

Tehnici Web

CURSUL 3

Semestrul I, 2017-2018
Carmen Chirita

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

CSS Layout – proprietatea **position**

Specifica tipul de pozitionare al unui element în pagina;
poate avea una din valorile:

- **position:static**
- **position:relative**
- **position:absolute**
- **position:fixed**
- **position:sticky**

Pentru pozitionare se folosesc proprietatile **left**, **right**, **top** și **bottom**
Se va specifica mai intai tipul de pozitionare cu proprietatea **position**

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

CSS Layout – proprietatea **position**

position: static

Este poziția implicită a oricărui element HTML; cu poziția static elementul va avea un flux normal în pagină;
Elementele cu poziția static nu sunt afectate de proprietățile left, right, top, bottom.

```
<style>
div{position:static;
    width:200px;
    border: 3px solid #73AD21;
    background-color: cyan;
}
</style>
</head>
<body>
<h2>Pozitionare statica</h2>
<p>Urmatorul element este un div
pozitionat<br> static</p>
<div class="static">
    Acesta este un element
    pozitionat static.
</div>
</body>
```

Pozitionare statica

Urmatorul element este un div pozitionat static

Acesta este un element
pozitionat static.

CSS Layout – proprietatea **position**

position: relative

Un element cu **position: relative** este poziționat relativ față de poziția pe care ar fi avut-o în mod normal în document;
Pentru poziționare se folosesc proprietățile left, right, top, bottom.

```
<style>
div{position:relative;
    left: 35px;
    width:200px;
    border: 3px solid #73AD21;
    background-color: cyan;
}
</style>
</head>
<body>
<h2>Poziționare relativa</h2>
<p>Următorul element este un div
poziționat<br>relativ</p>
<div class="relativ">
    Acesta este un element poziționat relativ.
</div>
```

Poziționare relativa

Următorul element este un div poziționat relativ

Acesta este un element
poziționat relativ.

CSS Layout – proprietatea **position**

position: absolute

Elementul este pozitionat intr-o pozitie relativa fata de primul parinte care are **position:absolute/relative/fixed**; elementul este **scos din fluxul documentului**.

```
<style>
div{position:absolute;
  left: 35px; top:20px;
  width:200px;
  border: 3px solid #73AD21;
  background-color: cyan;
}
</style>
</head>
<body>
<h2>Pozitionare absoluta</h2>
<p>Urmatorul element este un div
pozitionat<br>
absolut</p>
<div class="absolut">
  Acesta este un element pozitionat absolut.
</div>
```

Po Acesta este un element **uta**
pozitionat absolut.

Urmatorul element este un div pozitionat absolut

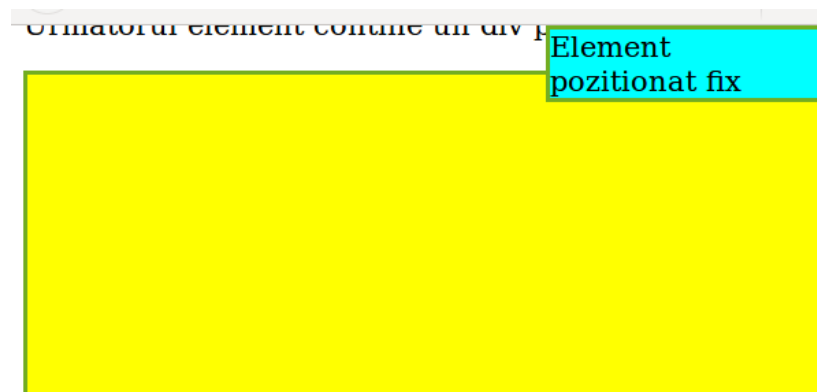
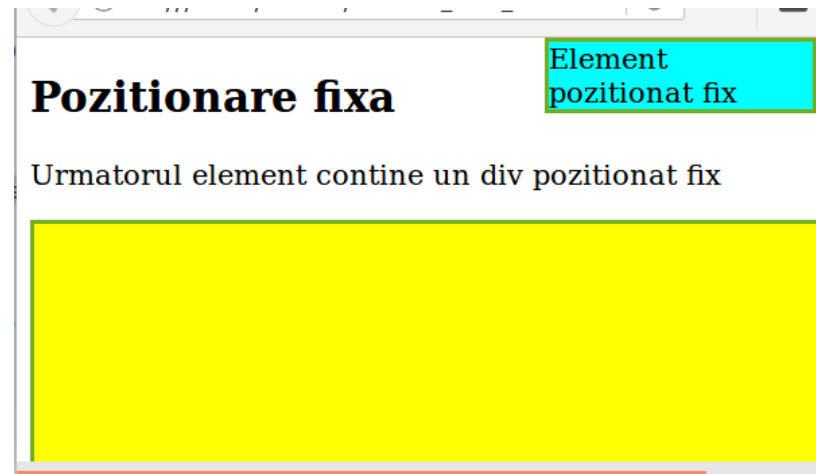
CSS Layout – proprietatea **position**

position: fixed

Elementul are o pozitie fixa in fereastra de browser; este scos din fluxul documentului si nu e afectat de scroll.

