

# Tehnici Web

## CURSUL 4

Semestrul II, 2023-2024  
Carmen Chirita

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

# 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:

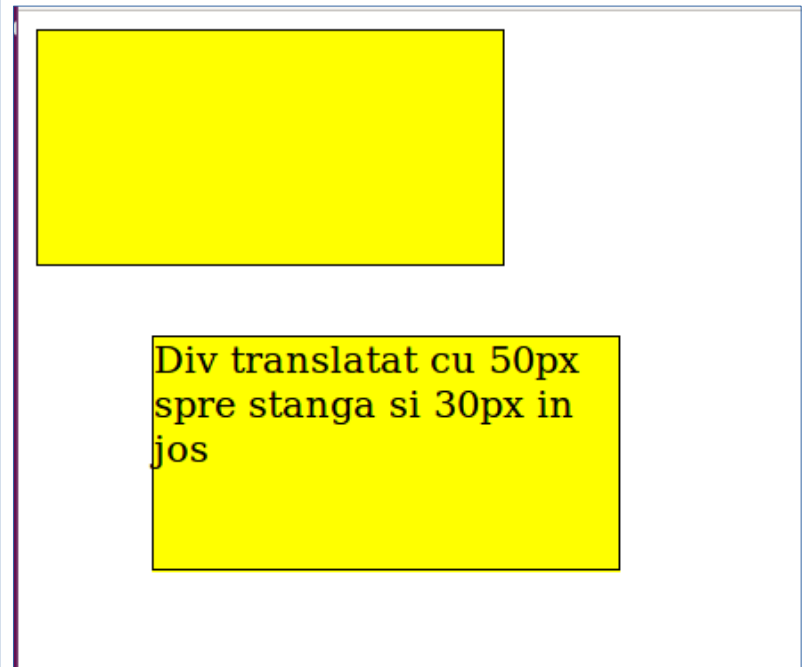
`translate()`, `rotate()`, `scale()`, `skew()`, `matrix()`

<https://css-tricks.com/almanac/properties/t/transform/>

```
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)

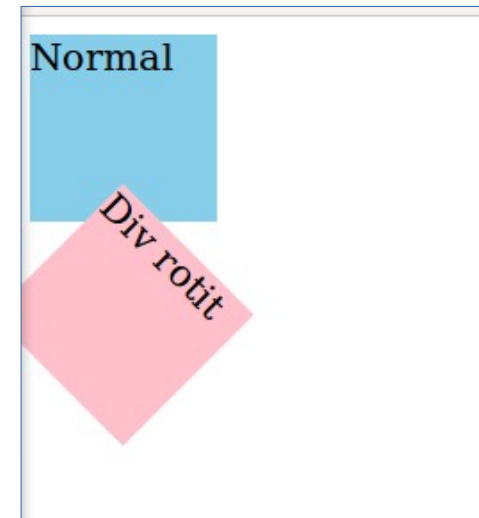
```
<style>  
div {  
  width: 200px;  
  height: 100px;  
  background-color: yellow;  
  border: 1px solid black;  
}  
div.translatat{  
  transform: translate(50px,30px);  
}  
</style>  
</head>  
<body>  
<div class="normal">  
</div>  
<div class="translatat">  
Div translatat cu 50px spre stanga 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

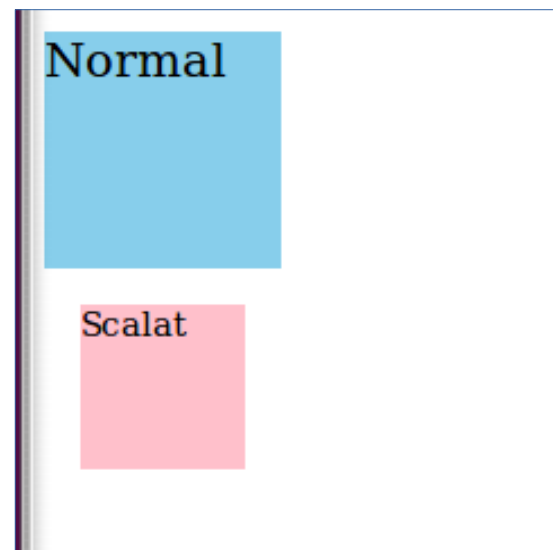
```
div {  
  width: 80px;  
  height: 80px;  
  background-color: skyblue;  
}  
.rotit{  
  transform: rotate(45deg);  
  background-color: pink;  
}  
</style>  
<body>  
<div> Normal</div>  
<div class="rotit">Div rotit</div>  
</body>
```



```
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;  
}  
.scalat{  
  transform: scale(0.7);  
  background-color: pink;  
}  
</style>  
<body>  
<div>Normal</div>  
<div class="scalat">Scalat</div>  
</body>
```



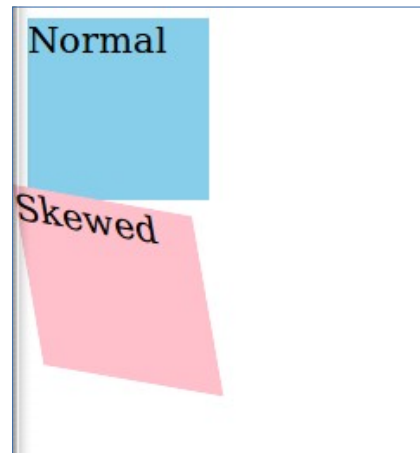
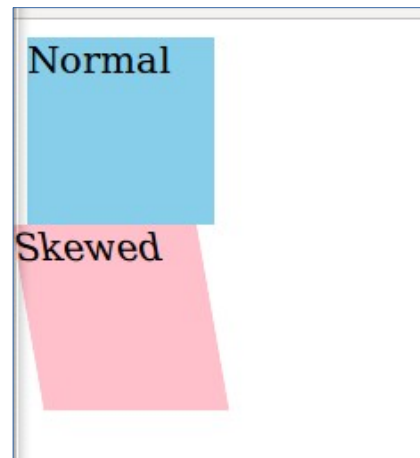
transform: skew(ax,ay), skew(ax) (ay=0),  
skewX(ax), skewY(ay)

