

# Tehnici Web

## CURSUL 2

Semestrul II, 2022-2023  
Carmen Chiriță

<https://sites.google.com/site/fmitehniciweb/>

# HTML- METADATA

<head>

<title>Titlu </title>

<meta charset="utf-8">

<meta name="autor" content="nume-autor">

<meta name="description" content="descriere">

<meta name="keywords" content="cuvinte cheie">

<base href="base URL for relative paths" >

<style type="text/css"> .... </style>

<script type="text/javascript"> ...</script>

<script type="text/javascript" src="fis.js"></script>

<link href="fis.tip" type="image/tip" rel="shortcut icon">

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

<head>

# Elementul **base**

Defineste un URL de baza pentru toate URL-urile relative din document;

Poate exista un singur element **base** în document

```
<head>
<title>Pagina mea</title>
<base href="/home/carmen/TEHNICI_WEB_CURSURI/EXEMPLE/"
target="_blank">
</head>
<body>
<a href="exemplul1.html" >Exemplul 1</a>
<a href="exemplul2.html" >Exemplul 2</a>

</body>
```

# CSS = CASCADING STYLE SHEETS

- Folosit la definirea aspectului paginii:
  - Stilizarea elementelor: culoare, marime, font, etc.
  - Așezarea elementelor în pagina
  - Vizibilitate
- De asemenea, este folosit în animatie

# Legătura cu fișierul de stil

```
<head>
```

```
...
```

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

```
...
```

```
</head>
```

Fișierul de stil conține una sau mai multe reguli CSS

### Regula CSS

selector

```
{ prop1:valoare1;  
  prop2:valoare2;  
  prop3:valoare3;  
  ...  
}
```

### Exemplu

selector

```
p {  
    color: red;  
    font-family: sans-serif;  
    font-size: 14pt;  
}
```

proprietate

valoare

## CSS se poate introduce în trei moduri

<b>Inline</b>	<pre>&lt;section style="width:80% ; background-color:white;"&gt; continut-sectiune &lt;/section&gt;</pre>
<b>In file</b>	<pre>&lt;head&gt; &lt;style type="text/css"&gt; section {width:80%; background-color:white;} &lt;/style&gt; &lt;/head&gt;</pre>
<b>In fisier separat</b>  <b>varianta recomandata</b>	<pre>&lt;head&gt; &lt;link href="filename.css" type="text/css" rel="stylesheet"&gt; &lt;/head&gt;</pre>

Acceasi pagina poate fi stilizata diferit prin modificarea fisierului de stil.

```
<head>
<link href="style1.css" type="text/css" rel="stylesheet" >

<!-- <link href="style2.css" type="text/css" rel="stylesheet" > -->
</head>
<body>
<header> PRIMA PAGINA

<nav>
  <a href="#">ACASA</a>
  <a href="#">CONTACT</a>
</nav>
</header>
<div id="container">

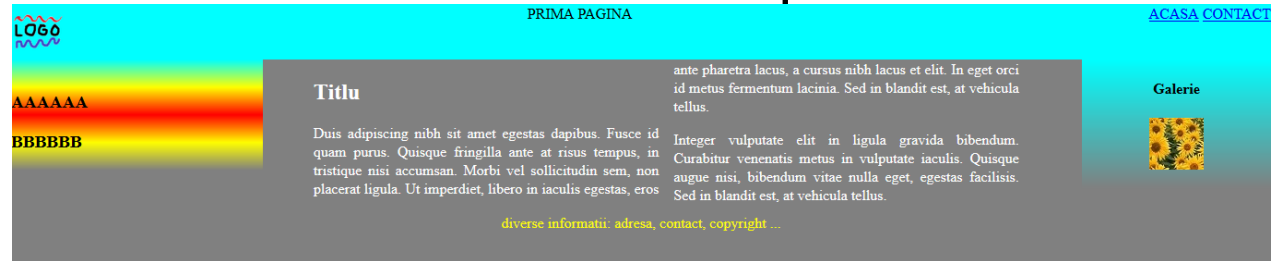
<section id="baralat">
<p>AAAAAA</p>
<p>BBBBBB</p>
</section>

<section id="main">
<h2> Titlu</h2>
<p>Duis adipiscing ...
</p>
<p>Integer vulputate elit ...</p>
</section>
</div>
<aside>
<p>
<h4> Galerie </h4>

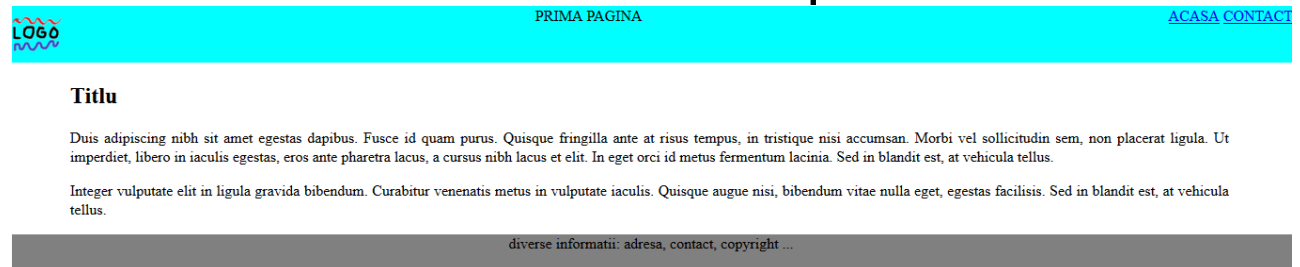
</p>
</aside>

</body>
```

style1.css



style2.css





# Precedenta stilurilor

precedenta  
cea mai  
mica



precedenta  
cea mai  
MARE

Stilul implicit al browser-ului

Fisierul css extern (<link> in header)  
si stiluri "in file" (<style> in header)  
(se aplica stilul care apare scris ultimul în <head>)

Stiluri inline

# Selectorii CSS

<http://www.w3.org/TR/css3-selectors/>

```
selector1, selector2 { proprietate1:valoare1; }
```

```
selector1 { proprietate2:valoare2; }
```

Aceluiasi element i se pot aplica mai multe reguli;  
este bine sa se grupeze elementele care au un stil comun, iar stilurile  
specifice sa fie scrise in reguli separate.

Unele stiluri sunt mostenite de la elementele parinte (in arborele asociat  
documentului), dar pot fi schimbate folosind reguli specifice.

