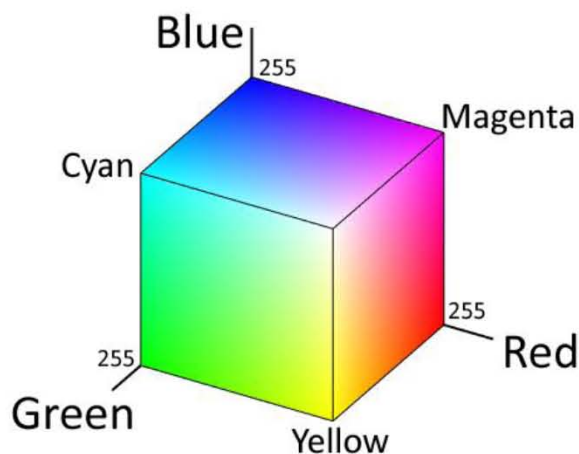
Model RGB (Red-Green-Blue)



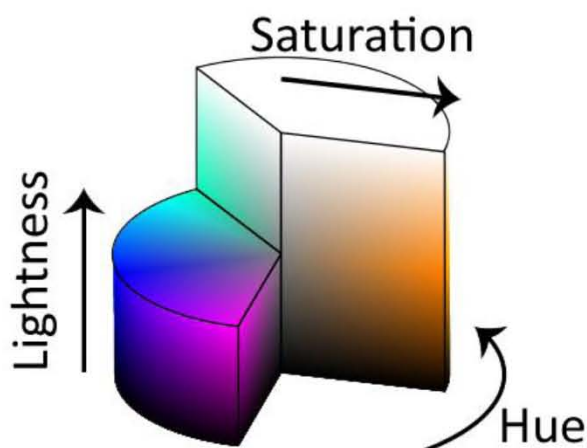
Model CMYK (Cyan-Magenta-Yellow-Key)

Modelul de culori RGB poate fi reprezentat atât în sistemul de coordonate carteziene tridimensional (sub formă de cub), cât și în sistemul de coordonate cilindrice, mult mai intuitiv și relevant. În acest ultim caz cele mai comune moduri de reprezentare sunt:

- **spațiul de culori HSL** folosește termenii *Hue* (nuanță), în grade ($0...359^\circ$), *Saturation* (saturație) și *Lightness* (luminozitate), exprimate în procente ($0...100\%$);
- **spațiul de culori HSV**, care descrie o culoare folosind termenii *Hue*, în grade, *Saturation* și *Value* (valoare, similară strălucirii - *Brightness*), indicate în procente.

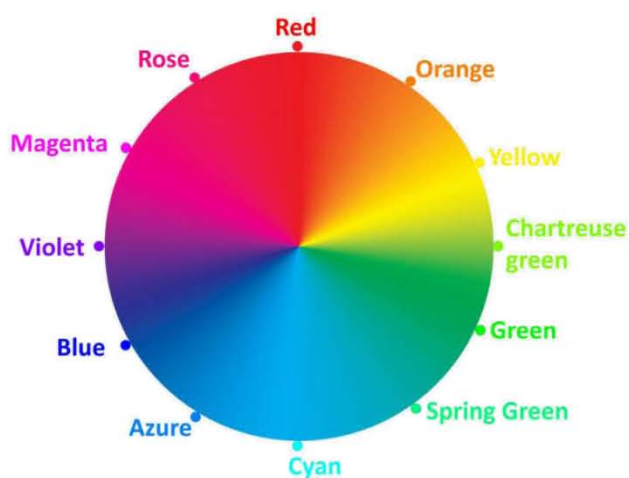


(a)



(b)

Unghi	Culoare	R, G, B	HEX
0°	Red	255, 0, 0	#FF0000
30°	Orange	255, 127, 0	#FF7F00
60°	Yellow	255, 255, 0	#FFFF00
90°	Chartreuse green	127, 255, 0	#7FFF00
120°	Green	0, 255, 0	#00FF00
150°	Spring green	0, 255, 127	#00FF7F
180°	Cyan	0, 255, 255	#00FFFF
210°	Azure	0, 127, 255	#007FFF
240°	Blue	0, 0, 255	#0000FF
270°	Violet	127, 0, 255	#7F00FF
300°	Magenta	255, 0, 255	#FF00FF
330°	Rose	255, 0, 127	#FF007F



(c)

Modelul de culori RGB: a) Cub RGB; b) Cilindru HSL; c) Roata culorilor (*color wheel*)

Toate editoarele vizuale pentru pagini Web sau programele pentru prelucrări grafice includ o aplicație de tip "*color picker*" utilă pentru alegerea vizuală a unei culori și obținerea codului în diverse formate. Printre programele freeware dedicate obținerii codului unei culori se pot enumera ColorPic (iconico.com), ColorExpress (tembolab.pl) sau Absolute Color Picker (www.eltime.com). Printre cele mai complete aplicații online, cu facilități complexe în combinarea culorilor, se află:

- <http://www.dematte.at/colorPicker/>
- <http://www.workwithcolor.com/hsl-color-picker-01.htm>
- <http://hslpicker.com/>
- http://www.w3schools.com/tags/ref_colorpicker.asp
- <http://www.colorpicker.com/>
- <http://www.eyecon.ro/colorpicker/>
- <http://www.rapidtables.com/web/color/color-picker.htm>
- <http://html-color-codes.info/>

În CSS există mai multe moduri pentru a specifica valoarea unei culori:

- **sub formă de nume predefinite:** browserele moderne suportă 140 nume de culori (prezentate în Anexa 1).

Exemplu: `body { background-color: magenta; }`

- **în format hexazecimal:** se folosește formatul `#RRGGBB`, în care *RR* (Red - roșu), *GG* (Green - verde), *BB* (Blue - albastru) sunt cifre în sistemul de numerație hexazecimal (0, ..., 9, A, ..., F) care specifică valorile componentelor primare.

Exemplu: `body { background-color: #FF00FF; }` sau `body { background-color: #F0F; }`

- **în format RGB:** valoarea este exprimată în formatul:
`rgb(red, green, blue)`

unde parametrii *red*, *green*, *blue* definesc intensitatea culorii și au valori întregi între 0 ... 255 sau valori procentuale.

Exemplu: `body { background-color: rgb(255, 0, 255); }`

- **în format RGBA:** este extensia formatului RGB la care se adaugă valoarea *alpha channel* (referitoare la opacitatea elementului). Formatul este:
`rgba(red, green, blue, alpha)`

și este recunoscut în noile versiuni ale browserelor (IE9+, Firefox 3+, Chrome, Safari, Opera 10+). Valorile parametrului *alpha* sunt numere zecimale între 0 (total transparent) și 1 (total opac).

Exemplu: `body { background-color: rgba(255, 0, 255, 0.4); }`

- **în format HSL:** este recunoscut de anumite versiuni ale browserelor (IE9+, Firefox, Chrome, Safari, Opera 10+) și conține valorile pentru *Hue* (nuanță), *Saturation* (saturație), *Lightness* (luminozitate). Formatul este:
`hsl(hue, saturation, lightness)`

unde:

- *hue* - numărul de grade pe roata culorilor (*color wheel*) relativ la culoarea Red - valori numerice de la 0 la 359, unde 0 - Red, 120 - Green, 240 - Blue;
- *saturation* - mărime procentuală de la 0% - ton de gri până la 100% - saturație completă (culoarea nealterată);
- *lightness* - mărime procentuală de la 0% - Black până la 100% - White, iar 50% - o valoare "normală".

Exemplu: `body { background-color: hsl(300,100,50); }`

- **în format HSLA:** este extensia formatului HSL, la care se adaugă informații despre opacitate (parametrul *alpha*). Formatul este:
`hsla(hue, saturation, lightness, alpha)`

unde parametrul *alpha* ia valori zecimale între 0 (total transparent) și 1 (total opac). Formatul este recunoscut de browserele IE9+, Firefox 3+, Chrome, Safari și Opera 10+.

Exemplu: `body { background-color: hsla(300,100,50,0.4); }`

4.1.5. URL

Se indică sub forma:

url (sursă);

Observații:

- opțional sursa se poate insera între ghilimele simple (' ') sau ghilimele duble (" ");
- simbolurile speciale din numele sursei (paranteze, virgule, spații, ghilimele simple sau duble) trebuie precedate de un \ (*backslash*);
- adresa specificată parțial este relativă la fișierul .css și nu la pagina Web în care este folosit fișierul extern de stiluri. Este indicat să se specifice adresa completă a sursei în locul adresei parțiale.