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);  
}
```

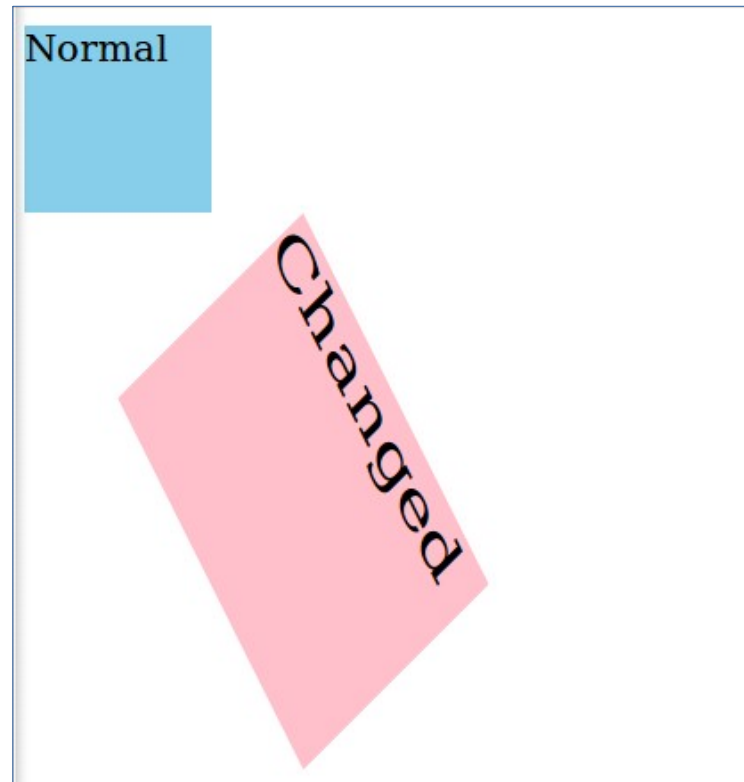
În exemplul 2 se va face doar ultima transformare  
**transform: scale(2).**

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)

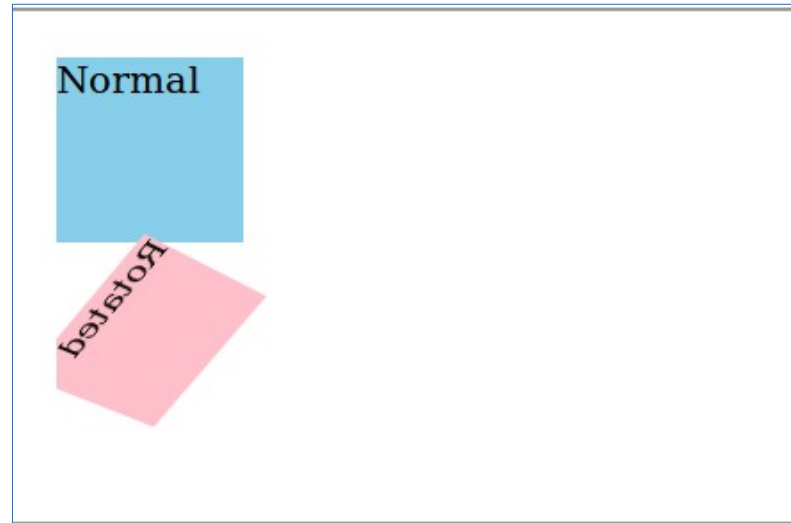
matrix3d(n,n,n,.....,n)  
16 valori

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

Proprietatile necesare pentru a defini o tranzitie

```
transition-property: proprietatea CSS care va fi schimbata;  
transition-duration: 2s (durata tranzitiei);  
transition-timing-function: linear; (functii predefinite)  
/* ease, ease-in, ease-out, step-start, step-end */  
transition-delay: 1s; (durata de asteptare înainte de inceperea  
tranzitiei)
```

Se poate folosi în varianta prescurtata:

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

Proprietati carora li se poate aplica o tranzitie

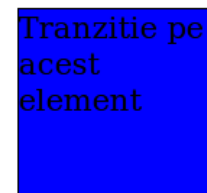
[https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\\_animated\\_properties](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_animated_properties)

## Exemplu (ex1-tranzitie.html din exemple-curs4)

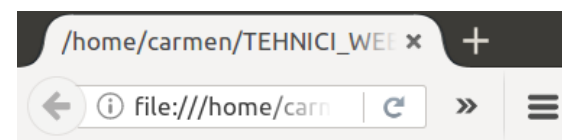
```
<style>
div {
  border-style: solid;
  border-width: 1px;
  display: block;
  width: 100px;
  height: 100px;
  background-color: #0000FF;
  transition: width 2s, height 2s, background-color
            2s, transform 2s;
}
div:hover {
  background-color: #FFCCCC;
  width: 200px;
  height: 200px;
  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>
```



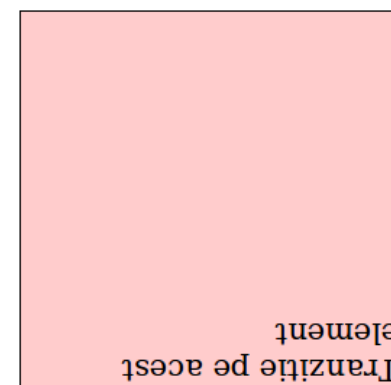
### Tranzitie CSS



La hover pe div incepe tranzitia



### Tranzitie CSS



La hover pe div incepe tranzitia

# CSS-animatii

```
@keyframes nume-animatie{  
  0% {background-color: red;}  
  25% {background-color: green;} //pas intermediar  
  .....  
  75% {background-color: blue; }  
  100% {background-color: yellow;}  
}
```

```
@keyframes nume-animatie{  
  from {background-color:red;} //echivalent cu 0%  
  
  to {background-color: yellow;} //echivalent cu 100%  
}
```

# CSS-animatii

Exemplu (ex2-animatie.html din exemple-curs4)

```
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-fill-mode: none /* forwards /  
backwards / both */
```

```
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;}  
}
```

```
animation: nume-animatie 5s linear 2s infinit alternate;  
// scurtatura pentru proprietatile de mai sus
```

# Exemple: CSS-tranzitii și animatii

Tranzitie: Daca se ajunge cu mouse-ul pe un li dintr-o sublista, daca li-ul este pe o pozitie para, isi schimba culoarea de background treptat pe parcursul a doua secunde de la transparent la verde (si invers, cand se ia cursorul de pe li)

```
li li:nth-child(even) {  
    transition: background-color 2s;  
}  
  
li li:nth-child(even):hover {  
    background-color:green;  
}
```

Animatie: Ultimul element al divului cu id-ul „parinte” să-și schimbe opacitatea de la 1 la 0.5 in timp de 4s, aceasta repetandu-se la infinit

```
div#parinte > :last-child {  
    animation-name: myanimation;  
    animation-duration: 4s;  
    animation-iteration-count:infinite;  
}
```