# Proprietăți CSS

**color** : pentru culoarea textului

**background-color**: pentru culoarea fondului

Culorile se pot specifica prin:

- Numele culorii: **red, green, blue, ...**

<https://codepo8.github.io/10kb-CSS-colour-game/>

# Culori CSS

- Valoarea `rgb(red,green, blue)`  
`red, green, blue` au valori între 0-255
- Valoarea hexazecimală `#rrggbb`  
`rr,gg,bb` au valori între 00-ff

`rgb(255, 0, 0)`

`rgb(0, 0, 255)`

`rgb(60, 179, 113)`

`rgb(238, 130, 238)`

`#ff0000`

`#0000ff`

`#3cb371`

`#ee82ee`

`#ffa500`

`#6a5acd`

# Culori CSS (noua în CSS3)

- Valoarea `hsl(hue, saturation, lightness)`,  
`hue` reprezintă gradul culorii (intre 0-360)  
`saturation` reprezintă intensitatea (intre 0%-100%)  
`lightness` reprezintă luminozitatea (intre 0%-100%)

`hsl(120,100%,50%)`

`hsl(120,100%,75%)`

`hsl(120,100%,25%)`

`hsl(120,60%,70%)`

`hsl(290,100%,50%)`

`hsl(290,60%,70%)`

# Culori CSS (noi în CSS3)

- Valoarea `rgba(red,green,blue, alpha)`, este o extensie a valorii `rgb` cu valoarea `alpha` care reprezinta opacitatea (intre 0.0-1.0)
- Valoarea `hsla(hue, saturation, lightness, alpha)` este o extensie a valorii `hsl` cu valoarea `alpha`

**`rgba(255, 99, 71, 0)`**

**`rgba(255, 99, 71, 0.2)`**

**`rgba(255, 99, 71, 0.4)`**

**`rgba(255, 99, 71, 0.6)`**

**`rgba(255, 99, 71, 0.8)`**

**`rgba(255, 99, 71, 1)`**

**`hsla(9, 100%, 64%, 0)`**

**`hsla(9, 100%, 64%, 0.2)`**

**`hsla(9, 100%, 64%, 0.4)`**

**`hsla(9, 100%, 64%, 0.6)`**

**`hsla(9, 100%, 64%, 0.8)`**

**`hsla(9, 100%, 64%, 1)`**

## Distanța/marimea în CSS

pot fi măsurate absolut sau relativ.

Unități absolute: **mm**, **cm**, **px**, **in**, **pt**

Unități relative: **em** = dimensiunea fontului părinte

**%** = procente din dimensiunea  
părintelui / fontului curent

**vw**, **vh** = 1% din lățimea/înălțimea  
viewportului

**rem** = fontul elementului rădăcină (html)

```
<section>
<h2>Titlul secțiunii</h2>
<p>Un paragraf</p>
</section>
```

```
<style>
section {width:80vw;
        }
h2 { font-size: 30px;
    }
p {width:50%;
  }
</style>
```

# Selectori CSS

- Selectorul universal \*

Exemplu: `* {border: 1px solid blue;}`

**Titlu cu h1**

Paragraful1 in div

Paragraful2 in div

Paragraful3 in div

Paragraful4

- Selectorul de tip (type selector) :

este numele unui element din documentul HTML: **h1, p, div,...**

Exemplu:

```
p {color: green;
margin-left: 25px;
}
```

HTML

```
<body>
<h1>Titlu cu h1</h1>
<div>
<p>Paragraful1 in div</p>
<p>Paragraful2 in div</p>
<p>Paragraful3 in div</p>
<div>
<p>Paragraful4 </p>
</body>
```

**Titlu cu h1**

Paragraful1 in div

Paragraful2 in div

Paragraful3 in div

Paragraful4



# Selectori CSS

- Selectorul prin **id**:  
**#myid** selecteaza elementul cu **id="myid"**

## Element HTML

`<tag id="myid"> continut </tag>`

(un document HTML poate contine un singur atribut **id** cu valoarea "myid")

## Exemplu

### HTML

`<div id="rosu"> continut </div>`

### CSS

`#rosu {color: red;}`

sau

`div#rosu {color: red;}`

# Selectorii CSS

- Selectorul prin clasa (**class selector**):  
.newcl selecteaza elementele cu class="newcl"

## Elemente HTML

```
<tag1 class="newcl"> continut </tag1>  
<tag2 class="newcl"> continut </tag2>
```

cu atributul **class** grupam mai multe elemente, care vor avea un stil comun;

poate aparea de mai multe ori in acelasi document HTML.

## Exemplu

### HTML

```
<ul>  
<li class="par"> continut </li>  
<li> continut </li>  
<li class="par"> continut </li>  
<li> continut </li>  
<li class="par"> continut </li>  
</ul>
```

### CSS

```
.par {color: green;}  
sau  
li.par {color: green;}
```

# Exemple

## HTML

```
.....  
<div class="a  
b">div</div>  
<p class="a b">p1</p>  
<p class="a">p2</p>  
<p>p3</p>  
<p class="b">p4</p>  
<p>p5</p>  
<p class="a b"> p6</p>  
.....
```

## CSS

```
p.a.b {  
    font-size:3em;  
}  
p.a, p.b {  
    border: 1px solid  
black;  
}  
p.a:not(.b){  
    color: red;  
}  
p.b:not(.a){  
    color: blue;  
}
```

div

p1

p2

p3

p4

p5

p6

Selectarea prin **id** are specificitate mai mare decat selectarea prin **class**

```
#id {color:green;}  
.class {color: blue;}
```

Exemplu

```
<p class="class" id="id"> Exemplu </p>
```

Cand doua stiluri au aceeasi specificitate, castiga cel care este scris ultimul in fisierul CSS.

```
.class1 {color: blue;}  
.class2 {color: green;}
```

Exemplu

```
<p class="class2 class1">Exemplu</p>
```

## Observatii:

Proprietatile marcate **!important** sunt aplicate indiferent de specificitatea lor

```
p { color: #ff0000 !important; }
```

Unele proprietati pot lua valoarea **inherit**, aceasta insemnand ca proprietatea este mostenita de la elementul parinte; este folosita pentru a suprascrie un alt stil si a forta comportamentul implicit.

```
<p>Paragraf în afara divului</p>
```

```
<div>
```

```
<p>Paragraf în div</p>
```

```
</div>
```

```
div{color:red;}
```

```
p{color:blue;}
```

```
div>p{color:inherit;}
```

# Combinarea selectorilor

Selectori pentru descendenți

$E \ F$  selectează toate elementele  $F$  care sunt descendenți ai lui  $E$

Selectori pentru copii

$E > F$  selectează elementele  $F$  care sunt copii ai lui  $E$

Selectori pentru frați

$E + F$  selectează toate elementele  $F$  care sunt frați cu  $E$  și care urmează imediat după  $E$

$E \sim F$  selectează toate elementele  $F$  care sunt frați cu  $E$  și care urmează după  $E$

## Exemple

```
div p {  
  background-color: yellow;  
}
```

## HTML

```
<body>  
<div>  
  <p>Paragraph 1 in the div.</p>  
  <p>Paragraph 2 in the div.</p>  
  <aside><p>Paragraph 3 in the div.</p></aside>  
</div>  
<p>Paragraph 4. Not in a div.</p>  
<p>Paragraph 5. Not in a div.</p>  
</body>
```

```
div > p {  
  background-color: yellow;  
}
```

## Descendent

Paragraph 1 in the div.