Exemple:

```
body { background: url("background1.jpg"); }
```

```
body { background: url("http://www.example.com/background1.jpg"); }
```

```
body { background: url("\picture1\".bmp); }
```

/* dacă numele fișierului conține ghilimele duble - "picture1".bmp - se folosește backslash ca secvență escape */

4.1.6. Comentarii

Pentru a adăuga o explicație referitoare la cod, proprietăți sau valori, sau pentru a inactiva un bloc de parametri, un comentariu se introduce între semnele /* și */, fiind ignorat de browser. Un comentariu se poate întinde pe mai multe linii.

Exemplu:

```
/* Acesta este  
    un comentariu  
pe mai multe linii */
```

Observație: Pentru versiunile Internet Explorer 8.0 și mai vechi trebuie specificat tipul documentului (marcajul <!DOCTYPE>) pentru ca proprietățile CSS să fie corect implementate. Pentru HTML5 se folosește expresia: <!DOCTYPE html>

4.2. Proprietăți ale textului

În acest capitol se va prezenta formatarea textului folosind proprietăți care controlează tipurile de fonturi, aspectul lor, culoarea, alinierea, spațierea și alte proprietăți specifice.

4.2.1. Font

Proprietățile CSS la nivel de font se referă la stabilirea familiei de fonturi, dimensiunea și stilul textului.

4.2.1.1. Familii de fonturi

În CSS familiile de fonturi pot fi:

- o familie generică - grup de familii de fonturi cu aspect similar: "Serif", "Sans-serif", "Monospace";
- o familie specifică ("Times New Roman", "Arial", etc.)

Proprietatea:

■ font-family: "..."

Valori posibile:

- orice nume de font (dacă acesta conține mai multe cuvinte, se plasează între ghilimele duble - "Courier New");
- o familie generică: sans-serif, serif, monospace, cursive, fantasy.

Se pot indica mai multe fonturi separate prin virgulă. Dacă primul font nu este găsit în sistem, browserul încearcă să afișeze textul cu al doilea font din listă și așa mai departe. De obicei la sfârșitul listei se pune o familie generică de fonturi, dacă niciun font nu este disponibil.

Exemplu:

```
<!DOCTYPE html>
<html>
<head>
<style>
p.monospatiat { font-family: "Courier New", "Lucida Console", monospace; }
</style>
</head>
<body>
<p class="monospatiat">Paragraf scris cu un font monospatiat.</p>
</body>
</html>
```

4.2.1.2. Dimensiune

Pentru dimensiunea fontului se poate indica:

- o valoare absolută: fontul va avea o dimensiune fixă care nu va putea fi modificată în browser (dezavantaj din punctul de vedere al accesibilității). Valoarea absolută se folosește atunci când se cunoaște dimensiunea fizică a mediului de ieșire (de exemplu, la tipărire).
- o valoare relativă: aceasta va permite utilizatorului modificarea mărimii textului în browser. Valoarea este relativă la dimensiunea altor elemente.

Dacă nu se specifică o mărime a fontului, se va folosi valoarea implicită (16 px = 1 em).

Proprietatea:

- font-size: "..."

Valori posibile: xx-small | x-small | small | medium | large | x-large | xx-large | smaller | larger | length | %

Exemplu:

```
p { font-size: small; }
p.clasa1 { font-size: 26px; }
p.clasa2 { font-size: 2em; }
...
<p>Paragraf 1</p>
<p class="clasa1">Paragraf 2</p>
<p class="clasa2">Paragraf 3</p>
```

4.2.1.3. Stil (*italic*)

Pentru stilul înclinat (*italic*/oblic) al textului se folosesc 3 valori: normal (este implicit), italic (textul este înclinat) și oblique (similar cu italic, mai puțin suportat de către browsere).

Proprietatea:

- font-style: "..."

Valori posibile: normal | italic | oblique

Exemplu:

```
p { font-family: verdana; font-size: large; font-style: italic; }
...
<p>Paragraf</p>
```


4.2.1.4. *Grosime (bold)*

Pentru grosimea textului se pot specifica atât cuvinte cheie (normal, bold, bolder, lighter), cât și valori numerice care dau gradul de îngroșare.

Proprietatea:

- font-weight: "..."

Valori posibile: normal | bold | bolder | lighter | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900

Exemplu:

```
p { font-weight: bold; }
```

```
p.clasa1 { font-weight: 700; }
```

...

```
<p>Paragraf 1</p>
```

```
<p class="clasa1">Paragraf 2</p>
```

4.2.1.5. *Variant (small caps)*

Proprietatea afișează textul în format SMALL CAPS (toate literele majuscule, cu o dimensiune mai mică decât majusculele uzuale - pentru comparație, textul SMALL CAPS este scris cu majuscule obișnuite).

Proprietatea:

- font-variant: "..."

Valori posibile: normal | small caps

Exemplu:

```
p { font-variant: small-caps; }
```

4.2.1.6. *Toate proprietățile într-o singură declarație*

Pentru a stabili mai multe proprietăți într-o singură declarație se folosește proprietatea font, urmată de specificarea (în ordine): *font-style font-variant font-weight font-size/line-height font-family*, din care *font-size* și *font-family* sunt obligatorii (în lipsa celorlalte se vor folosi valorile implicite). *line-height* specifică spațiul dintre linii.

Proprietatea:

- font: font-style font-variant font-weight font-size/line-height font-family;

Exemplu:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
p.clasa1 { font: 15px Arial, sans-serif; }
```

```
p.clasa2 { font: italic bolder 2em/40px Georgia, "Times New Roman", serif; }
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p class="clasa1">Paragraf. Paragraf. Paragraf. Paragraf. Paragraf. Paragraf. Paragraf. Paragraf.</p>
```

```
<p class="clasa2">Paragraf. Paragraf. Paragraf. Paragraf. Paragraf. Paragraf. Paragraf. Paragraf.</p>
```

```
</body>
```

```
</html>
```

În cadrul proprietății *font* se poate specifica individual utilizarea fonturilor folosite de sistemul de operare pentru diferite elemente:

Proprietatea:

- font: caption | icon | menu | message-box | small-caption | status-bar | initial | inherit

unde:

caption - fontul folosit în controale (butoane, controale drop-down etc.);

icon - fontul folosit în etichetele icon-urilor;

menu - fontul folosit în meniuri de tip drop-down;

message-box - fontul folosit de casetele de dialog;

small-caption - un font mai mic decât fontul *caption*;

status-bar - fontul folosit în bara de stare (*status bar*) a ferestrei browserului.

Aceste cuvinte cheie nu pot fi folosite împreună cu proprietățile de mai sus (font-family, font-size etc.). Rezultatul acestor proprietăți depinde de browserul folosit.

Exemplu:

```
body { font: menu; }
```

4.2.2. Culoare

Cele mai uzuale valori pentru culoare se specifică în format hexazecimal ("#0000FF" pentru albastru), ca valoare RGB ("rgb(0,0, 255)") sau ca nume de culoare ("blue") - vezi capitolul 2.3 pentru mai multe opțiuni privind modul de indicare a culorilor. Culoarea implicită se specifică în <body>, iar culorile particularizate în cadrul fiecărui element.

Proprietatea:

- color: "..."

Exemplu:

```
body { color: #008000; }
```

```
h1 { color: brown; }
```

...

```
<h1>Titlu maro</h1>
```

```
<p>Paragraf cu verde</p>
```

4.2.3. Aliniere

CSS oferă posibilitatea de a alinia textul atât pe orizontală, cât și pe verticală (relativ la un element înconjurător, de exemplu o imagine).

4.2.3.1. Aliniere pe orizontală

Alinierea textului pe orizontală poate fi: *left* (la stânga, implicită dacă direcția textului este left-to-right), *center* (la centru), *right* (dreapta) sau *justify* (la stânga și la dreapta, textul egal distribuit).

Proprietatea:

- text-align: "..."

Valori posibile: left | center | right | justify

Exemplu:

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

...