Este pozitionat folosind proprietatile left, top, right, bottom.

```
<style>
div.parinte{width:500px; height:300px;
  border: 3px solid #73AD21;
  background-color: yellow
}
div.fix{position:fixed;
  right:0; top:0;
  width:150px;
  border: 3px solid #73AD21;
  background-color: cyan;
}
</style>
</head>
<body>
<h2>Pozitionare fixa</h2>
<p>Urmatorul element contine un div pozitionat
fix</p>
<div class="parinte">
<div class="fix">
  Element pozitionat fix
</div>
</div>
```



CSS Layout – proprietatea **position**

position: sticky

Elementul comuta între poziționarea relativă și fixă în funcție de poziția scrolului; dacă poziția scrolului nu depășește elementul acesta va fi poziționat relativ, altfel se comporta ca un element fix

```
<style>
div.sticky {
  position: sticky;
  top: 0;
  padding: 5px;
  background-color: lightgreen;
  border: 2px solid black;
}
</style>
</head>
<body>
<p>Paragraf 1</p>
<p>Paragraf 2</p>
<p>Paragraf 3</p>
<div class="sticky">Element
sticky</div>
<div>
<p>Paragraf 4</p>
<p>Paragraf 5</p>
<p>Paragraf 6</p>
<p>Paragraf 7</p>
.....
```

Paragraf 1

Paragraf 2

Paragraf 3

Element sticky

Paragraf 4

Paragraf 5

Element sticky

Paragraf 4

Paragraf 5

Paragraf 6

Paragraf 7

Suprapunerea elementelor: proprietatea **z-index**

Prin pozitionare elementele se pot suprapune, se pot crea stive de elemente; ordinea elementelor în stiva este data de **z-index**; Elementele cele mai vizibile au **z-index** mai mare.

Proprietatea **z-index** poate fi setata numai pentru elementele care sunt pozitionate **absolute**, **relative** sau **fixed**.

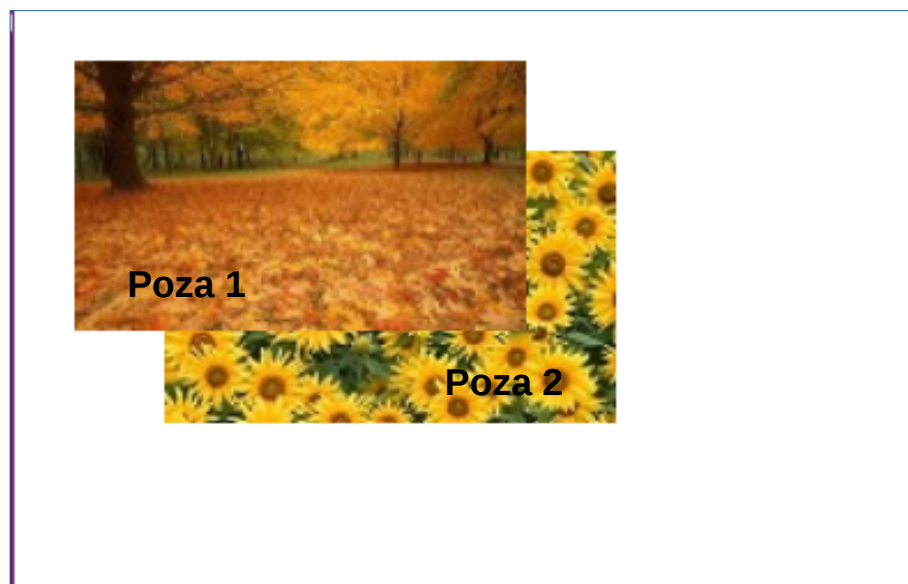
z-index: ..., -100 /* 0, 100, ...


```

<style>
#poza1{position:absolute;
    left:20px; top:20px;
    z-index:10;
}
#poza2{position:absolute;
    left:50px; top:50px;
    z-index:2;
}
</style>
</head>
<body>


</body>

```



```

.....
#poza3{position:absolute;
    left:100px; top:100px;
    z-index:10;
}
</style>
</head>
<body>
.....

</body>

```



CSS layout: proprietatile **float** și **clear**

```
float:left /*right, none*/  
clear:both /*left, right*/
```

Elementele cu proprietatea **float** sunt scoase din fluxul documentului și pozitionate conform valorii, la stanga sau la dreapta. Ele afecteaza celelalte elemente care se aranjeaza inconjurand (“wrapping”) elementele **float**.