Paragraph 2 in the div.

Paragraph 3 in the div.

Paragraph 4. Not in a div.

Paragraph 5. Not in a div.

## Copil

Paragraph 1 in the div.

Paragraph 2 in the div.

Paragraph 3 in the div.

Paragraph 4. Not in a div.

Paragraph 5. Not in a div.

## Adjacent sibling

---

```
<body>
<div>
  <p>Paragraph 1 in the div.</p>
  <p>Paragraph 2 in the div.</p>
</div>

<p>Paragraph 3. Not in a div.</p>
<p>Paragraph 4. Not in a div.</p>

</body>
```

```
div + p {
  background-color: yellow;
}
```

Paragraph 1 in the div.

Paragraph 2 in the div.

Paragraph 3. Not in a div.

Paragraph 4. Not in a div.

## General sibling

```
<body>

<p>Paragraph 1.</p>

<div>
  <code>Some code.</code>
  <p>Paragraph 2.</p>
</div>

<p>Paragraph 3.</p>
<code>Some code.</code>
<p>Paragraph 4.</p>

</body>
```

```
div ~ p {
  background-color: yellow;
}
```

Paragraph 1.

Some code.

Paragraph 2.

Paragraph 3.

Some code.

Paragraph 4.



# Selectorii CSS

## Selectorii de atribut

`e[attr]` selectează elem. `e` care au `attr` ca atribut

`e[attr="val"]` selectează elem. `e` cu `attr="val"`

`e[attr~="val"]` selectează elem. `e` cu proprietatea ca valoarea lui `attr` conține cuvântul `val`

`e[attr*="val"]` selectează elem. `e` cu proprietatea ca valoarea lui `attr` conține subsirul `val`

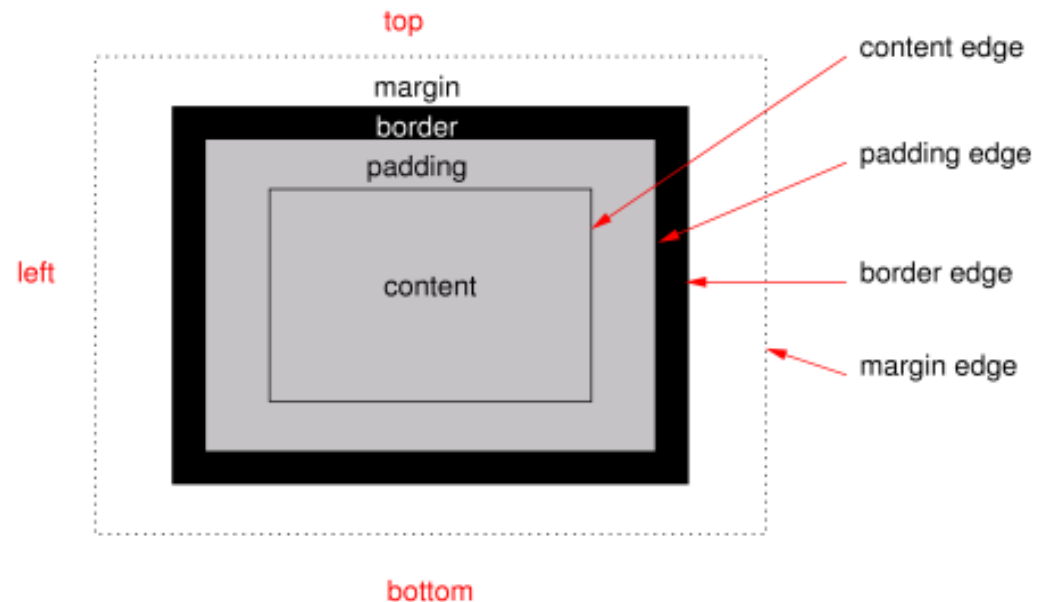
Exemplu: `a[target="_blank"]`

# “Box Model” pentru elemente HTML

<http://www.w3.org/TR/css3-box/>

- este o modalitate de a reprezenta spațiul pe care un element îl ocupă pe o pagină și modul în care acesta se raportează la alte elemente.

- Width/Height
- Padding
- Border
- Margin



[https://www.w3schools.com/cssref/css\\_default\\_values.asp](https://www.w3schools.com/cssref/css_default_values.asp)

# Proprietati CSS: border

`border: grosime stil culoare;`  
`border-left: grosime stil culoare;`  
`border-right: grosime stil culoare;`  
`border-top: grosime stil culoare;`  
`border-bottom: grosime stil culoare;`

`border-color:`  
`border-style:`  
`border-width:`

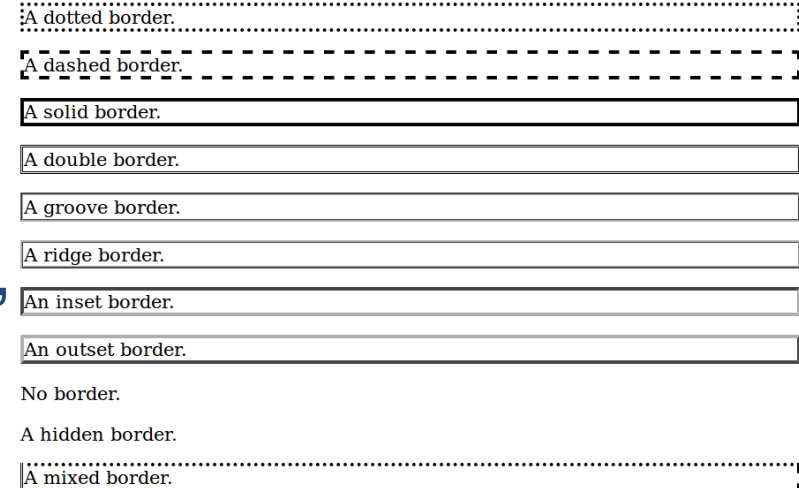
`border-left-color:`  
`border bottom-width:`  
...

grosime: exprimata in `px`, `pt`, `em`, sau  
`thin`, `medium`, `thick`

stil: **`none`**, `solid`, `dotted`, `dashed`, `inset`,  
`outset`, `double`, `groove`, `ridge`,  
**`hidden`**

`border-radius: 12px;`

margini cu colturi rotunde



Proprietatea `border-color` nu funcționează singura, trebuie obligatoriu și `border-style`

## CSS

```
<style>  
body { border: thick ridge red;  
        width:30%;  
      }  
</style>
```

# **Doua exemple**

[Sectiuni cu articole](#)

[Articol cu sectiuni](#)