```
<p>Paragraf aliniat la centru.</p>
```

4.2.3.2. Aliniere pe verticală

Proprietatea aliniază textul pe verticală față de laturile de sus/jos ale unui obiect înconjurător (o imagine de exemplu).

Proprietatea:

- vertical-align: "..."

Valori posibile: baseline | sub | super | top | text-top | middle | bottom | text-bottom | distanță | % | initial | inherit

Exemplu:

```
img.clasa1 { vertical-align: text-top; }
img.clasa2 { vertical-align: text-bottom; }
...
Imagine cu text aliniat implicit (baseline).</p>
<p>Imagine cu text aliniat sus.</p>
<p>Imagine cu text aliniat jos.</p>
```

4.2.4. Decorare (subliniere)

Se folosește pentru a sublinia textul sau a elimina această subliniere (de exemplu, la linkuri).

Valoarea *blink* depinde de browser (în general nu mai este implementată).

Proprietatea:

- text-decoration: "..."

Valori posibile: none | underline | overline | line-through | blink

Exemplu:

```
p.clasa1 { text-decoration: line-through; }
p.clasa1 { text-decoration: overline underline; }
...
<p class="clasa1">Text taiat cu o linie orizontala.</p>
<p class="clasa2">Text subliniat sus si jos.</p>
```

4.2.5. Transformare

Se folosește pentru a prelucra automat textul și a-l afișa în formatul: toate majuscule, toate minuscule sau prima literă a fiecărui cuvânt majusculă.

Proprietatea:

- text-transform: "..."

Valori posibile: none | capitalize | uppercase | lowercase

Exemplu:

```
p.majuscule { text-transform: uppercase; }
p.minuscule { text-transform: lowercase; }
p.primamajuscula { text-transform: capitalize; }
...
<p class="majuscule">Paragraf cu text prelucrat.</p>
<p class="minuscule">Paragraf cu text prelucrat.</p>
<p class="primamajuscula">Paragraf cu text prelucrat.</p>
```

4.2.6. Indentare

Este folosită pentru a specifica indentarea (aliniatul) primei linii a paragrafului.

Proprietatea:

- text-indent: "..."

Valori posibile: distanță | %

Exemplu:

```
p { text-indent: 30px; }
...
<p>Paragraful are prima linie la o distanta de 30 pixeli fata de marginea ferestrei.
<br>Linia a 2-a incepe de la margine.</p>
```

4.2.7. Spațiere orizontală

Este folosită pentru a specifica distanța dintre caractere pe orizontală.

Proprietatea:

- letter-spacing: "..."

Valori posibile: normal | distanță

Exemplu:

```
p.clasa1 { letter-spacing: 5px; }
```

```
p.clasa2 { letter-spacing: -2px; }
```

...

```
<p class="clasa1">Paragraf.</p>
```

```
<p class="clasa2">Paragraf.</p>
```

4.2.8. Spațiere verticală

Este folosită pentru a specifica distanța pe verticală dintre linii. Distanța se poate preciza în cuvinte cheie (*normal*, *none*), valori zecimale, unități de lungime sau procente. Distanța implicită între linii în majoritatea browserelor este 110%-120% (relativ la mărimea fontului)

Proprietatea:

- line-height: "..."

Valori posibile: normal | distanță

Exemplu:

```
body { font-size: 20px; line-height: normal; }
```

```
p.clasa1 { line-height: 0.7; }
```

```
p.clasa2 { line-height: 200%; }
```

...

```
<p>Paragraf cu distanta implicita intre linii.<br>Paragraf cu distanta implicita intre linii.</p>
```

```
<p class="clasa1">Paragraf cu distanta mai mica intre linii.<br>Paragraf cu distanta mai mica intre linii.</p>
```

```
<p class="clasa2">Paragraf cu distanta mai mare intre linii.<br>Paragraf cu distanta mai mare intre linii.</p>
```

4.2.9. Spațiere între cuvinte

Este folosită pentru a specifica distanța dintre cuvinte pe orizontală.

Proprietatea:

- word-spacing: "..."

Valori posibile: normal | distanță

Exemplu:

```
p.clasa1 { word-spacing: 15px; }
```

```
p.clasa2 { word-spacing: -5px; }
```

...

```
<p>Paragraf cu distanta implicita intre cuvinte. </p>
```

```
<p class="clasa1">Paragraf cu distanta mai mare intre cuvinte.</p>
```

```
<p class="clasa2">Paragraf cu distanta mai mica intre cuvinte.</p>
```

4.2.10. Direcția textului

Este folosită pentru a modifica direcția textului într-un element.

Proprietatea:

- direction: "..."

Valori posibile: ltr | rtl

Exemplu:

```
p.clasa1 { direction: rtl; }
```

...

<p>Paragraf cu directia implicita a textului (stanga spre dreapta). </p>
 <p class="clasa1">Paragraf cu directia textului de la dreapta spre stanga.</p>

4.3. Liste

Prin CSS se pot particulariza elementele de enumerare ale listelor ordonate și neordonate: aspectul marcatorelor și amplasarea lor.

4.3.1. Tipuri de marcatori

Similar listelor ordonate și neordonate din HTML, se poate indica forma marcatorelor (simboluri grafice, cifre - romane, arabe, litere - latine, grecești).

Proprietatea:

- list-style-type: "..."

Valori posibile: disc | circle | square | decimal | lower-roman | upper-roman | lower-alpha | upper-alpha | lower-greek | lower-latin | upper-latin | none

Exemplu:

```
ul.clasa1 { list-style-type: square; }
```

```
ol.clasa2 { list-style-type: lower-greek; }
```

...

```
<ul class="clasa1"><li>Elem1</li><li> Elem2</li><li> Elem3</li></ul>
```

```
<ol class="clasa2"><li>Elem1</li><li> Elem2</li><li> Elem3</li></ol>
```

4.3.2. Imagini ca marcatori

O imagine poate fi folosită ca marcator pentru o listă.

Proprietatea:

- list-style-image: "..."

Valori posibile: url | none |

Exemplu:

```
ul.imagine { list-style-image: url('/img/imagine1.jpg'); }
```

...

```
<ul class="imagine"><li>Elem1</li><li> Elem2</li><li> Elem3</li></ul>
```

4.3.3. Poziția marcatorelor

Se poate controla poziția marcatorelor (în interiorul sau în exteriorul fluxului textului).

Proprietatea:

- list-style-position: "..."

Valori posibile: inside | outside |

Exemplu:

```
ol.clasa1 { list-style-position: inside; }
```

...

```
<ol class="clasa1"><li>Elem1</li><li> Elem2</li><li> Elem3</li></ol>
```

4.3.4. Toate proprietățile într-o singură declarație

Când se folosesc toate proprietățile într-o singură declarație, ordinea acestora este: *list-style-type list-style-position list-style-image*. Dacă una dintre valori lipsește, se folosește valoarea implicită.

Proprietatea:

- list-style: list-style-type list-style-position list-style-image;

Exemplu:

```
ol.clasa1 { list-style: upper-alpha inside; }
```

...

```
<ol class="clasa1"><li>Elem1</li><li> Elem2</li><li> Elem3</li></ol>
```

4.4. Linkuri

Linkurile se pot stiliza ca orice alt element, prin aplicarea culorilor, schimbarea familiei de fonturi și a stilurilor pentru text. În plus, proprietățile se pot adăuga în funcție de starea în care sunt linkurile:

- *a:link* - link nevizitat (normal la accesarea unei pagini);
- *a:visited* - link vizitat;
- *a:hover* - cursorul mouse-ului plasat deasupra linkului;
- *a:active* - link activ (în momentul în care se dă click pe link).

În definirea acestor proprietăți trebuie să se țină cont de următoarele reguli:

- *a:hover* trebuie definit DUPĂ *a:link* și *a:visited*;
- *a:active* trebuie definit DUPĂ *a:hover*.

Exemplu:

```
<style>
```

```
a:link { color: red; }
```

```
a:visited { color: green; }
```

```
a:hover { color: hotpink; }
```

```
a:active { color: blue; }
```

```
</style>
```

...

```
<a href="http://www.example.com">www.example.com</a>
```

4.5. Fundaluri

Proprietățile fundalului unui element constau în alegerea unei culori sau a unei imagini care să se repete într-un anumit mod sau care să defileze odată cu textul.

4.5.1. Culoare

Proprietatea se poate indica printr-o culoare în hexa, o valoare RGB sau un nume valid de culoare.

Proprietatea:

- background-color: "..."

Valori posibile: color-rgb | color-hex | nume culoare | transparent

Exemplu:

```
body { background-color: #ff00ff; }
```

```
p { background-color:rgb(255,0,0); }
```

```
p.clasa1 { background-color: yellow; }
```

...