```
@keyframes myanimation {  
    from {opacity: 1;}  
    to {opacity: 0.5;}  
}
```

# Ce înseamnă site responsive?

- Adaptarea dimensiunii și a layoutului paginii web la dispozitivele utilizatorilor (ecran de telefon, tableta, calculator).
- Exista doua metode:
  - Adaptive Mobile Friendly (Exemplu: m.digi24.ro)
  - Responsive Web Design

<http://torquemag.io/responsive-design-vs-adaptive-design/>



# Pagini web responsive

Ethan Marcotte (2010)

<http://alistapart.com/article/responsive-web-design>

## Avantaje:

- O singură versiune de site care funcționează bine pe orice dispozitiv.
- Posibilitatea de sharing a aceluiași URL între diverse dispozitive.
- Layout-ul se adaptează la browser, eliminând astfel scroll-ul orizontal.

# Media queries

- **Media query** are doua componente:
  - **media type**: screen, print, speech, all.
  - **media features**: width, height, max-height, max-width, orientation: portrait, landscape, resolution, aspect-ratio, etc.

## Media-query -sintaxa

media type and (med-feature1:value) and (med-feature2:value)...

# Media query - HTML

- HTML conține atributul **media**
  - Permite folosirea unor stiluri CSS diferite în funcție de proprietatile dispozitivului media care prelucrează pagina (de ex. stiluri diferite pentru printare și afisare).

```
<head>  
  
<link rel="stylesheet" type="text/css" media="screen and (max-width: 600px)"  
  href="mic.css">  
  
<link rel="stylesheet" type="text/css" media="print" href="print.css">  
</head>
```

# Media query - CSS

- CSS at-rule `@media`
  - `@media media-query1, media-query2 { }`

Exemplu:

```
@media screen and (max-width: 960px) and  
(min-width: 500px)  
{  
  body { background-color: pink; }  
}
```

## Operatorii: **and**, **only**, **not** și **", "**

ordinea aplicarii operatorilor este: and, not, ", "

Operatorul **and** cere ca ambele conditii pe care le leaga sa fie indeplinite.

```
@media screen and (min-width:250px) and (max-width:700px)
{ body{ background-color:blue; } }
```

Operatorul **only** e folosit mai mult pentru browsere mai vechi care nu-l recunosc si astfel css-ul din media query e ignorat.

**@media only screen** este echivalent cu **@media screen**

Operatorul **not** neaga conditia data

```
@media not all and (min-width:350px) and (max-width:550px)
{body{background-color:blue;}}
```

Operatorul **", "** are rol de "sau"

```
@media (min-width:500px) , (max-width:300px) {body{background-
color:blue;}}
```

# CSS-media query

Permite definirea unui cod css care se va aplica doar in anumite conditii specificate de query

Exercitiu:

La latimea paginii sub 500px, divurile continute în elementul cu id-ul „container” trebuie sa se aseze unele sub altele si sa aiba latimea egala cu jumatate din latimea viewportului (nu a containerului).

De asemenea divul cu id-ul „d3” nu trebuie sa se mai afiseze.

# CSS-media query (media\_query.html din exemple-curs4)

## CSS:

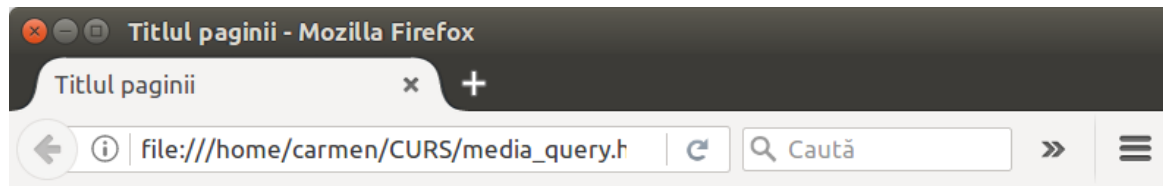
```
#continut {background-color: grey;}
#menu {background-color : pink;}
```

```
@media screen and (min-width: 768px) {
#menu {width: 100%;}
#menu li {display: inline; margin-left: 20px;
margin-right: 20px;}
#continut {width: 100%;}
}
```

```
@media screen and (max-width: 768px) {
#menu {width: 25%; float: left;}
#menu ul {list-style-type: none;}
#menu li {margin-top: 10px; margin-bottom:
10px;}
#continut {width: 70%; float: right; padding-left:
5px;}

}
```

```
<body>
<nav id="menu">
  <ul>
    <li>HTML</li>
    <li>CSS</li>
    <li>JavaScript</li>
  </ul>
</nav>
<div id="continut">
  Sed ut perspiciatis unde omnis iste
  natus error sit voluptatem accusantium
  doloremque laudantium, totam rem
  aperiam, eaque ipsa quae ab
  .....
</div>
</body>
```

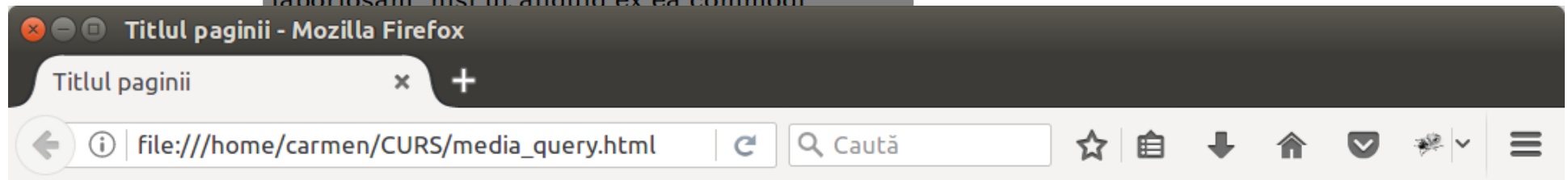


HTML

CSS

JavaScript

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi



HTML

CSS

JavaScript

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?



# Setarea viewport-ului