# Proprietati CSS: margin, padding

```
padding: 15px;  
padding-left: 10%;  
padding-right: 20%;  
padding-top: 15px;  
padding-bottom: 15px;
```

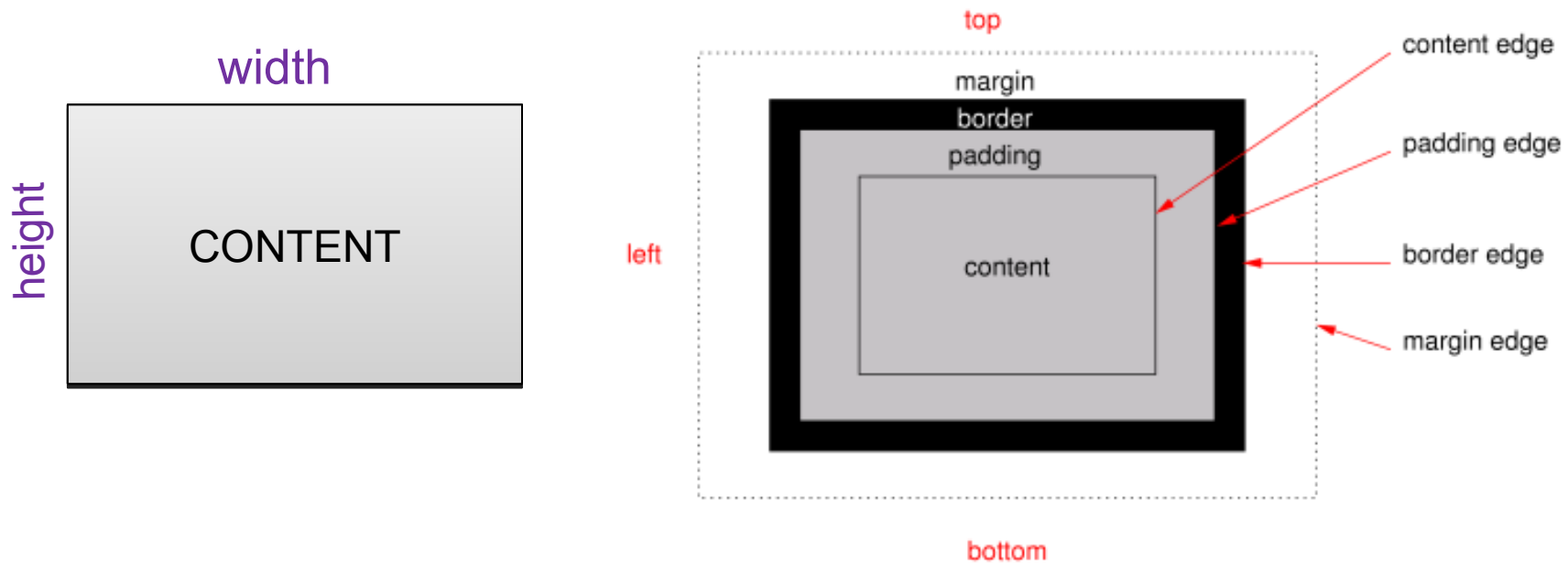
```
margin: 15px;  
margin-left: 10%;  
margin-right: 20%;  
margin-top: 15px;  
margin-bottom: 15px;
```

padding și margin au background transparent  
% din latimea elementului

```
margin: 15px 20% 15px 10%;  
padding: 15px 20% 15px 10%;
```

↓   ↓   ↓   ↓  
top right bottom left

# Proprietati CSS: width, height



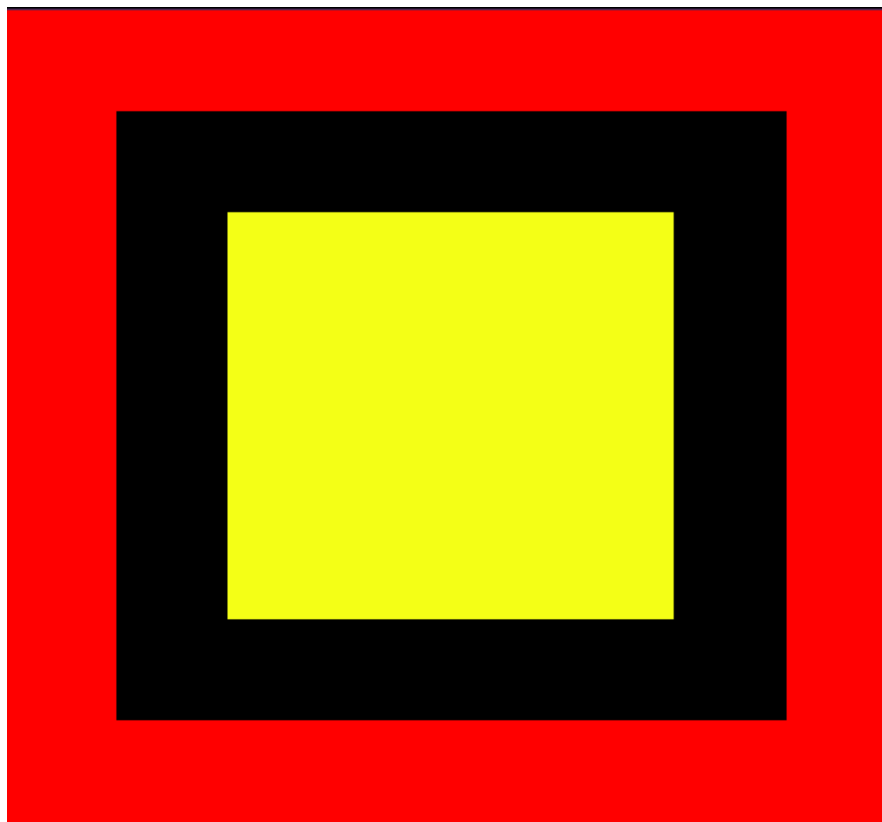
```
width: 500px /* 50% */  
/* height, max-width, min-width,  
   max-height, min-height */
```

# Exercitiu

Pentru codul html, scrieți cod css astfel încât sa obtineti imaginea de mai jos:

```
<body>
<div id="mare">
<div id="mic">
</div>
</div>
</body>
```

- portiunea galbena are latimea si inaltimea de 200px
- portiunile rosie si neagra au grosimea de cate 50px
- intreaga cutie va avea latimea si inaltimea de 400px