Elementele cu proprietatea **clear** nu se aranjeaza in jurul elementelor **float** ci se deplaseaza sub acestea.

Elementele pozitionate absolut ignora proprietatea float

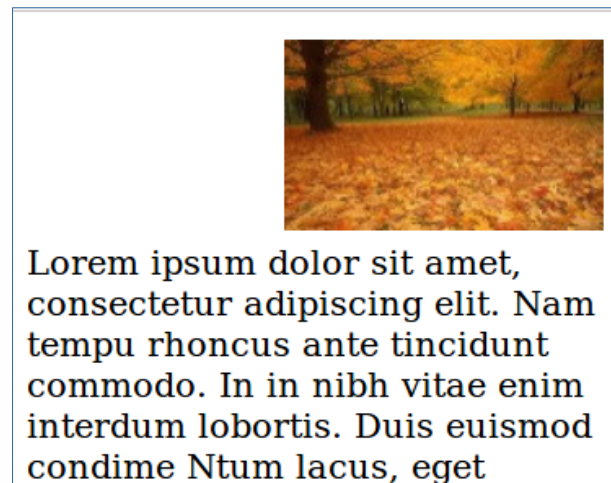
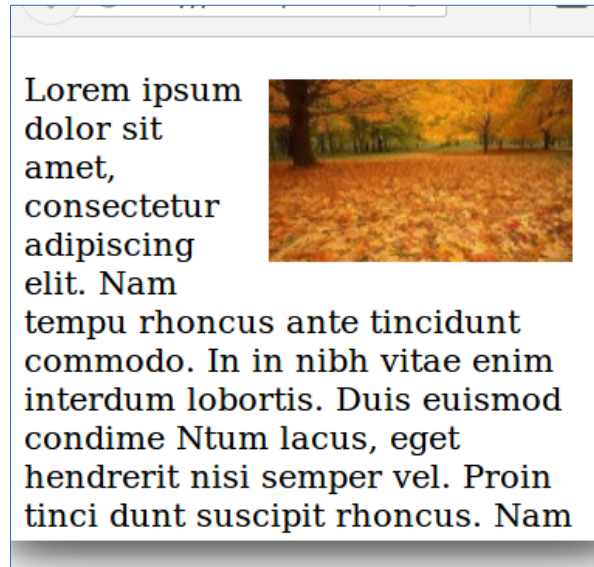
```

<head>
<style>
#poza1 {float:right;
        margin:5px;
}
</style>
</head>
<body>

<p>Lorem ipsum dolor sit
a.....</p>
</body>

```

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



CSS layout: proprietatea **display**

Precizeaza modul cum este afisat un element în pagina

display: inline/* inline-block, 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;}
```

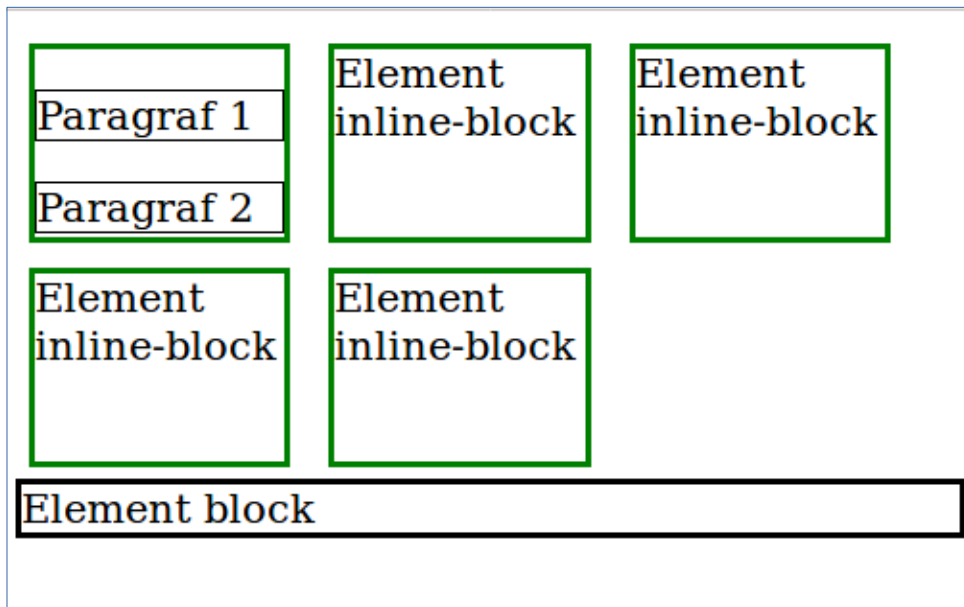
Item 1 Item 2 Item 3 Item 4

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

`display: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: 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: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>
```