```
<head>
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

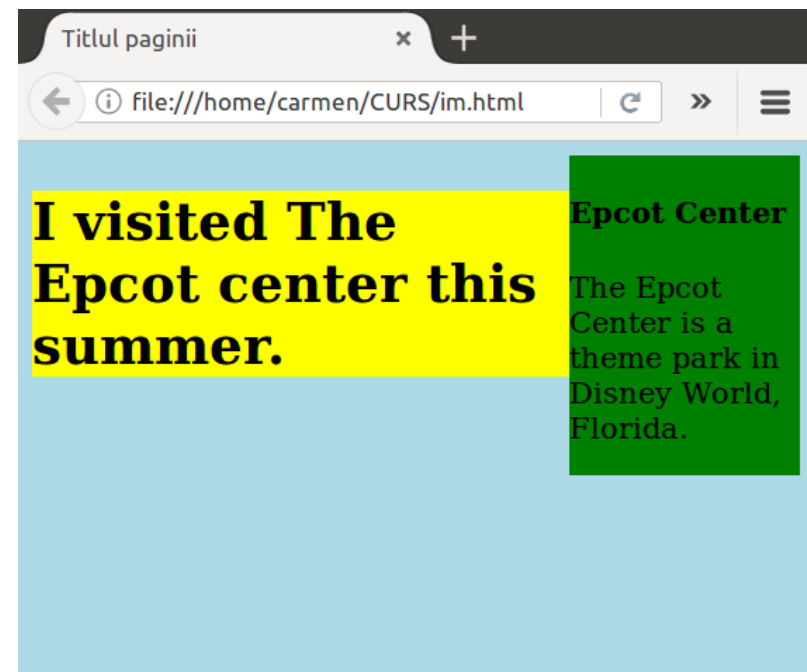
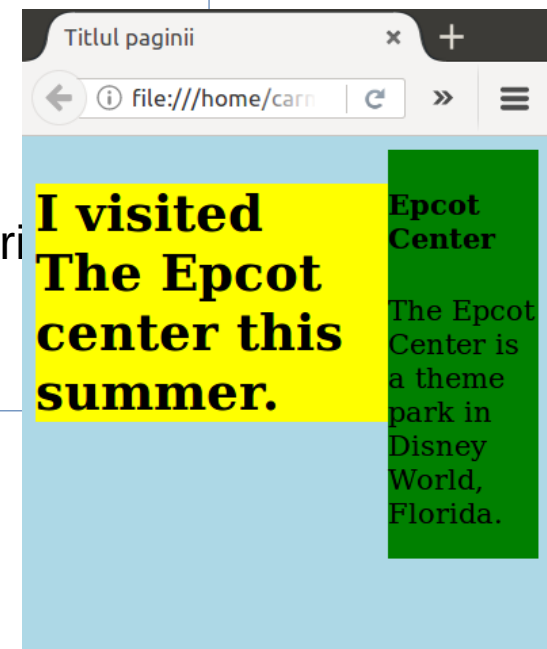
```
</head>
```

Acesta indică browser-ului că pagina trebuie să fie scalată pentru a se potrivi pe dimensiunea ecranului.

# Unități de măsură relative: procente, ems

```
<body>
<h1>I visited The Epcot center this summer.</h1>
<aside>
  <h4>Epcot Center</h4>
  <p>The Epcot Center is a theme park in Disney World, Florida.</p>
</aside>
</body>
```

```
body {background-color: lightblue;
      font-size: 0.5em;
}
h1 {
  width: 70%;
  background-color: yellow;
  float: left;
  font-size: 3.5em;
}
aside {
  width: 30%;
  background-color: green;
  float: left;
  font-size: 2em;
}
```



# Text responsive

## Folosind media queries

```
@media screen and (min-width:768px){  
  body{font-size:20px;}  
}
```

```
@media screen and (max-width:768px){  
  body{font-size:12px;}  
}
```

```
h1 {font-size: 3em;}
```

```
h2 {font-size: 1.2em;}
```

```
p {font-size: 0.87em;}
```

```
<body>
```

```
<h1>This is heading 1</h1>
```

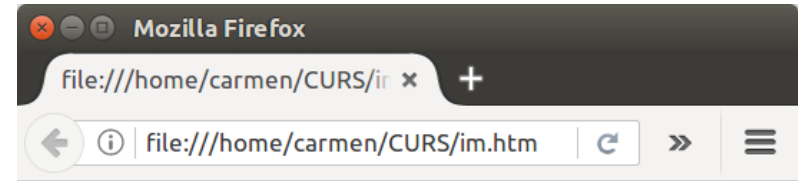
```
<h2>This is heading 2</h2>
```

```
<p>This is a paragraph.</p>
```

```
<p>Specifying the font-size in  
em allows all major browsers  
the text.
```

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

```
</body>
```

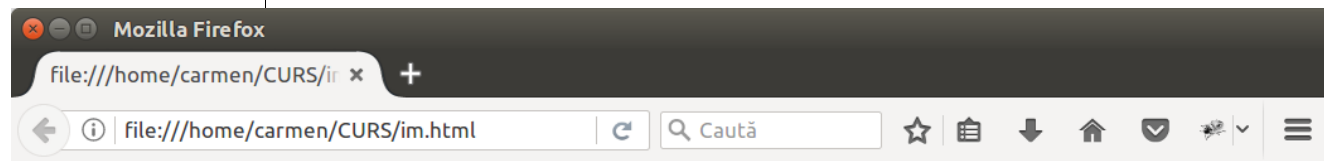


## This is heading 1

### This is heading 2

This is a paragraph.

Specifying the font-size in em allows all major browsers to resize the text. Unfortunately, there is still a problem with older versions of IE. When resizing the text, it becomes larger/smaller than it should.



## This is heading 1

### This is heading 2

This is a paragraph.

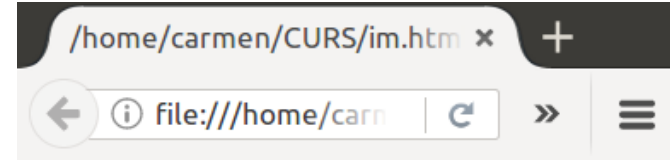
Specifying the font-size in em allows all major browsers to resize the text. Unfortunately, there is still a problem with older versions of IE. When resizing the text, it becomes larger/smaller than it should.

# Text responsive

Folosind unități de măsură ale viewportului: vw, vh, vmin, vmax

```
h1 {font-size: 5vw;}  
h2 {font-size: 4vw;}  
p  {font-size: 2vw;}
```

```
<body>  
<h1>This is heading 1</h1>  
<h2>This is heading 2</h2>  
<p>This is a paragraph.</p>  
<p>Specifying the font-size în  
em allows all major browsers to  
resize the text.  
.....  
</p>  
  
</body>
```

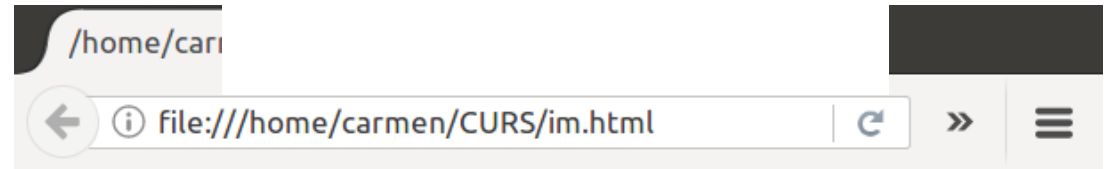


**This is heading 1**

**This is heading 2**

This is a paragraph.