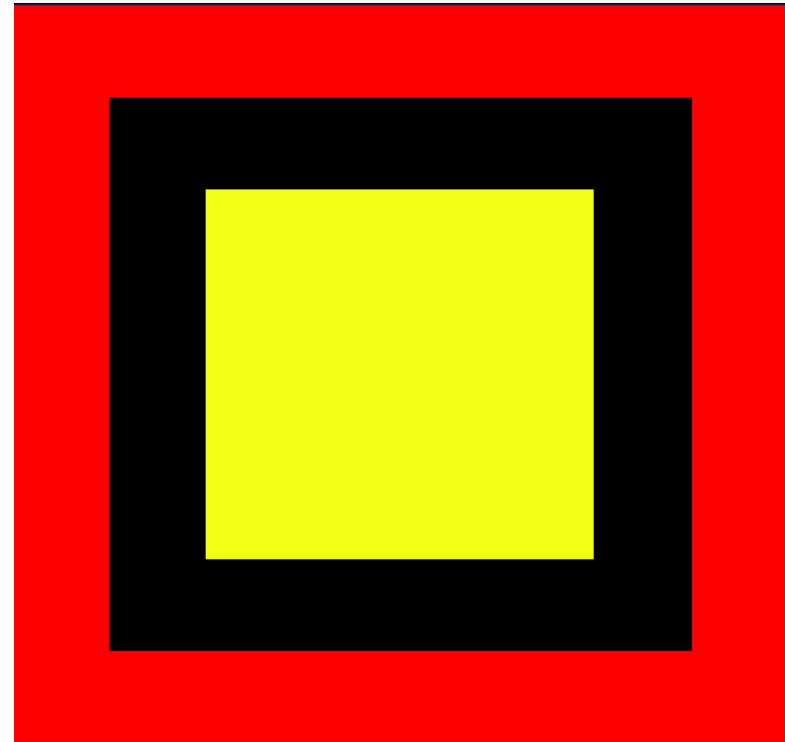


## Soluție 1

```
#mic {  
  background-color: yellow;  
  border: 50px solid black;  
  height: 200px;  
  width: 200px;  
}  
  
#mare {  
  background-color: red;  
  height: 300px;  
  padding: 50px;  
  width: 300px;  
}
```

## Soluție 2

```
#mic {  
  background-color: yellow;  
  height: 200px;  
  margin: 50px;  
  width: 200px;  
}  
  
#mare {  
  background-color: black;  
  border: 50px solid red;  
  height: 300px;  
  width: 300px;  
}
```





# Proprietati CSS: fonturi

<http://www.w3.org/Style/Examples/007/fonts>

```
font-family: Helvetica,"Times New Roman",serif;  
font-size: 16px /* 16pt, 1.25em, 125%, x-large*/  
font-weight: normal /* bold /* lighter /* bolder /* 100,  
..., 900 */  
font-style: normal /* italic /* oblique*/
```

## font-family:

Fonturi generice: serif, sans-serif, monospace, cursive, fantasy

Fonturi specifice: Times New Roman, Arial, Georgia, Verdana

## font-size:

Unitati: px, pt, em (m-size: 1em = dimensiunea fontului curent)

Valori: xx-small, x-small, smal, medium, large, x-large, xx-large,...

Procente: procent din fontul parintelui

at-rules sunt reguli folosite pentru a defini fonturi custom

@font-face - importam un font

<http://www.w3.org/TR/css-syntax-3/>

```
<style text="text/css">
```

```
@font-face{ font-family: nume-font;  
            src:url(fis-font.ext);  
            }
```

```
selector { font-family: nume-font;}  
</style>
```

```
<style type="text/css">  
@font-face{font-family:fontul-meu;  
            src:url(Atlantic%20Bentley.ttf)}  
  
h1 {font-family:fontul-meu;}  
  
</style>
```

***Titlu scris cu font Atlantic Bentley***

[www.dafont.com/](http://www.dafont.com/)

# Proprietati CSS: proprietati ale textului

<http://www.w3.org/TR/CSS2/text.html>

```
color: nume-culoare /* valori */;
text-align: center /*left, right, justify */
text-decoration: underline /* overline, line-through */
text-shadow: -2px 5px 2px gray /*none/*initial;
               /* h-shadow, v-shadow, blur-radius, culoare */
text-indent: 3em /* aliniere prima linie */
text-transform: uppercase /* lowercase /* capitalize
line-height: 150% /* 1.5em , 30px */
word-spacing: 1em /* +15mm */
letter-spacing: 0.1em /* -1px */
```

The default line height in most browsers is about 110% to 120%.

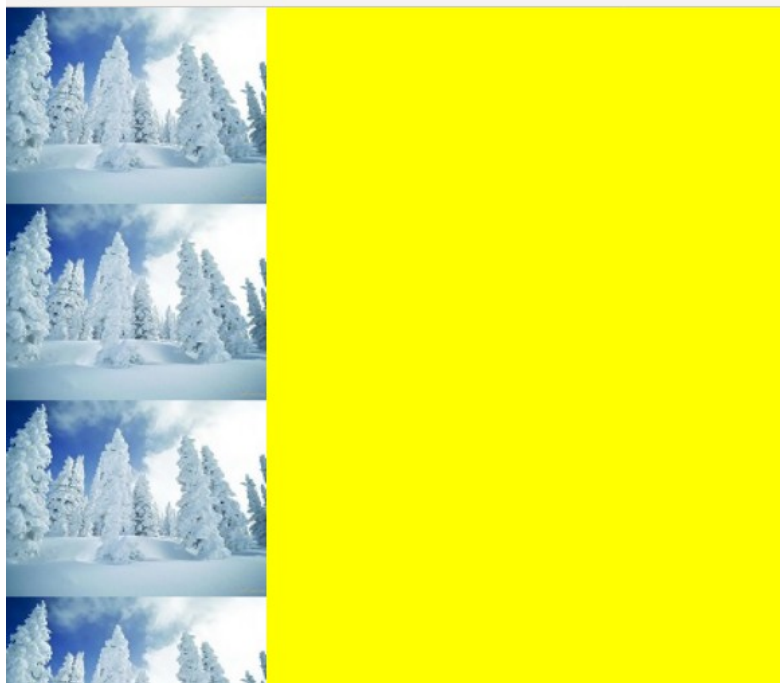
## Proprietati CSS: background

```
background-color: culoare;  
background-image: url(adresa-fisier);  
background-repeat: repeat (implicit)  
    /*repeat-x, repeat-y, or no-repeat*/  
background-position: 370px 20px;  
    /* top, left, right, bottom, center, procente */  
background-size: 300px 100px(dimensiunea);  
    /*contain, cover, 100% 100%, auto*/  
background-origin: padding-box (implicit)  
    /*border-box, content-box*/  
background-attachment: fixed /* scroll  
(implicit)*/
```