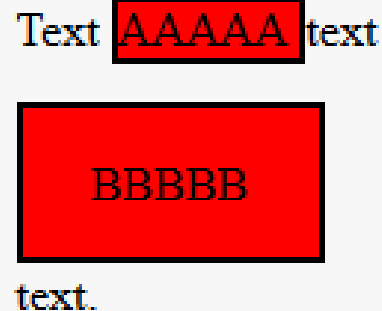
```
.cl {background: red; width: 100px; height: 50px; line-height: 50px;  
text-align: center; border: solid 2px black;}
```

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



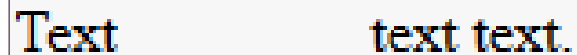
Text AAAAAA text BBBBBB text.

```
#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

Vendor specific properties:

proprietati CSS care nu sunt standardizate,
dar browserele ofera suport pentru ele

```
proprietate: valoare;  
-ms-proprietate:valoare; /* IE */  
-moz-proprietate:valoare; /* Firefox */  
-webkit-proprietate:valoare; /* Chrome si Safari*/  
-o-proprietate:valoare; /* Opera */
```

CSS3 - transformari

Permit ca un element să-și schimbe poziția, forma și mărimea.
CSS3 suporta transformari 2D și 3D.

Sintaxa: `transform:metoda(valori)`

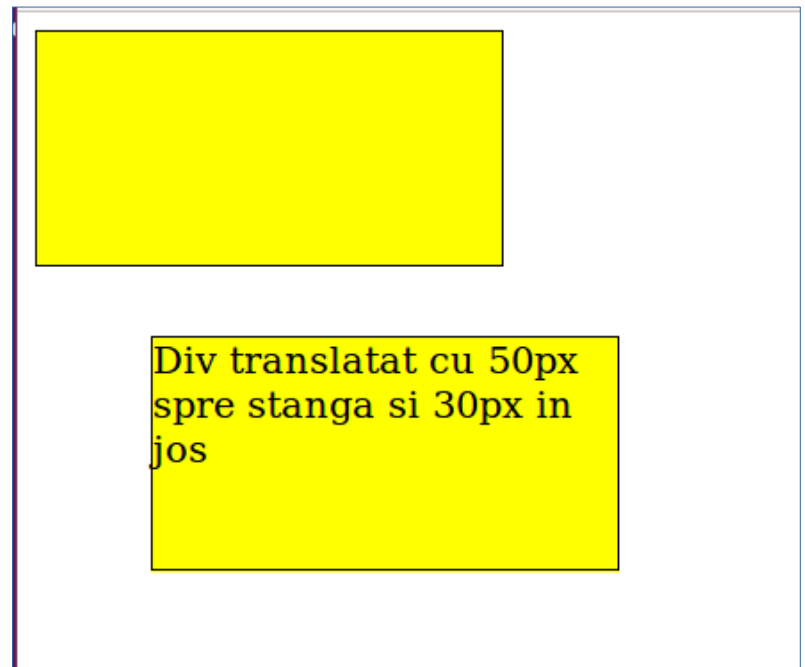
Metode de transformare 2D:

`translate()`, `rotate()`, `scale()`, `skewX()`, `skewY()`, `matrix()`

**transform: translate(tx,ty), translate(tx)(ty=0),
translateX(tx), translateY(ty)**

Muta elementul de la pozitia initiala, in functie de parametri dati pentru pozitionare: (axa-X) si (axa-Y)

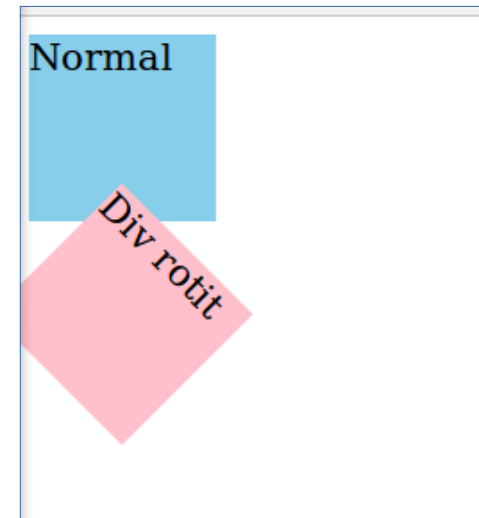
```
<style>
div {
  width: 200px;
  height: 100px;
  background-color: yellow;
  border: 1px solid black;
}
div.translatat{
  -ms-transform: translate(50px,30px); /* IE 9 */
  -webkit-transform: translate(50px,30px); /* Safari */
  transform: translate(50px,30px); /* Standard syntax */
}
</style>
</head>
<body>
<div class="normal">
</div>
<div class="translatat">
Div translatat cu 50px spre stanga si 30px in jos
</div>
</body>
```



transform: rotate(grade) (ex. 45deg, -45deg)

Rotește elementul în sensul acelor de ceas dacă valoarea **grade** este pozitivă și invers dacă valoarea este negativă

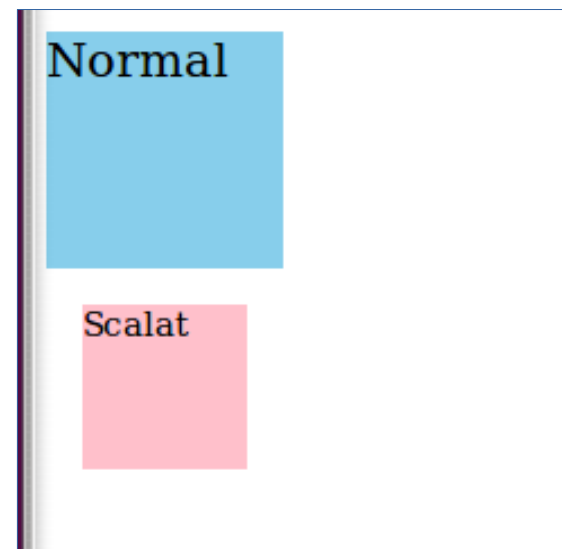
```
div {  
  width: 80px;  
  height: 80px;  
  background-color: skyblue;  
}  
.rotated {  
  transform: rotate(45deg);  
  background-color: pink;  
}  
</style>  
<body>  
<div> Normal</div>  
<div class="rotated">Div rotit</div>  
</body>
```