Specifying the font-size in em allows all major browsers to resize the text. Unfortunately, there is still a problem with older versions of IE. When resizing the text, it becomes larger/smaller than it should.



**This is heading 1**

**This is heading 2**

This is a paragraph.

Specifying the font-size in em allows all major browsers to resize the text. Unfortunately, there is still a problem with older versions of IE. When resizing the text, it becomes larger/smaller than it should.

# Imagini responsive

Elementele img, video, iframe nu se dimensioneaza în HTML.

Exemplu:

```
#container{
    width:80%;
    border: 2px solid red;
    float: left;
}
```

```
<html>
<div id="container">



<p>Croaziere in Marea Mediterana,
Nordul Europei, Caraibe, America de
Sud, Oceanul Indian, Asia.
Planuiesti o coaziera de vis?
</p>

</div>
</html>
```



# Imagini responsive

## Rezolvare

```
img{  
  width:100%;  
  height:auto;  
}
```



Croaziere in Marea Mediterana, Nordul Europei, Caraibe, America de Sud, Oceanul Indian, Asia. Planuiesti o coaziera de vis?

```
img{  
  max-width:100%;  
  height:auto;  
}
```



Croaziere in Marea Mediterana, Nordul Europei, Caraibe, America de Sud, Oceanul Indian, Asia. Planuiesti o coaziera de vis?

# Imagini responsive

Pentru dimensiuni diferite se vor afisa imagini diferite;  
se vor folosi elementele html <picture> și <source>

```
<picture>  
  <source srcset="mica.jpg" media="(max-width:500px)"/>  
  <source srcset="mijlocie.jpg" media="(max-width:1000px)"/>  
    
</picture>
```

[https://developer.mozilla.org/en-US/docs/Learn/HTML/Multimedia\\_and\\_embedding/Responsive\\_images](https://developer.mozilla.org/en-US/docs/Learn/HTML/Multimedia_and_embedding/Responsive_images)

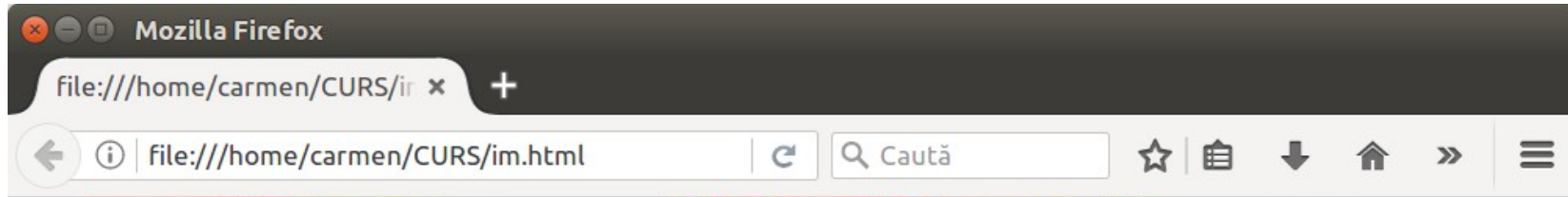
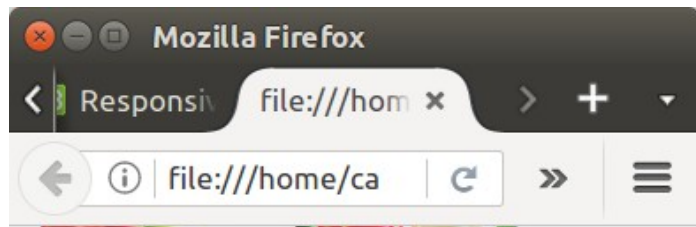
# Imagini de background responsive

```
background-image:url (adresa-imagine);  
background-size:cover;
```

→ imaginea acopera intregul container

```
body {margin-left:auto; margin-right:auto; width:80%;  
      background-image:url(Flori.jpg);  
      background-position: center center;  
      background-repeat: no-repeat;  
      background-size:cover;}
```

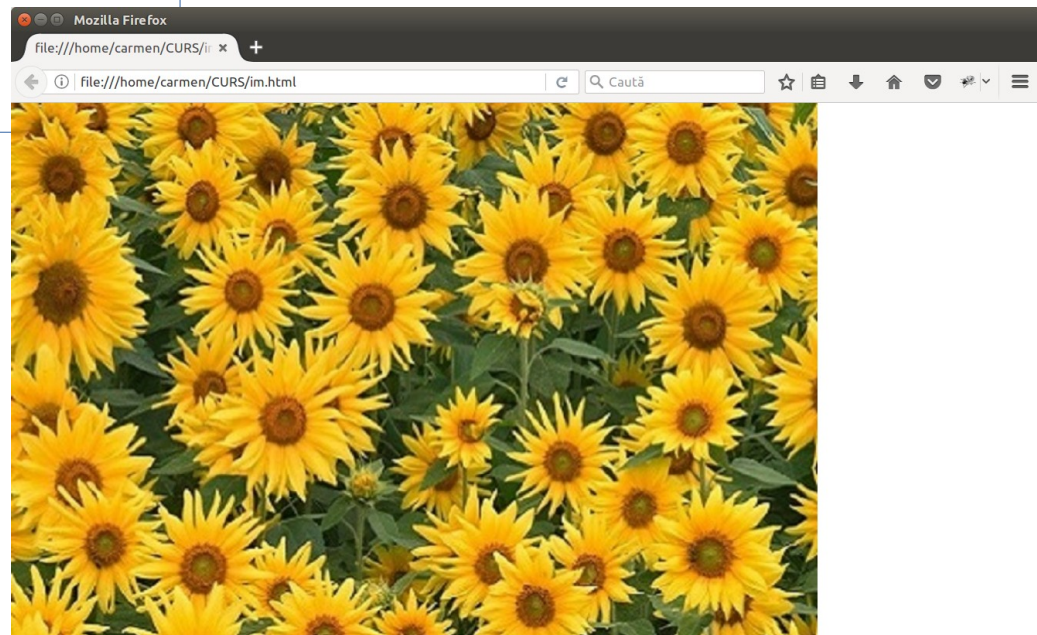
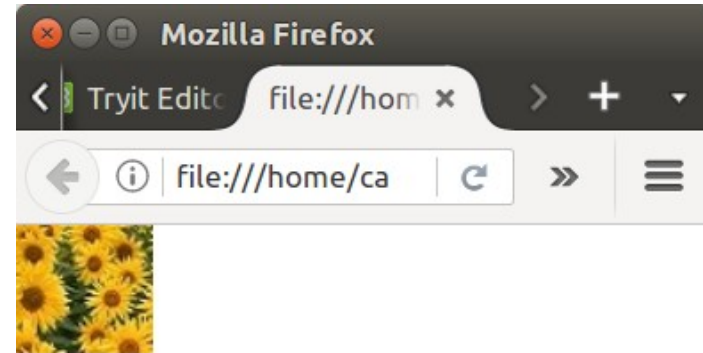