## Proprietati CSS: multiple backgrounds

```
background-image: url(img3), url(img2), url(img1);  
background-position: center center, 20% 80%, top left;  
background-repeat: no-repeat;  
/* no-repeat, no-repeat*/
```

```
<style type="text/css">
body{ background-color: yellow;
background-image: url(larna.jpg);
background-repeat: repeat-y;
}
</style>
```



```
<style type="text/css">
body{ background-color: yellow;
      background-image: url(Primavara.jpg),
      url(Vara.jpg), url(larna.jpg);
      background-repeat: repeat-y, repeat-x, repeat;
}
</style>
```



```
<style type="text/css">
body{ background-color: yellow;
      background-image: url(Primavara.jpg), url(larna.jpg),
url(Vara.jpg);
      background-repeat: no-repeat;
      background-position: 20px 20px, 100px 100px, 200px 200px;
}

</style>
```





# Pseudo-clase CSS

Sunt folosite pentru a defini starea unui element:

- Stilizarea unui element la trecerea cu mouse-ul
- Stilizarea diferita a linkurilor vizitate și nevizitate
- Stilizarea unui element când are focus (ex. pentru elemente de tip input)

# Pseudo-clase CSS

Sintaxa:

```
selector:pseudo-class  
    { proprietate:valoare;  
    }
```

- :hover** elemente peste care se trece cu mouse-ul
- :link** legaturi care nu au fost vizitate
- :visited** legaturi care au fost vizitate
- :active** legături active
- :not()** elementele care nu apar în argumentul lui not

```
a:link {text-decoration:none; color: blue;}  
a:visited {text-decoration:none; color:green;}  
p:hover {text-decoration:underline; color:red;}  
li:not(.abc){font-style:italic;}
```

# Pseudo-clase CSS

**:first-child** selecteaza elementul specificat care este primul copil al altui element

**:last-child** selecteaza elementul specificat care este ultimul copil al altui element

```
<head>
<style type="text/css">
p:first-child {color:red;}
p:last-child {color: blue;}
</style>
</head>

<body>
<p>Paragraful 1</p>
<p>Paragraful 2</p>
<p><b>Acest paragraf</b> contine text ingrosat</p>
<div>
<p>Paragraful 3</p>
<p>Paragraful 4</p>
</div>
</body>
```

Paragraful 1

Paragraful 2

**Acest paragraf** contine text ingrosat

Paragraful 3

Paragraful 4

# Example

## HTML

```
<body>
<p>Ion are o <b>pisica</b>, un <b>catel</b>, un <b>
papagal</b>. </p>
<p>George are o <b>bicicleta</b> si un
<b>scuter</b>.</p>
<p>Ana are <b>mere</b> si <b>pere</b>.</p>
</body>
```

```
p:first-child b
{
  color: blue;
}
```

Ion are o **pisica**, un **catel**, un **papagal**.  
George are o **bicicleta** si un **scuter**.  
Ana are **mere** si **pere**.

```
p b:first-child
{
  color: blue;
}
```

Ion are o **pisica**, un **catel**, un **papagal**.  
George are o **bicicleta** si un **scuter**.  
Ana are **mere** si **pere**.

# Pseudo-clase

**:first-of-type** selectează elementul care este primul copil de tipul specificat al altui element

**:nth-of-type(n)** selectează elementul care este al n-lea copil de tipul specificat al altui element

**:nth-child(n)** selectează elementul de tipul specificat care este al n-lea copil al altui element

**n** poate fi:    număr natural  
                  cuvânt cheie (odd, even)  
                  formula:  $a \cdot n + b$  (n pornește de la 0)

## Exemplu

```
.....
<style type="text/css">
p:nth-of-type(2){color:red;}
p:nth-of-type(odd){color: blue;}
p:nth-of-type(4n+0){color: green;}
</style>

</head>
<body>
<p>Paragraful 1</p>
<p>Paragraful 2</p>
<p>Paragraful 3</p>
<p>Paragraful 4</p>
<p>Paragraful 5</p>
<p>Paragraful 6</p>
<p>Paragraful 7</p>
<p>Paragraful 8</p>
<p>Paragraful 9</p>
<p>Paragraful 10</p>
<p>Paragraful 11</p>
<p>Paragraful 12</p>

</body>
```

Paragraful 1

Paragraful 2

Paragraful 3

Paragraful 4

Paragraful 5

Paragraful 6

Paragraful 7

Paragraful 8

Paragraful 9

Paragraful 10

Paragraful 11

Paragraful 12

## Exemplu

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

```
p:nth-child(3) {  
    background: red;  
}
```

```
p:nth-of-type(3) {  
    background: green;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div>Urmeaza 12 paragrafe</div>
```

```
<p>Paragraful 1</p>
```

```
<p>Paragraful 2</p>
```

```
<p>Paragraful 3</p>
```

```
<p>Paragraful 4</p>
```

```
<p>Paragraful 5</p>
```

```
<p>Paragraful 6</p>
```

```
<p>Paragraful 7</p>
```

```
<p>Paragraful 8</p>
```

```
<p>Paragraful 9</p>
```

```
<p>Paragraful 10</p>
```

```
<p>Paragraful 11</p>
```

```
<p>Paragraful 12</p>
```

```
</body>
```

Urmeaza 12 paragrafe

Paragraful 1

Paragraful 2

Paragraful 3

Paragraful 4

Paragraful 5

Paragraful 6

Paragraful 7

Paragraful 8

Paragraful 9

Paragraful 10

Paragraful 11

Paragraful 12

## Exemplu

```
<style>
table {
  font-family: arial, sans-serif;
  border-collapse: collapse;
  width: 30%;
}

td, th {
  border: 2px solid black;
  text-align: left;
  padding: 8px;
}

tr:nth-child(even) {
  background-color: #dddddd;
}
</style>
```

---

Nume	Prenume	Varsta
Popescu	Maria	30
Ionescu	Bogdan	26
Sandu	Mihai	14
Badea	Ana	24
Preda	Dan	20
Iancu	Mara	40



# Pseudo-elemente CSS (::)

elemente virtuale, create cu CSS,  
care nu apar in arborele HTML al paginii

::first-letter    prima litera a unui text

::first-line    prima linie a unui element bloc

::before    sunt folosite pentru a adauga

::after    continut inainte/dupa un element  
HTML; se folosesc cu proprietatea  
content

```
<body>
<h1>Pseudo-elemente</h1>
<p>Lorem ipsum, dolor sit amet consectetur adipisicing elit.... </p>
<p>Lorem ipsum dolor sit, amet consectetur adipisicing elit..... </p>
<p>Lorem, ipsum dolor sit amet consectetur adipisicing elit.....</p>
</body>
```

```
<style>
h1::first-letter {font-size:200%; color: red;}
p::first-line {font-variant: small-caps; color: blue;}
</style>
```

# Pseudo-elemente

LOREM IPSUM, DOLOR SIT AMET CONSECTETUR ADIPISICING ELIT.

Dolorem natus animi provident id quam consequuntur est  
similique neque, ducimus, assumenda corrupti?

LOREM IPSUM DOLOR SIT, AMET CONSECTETUR ADIPISICING ELIT.