```
<p>Paragraf cu fundal rosu.</p>
```

```
<p class="clasa1">Paragraf cu fundal galben.</p>
```

4.5.2. Imagine

Proprietatea specifică o imagine care se repetă automat pe fundal astfel încât să umple fie întreaga pagină - elementul <body> -, fie fundalul unui alt element.

Proprietatea:

- `background-image: "..."`
Valori posibile: `<url>` | `none`

Exemplu:

```
body { background-image: url("imagini/img1.jpg"); }  
h1 { background-image: url("http://www.example.com/images/img1.jpg"); }  
...  
<body>  
<h1>Titlu cu fundal </h1>  
</body>
```

4.5.3. Repetarea imaginii

Implicit imaginea se repetă pe ambele axe (orizontală și verticală). În anumite cazuri este necesară repetarea pe o singură direcție.

Proprietatea:

- `background-repeat: "..."`
Valori posibile: `repeat` | `repeat-x` | `repeat-y` | `no-repeat`

Exemplu:

```
body { background-image: url("image1.jpg"); background-repeat: repeat-x; }  
Dacă se dorește afișarea imaginii de fundal o singură dată (fără repetare), se folosește valoarea no-repeat.
```

Exemplu:

```
body { background-image: url("image1.jpg"); background-repeat: no-repeat; }
```

4.5.4. Poziția imaginii

Proprietatea schimbă poziția unei imagini de fundal, astfel încât să fie așezată optim față de text.

Proprietatea:

- `background-position: "..."`
Valori posibile: `procent` | `dimensiune` | `[top|center|bottom]` | `[left|center|right]`

Exemplu:

```
body { background-image: url("image1.jpg"); background-repeat: no-repeat; background-position: right center; }
```

4.5.5. Imagine fixă

Proprietatea păstrează fixă o imagine de fundal, astfel încât să nu se deplaseze odată cu textul atunci când se face scroll.

Proprietatea:

- `background-attachment: "..."`
Valori posibile: `scroll` | `fixed`

Exemplu:

```
body { background-image: url("image1.jpg"); background-repeat: repeat-x; background-attachment: fixed; }
```

4.5.6. Toate proprietățile într-o singură declarație

Pentru a scrie mai puțin cod este posibil ca toate proprietățile să fie înglobate în una singură. Ordinea când se folosește o singură proprietate este: *background-color background-image background-repeat background-attachment background-position*.

Proprietatea:

▪ background: background-color background-image background-repeat background-attachment background-position;

Exemplu:

```
body { background: yellow url("image1.jpg") no-repeat fixed top; }
```

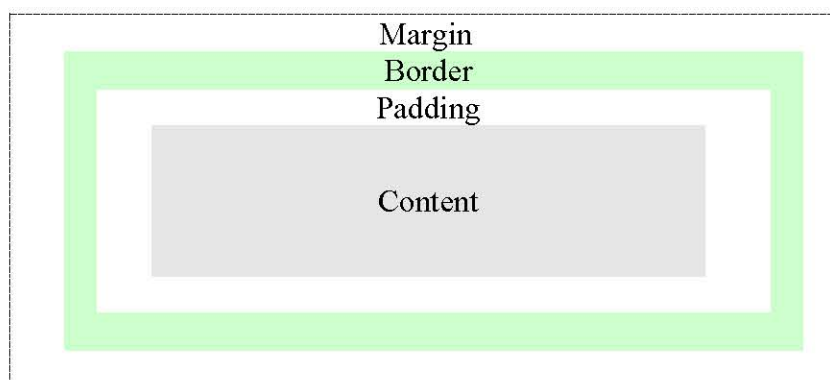
4.6. Modelul Casetă

Modelul Casetă se poate aplica tuturor elementelor HTML. O astfel de casetă înconjoară fiecare element, iar proprietățile care pot fi modificate se referă la chenar, spațiile exterioare între elemente și distanțele interioare.

Elementele componente ale modelului Box (casetă) sunt:

- **Content** - este conținutul efectiv al casetei (în care se pot găsi text, imagini, tabele, obiecte sau alte elemente);
- **Padding** - o zonă lăsată liberă, transparentă, care înconjoară conținutul (*Content*). Această distanță interioară separă părțile *Content* și *Border*.
- **Border** - chenarul înconjoară cele două zone, *Content* și *Padding*. Pentru evidențierea elementului se mai poate adăuga un chenar suplimentar (*Outline*).
- **Margin** - o zonă lăsată liberă, transparentă, situată în exteriorul chenarului (*Border*). Această zonă delimitează practic spațiul dintre elementele paginii Web.

Suplimentar se pot adăuga proprietăți care controlează dimensiunile unui element, poziția lui în pagină și ordinea în cadrul elementelor suprapuse.



4.6.1. Chenar

Proprietatea permite specificarea stilului, lășimii și culorii pentru chenarul unui element.

4.6.1.1. Stil

Proprietatea permite specificarea stilului de linie pentru chenar (linie continuă, punctată, întreruptă, dublă etc.). Proprietatea *border-style* poate avea între o valoare (și atunci cele 4 laturi ale elementului au același tip de linie) și 4 valori (pentru a specifica în ordine stilul de linie pentru marginile *top*, *right*, *bottom* și *left*). Dacă sunt specificate doar 2 valori, prima va fi aplicată pentru *top* și *bottom*, iar a doua pentru *right* și *left*. Dacă sunt specificate doar 3 valori, valoarea din mijloc (corespunzătoare marginii *right*) se aplică și marginii din stânga (*left*).

Proprietatea:

▪ border-style: "..."

Valori posibile: none | hidden | dotted | dashed | solid | double | groove | ridge | inset | outset | initial | inherit

Exemplu:

```
p.clasa1 { border-style: solid; }
```

```
p.clasa2 { border-style: solid dotted dashed none; }
...
<p class="clasa1">Paragraf cu chenar continuu.</p>
<p class="clasa2">Paragraf cu stiluri diferite de chenare.</p>
```

4.6.1.2. Lățime

Lățimea (grosimea) celor 4 chenare poate fi exprimată în unități de măsură (px, pt, cm, em) sau folosind cuvintele cheie *thin*, *medium*, *thick*. Proprietatea *border-width* poate avea între o valoare (și atunci cele 4 laturi ale elementului au aceeași grosime) și 4 valori (pentru a specifica în ordine grosimea chenarului pentru marginile top, right, bottom și left).

Proprietatea:

- border-width: "..."

Valori posibile: thin | medium | thick | dimensiune

Observație: Proprietatea *border-width* funcționează doar dacă se specifică împreună cu *border-style*.

Exemplu:

```
p.clasa1 { border-style: double; border-width: 4px; }
p.clasa2 { border-style: solid; border-width: thin 0 thick 2px; }
...
<p class="clasa1">Paragraf cu chenar de acelasi tip.</p>
<p class="clasa2">Paragraf cu latimi diferite de chenare.</p>
```

4.6.1.3. Culoare

Proprietatea permite stabilirea culorii pentru cele 4 chenare ale unui element. Culoarea poate fi specificată prin numele său (exemplu: "blue"), o valoare hexa (exemplu: "#000000") sau o valoare RGB (exemplu: "rgb(0,0,255)"). Dacă o culoare nu este specificată, ea moștenește culoarea elementului.

Proprietatea:

- border-color: "..."

Valori posibile: color-rgb | color-hex | nume culoare

Observație: Proprietatea *border-color* funcționează doar dacă se specifică împreună cu *border-style*.

Exemplu:

```
p.clasa1 { border-style: solid; border-color: green; }
p.clasa2 { border-style: solid; border-color: #ff0000 #00ff00 blue rgb(255,0,255); }
...
<p class="clasa1">Paragraf cu chenar de aceeași culoare (verde).</p>
<p class="clasa2">Paragraf cu chenare de culori diferite.</p>
```

4.6.1.4. Proprietăți individuale pentru cele 4 chenare

Pentru fiecare dintre cele 4 chenare (top, right, bottom, left) se pot specifica proprietăți individuale (stilul, lățimea și culoarea).

Proprietatea:

- border-top-style: "..."
- border-right-style: "..."
- border-bottom-style: "..."
- border-left-style: "..."

Valori posibile: none | dotted | dashed | solid | double | groove | ridge | inset | outset | hidden

Proprietatea:

- border-top-width: "..."
- border-right-width: "..."

- border-bottom-width: "..."
- border-left-width: "..."

Valori posibile: thin | medium | thick | dimensiune

Proprietatea:

- border-top-color: "..."
- border-right-color: "..."
- border-bottom-color: "..."
- border-left-color: "..."

Valori posibile: color-rgb | color-hex | nume culoare

Exemplu:

```
p.clasa1 {
  border-top-style: dotted;
  border-top-width: thin;
  border-top-color: magenta;
  border-right-style: solid;
  border-right-width: thick;
  border-right-color: blue;
  border-bottom-style: dashed;
  border-bottom-width: 2px;
  border-bottom-color: #aa0521;
  border-left-style: double;
  border-left-width: 6px;
  border-left-color:rgb(255, 127, 80);
}
...
<p class="clasa1">Exemplu de paragraf.</p>
```

4.6.1.5. Toate proprietățile într-o singură declarație

Există proprietăți care comasează proprietățile individuale referitoare la stilul, lățimea și culoarea chenarelor. Astfel, se pot specifica simultan aceste proprietăți simultan fie doar pentru unul dintre chenare (*border-top*, *border-right*, *border-bottom*, *border-left*), fie pentru toate chenarele într-o singură proprietate, *border*.

Proprietatea:

- border-top: "..."
- border-right: "..."
- border-bottom: "..."
- border-left: "..."
- border: "..."

Valori posibile: dimensiune | stil | culoare

Exemplu:

```
p.clasa1 {
  border-top: thick dotted firebrick;
  border-right: thin double #ff0000;
  border-bottom: 5px dashed rgb(100,256,100);
  border-left: medium solid blue;
}
p.clasa2 { border: thick double blue; }
...
<p class="clasa1">Exemplu de paragraf.</p>
```

<p class="clasa2">Exemplu de paragraf.</p>

4.6.2. Chenar pentru evidențiere

Un chenar suplimentar (*Outline*) poate fi trasat în afara chenarului tipic (*Border*) pentru a sublinia importanța elementului și a-l face să iasă în evidență. Deși este diferit de chenarul uzual, chenarul suplimentar suportă aceleași proprietăți (stil, lățime și culoare) și nu afectează dimensiunea efectivă a elementului (formată din dimensiunea zonei Content la care se adaugă dimensiunile specificate în proprietățile padding, border și margin).

4.6.2.1. Stil

Proprietatea:

- `outline-style: "..."`

Valori posibile: none | hidden | dotted | dashed | solid | double | groove | ridge | inset | outset | initial | inherit

Exemplu:

```
p { border: 1px solid red; }
```

```
p.clasa1 { outline-style: dotted; }
```

...

```
<p>Paragraf cu chenar rosu.</p>
```

```
<p class="clasa1">Paragraf cu chenar rosu si chenar suplimentar.</p>
```

4.6.2.2. Lățime

Proprietatea:

- `outline-width: "..."`

Valori posibile: thin | medium | thick | dimensiune

Observație: Proprietatea *outline-width* funcționează doar dacă se specifică împreună cu *outline-style*.

Exemplu:

```
p { border: 1px solid red; }
```

```
p.clasa1 { outline-style: dotted; outline-width: 4px; }
```

...

```
<p>Paragraf cu chenar rosu.</p>
```

```
<p class="clasa1">Paragraf cu chenar rosu si chenar suplimentar.</p>
```

4.6.2.3. Culoare

Proprietatea:

- `outline-color: "..."`

Valori posibile: color-rgb | color-hex | nume culoare

Observație: Proprietatea *outline-color* funcționează doar dacă se specifică împreună cu *outline-style*.

Exemplu:

```
p { border: 1px solid red; }
```

```
p.clasa1 { outline-style: dashed; outline-color: lime; }
```

...

```
<p>Paragraf cu chenar rosu.</p>
```

```
<p class="clasa1">Paragraf cu chenar rosu si chenar suplimentar.</p>
```

4.6.2.4. Offset

Proprietatea adaugă un spațiu suplimentar între chenar (*Border*) și chenarul suplimentar (*Outline*).

Proprietatea:

- `outline-offset: "..."`

Valori posibile: lungime | initial | inherit

Observație: Proprietatea *outline-offset* funcționează doar dacă se specifică împreună cu *outline-style*.

Exemplu:

```
p { border: 1px solid red; }
p.clasa1 { outline-style: double; outline-offset: 5px; }
...
<p>Paragraf cu chenar rosu.</p>
<p class="clasa1">Paragraf cu chenar rosu si chenar suplimentar.</p>
```

4.6.2.5. Toate proprietățile într-o singură declarație

Proprietățile individuale specificate pentru Outline pot fi reunite într-o singură proprietate generală. Trebuie respectată următoarea ordine: *outline-color outline-style outline-width*.

Proprietatea:

- outline: "..."

Valori posibile: outline-color outline-style outline-width | initial | inherit

Exemplu:

```
p { border: 1px solid red; }
p.clasa1 { outline: blue groove 10px; }
...
<p>Paragraf cu chenar rosu.</p>
<p class="clasa1">Paragraf cu chenar rosu si chenar suplimentar.</p>
```

4.6.3. Margini

Proprietatea este folosită pentru a crea spații libere în jurul elementelor. Aceste margini sunt în afara chenarului (*Border*) și sunt transparente (nu pot avea culori pe fundal).

4.6.3.1. Margini individuale

Proprietatea:

- margin-top: "..."
- margin-right: "..."
- margin-bottom: "..."
- margin-left: "..."

Valori posibile: lungime | procent | auto | inherit

Observații: Lungimea se exprimă în unități de măsură (precum px, pt, cm). Marginile pot avea și valori negative.

Exemplu:

```
p { background-color: LightGreen; }
p.clasa1 {
  margin-top: 50px;
  margin-right: 150px;
  margin-bottom: 50px;
  margin-left: 150px;
}
...
<p>Paragraf fara margini impuse.</p>
<p class="clasa1">Paragraf cu margini impuse.</p>
```

4.6.3.2. Toate proprietățile într-o singură declarație

Toate marginile se pot specifica într-o singură proprietate, *margin*. Ordinea este: *margin-top*, *margin-right*, *margin-bottom*, *margin-left*:

- dacă este specificată o singură valoare, toate cele 4 margini vor avea aceeași margine;
- dacă sunt specificate 2 valori, prima va fi pentru marginile sus și jos, iar a doua pentru marginile dreapta și stânga;
- dacă sunt specificate 3 valori, a doua valoare va fi aplicată și marginii din stânga.

Proprietatea:

- `margin: "..."`

Valori posibile: `lungime` | `auto` | `initial` | `inherit`

Exemplu:

```
p { margin: 150px 50px 150px; }
```

...

```
<p>Paragraf.</p>
```

4.6.4. Distanțe interioare

Proprietatea *padding* adaugă margini interioare, între chenar (*Border*) și zona de conținut (*Content*). Zona *padding* este afectată de culoarea de fundal a elementului.

4.6.4.1. Distanțe individuale

Proprietatea:

- `padding-top: "..."`
- `padding-right: "..."`
- `padding-bottom: "..."`
- `padding-left: "..."`

Valori posibile: `lungime` | `procent` | `inherit`

Observații: Lungimea se exprimă în unități de măsură (precum `px`, `pt`, `cm`).

Exemplu:

```
p.clasa1 {
  background-color: lightgreen;
  padding-top: 30px;
  padding-right: 40px;
  padding-bottom: 30px;
  padding-left: 40px;
}
```

...

```
<p class="clasa1">Paragraf cu margini interioare.</p>
```

4.6.4.2. Toate proprietățile într-o singură declarație

Toate proprietățile privind marginile interioare se pot specifica într-o singură proprietate, *padding*. Ordinea proprietăților este: *padding-top* *padding-right* *padding-bottom* *padding-left*:

- dacă este specificată o singură valoare, toate cele 4 margini vor avea aceeași distanță interioară;
- dacă sunt specificate 2 valori, prima va fi pentru distanțele interioare de sus și jos, iar a doua pentru distanțele interioare dreapta și stânga;
- dacă sunt specificate 3 valori, a doua valoare va fi aplicată și distanței interioare din stânga.

Proprietatea:

- `padding: "..."`

Valori posibile: `lungime` | `initial` | `inherit`

Exemplu:

```
p.clasa1 {  
    padding: 30px 100px;  
}  
...  
<p class="clasa1">Paragraf cu margini interioare.</p>
```

4.6.5. Dimensiunea elementelor

Proprietățile CSS permit controlul dimensiunii pentru un element HTML prin specificarea valorilor pentru lățime (proprietatea *width*) și înălțime (proprietatea *height*).

4.6.5.1. Lățime și înălțime

Valorile indicate pentru *width* și *height* se referă de fapt la dimensiunile zonei *Content* din Modelul Casetă. La aceste valori trebuie adăugate dimensiunile specificate în proprietățile *padding*, *border* și *margin* pentru a afla efectiv dimensiunile ocupate de un element. Implicit valorile pentru *width* și *height* sunt auto (calculate automat de către browser), însă se pot specifica valori exprimate în px, cm sau procente din blocul superior care conține acel element.

Proprietatea:

- `width: "..."`

Valori posibile: auto | lungime | % | initial | inherit

Exemplu:

```
p.clasa1 { width: 250px; }  
...  
<p class="clasa1">Exemplu de paragraf cu latimea de 250 pixeli.</p>
```

Proprietatea:

- `height: "..."`

Valori posibile: auto | lungime | % | initial | inherit

Exemplu:

```
img.clasa1 { height: 100px; }  
...  

```

4.6.5.2. Lățimi și înălțimi minime / maxime

Proprietățile *min-width*, *min-height*, *max-width*, *max-height* permit stabilirea unor dimensiuni minime, respectiv maxime, pentru un element.

Proprietatea:

- `min-width: "..."`
- `min-height: "..."`

Valori posibile: auto | lungime | % | initial | inherit

Exemplu:

```
p.clasa1 { min-width: 100px; }  
...  
<p class="clasa1">Exemplu de paragraf cu latimea minima de 100 pixeli.</p>
```

Proprietatea:

- `max-width: "..."`
- `max-height: "..."`

Valori posibile: auto | lungime | % | initial | inherit

Exemplu:

```
p.clasa1 { max-height: 200px; }
```

...

```
<p class="clasa1">Exemplu de paragraf cu inaltimea maxima de 200 pixeli.</p>
```

4.6.6. Amplasarea elementelor

Problemele care apar în modul de amplasare a diverselor elemente dintr-o pagină HTML vizează așezarea lor relativ la întreaga pagină sau la elementele înconjurătoare, distanțele față de margini, modul în care se suprapun și cum se aliniază textul și alte elemente față de aceste elemente.

4.6.6.1. Poziționare

Proprietatea specifică modul de poziționare a unui element:

- *static* - este poziționarea implicită a elementelor HTML. Acestea nu sunt amplasate în vreun mod special, ci conform fluxului normal al paginii.
- *relative* - amplasarea se face relativ la poziția normală a elementului. Proprietățile top, right, bottom, left vor ajusta distanța elementului față de respectivele margini normale.
- *fixed* - poziționarea este fixă, relativ la fereastra browserului. Chiar dacă se face scroll în pagină, acest element va rămâne în aceeași poziție.
- *absolute* - elementul cu această proprietate este poziționat relativ la cel mai apropiat strămoș (element de nivel superior în interiorul căruia se află), în loc să fie poziționat relativ la fereastra browserului (ca la proprietatea *fixed*). Dacă nu există strămoș, poziționarea absolută va fi relativă la corpul documentului, iar obiectul se va deplasa odată cu scroll-ul în pagină.

Proprietatea:

▪ position: "..."

Valori posibile: static | relative | fixed | absolute

Exemplu:

```
div.clasa1 { position: static; border: 2px solid green; }
```

```
div.clasa2 { position: fixed; bottom: 0; right: 0; width: 400px; border: 2px dashed rgb(0,0,255); }
```

```
div.clasa3 { position: relative; left: 30px; top: 150px; width: 500px; height: 300px; border: 2px solid red; }
```

```
div.clasa4 { position: absolute; top: 10px; right: 10px; width: 200px; height: 100px; border: 2px dotted #AABBCC; }
```

...

```
<div class="clasa1">Bloc cu proprietatea position: static</div>
```

```
<div class="clasa2"> Bloc cu proprietatea position: fixed</div>
```

```
<div class="clasa3"> Bloc cu proprietatea position: relative
```

```
<div class="clasa4"> Bloc cu proprietatea position: absolute</div>
```

```
</div>
```

4.6.6.2. Suprapunerea

Elementele se pot suprapune sub formă de stivă, astfel încât se poate controla ordinea în care ele sunt așezate unul peste altul. Proprietatea z-index specifică ordinea elementelor: un element cu o valoare mai mare pentru z-index va fi deasupra celor cu valoarea z-index mai mică. Valoarea z-index poate fi pozitivă sau negativă. Dacă două elemente se suprapun și nu au valoarea z-index specificată, ultimul element inserat în codul HTML va fi afișat deasupra.

Proprietatea:

▪ z-index: "..."

Valori posibile: auto | number

Exemplu:

```
img { position: absolute; left: 0px; top: 0px; z-index: -1; }
```

...

```

```

```
<p>Textul din paragraf va aparea deasupra imaginii care are z-index negativ.</p>
```

4.6.6.3. Distanțe față de margini

Pentru o poziționare absolută se pot specifica distanțele față de cele 4 margini (top, right, bottom, left) ale elementului ierarhic superior. Sunt permise valori exprimate în unități de lungime (px, cm) sau procente; valorile negative sunt de asemenea permise.

Proprietatea:

- top: "..."
- right: "..."
- bottom: "..."
- left: "..."

Valori posibile: auto | dimensiune | % | initial | inherit

Exemplu:

```
div { width: 400px; height: 300px; border: 3px solid blue; }
```

```
div.clas1 { position: absolute; top: 70px; left: 10%; width: 200px; height: 200px; border: 2px solid red; }
```

...

```
<div>Bloc de 400x300 pixeli cu pozitie implicita.</div>
```

```
<div class="clas1">Bloc cu pozitie absoluta la 70 pixeli fata de marginea de sus a paginii si 10% fata de marginea din stanga.</div>
```

4.6.6.4. Decupare

Proprietatea permite decuparea unui dreptunghi dintr-un element poziționat absolut, în cazul în care elementul este mai mare decât elementul ierarhic superior (elementul care îl conține).

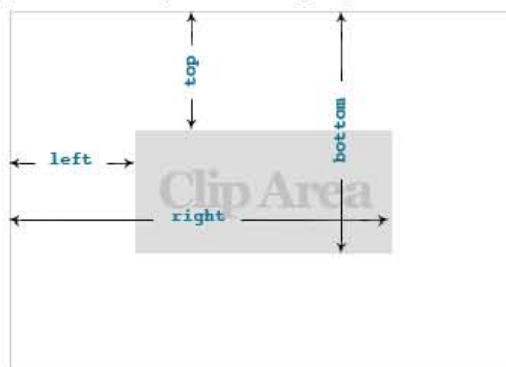
Proprietatea:

- clip: "..."

Valori posibile: shape | auto | initial | inherit

Observații:

- Valoarea *auto* înseamnă că nu se decupează (valoarea implicită);
- *shape* poate fi doar *rect* (*top*, *right*, *bottom*, *left*), în care se specifică un dreptunghi cu coordonatele date relativ la colțul din stânga-sus al elementului ce va fi decupat;
- proprietatea *clip* nu va funcționa dacă proprietatea *overflow* este *visible*.



Exemplu:

```
img.clas1 { position: absolute; clip: rect(0px, 70px, 80px, 20px); }
```

...

```

```

```
<p>
```

4.6.6.5. Depășire

Proprietatea arată ce se întâmplă când conținutul depășește chenarul elementului: restul conținutului este afișat, ascuns sau se aplică alte opțiuni. Se pot specifica aceste opțiuni doar pe o axă (*overflow-x*, *overflow-y*) sau pentru ambele axe (*overflow*).

Proprietatea:

- *overflow*: "..."
- *overflow-x*: "..."
- *overflow-y*: "..."

Valori posibile: *visible* | *hidden* | *scroll* | *auto* | *initial* | *inherit*

Exemplu:

```
div.clasa1 { width: 80px; height: 100px; background-color: yellow; }
```

```
div.clasa2 { width: 80px; height: 100px; overflow: scroll; background-color: yellow; }
```

```
div.clasa3 { width: 80px; height: 100px; overflow: hidden; background-color: yellow; }
```

...

```
<div class="clasa1">Bloc care poate fi controlat prin proprietatea overflow:visible (valoare implicită).</div>
```

```
<p><div class="clasa2">Bloc care poate fi controlat prin proprietatea overflow:scroll.</div>
```

```
<p><div class="clasa3">Bloc care poate fi controlat prin proprietatea overflow:hidden.</div>
```

4.6.6.6. Plutire

Proprietatea specifică dacă un element poate "pluti" (poate fi amplasat în stânga sau în dreapta paginii astfel încât textul să "curgă" în partea liberă). Elementele cu poziționare absolută ignoră proprietatea *float*.

Proprietatea:

- *float*: "..."

Valori posibile: *left* | *right* | *none* | *initial* | *inherit*

Exemplu:

```
img.clasa1 { float: right; }
```

...