```
transform: scale(sx,sy), scale(sx) (sy=sx),  
          scaleX(sx), scaleY(sy)
```

Mărește sau micșorează dimensiunile elementului în funcție de parametrii **sx** și **sy**.

```
<style>  
div {  
  width: 80px;  
  height: 80px;  
  background-color: skyblue;  
}  
.scaled {  
  transform: scale(0.7);  
  background-color: pink;  
}  
</style>  
<body>  
<div>Normal</div>  
<div class="scaled">Scalat</div>  
</body>
```



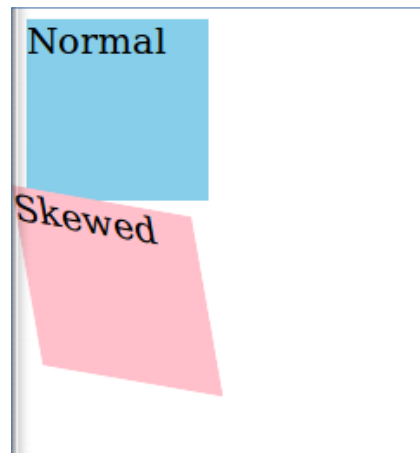
transform: skew(ax,ay), skew(ax) (ay=0),
skewX(ax), skewY(ay)

Distorsioneaza elementul (și conținutul) de-a lungul axelor X și Y
în funcție de parametrii specificați.

```
.skewed {  
  transform: skew(10deg);  
  background-color: pink;  
}
```

```
<body>  
<div>Normal</div>  
<div class="skewed">Skewed</div>  
</body>
```

```
.skewed {  
  transform: skew(10deg, 10deg);  
  background-color: pink;  
}
```



Observatii

Unui element i pot aplica mai multe transformari în același timp (exemplul 1).

Exemplul 1

```
element {  
    transform: translate(25px,30px) rotate(45deg) scale(2);  
}
```

Exemplul 2