# Exemplu

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

```
@media screen and (min-width: 400px)  
{  
    body {  
        background-image:  
            url(Vara-mare.jpg);  
    }  
}
```

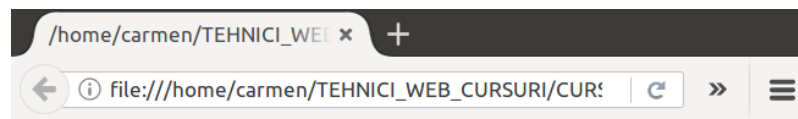


# CSS-aliniere pe orizontala: centrare

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 centrat
<p>Paragraf 1</p>
<p>Paragraf 2</p>
<p>Paragraf 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

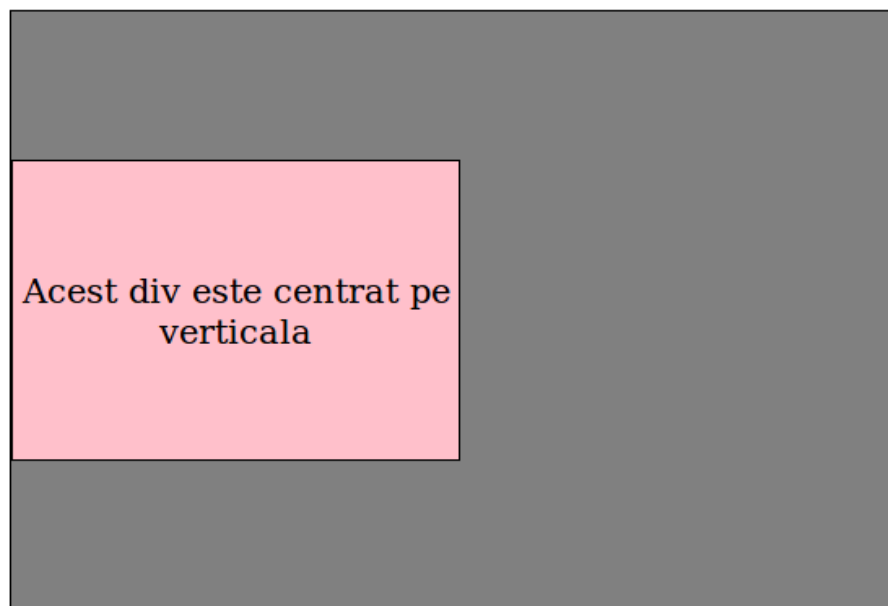


## CSS-centrare pe verticala

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





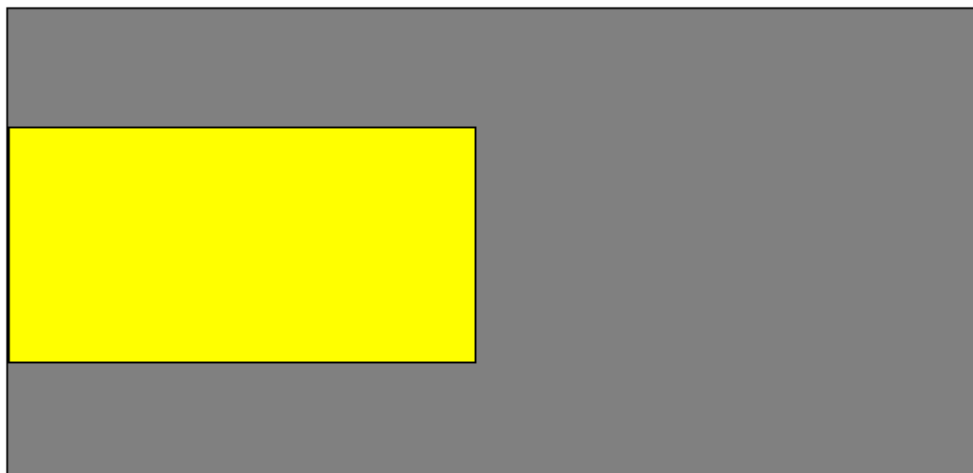
## CSS- aliniere pe verticala

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





## CSS-layout : multicolorane

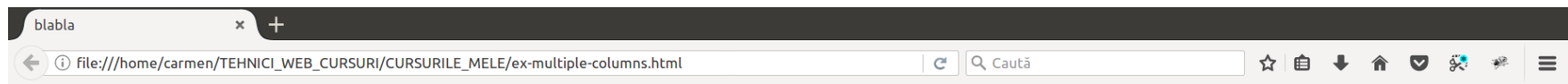
Permite aranjarea textului pe mai multe coloane.

`column-count` : nr.de coloane  
`column-gap`: spațiul dintre coloane  
`column-rule-style`: stilul liniei care desparte coloanele  
`column-rule-width`: grosimea liniei dintre coloane  
`column-rule-color`: culoarea liniei dintre coloane  
`column-rule`: grosime stil culoare  
`column-span`: cate coloane ocupa un element  
`column-width`: latimea unei coloane

`column-count` si `column-width` se determina una pe cealalta

```
#col1{ column-width: 100px;
        column-gap: 6px;
        column-rule: 3px solid red;
}
#col2{ column-count: 4;
        column-gap: 10px;
        column-rule: 4px dotted black;
```

```
<body>
<h1>multiple colums</h1>
<p id="col1">Lorem ipsum dolor
sit amet,....</p>
<p id="col2">Aliquam lectus
odio,.....<p>
</body>
```