```

```

```
<p>Paragraf 1.</p>
```

```

```

```
<p>Paragraf 2.</p>
```

4.6.6.7. Golire

Proprietatea specifică partea unui element pe care nu se pot amplasa alte elemente "plutitoare" (*float*).

Proprietatea:

- *clear*: "..."

Valori posibile: *left* | *right* | *none* | *both* | *initial* | *inherit*

Exemplu:

```
img.clasa1 { float: right; }
```

```
p.clasa2 { clear: both; }
```

...

```

```

```
<p>Paragraf 1.</p>
```

```
<p class="clasa2">Paragraf 2.</p>
```

4.6.7. Afișarea și vizibilitatea elementelor

Cele două proprietăți, *display* și *visibility*, au un rol important în CSS deoarece permit afișarea sau ascunderea unui anumit element.

4.6.7.1. Afișarea elementelor

Proprietatea *display* determină dacă un element este sau nu afișat. În funcție de tipul lor, elementele HTML au un tip implicit pentru modul de afișare:

- *block* (încep întotdeauna pe o linie nouă și ocupă lățimea maximă disponibilă): `<div>`, `<h1>...<h6>`, `<p>`, `<form>`, `<header>`, `<footer>`, `<section>`;
- *inline* (nu încep pe linie nouă și ocupă doar spațiul de care au nevoie) - ``, ``, `<a>`.

Proprietatea:

▪ `display: "..."`

Valori posibile: `none` | `inline` | `block` | `inline-block` | `flex` | `inline-flex` | `list-item` | `run-in` | `compact` | `marker` | `table` | `inline-table` | `table-row-group` | `table-header-group` | `table-footer-group` | `table-row` | `table-column-group` | `table-column` | `table-cell` | `table-caption` | `initial` | `inherit`

Observație: *display:none* este folosit în JavaScript pentru a ascunde și afișa elemente fără a le șterge și recrea ulterior.

Exemplu:

```
p { display: inline; }
```

...

```
<p>Paragraf 1.</p>
```

```
<p>Paragraf 2.</p>
```

4.6.7.2. Vizibilitatea elementelor

Proprietatea specifică dacă un element este sau nu vizibil. Elementele invizibile ocupă totuși un spațiu (gol) în pagină; pentru a crea elemente invizibile care nu ocupă spațiu se folosește proprietatea *display*.

Proprietatea:

▪ `visibility: "..."`

Valori posibile: `visible` | `hidden` | `collapse` | `initial` | `inherit`

Observație: *collapse* se folosește pentru un tabel: elimină o linie sau coloană fără a afecta macheta întregului tabel. Spațiul ocupat de elementele eliminate vor fi disponibile pentru alt conținut. Aplicată altui tip de element, valoarea *collapse* are efectul valorii *hidden* (elementul va fi ascuns, însă va ocupa spațiu).

Exemplu:

```
p.clasa1 { visibility: hidden; }
```

```
p.clasa2 { display: none; }
```

...

```
<p>Paragraf vizibil</p>
```

```
<p class="clasa1">Paragraf ascuns.</p>
```

```
<p class="clasa2">Paragraf neafisat.</p>
```

```
<p>Paragraf vizibil</p>
```

4.7. Tabele

Pentru formatarea tabelelor cu CSS (fie întregul tabel - marcajul `<table>`, fie la nivelul elementelor componente - `<th>`, `<tr>`, `<td>`) se folosesc atât proprietăți generale, prezentate în capitolele anterioare, cât și proprietăți specifice acestor tipuri de elemente.

Proprietăți generale pentru tabele:

- *border*: specificarea proprietăților pentru chenare;
- *width, height*: specificarea proprietăților pentru lățime, respectiv înălțime;
- *text-align*: specificarea proprietăților pentru alinierea conținutului pe orizontală;
- *vertical-align*: specificarea proprietăților pentru alinierea conținutului pe verticală;
- *padding*: specificarea proprietăților pentru distanțele interioare între conținut și marginile celulelor (padding);
- *color, background-color*: specificarea proprietăților pentru culoarea, respectiv fundalul elementelor.

Proprietăți specifice tabelelor:

- *border-collapse*: restrângerea chenarelor într-unul singur (valori posibile: collapse | separate);
- *border-spacing*: specifică distanța dintre chenarele celulelor adiacente (valori exprimate în unități de lungime - px, cm; valorile trebuie să fie pozitive);
- *caption-side*: specifică amplasarea legendei (*caption*) pentru tabel (valori posibile: top | bottom);
- *empty-cells*: specifică dacă se afișează chenarul și culoarea de fundal pentru celulele goale (valori posibile: show | hide);
- *table-layout*: specifică tipul de algoritm folosit pentru generarea tabelului (valori posibile: auto | fixed). Valoarea *auto* (implicită) generează tabelul în funcție de cel mai lung conținut care nu poate fi întrerupt, în timp ce valoarea *fixed* determină un tabel de dimensiunile specificate.

Exemplu:

```
<!DOCTYPE html>
<html>
<head>
<style>
table { border-collapse: separate; width: 100%; border: 1px solid black; border-spacing: 10px 10px;
}
td, th { border: 2px solid black; }
table.clasa1 { table-layout: fixed; width: 500px; border-collapse: collapse; caption-side: bottom; }
td.clasa1 { border: 2px solid red; text-align: center; padding: 10px; }
th { background-color: lightgreen; color: blue; height: 50px; vertical-align: bottom; }
</style>
</head>
<body>
<table>
<caption>Tabel 1</caption>
<tr>
<th>Antet 1</th>
<th>Antet 2</th>
</tr>
<tr>
<td width="20%">Celula 1</td>
<td class="clasa1">Celula 2</td>
</tr>
</table>
<p>
<table class="clasa1">
<caption>Tabel 2</caption>
<tr>
<td width="20%" class="clasa1">Celula 3</td>
```



```
<td>Celula 4</td>
</tr>
</table>
</body>
</html>
```

4.8. CSS3

Este cel mai recent standard publicat, cu proprietăți deja implementate în ultimele versiuni ale browserelor populare. CSS 3 este compatibil cu versiunile 1 și 2, dar aduce și noi proprietăți.

Fonturi

CSS3 permite un mecanism (*@font-face*) pentru utilizarea unor fonturi care nu sunt instalate în calculatorul utilizatorului. Formatele de fonturi disponibile pentru textul paginilor Web sunt:

- TTF (*TrueType Fonts*) - cel mai cunoscut format de fonturi, instalate în sisteme de operare Microsoft Windows și Apple Mac OS);
- OTF (*OpenType Fonts*) - format scalabil de font, introdus de Microsoft și bazat pe TTF; este folosit actual pe majoritatea platformelor.
- WOFF (*The Web Open Font Format*) - format dezvoltat în 2009 pentru pagini Web și recomandat de W3C; este bazat pe TTF și OTF, având în plus comprimarea informațiilor și metadate suplimentare.
- WOFF 2.0 (*The Web Open Font Format*) - format care aduce o rată de comprimare superioară față de WOFF, ideal pentru clienții cu constrângeri de trafic sau rate de transfer;
- SVG (fonturi și forme *Scalable Vector Graphics*) - specificațiile permit crearea fonturilor în cadrul documentelor SVG cărora li se pot aplica stiluri CSS;
- EOT (*Embedded OpenType Fonts*) - format compact al OTF folosit pentru înglobare în paginile Web.

Pentru a utiliza un font disponibil la o anumită adresă Web trebuie urmați pașii:

- se definește numele fontului (în proprietatea font-family);
- se indică sursa acestui font (adresa URL).

Exemplu:

```
<style>
@font-face { font-family: NumeFamilieFont; src: url("NumeFont.woff2"); }
div { font-family: NumeFamilieFont; }
</style>
...
<div>Text scris cu fontul NumeFamilieFont.</div>
```

@font-face este o tehnică de accesare a fonturilor stocate pe propriile servere. Pentru a utiliza fonturi stocate online pe alte servere este folosită regula *@import* prin care se specifică adresa fontului care se dorește a fi folosit (de exemplu, fonturi de la Google Fonts). Fonturile online se pot accesa și cu marcajul *<link>* prin care se creează o legătură externă spre un fișier de fonturi.

Exemplu:

```
<!DOCTYPE html>
<html>
<head>
<link href='https://fonts.googleapis.com/css?family=Tangerine' rel='stylesheet' type='text/css'>
<style>
@import url(https://fonts.googleapis.com/css?family=Oswald);
```