Nam ipsa hic omnis porro sunt dolor in? Veritatis placeat odit  
mollitia reiciendis harum porro eum eligendi alias eos iusto  
accusantium sint provident reprehenderit hic dolorum sequi,  
adipisci doloribus numquam eveniet vitae.

LOREM, IPSUM DOLOR SIT AMET CONSECTETUR ADIPISICING ELIT.

Totam, rem eum dolores ipsa adipisci odio consequatur  
quaerat omnis maxime optio possimus fugiat labore?

# Pseudo-elemente CSS (: sau ::)

```
<style>
```

```
.email-address:before { content: "Email address: " ;}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<ul>
```

```
<li class="email-address">myemail@gmail.com</li>
```

```
</ul>
```

- Email address: myemail@gmail.com

# Specificitate= valoare intreaga calculata pe baza sirului de selectori

- se pornește de la valoarea 0
- se aduna 1000 pentru stilizare inline (atributul style)
- se aduna 100 pentru fiecare selector prin id (#nume\_id)
- se aduna 10 pentru fiecare selector prin clasa (.nume\_clasa), atribut ([attr=..] sau pseudo-clasa (:hover, :nth-child(2n), :nth-of-type(2n),...))
- se aduna 1 pentru fiecare selector de tip element (nume\_tag) sau pseudo-selector (::before, ::after, ::first-line, ::first-letter)
- pentru selectorii combinati (descendenti, copii, frați, selectorul universal \* ) nu se adauga nimic

Selectarea cu specificitate mai mare este prioritara, chiar daca este urmata de o alta selectare cu o prioritate mai mica

```
div#a {background-color: green;}  
#a {background-color: yellow;}  
div[id="a"] {background-color: blue;}
```

```
<div id="a">This is a div</div>
```

Se va aplica prima regula  
(are cea mai mare specificitate)

## Proprietati CSS: visibility, overflow

`overflow: hidden; /* visible, scroll */`

descrie modul in care este redat textul care iese dintr-un cadru

`visibility: visible /* hidden */`

seteaza vizibilitatea unui anume element;  
chiar daca vizibilitatea este “hidden”  
elementul ocupa loc in pagina

## Exemplu : overflow

```
<style>
div {width:2cm; height:2cm; border:1px solid green;
margin-left: 25px;}

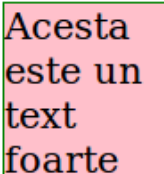
div#hidden{overflow: hidden; background-color:pink;}

div#scroll{overflow: scroll; background-color: grey;}

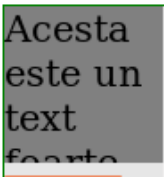
div#visible{overflow: visible;background-color: yellow;}

</style>
```

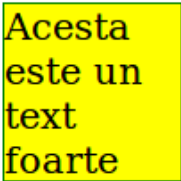
```
<body>
<div id="hidden">
  Acesta este un text foarte lung si important.
</div>
<hr>
<div id="scroll">
  Acesta este un text foarte lung si important.
</div>
<hr>
<div id="visible">
  Acesta este un text foarte lung si important.
</div>
```



---



---

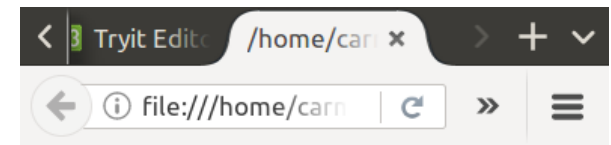


lung si  
important.

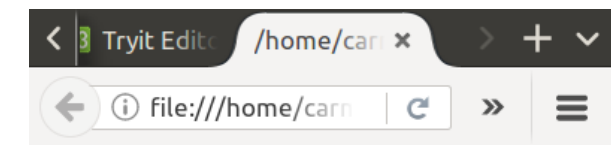
## Menu drop-down

```
ul ul {visibility:hidden;}  
ul li:hover ul {color:blue; visibility:visible;}
```

```
<ul>  
<li>MENIU 1  
    <ul><li> submenu 1.1</li>  
        <li> submenu 1.2 </li>  
    </ul> </li>  
<li>MENIU 2  
    <ul><li> submenu 2.1</li>  
        <li> submenu 2.2</li>  
    </ul> </li>  
</ul>
```



- MENIU 1
  - submenu 1.1
  - submenu 1.2
- MENIU 2



- MENIU 1
- MENIU 2
  - submenu 2.1
  - submenu 2.2

## Proprietăți CSS: `display`

Precizeaza modul cum este afisat un element în pagina

`display: inline/* block, inline-block, none */`

`display:inline /*` spatiul pe care il ocupa este lungimea elementului;  
Ignora proprietatile `width` si `height`,  
dar accepta proprietati precum  
`border, padding ...*/`

```
li { display:inline;  
      background-color:pink;  
      margin-left:5px;}
```

Item 1 Item 2 Item 3 Item 4

```
<ul>  
<li>Item 1</li>  
<li>Item 2</li>  
<li>Item 3</li>  
<li>Item 4</li>  
</ul>
```



`display: block; /*` permit stilizarea folosind box model; sunt plasate pe rand nou\*/

```
<style>
a {
  display: block;
}
</style>
</head>

<body>
<p>Meniu pe verticala</p>
<a href="index1.html" target="_blank">Link 1</a>
<a href="index2.html" target="_blank">Link 2</a>
<a href="index3.html" target="_blank">Link 3</a>
</body>
```

Meniu pe verticala

[Link 1](#)

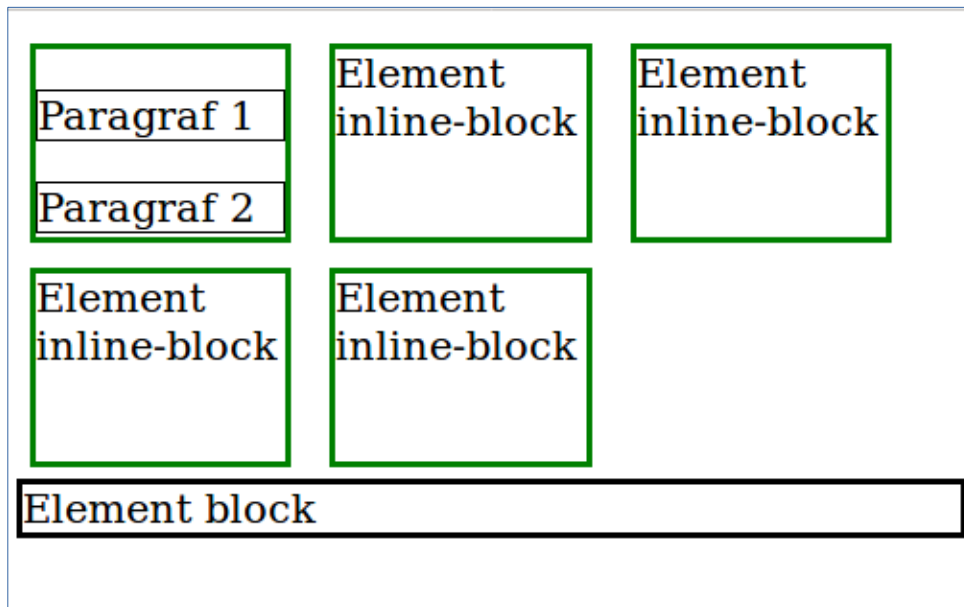
[Link 2](#)

[Link 3](#)

`display:inline-block; /* element positionat inline  
care poate fi stilizat ca un element bloc;  
accepta proprietatile width și height*/`

```
<style>
div.box { display: inline-block;
          width:100px; height:75px;
          border: 3px solid green;
          margin:5px;
          vertical-align:top;
        }
div.alt-div {border: 3px solid black;}
p {border: 1px solid black;}
</style>
</head>
```

```
<body>
<div class="box">
<p>Paragraf 1</p>
<p>Paragraf 2</p>
</div>
<div class="box">Element inline-block</div>
<div class="box">Element inline-block</div>
<div class="box">Element inline-block</div>
<div class="box">Element inline-block</div>
<div class="alt-div">Element block</div>
.....
```



`display:none` /\* elementul este eliminat din pagina,  
dar este prezent in DOM \*/

```
<p>Text <span id="ex1" class="cl"> AAAAAA </span>  
text <span id="ex2" class="cl"> BBBBBB </span> text.  
</p>
```

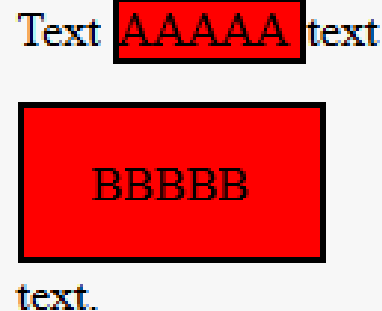
```
.cl {background: red; width: 100px; height: 50px; line-height: 50px;  
text-align: center; border: solid 2px black;}
```

```
#ex2 {display:inline-block;}
```



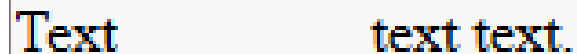
Text AAAAAA text BBBBBB text.

```
#ex2 {display:block;}
```



Text AAAAAA text  
BBBBBB  
text.

```
#ex1 {visibility:hidden}  
#ex2 {display:none;}
```



Text text text.

MENIU 1

MENIU 2

- submenu 2.1
- submenu 2.2

```
ul ul {visibility:hidden;}  
ul li:hover ul {color:blue; visibility:visible;}  
.cl {display: inline-block; vertical-align: top;}
```

```
<ul>  
<li class="cl">MENIU 1<ul><li> submenu 1.1</li>  
  <li> submenu 1.2 </li> </ul> </li>  
<li class="cl">MENIU 2 <ul><li> submenu 2.1</li>  
  <li> submenu 2.2</li> </ul> </li>  
</ul>
```

```
<section id="galerie">
<h2> Galerie de imagini I</h2>
<figure class="ig">

<figcaption> peisaj de iarna </figcaption>
</figure>
```

```
<figure class="ig">

<figcaption> flori de vara </figcaption>
</figure>
```

```
<figure class="ig">
 toamna </figcaption>
</figure>
```

```
<figure class="ig">

<figcaption> vara </figcaption>
</figure>
</section>
```

```
#galerie {
  margin-left :auto;
  margin-right:auto;
  margin-top:200px;
  width:400px; }
```

```
figcaption, h2, section {text-align: center;}
```

```
.ig {margin:10px; display: inline-block;}
```

## Galerie de imagini I



peisaj de iarna



flori de vara



toamna



primavara

# Variabile CSS

Variabilele sunt proprietăți CSS *custom* (nestandard, definite de utilizator), definite pentru a fi reutilizate în cadrul unui document

Globale: valabile în întreg documentul

```
:root {  
  --nume-proprietate: valoare;  
}
```

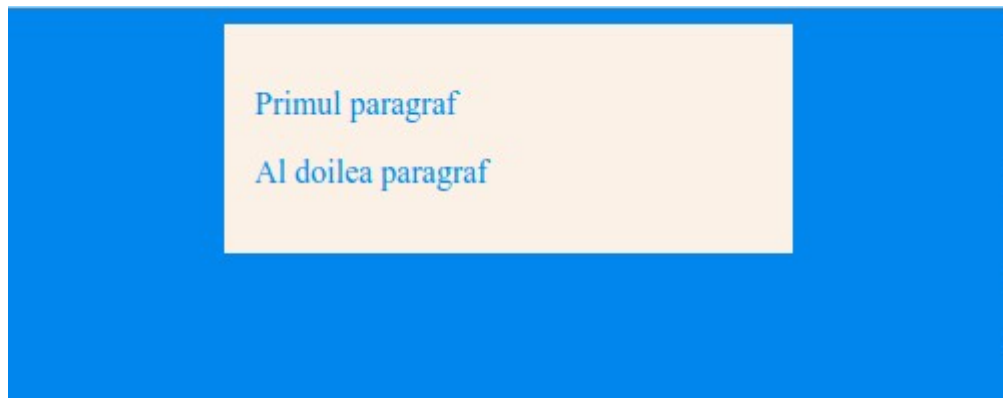
Locale: valabile în interiorul (toți descendenții) elementului în care au fost definite

```
element {  
  --nume-proprietate: valoare;  
}
```

Se folosesc cu funcția **var(--nume-proprietate)**

```
body {  
  --culoare1: #2496ed;  
  --culoare2: #faf0e6;  
  --margin-container: 100px;  
}  
body {  
  background-color: var(--culoare1);  
}  
.container {  
  background-color: var(--culoare2);  
  padding: 15px;  
  margin-left: var(--margin-container);  
  margin-right: var(--margin-container);  
}  
p {  
  color: var(--culoare1);  
}
```

```
<div class="container">  
  <p>Primul paragraf</p>  
  <p>Al doilea paragraf</p>  
</div>
```



# Exercițiu

**Scrieți cod CSS pentru cerinta:**

Cand se vine cu cursorul pe un paragraf, toate elementele de dupa el din pagina (frați) care au clasa „ceva” cat si elementele din subarborii fratilor care au clasa „altceva” sa aiba font-size-ul de 2 ori mai mare decat containerul

# Rezolvare

Cand se vine cu cursorul pe un paragraf, toate elementele de dupa el din pagina (frați) care au clasa „ceva” cat si elementele din subarborii fratilor care au clasa „altceva” sa aiba font-size-ul de 2 ori mai mare decat containerul

```
p:hover ~ .ceva , p:hover ~ * .altceva  
  {  
    font-size:2em;  
  }
```