```
element {  
    transform: translate(25px,30px);  
    transform: rotate(45deg);  
    transform: scale(2);  
}
```

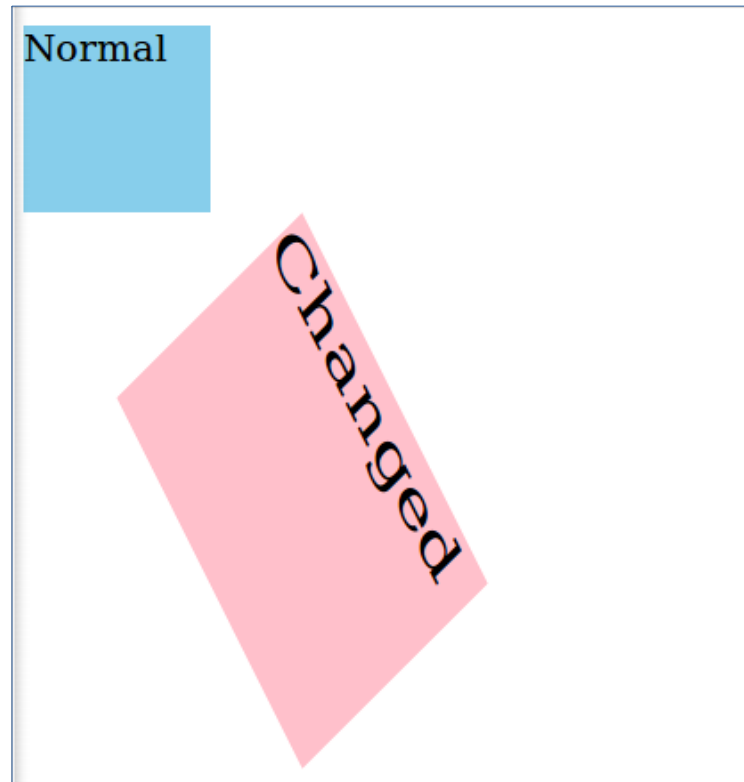
În exemplul 2 se va face doar ultima transformare
transform: scale(2).

transform: matrix(a,b,c,d,tx,ty)

```
<style>
div {
  width: 80px;
  height: 80px;
  background-color: skyblue;
}

.changed {
  transform: matrix(1, 2, -1, 1, 80, 80);
  background-color: pink;
}

</style>
</head>
<body>
<div>Normal</div>
<div class="changed">Changed</div>
</body>
```



CSS transformari 3D

Sunt extensii ale transformarilor 2D.

Efectul 3D se poate obtine folosind proprietatea
perspective = distanta pe axa oz

translateZ(tz)
translate3d(tx,ty,tz)

rotateZ(az)
rotate3d(x,y,z,grade)

scaleZ(sz)
scale3d(sx,sy,sz)

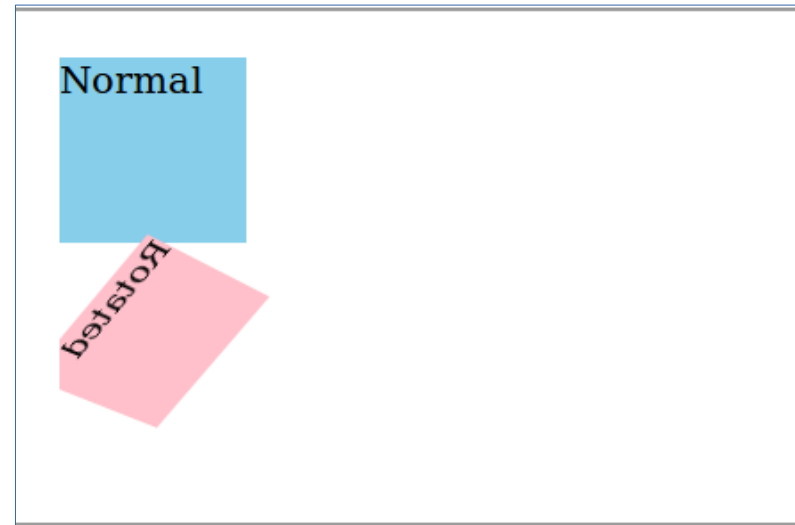
matrix3d(n,n,n,.....,n)
16 valori

perspective(n)

```
body {  
  perspective: 800px;  
}  
div {  
  width: 80px;  
  height: 80px;  
  background-color: skyblue;  
}  
.rotated {  
  transform: rotate3d(0, 1, 0, 60deg);  
  background-color: pink;  
}  
</style>  
</head>  
<body>  
<div>Normal</div>  
<div class="rotated">Rotated</div>  
</body>
```



```
.rotated {  
  transform: rotate3d(1, 2, -1, 192deg);  
  background-color: pink;  
}
```



CSS3 - tranzitii

O tranzitie este declansata de un element
:hover, :focus, :active

```
transition-property: proprietate CSS;  
transition-duration: 2s (durata tranzitiei);  
transition-timing-function: linear;  
/* ease, ease-in, ease-out, step-start, step-end */  
transition-delay: 1s; (cand incepe tranzitia  
dupa schimbarea proprietatii)
```

Se poate folosi în varianta prescurtata:

```
transition: margin-right 4s ease-in-out 1s;
```

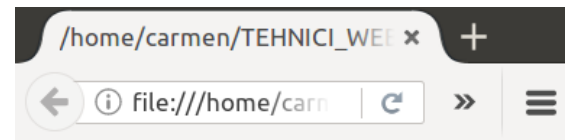
Proprietati carora li se poate aplica o tranzitie

<https://www.w3.org/TR/css3-transitions/#animatable-properties>

```

<style>
div {
  border-style: solid;
  border-width: 1px;
  display: block;
  width: 100px;
  height: 100px;
  background-color: #0000FF;
  transition: width 2s, height 2s, background-color
              2s, transform 2s;
}
div:hover {
  background-color: #FFCCCC;
  width: 200px;
  height: 200px;
  transform: rotate(180deg);
}
</style>
</head>
<body>
<p>Tranzitie CSS</p>
<div>Tranzitie pe acest element</div>
<p>La hover pe div incepe tranzitia</p>
</body>

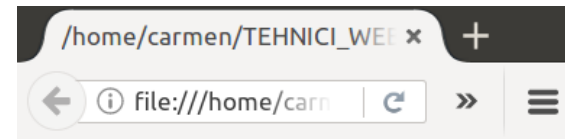
```



Tranzitie CSS

Tranzitie pe
acest
element

La hover pe div incepe tranzitia



Tranzitie CSS

Tranzitie pe acest
element

La hover pe div incepe tranzitia

CSS-animatii

```
@keyframes nume-animatie{
  0% { } /* from {stil initial} */
  25% { }
  ....
  75% { }
  100% { } /* to {stil final} */
}
```

```
animation-name: nume-animatie;
animation-duration: 5s (obligatorie);
animation-timing-function: linear
/* ease, ease-in, ease-out, ....*/
animation-delay: 2s
animation-iteration-count: 10; /* infinite */
animation-direction: normal /* alternate */
reverse */
```

```
animation: myname 5s linear 2s..
-webkit-animation:
-moz-animation:
-ms-animation
-o-animation:
```

```
@keyframes myname {}
@-webkit-keyframes myname{}
@-moz-keyframes myname{}
@-ms-keyframes myname{}
@-o-keyframes myname{}
```

```
animation: myname 5s linear 2s infinit alternate;
```

```
div {  
  width: 100px;  
  height: 100px;  
  background-color: red;  
  animation-name: exemplu;  
  animation-duration: 4s;  
  animation-iteration-count: infinite;  
}  
  
@keyframes exemplu {  
  0% {background-color: red;}  
  25% {background-color: yellow;}  
  50% {background-color: blue;}  
  100% {background-color: green;}  
}
```

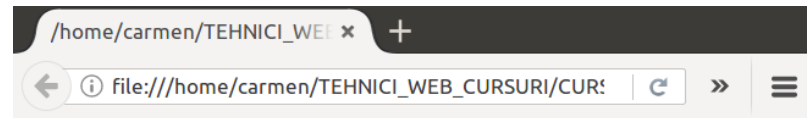
Exemplu. animatie.html

CSS-aliniere pe orizontala: centrare

Pentru elemente block se folosesc proprietatile margin și width cu valorile:

margin-left: auto
margin-right: auto
width setat, diferit de 100%

```
<style>
div {
  margin-left:auto;
  margin-right:auto;
  width: 300px;
  height: auto;
  background-color: skyblue;
}
h1{text-align:center;}
</style>
</head>
<body>
<h1>Centrare element block</h1>
<div>Div centrato
<p>Paragrafo 1</p>
<p>Paragrafo 2</p>
<p>Paragrafo 3</p>
</div>
</body>
```



Centrare element block

Div centrat
Paragraf 1
Paragraf 2
Paragraf 3

CSS-aliniere pe orizontala: centrare

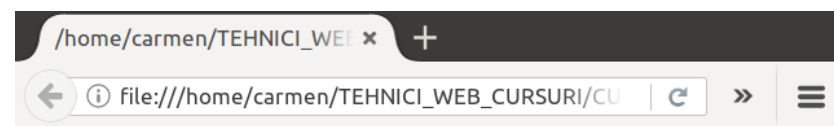
Pentru elemente inline se foloseste proprietatea display cu valoarea **display:block** după care se aliniaza ca un element block