```
p.clasa1 { font-family: 'Oswald', sans-serif; }
p.clasa2 { font-family: 'Tangerine', sans-serif; font-size: 40px; }
</style>
</head>
<body>
<p class="clasa1">Textul paragrafului 1.</p>
<p class="clasa2">Textul paragrafului 2.</p>
</body>
</html>
```

Colțuri rotunde

Proprietatea *border-radius* creează pentru modelul casetă colțuri rotunjite.

Exemplu:

```
#colturirotunde { border-radius: 30px; border: 2px solid red; background: blue; padding: 10px; width:
300px; height: 200px; color: yellow; }
...
<p id="colturirotunde">Paragraf cu colturi rotunde</p>
```

Imagini în chenar

Proprietatea *border-image* setează o imagine pe chenarul care înconjoară un element.

Exemplu:

```
#id1 { border: 10px solid transparent; padding: 25px; border-image: url(imagine1.jpg) 30 round; }
...
<p id="id1">Paragraf.</p>
```

Noi proprietăți pentru fundal

Pentru a controla mai bine proprietățile imaginii de fundal au fost adăugate proprietățile *background-size*, *background-origin* și *background-clip*.

Exemplu:

```
#fundal { border: 1px solid black; background:url(imagine1.jpg); background-size: 100px 80px;
background-repeat: no-repeat; padding:15px; background-origin: border-box; background-clip:
content-box; }
...
<div id="fundal">Text</div>
```

Culori în noi formate

Suplimentar modului de definire a culorilor din CSS2, în CSS3 este introdus suport pentru noile formate HSL (*Hue, Saturation, Lightness*), RGBA, HSLA (*A - alpha channel*, opacitatea unei culori).

Exemplu:

```
#id1 { background-color:rgba(255, 0, 0, 0.4); }
#id2 { background-color:hsl(120, 100%, 50%); }
#id3 { background-color:rgb(0, 0, 255); opacity:0.6; }
...
<p id="id1">Rosu cu opacitate</p>
<p id="id2">Verde</p>
<p id="id3">Albastru cu opacitate</p>
```

Gradiente

Pentru a crea treceri fine între două culori au fost create proprietățile *linear-gradient*, *radial-gradient* care generează gradiente în format liniar sau radial.

Exemplu:

```
#gradient1 { height: 200px; background: linear-gradient(to bottom left, red , blue); }
```

Efecte de umbrire

Textului sau casetelor li se pot adăuga acum efecte vizuale sub forma unor umbriri (proprietățile *text-shadow*, *box-shadow*).

Exemplu:

```
p { text-shadow: 2px 2px yellow; }
div { box-shadow: 5px 5px yellow; width: 200px; height: 200px; border: solid; }
...
<div>
<p>Text cu umbre</p>
</div>
```

Întrerupere text

Textul care nu încapă în casetă poate fi afișat întrerupt sau cu puncte de suspensie. Proprietățile care permit aceste facilități sunt *text-overflow*, *word-wrap*, *word-break*.

Exemplu:

```
p { white-space: nowrap; width: 200px; border: 1px solid #000000; overflow: hidden; }
p.clasa1 { text-overflow: ellipsis; }
...
<p>Textul din paragraf nu incapă in caseta.</p>
<p class="clasa1">Textul din paragraf nu incapă in caseta.</p>
```

Suport pentru transformări 2D și 3D

Elementele pot suferi diverse efecte - deplasare, rotire, scalare și deformare - proprietatea *transform* și funcțiile *translate()*, *rotate()*, *scale()*, *skewX()*, *skewY()*, *matrix()*, *rotateX()*, *rotateY()*, *rotateZ()*.

Exemplu:

```
div { width: 300px; height: 100px; background-color: yellow; border: 1px solid black; }
div.clasa1 { transform: translate(150px, 130px); transform: scale(0.75,1.25); }
div.clasa2 { transform: rotate(20deg); }
div.clasa3 { transform: skew(20deg, -10deg); }
...
<div>Text initial</div>
<div class="clasa1">Text din blocul 1</div>
<div class="clasa2">Text din blocul 2</div>
<div class="clasa3">Text din blocul 3</div>
```

Tranziții

Prin proprietatea *transition* se adaugă efecte de modificare dinamică a unor proprietăți (de exemplu, dimensiunile) într-un timp dat.

Exemplu:

```
div { width: 100px; height: 50px; background: blue; transition: width 2s, height 4s; }
div:hover { width: 250px; height: 250px; }
...
<div></div>
```

Animații

Constau în modificarea proprietăților și deplasarea elementelor fără a folosi alte tehnologii precum JavaScript sau Flash. Se folosesc proprietățile *@keyframes* și *animation* având ca parametri numeroase alte proprietăți care suportă utilizarea în animații și tranziții și care conțin valori numerice, procente sau culori.

Exemplu:

```
div { width: 250px; height: 250px; background-color: blue; animation-name: animatie; animation-
duration: 3s; animation-delay: 2s; animation-iteration-count: infinite; animation-direction: alternate;
}
@keyframes animatie { from { background-color: blue; } to { background-color: red; } }
...
<div></div>
```

Coloane tip publicație

Împărțirea automată a textului sub formă de multiple coloane de publicație se face folosind proprietatea *columns*.

Exemplu:

```
div.clasa1 { column-count: 3; column-gap: 40px; column-rule: 2px solid lightblue; }
...
<div class="clasa1">
```

Paragraf cu text foarte lung. Paragraf cu text foarte lung. Paragraf cu text foarte lung. Paragraf cu text foarte lung. Paragraf cu text foarte lung. Paragraf cu text foarte lung. Paragraf cu text foarte lung. Paragraf cu text foarte lung....</div>

Redimensionarea interactivă a casetelor elementelor

Proprietatea *resize* permite redimensionarea de către utilizator a casetelor elementelor.

Exemplu:

```
div { border: 3px dashed; padding: 20px; width: 200px; resize: both; overflow: auto; }
...
<div>Casetă redimensionabilă. Textul se va aseza în funcție de dimensiunile casetei.</div>
```

Alte proprietăți

Alte proprietăți nou introduse de CSS3 permit:

- un nou concept de machetă cu casete flexibile (*flexbox*);
- îmbunătățirea conceptului de stiluri particularizate în funcție de *tipul* dispozitivului (*media type*, concept existent în CSS2). Regula *@media* în CSS3 interoghează echipamentul (*media query*) și extrage informații referitoare la *capabilitățile* acestuia (dimensiunile dispozitivului/ecranului, orientarea - portrait/landscape, rezoluția).

4.9. CSS Responsive

În momentul de față diversitatea dispozitivelor care permit accesarea paginilor Web face ca un nou concept să devină esențial în proiectarea machetelor paginilor: *Responsive Web Design*

(RWD). Indiferent de dispozitivul folosit (desktop/laptop, tabletă sau smartphone), este important ca aspectul unei pagini Web să fie la fel de atractiv vizual și ușor de accesat, iar conținutul său să se adapteze la orice display, browser și tip de dispozitiv.

Câteva elemente de care trebuie să se țină cont la proiectarea unui site *responsive* sunt:

- mărimea zonei vizibile din pagină (așa-numitul *viewport*): folosind marcajul meta de mai jos se oferă browserului instrucțiuni despre cum să gestioneze dimensiunile paginii și scalarea:
<meta name="viewport" content="width=device-width, initial-scale=1.0">
- împărțirea paginii într-o grilă formată standard din 12 coloane / lățime 100% (care se va redimensiona în funcție de fereastra browserului) și plasarea elementelor funcție de aceste coloane;
- utilizarea media queries (regula @media) astfel încât să fie aplicate stiluri diferite pentru ecrane diferite (mari sau mici) și orientări portrait/landscape;
- pentru imagini/video:
 - o lățime de 100% va scala imaginea/video funcție de ecran;
 - utilizarea unor imagini diferite funcție de dispozitivul folosit;
 - marcajul <picture> din HTML5 încarcă imaginea funcție de lățimea ecranului:

Exemplu:

```
<picture>
<!-- Pentru desktop -->
<source media="(min-width: 64em)" src="image-mare.jpg">
<!-- Pentru tableta -->
<source media="(min-width: 37.5em)" src="image-medie.jpg">
<!-- Pentru smartphone -->
<source src="image-mica.jpg">
<!-- Pentru browsere incompatibile -->

<p>Text pentru browserele incompatibile.</p>
</picture>
```