## multiple columns

<p>             Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam tempus rhoncus ante tincidunt commodo. In in nibh           </p>	<p>             vitae enim interdum lobortis. Duis euismod condimentum lacus, eget hendrerit nisi semper vel. Proin tincidunt tincidunt suscipit           </p>	<p>             rhoncus. Nam at iaculis dolor. Proin at facilisis leo, non ullamcorper urna. Mauris auctor sagittis nibh, ac           </p>	<p>             viverra nisl lobortis ut. Morbi et arcu eget purus tristique Duis ac urna leo. Maecenas sagittis elementum enim ac           </p>	<p>             suscipit porttitor. Sed turpis nunc, luctus id interdum nec, suscipit sed lectus. Nam sagittis elementum enim           </p>	<p>             imperdiet ullamcorper. Fusce laoreet fermentum dui in tempor. Nam rhoncus, tellus sed rutrum porta, nisi           </p>	<p>             leo euismod ante, et luctus elit sem eu mi. Duis adipiscing nibh sit amet egestas dapibus. Fusce id quam           </p>	<p>             placerat ligula. Ut imperdiet, libero in iaculis egestas, eros ante pharetra lacus, a cursus nibh lacus et elit. In eget orci           </p>	<p>             id metus fermentum lacinia. Sed in blandit est, at vehicula tellus. Integer vulputate elit in ligula gravida bibendum.           </p>	<p>             Curabitur venenatis metus in vulputate iaculis. Quisque augue nisi, bibendum vitae nulla eget, egestas facilisis           </p>	<p>             sem. Morbi malesuada, dui vel molestie congue, sem dolor diam, eget iaculis mi leo quis arcu. Cras quis massa           </p>	<p>             neque. Donec molestie felis vel mi mollis, nec suscipit risus malesuada.           </p>
---	---	---	---	--	---	---	--	---	---	--	---

<p>             Aliquam lectus odio, dignissim nec auctor iaculis, tristique sit amet dolor. Donec a velit consectetur, tincidunt nisl non, luctus ligula. Interdum et malesuada fames ac ante ipsum primis in faucibus. Integer consectetur justo porttitor urna suscipit, a bibendum velit semper. Vestibulum fringilla fringilla odio, id sagittis justo iaculis ac. Nunc id faucibus velit, eu pretium nisi. Pellentesque facilisis dictum augue sit amet sodales. Nunc sed ornare diam, quis blandit nulla. In ut arcu elit. Fusce molestie sagittis ipsum in ultrices. Nunc sapien nibh, malesuada eget malesuada           </p>	<p>             :ut, elementum at nibh. Ut sodales est ut :elit tincidunt congue. Fusce tincidunt :nunc ut elit sollicitudin interdum ut :euismod elit. Nulla molestie tincidunt :lectus et viverra. Cras arcu risus, :faucibus et tellus in, condimentum :tincidunt risus. Vivamus imperdiet odio :et nisl mattis eleifend sed sagittis ipsum. :Morbi arcu dolor, ultricies eu orci ac, :pharetra vulputate enim. Mauris cursus :tempor rutrum. In hac habitasse platea :dictumst. In ut placerat felis, ac tempor :erat. Curabitur a leo eget mi semper :consequat sed sit amet diam. Proin :pulvinar fringilla est, ut gravida leo           </p>	<p>             :porttitor vel. In non eros a sapien :adipiscing scelerisque. Maecenas nisl :leo, scelerisque nec orci suscipit, :tristique pharetra ligula. Sed et eleifend :felis. Class aptent taciti sociosqu ad :litora torquent per conubia nostra, per :inceptos himenaeos. Integer sit amet :lectus sit amet leo blandit malesuada :scelerisque eu eros. In urna turpis, :vehicula eu orci semper, dapibus :rhoncus neque. Mauris non tortor enim. :Suspendisse hendrerit purus non ipsum :feugiat, quis fermentum dui ultricies. :Aliquam eu sapien elit ullamcorper :venenatis. Vestibulum tincidunt odio           </p>	<p>             :vitae quam condimentum, sit amet :fringilla enim vehicula. Nulla :ullamcorper mauris nec tellus rutrum, :quis facilisis urna pulvinar. Nulla urna :turpis, consequat eget posuere quis, :dictum sed nibh. Sed tristique eros vitae :facilisis pulvinar. Maecenas est risus, :posuere in arcu et, auctor sagittis metus. :Sed augue leo, placerat quis leo in, :semper vehicula dolor. Nam sit amet :consectetur velit, eget tempor purus. :Cras urna purus, feugiat ut pharetra in, :tempus ac augue. Integer libero risus, :venenatis eu sodales at, cursus pharetra :ligula.           </p>
--	--	--	--

## CSS3- gradienti

Gradient : tranzitie de la o culoare la alta

### Linear gradient

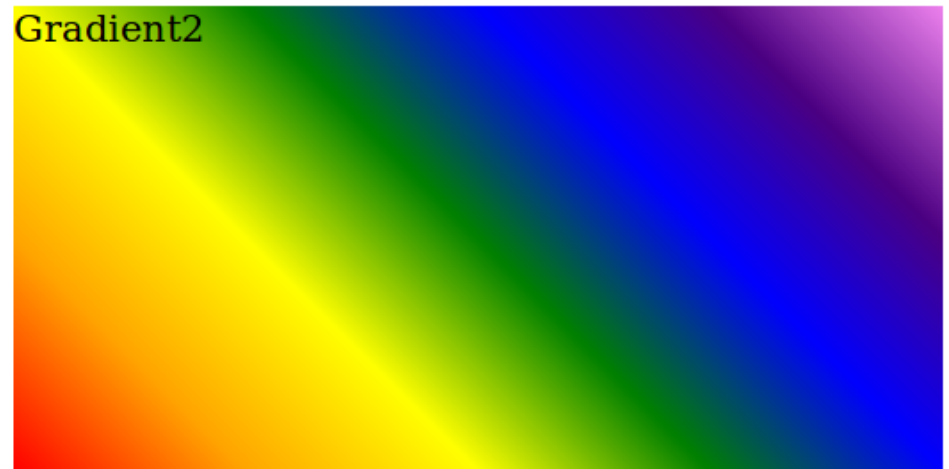
```
background: linear-gradient(directia, culoare1,culoare 2,...)
```

```
directia : to bottom (implicit) /*  
           to right ,  
           to left,  
           to top,  
           to bottom right,  
           unghi( 45deg) */
```

```
background: linear-gradient(to right, black, white);  
background: linear-gradient(45deg, black, white);  
background: repeating-linear-gradient(red, yellow 10%, green 20%);
```

```
div{ width: 400px; height:200px;}
#grad1 {
  background: linear-gradient(to right
bottom, red, orange, yellow, green,
blue, indigo, violet);
}
#grad2 {
  background: linear-gradient(45deg,
red, orange, yellow, green, blue, indigo,
violet);
}
#grad3 {
  background: repeating-linear-
gradient(red, orange 5%, green 10%);
}
```

```
<body>
<div id="grad1">
Gradient1 </div>
<hr>
<div id="grad2">
Gradient2 </div>
<hr>
<div id="grad3">
Gradient3 </div>
</body>
```