```
<style>
h1,p {text-align:center;}

a {display:block; margin:auto;
  width:100px;}

img {display:block; margin:auto;
     width:150px; height:90px;}
</style>
</head>
<body>
<h1>Centrare elemente inline</h1>
<a href="toamna.html">Toamna</a>
<p>O poza de toamna</p>

</body>
```



Centrare elemente inline

[Toamna](#)

O poza de toamna



CSS-aliniere pe orizontala: stânga, dreapta

O metoda pentru aliniere de elemente la stânga sau la dreapta:

```
position: absolute;  
right: 0px sau left: 0px
```

```
<style>  
  h1 {text-align:center;}  
  img {position:absolute; right:0px;  
        width:150px; height:90px;}  
</style>  
</head>  
<body>  
  <h1>Aliniere imagine la dreapta</h1>  
  <p>O poza de toamna</p>  
    
</body>
```

Aliniere imagine la dreapta

O poza de toamna



CSS-aliniere pe orizontala: stânga, dreapta

O alta metoda pentru aliniere de elemente la stânga sau la dreapta:

```
float: left /*right*/;
```

```
<style>
body{margin:0px; padding:0px;}
h1 {text-align:center;}
img {float:left;
     width:150px; height:90px;}
</style>
</head>
<body>
<h1>Aliniere imagine la stanga</h1>
<p>O poza de toamna</p>

</body>
```



Aliniere imagine la stanga

O poza de toamna

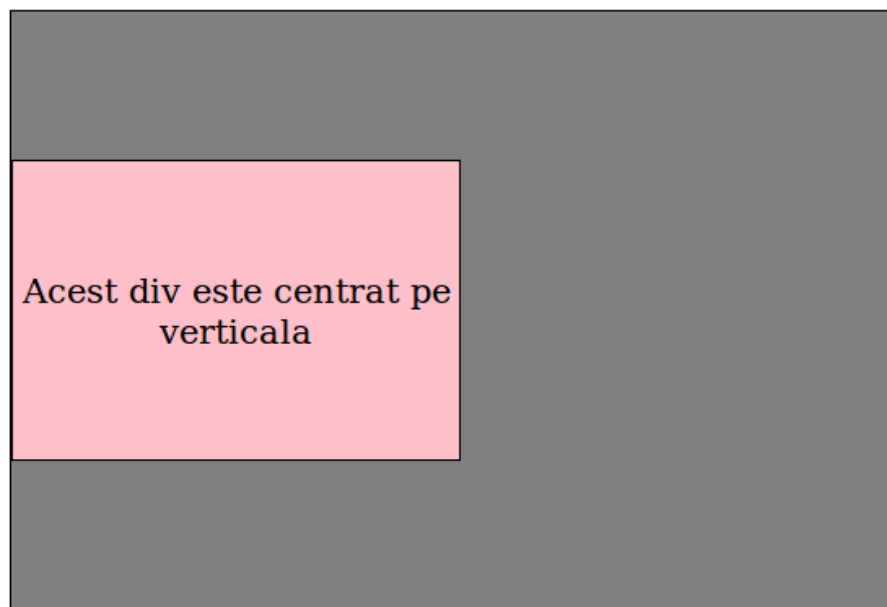


CSS-centrare pe verticala

O metoda pentru centrarea pe verticala este cu ajutorul proprietatii **padding** cu **padding-top**, **padding-bottom** egale

```
<style>
div.parinte{ width: 30%;
              padding: 70px 0 ;
              border: 1px solid black;
              margin-left:10px;
              background-color:grey;
            }
div.copil{padding: 35px 0;
          width:50%;
          border: 1px solid black;
          background-color:pink;
          text-align:center;
        }
</style>
</head>
<body>
<h1>Centrare pe verticala</h1>
<div class="parinte">
<div class="copil">
<p> Acest div este centrat pe
verticala</p>
</div>
</div>
</body>
```

Centrare pe verticala



CSS- aliniere pe verticala

O alta metoda pentru centrarea pe verticala este cu proprietatile **line-height** și **vertical-align**:

line-height și **height** egale pentru elementul părinte
vertical-align: middle/* top/* bottom pentru elementul copil
display: inline-block; pentru elementul copil de tip block

```
<style>
.parinte {line-height:200px;
          height:200px; width:30%;
          border: 1px solid black;
          margin-left:10px;
          background-color:grey;
        }
.copil {display:inline-block;
        vertical-align:middle; /*top, bottom*/
        width:200px;height:100px;
        border: 1px solid black;
        background-color:yellow; }
</style>
</head>
<body>
<h1>Centrare pe verticala</h1>
<div class="parinte">
<div class="copil"></div>
</div>
</div>
```

Centrare pe verticala