## Radial gradient

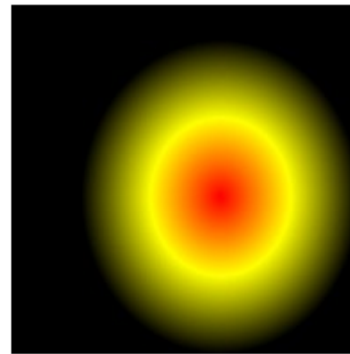
background: radial-gradient (shape size at position, start-color, ..., last-color)

shape : ellipse (implicit) /\* circle\*/

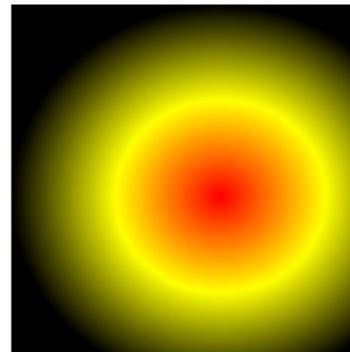
size : closest-side,  
farthest-side,  
closest-corner,  
farthest-corner (implicit)

position: center (implicit)  
10%,60%

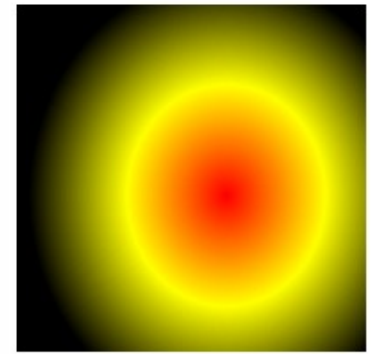
**closest-side:**



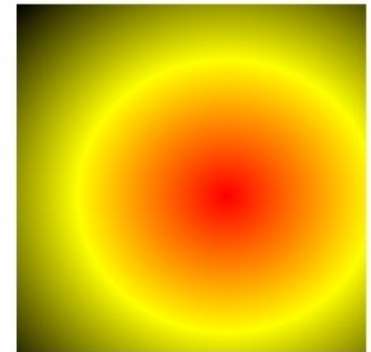
**farthest-side:**



**closest-corner:**



**farthest-corner (t**



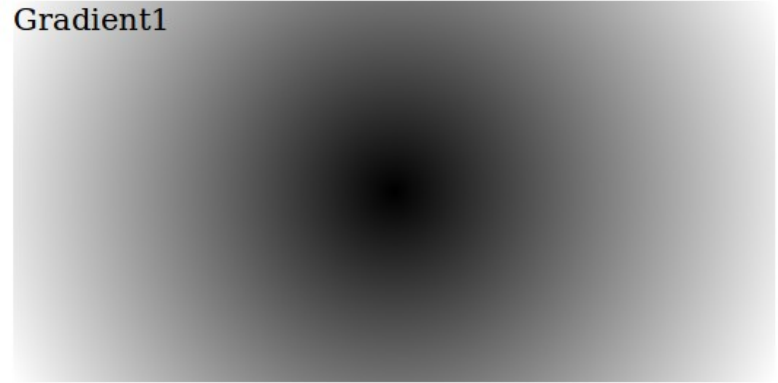
```
#gr1 {background:  
radial-gradient (circle at 50% 50%,  
black, white);}
```

```
#gr2 {  
background: radial-gradient( ellipse,  
red, blue, aqua,white);}
```

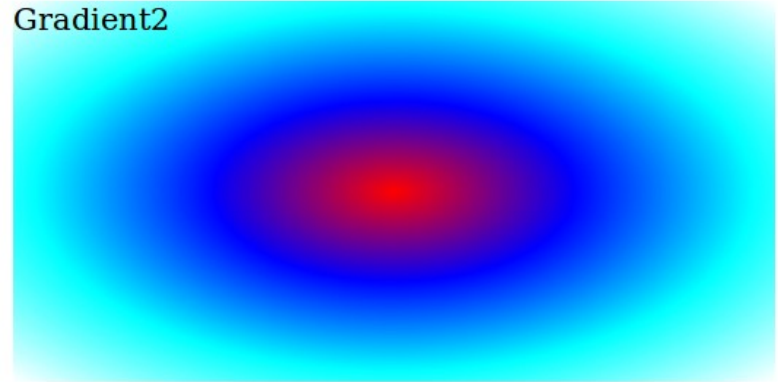
```
#gr3{ background: repeating-radial-  
gradient(red, yellow 10%, green 15%);}
```

```
<body>  
<div id="gr1">  
Gradient1 </div>  
<hr>  
<div id="gr2">  
Gradient2 </div>  
<hr>  
<div id="gr3">  
Gradient3 </div>  
</body>
```

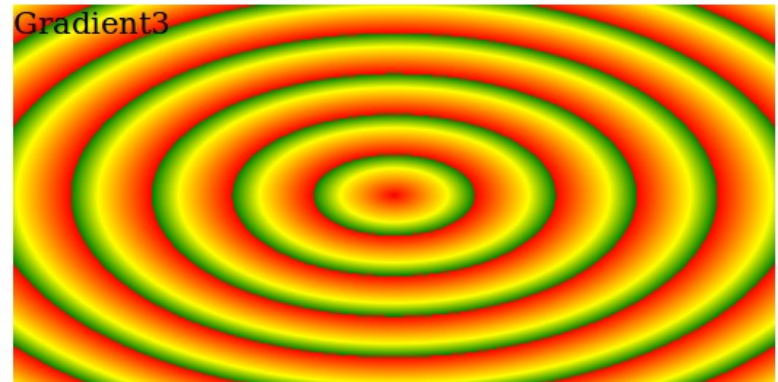
Gradient1



Gradient2



Gradient3



# CSS3: border-radius (border cu colturi rotunjite)

**border-radius:** reuniune a proprietatilor  
**border-top-left-radius,**  
**border-top-right-radius,**  
**border-bottom-right-radius,**  
**border-bottom-left-radius.**

Patru valori: 15px 50px 30px 5px:



Trei valori: 15px 50px 30px:



Doua valori: 15px 50px:



O valoare: 30px:

