Codurile CSS pot fi scrise in interiorul paginii sau intr-un fisier extern cu extensia".css".  
Codul CSS, ca forma generala, este alcatuit din: obiectul care va fi formatat. proprietatile acestuia si valoarea (sau atributele) fiecarei proprietati.  
Cand este adaugat in documentul HTML, trebuie scris in cadrul unui tag **<style>** in sectiunea HEAD a documentului HTML, dupa cum se vede in formula generala de mai jos;

<html>

<head>

<title>Titlul</title>

<style>

**.element {**

**proprietate:valoare;**

**alta\_proprietate:valoare;**

**}**

</style>

</head>

<body>

... Continut ...

</body>

</html>

- Observati sintaxa codului CSS. Perechile **proprietate:valoare** se scriu intre acoladele selector-ului CSS pe care-l definesc (aici .element), intre "proprietate" si "valoare" trebuie sa fie un caracter doua-puncte (:) iar la sfarsitul perechii se adauga un caracter punct-si-virgula (**;**)  
- Cand este adaugat intr-un fisier extern ".css", codul CSS se scrie la fel, dar nu se mai adauga tag-ul <style>.

**Obiectele (regulile) CSS**

Exista trei tipuri principale de obiecte CSS: selector, clasa si identificator.

**Selectorul HTML**

Selectorul HTML se foloseste pentru a redefini modul de afisare a continutului etichetei HTML.  
Un selector HTML reprezinta tipul etichetei HTML, numele tag-ului.  
Definirea unui selector HTML in CSS are ca rezultat redefinirea stilului acelui tag HTML.  
Iata un exemplu:

<!DOCTYPE html>

<html>

<head>

<title>Titlul</title>

<style>

p {

color:#0088fe;

font-size:16px;

}

</style>

</head>

<body>

<p>Un text oarecare in paragraf.</p>

<p>Alt paragraf. Tutorial css.</p>

</body>

</html>

Incercati codul

- Aici selectorul este "p" (paragraf).

**Clasa**

Clasa este un obiect care poate fi aplicat oricarei etichete HTML.  
O clasa trebuie creata in interiorul etichetei HTML inainte de a fi definita intr-un cod CSS. Crearea clasei in interiorul etichetei se face simplu, prin specificarea cuvantului **class** si numele clasei, ca in exemplu de mai jos:

**<h3 class="nume\_clasa"> Text </h3>**

- "nume\_clasa", poate fi orice nume dorim.  
- Apoi in interiorul codului CSS, clasa trebuie definita prin adaugarea unui caracter punct (.) inaintea numelui clasei:

**.nume\_clasa { color:#0088fe; font-size:18px; }**

Numele aceleasi clase poate fi atribuit mai multor etichete HTML din aceeasi pagina, si toate vor prelua acelasi stil css.  
- Exemplu:

<!DOCTYPE html>

<html>

<head>

<title>Titlul</title>

<style>

.cls\_red {

color:#ee0000;

font-size:16px;

}

</style>

</head>

<body>

<p class='cls\_red'>Un text oarecare in paragraf cu clasa 'cls\_red'.</p>

<p>Alt paragraf, fara class.</p>

<div class='cls\_red'>Un Div cu aceeasi clasa: cls\_red.</div>

</body>

</html>

Incercati codul

**Identificator**

Identificatorul (ID) este asemanator cu clasa. Poate fi aplicat oricarei etichete HTML, dar spre deosebire de clase, numele unui ID trebuie sa fie unic (dat unei singure etichete HTML dintr-o pagina), pentru alta eticheta se adauga un ID cu alt nume, diferit.  
Ca si clasa, ID-ul trebuie intai creat in interiorul etichetei HTML. Modul de creare este simplu, prin specificarea cuvantului **id** si numele, ca in exemplu de mai jos:

**<div id="nume\_id"> Text </div>**

- "nume\_id", poate fi orice nume dorim.  
In interiorul codului CSS, identificatorul este definit prin adaugarea unui caracter diez (#) inaintea numelui:

**#nume\_id { color:blue; font-size:1px; }**

- Exemplu:

<!DOCTYPE html>

<html>

<head>

<title>Titlul</title>

<style>

#dv\_id {

color:#1234ee;

text-align:center;

}

</style>

</head>

<body>

<div id='dv\_id'>Div cu ID: dv\_id.</div>

<div>Alt Div, fara ID.</div>

</body>

</html>

Incercati codul

**Componentele unui obiect CSS**

Obiectele CSS, indiferent de tipul lor, au in componenta urmatoarele elemente:

* **Selectorii** - identifica un obiect; pot fi selectori de etichete HTML, clase sau identificatori.
* **Proprietatile** - identifica o proprietate de stil a obiectului.
* **Valorile** - definesc valoarea unei proprietati; pot fi cuvinte cheie, valori numerice sau procentuale, tipul valorii depinzand de proprietate.

Sintaxa generala a unei reguli CSS este urmatoarea:

* **selector {proprietate: valoare; }**

Proprietatile si valorile constituie definitia elementului CSS. Acestea sunt perechi (nume:valoare;), despartite de caracterul doua puncte ":", si terminate cu un caracter punct si virgula ";".

# Creare Foi de Stil

## Curs Css

[**Home**](https://marplo.net/)[**Engleza**](https://marplo.net/engleza)[**Spaniola**](https://marplo.net/spaniola)[**Html**](https://marplo.net/html)[**JavaScript**](https://marplo.net/javascript)[**PHP-MySQL**](https://marplo.net/php-mysql)[**Ajax**](https://marplo.net/ajax)[**Blog**](https://marplo.net/blog)[**Forum**](https://marplo.net/forum/)[**Games**](https://gamv.eu/)

Top of Form



Bottom of Form

### Etichete HTML si stiluri CSS

CSS ofera posibilitatea de a schimba aspectul fiecarei etichete in parte, prin stabilirea unui anume stil in interiorul ei, cu atributul: **style**. Acest lucru este util mai ales pentru a anula alte stiluri ale elementului respectiv sau de a da elemente grafice de stil doar etichetei respective.  
Sintaxa pentru definirea stilurilor in interiorul unei etichete HTML este urmatoarea:

**<eticheta style="proprietate:valoare;"> text ... </eticheta>**

CSS permite si definirea unor reguli de stil generale intr-o pagina web. Acest set de reguli trebuie scris in sectiunea ***<head>*** a paginii, in cadrul tag-ului <style>.  
Sintaxa pentru definire CSS in interiorul etichetei <head> </head>, este urmatoarea:

**<style>**

**selector\_1 {proprietate1:valoare1; proprietate2:valoare2; ... }**

**...**

**selector\_n {proprietate1:valoare1; proprietate2:valoare2; ... }**

**</style>**

- Definirea tuturor stilurilor intr-o locatie comuna usureaza modificarea mai rapida a unei pagini.  
Iata un exemplu practic de cod css:

<!DOCTYPE html>

<html>

<head>

<title>Titlul</title>

<style>

h1 {

color:#1111ff;

font-size:21px;

text-decoration:underline;

}

p {

font-family:Calibri;

font-size:16px;

color:#00be00;

}

</style>

</head>

<body>

<h1>Titlu paginii</h1>

<p>Paragraf, cu stilul css definit in tagul STYLE;.</p>

<p>Alt paragraf, preia acelasi stil.</p>

</body>

</html>

Incercati codul

Foile de stil pot fi utilizate nu doar la nivel de pagina web, ci si la nivel de intreg site. Astfel, trebuie creat un fisier cu extensia "***.css***", care va contine codul CSS si poate fi inclus in paginile HTML prin doua procedee: legatura sau import.

*Crearea unei foaie de stil externe se face scriind codul CSS intr-un fisier text cu ajutorul unui editor de texte simplu (de exemplu Notepad++). In fisierul extern creat se adauga reguli CSS, fara insa ca aceste reguli sa fie incadrate in etichete STYLE.*

Dupa ce a fost creat fisierul ".css", poate fi folosit in paginile din site utilizand urmatoarea sintaxa, in interiorul tag-ului ***<head> </head>***:

**<link rel="stylesheet" href="nume\_fisier.css" type="text/css">**

- Atributele folosite transmit navigatorului tipul de legatura ("rel" – legatura cu o foaie de stil, "type" - tipul codului din fisier), la "**href**" locatia fisierului ".css".  
  
O alta modalitate de utilizare a foilor de stil externe intr-un document HTML o reprezinta importul acestora folosind comanda **@import**.

**<style>**

**@import url(adresa/nume\_fisier.css);**

**</style>**

- "adresa/nume\_fisier.css" reprezinta calea si numele fisierului ce contine regulile CSS definite.  
Alaturi de instructiunea "@import", in cadrul etichetei STYLE, pot fi adaugate definiti si selectori suplimentari.  
Legatura la un fisier CSS extern sau importul acestuia intr-un document HTML are acelasi efect ca si cum stilurile incluse ar fi fost definire direct in eticheta STYLE din sectiunea HEAD a paginii.

### Definirea selectorilor

- Selectorii HTML pot fi definiti prin adaugarea unui numar de definitii compatibilie cu eticheta HTML la care se refera, avand urmatoarea forma generala:

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

Dupa definirea selectorului HTML in CSS, stilurile etichetelor respective din documentul HTML vor fi modificate conform proprietatilor din CSS.  
  
Utilizarea selectorilor de tip clasa ofera posibilitatea configurarii unor stiluri care se pot aplica doar acelor elemente care au clasa respectiva. Sintaxa generala de definire a unei clase in CSS este:

**.nume\_clasa { proprietate1:valoare1; proprietate2:valoare2; ... }**

In codul CSS o clasa se poate asocia direct unui selector HTML. In acest caz, stilul definit va afecta doar elementele HTML specificate care au acea clasa. Pentru a defini o clasa care sa afecteze in mod direct un anume selector HTML, se foloseste urmatoarea sintaxa:

**tag\_Html.nume\_clasa { proprietate1:valoare1; proprietate2:valoare2; ... }**

- Asemanator cu selectorii de clasa se definesc si identificatorii (id-ul). Acestia sunt folositi pentru crearea de stiluri care pot fi aplicat unei singure etichete HTML dintr-o pagina.

**#id\_ul { proprietate1:valoare1; proprietate2:valoare2; ... }**

### Utilizare <span> si <div> cu CSS

- Elementul **<span>** nu are un stil propriu, reprezinta doar o locatie vida care creaza o eticheta in linie.  
Pentru a configura o eticheta <span> trebuie sa aibe o clasa (class) sau ID la care sa poata aplica proprietati in css.  
Iata un exemplu cu elemente <span> stilizate in css:

<!DOCTYPE html>

<html>

<head>

<title>Titlul</title>

<style>

.cls\_1 {

color:#ee0000;

font-weight:700;

}

#id\_sp {

color:#0000ee;

font-size:17px;

}

</style>

</head>

<body>

Un text cu <span class='cls\_1'>tag span cu class cls\_1</span> in cadrul unei linii.<br>

Alt text cu <span id='id\_sp'>tag span cu id id\_sp</span> in cadrul liniei.<br>

Alta linie de text cu <span class='cls\_1'>span, cu class cls\_1</span> in interior.

</body>

</html>

Incercati codul

#### CSS la element DIV

- Elementul **<div>** creaza o zona proprie in pagina, cu linie noua atat deasupra sa cat si dedesubtul lui, fara a avea un stil propriu.  
Elementele DIV nu se creaza in interiorul unui text, ci in loc liber, pentru a contine text si alt tip de continut (imagini, alte tag-uri) la care se vor aplica stilul CSS.  
Si la <div> se pot adauga atribute "class" si "id", la care in CSS se pot atribui proprietati de stil.

- Exemplu:

<!DOCTYPE html>

<html>

<head>

<title>Titlu</title>

<style>

.cntr {

border:1px solid blue;

text-align:center;

}

#id\_dv {

border:1px solid red;

color:#0000ee;

height:80px;

width:250px;

}

</style>

</head>

<body>

<div class='cntr'>Continut in Div cu class .cntr.</div>

<div id='id\_dv'>Cele bune sa se adune.</div>

<div class='cntr'>Intre doua nu te ploua, dar mai bine singur.</div>

</body>

</html>

Incercati codul

*Vedeti si tutorialul HTML ->*[*DIV si SPAN*](https://marplo.net/html/div_span.html)*.*

### Definirea de reguli similare

Daca mai multi selectori folosesc aceleasi definitii css, acestia pot avea aceeasi lista de proprietati, fiind scrisi separat prin virgule. Sintaxa generala pentru definirea unei liste cu mai multi selectori este urmatoarea:

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

In urmatorul exemplu se aplica aceleasi stiluri css la clasa "o\_clasa", la tagurile "h3" care au clasa "cls\_2" si la elementul cu id "un\_id":

.o\_clasa, h3.cls\_2, #un\_id {

background:#fcfcfe;

font-size:18px;

margin:5px auto;

}

### Sitluri la elemente imbricate

Cand un element Html este in interiorul altuia, spunem ca aceste etichete sunt imbricate. Elementul exterior se numeste, tag parinte, iar cel din interior, copil. Se pot crea reguli CSS care revin numai etichetelor copil, sintaxa generala a unei astfel de reguli fiind:

**selector\_Parinte selector\_Copil { proprietate:valoare; proprietate2:valoare2; ... }**

Lista de selectori imbricati poate fi mai mare de doi, ultimul selector din lista este cel care primeste toate stilurile incluse in regula si in plus le mosteneste si pe cele ale parintilor.  
Ca si selectorii singuri, selectorii imbricati pot include in lista clase, id\_uri sau selectori HTML.  
Toate etichetele HTML, cu exceptia etichetei BODY, au cel putin o eticheta parinte care le inconjoara. De cele mai multe ori stilurile etichetelor din interior preiau stilurile etichetelor parinte (exista insa cazuri in care proprietatile nu sunt mostenite de etichetele imbricate). Acest mecanism se numeste mostenirea stilurilor.  
Proprietatile mostenite pot fi anulate prin redefinirea acestora in lista de definitii a etichetei imbricate.

### Utilizarea instructiunii !important

Valoarea **!important** adaugata unei definitii CSS atribuie maximum de prioritate in determinarea ordinii unei executii.  
Valoarea **!important** trebuie plasata la urma, inaintea caracterului ’;’, ca in exempul urmator:

**selector { proprietate:valoare !important; }**

Exemplu, paragraful cu clasa "clr\_b" va avea culoarea blue, chiar daca in tagul lui "style" si la selectorul "p" e definita alta culoare.

<!DOCTYPE html>

<html>

<head>

<title>Titlul</title>

<style>

p { color:#00d000;}

.clr\_b {color:blue !important;}

</style>

</head>

<body>

<p class='clr\_b' style='color:red;'>Va avea culoarea de la valoarea cu !important.</p>

</body>

</html>

Incercati codul

### Determinarea ordinii executiei

Deoarece exista mai multe moduri de a aplica stilurile, pot apare situatii in care unei etichete sa-i fie aplicate mai multe stiluri din foi de stil (fisiere css) diferite. Exista cateva reguli care determina ordinea executiei (numita si cascada), acestea sunt:

* 1. Regulile CSS scrise in interiorul paginii HTML, in cadrul etichetei <style> din sectiunea "head" au o prioritate mai mare decat cele scrise intr-un fisier css extern, iar regulile scrise in interiorul etichetelor, la atributul "style" au o prioritate mai mare decat cele din tag-ul <style>.  
  2. Existenta valorii **!important** – confera prioritate maxima la afisare definitiei in care este utilizat.  
  3. **Momentul aparitiei** – cu cat un stil apare mai tarziu, cu atat importanta lui este mai mare. Astfel, definitiile unei etichete copil au prioritatea mai mare si anuleaza toate stilurile precedente cu care intra in conflict.

### Adaugarea comentariilor la CSS

Comentariile ajuta la intelegerea codului, facandu-se astfel cunoscut, pentru mai tarziu, rolu anumitor elemente din codul CSS.  
Comentariile nu au nici un efect si pot fi plasate in jurul regulilor.  
Pentru a adauga comentarii in regulile de stil avem la dispozitie doua modalitati:

* In cazul in care comentariul se intinde pe un singur rand este suficienta folosirea perechii de caractere ’**//**’ urmata de comentariu.
* In cazul in care comentariul contine mai multe linie, trebuie plasat intre "**/\***" care deschide zona de comentariu si "**\*/**" care indica sfarsitul comentariului.

Iata un exemplu in care puteti vedea cum pot fi adaugate comentariile:

/\* Comentariu,

pe mai

multe linii \*/

.nume\_clasa { proprietate1:valoare1; proprietate2:valoare2; ... }

// Comentariu pe o singura linie

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

# Configurare fonturi

## Curs Css

[**Home**](https://marplo.net/)[**Engleza**](https://marplo.net/engleza)[**Spaniola**](https://marplo.net/spaniola)[**Html**](https://marplo.net/html)[**JavaScript**](https://marplo.net/javascript)[**PHP-MySQL**](https://marplo.net/php-mysql)[**Ajax**](https://marplo.net/ajax)[**Blog**](https://marplo.net/blog)[**Forum**](https://marplo.net/forum/)[**Games**](https://gamv.eu/)

Top of Form



Bottom of Form

Exista cinci familii de fonturi de baza:

* **serif** – au un ornament (serif) plasat la terminatia literei, care ii ofera o distinctie speciala. Sunt folosite pentru tiparire, chiar daca textele sunt mai mari sau mai mici. Nu sunt indicate pentru afisarea textelor pe ecran.
* **sans serif** – sunt fonturi care nu folosesc serife, fiind indicate pentru continut text general.
* **monospace** – fonturile monospatiate pot avea serife, ele se deosebesc prin faptul ca fiecare litera ocupa aceeasi cantitate de spatiu. Sunt cele mai indicate pentru textele care trebuie citite cu exactitate, ca de exemplu liniile de program.
* **cursive** – imita scrisul de mana, intr-o maniera stilizata. Sunt indicate in scopuri decorative, nefiind recomandate pentru scrierea unor texte mai lungi.
* **fantasy** – nu se incadreaza in nici una dintre categoriile de mai sus, fiind fonturi care au un caracter predominant ornamental (reprezentand ilustratii sau pictograme).

### 1. Stabilirea fontului

Fontul folosit pentru afisarea unui text poate fi stabilit cu proprietatea **font-family** urmata de numele fontului sau a fonturilor (despartite prin virgula). Este bine ca numele fonturilor sa fie incadrate intre ghilimele simple sau duble, mai ales daca numele acestora contine spatii.  
Forma generala fiind urmatoarea:

**selector { font-family:"nume\_font1", "nume\_font2", ..., nume\_generic }**

- Dupa ultima virgula se pot folosi urmatoarele nume generice de fonturi: "serif", "sans-serif", "cursive", "monospace" sau "fantasy". Includerea unei asemenea valori este optionala.  
Iata un exemplu practic:

h1 { font-family:"Arial", "Helvetica", sans-serif }

Cand este specificata o lista de fonturi, navigatorul incearca sa foloseasca primul font din lista, daca nu este gasit parcurge lista pana in momentul in care intalneste un font instalat. Daca nu exista fonturi echivalente, textul va fi afisat cu fontul prestabilit de browser. Daca este specificat un nume generic, textul va fi afisat cu un font apartinand aceluiasi stil.

#### Utilizare fonturi personale

Exista posibilitatea folosirii unei palete largi de fonturi in crearea paginilor web, nu doar a celor din lista limitata a fonturilor compatibile cu navigatoarele. Solutia consta in inglobarea fontului in cadrul paginii si trimiterea lui in mod automat in browser-ul vizitatorului.  
Pentru a include un font intr-o pagina care sa fie transmis la vizitator, se foloseste intructiunea **@font-face**. Aceasta trebuie sa includa proprietatea **font-family**, cu numele fontului si proprietatea **src** cu locatia **url** a fontul respectiv.  
Exemplu:

@font-face {

font-family: 'Sansation\_Light';

src: url('Sansation\_Light.ttf');

}

/\* Dupa ce fontul e inclus, se poate folosi \*/

h1 {

font-family: 'Sansation\_Light', sans-serif;

}

Un font ca sa fie inclus trebuie sa respecte un anume format. Internet Explorer foloseste fonturi in format "eof" (obtinute cu ajutorul unui program numit WEFT), iar Mozilla, Chrome, Safari, si Opera support fonturi de tip OpenType Face (.otf) si TrueType Format (.ttf).

*Google Font Directory ofere donturi care pot fi utilizate in****@font-face****:*[*Google Font Directory*](https://code.google.com/apis/webfonts/)*.*

### Dimensiunea fontului

Cu ajutorul CSS se poate stabili dimensiunea fontului folosind valori absolute (exprimate in diferite unitati de masura: pixeli, centimetri), procentuale, sau chiar relative.  
Pentru a defini dimensiunea fontului se foloseste proprietatea **font-size** urmata de o valoare a dimensiunii care poate lua una din urmatoarele tipuri de valori:

* **- unitate de masura** - exprimata in pixeli (px), puncte (pt), em (ca raport la dimensiunea din elementul parinte) sau centimetri (cm)
* **- expresie absoluta** - xx-small, x-small, small, medium, large, x-large si xx-large
* - expresiile **smaller** sau **larger**, ca raport cu elementul parinte
* **- procent** - un numar exprimat in procente (cu %), care indica marimea textului in raport cu dimensiunea elementului parinte.

Forma generala a codului CSS pentru stabilirea dimensiunii fontului:

**selector { font-size:valoare; }**

Pixelii (px) reprezinta, cea mai des utilizata unitate de masura pentru stabilirea dimensiunii fontului, ofera o dimensiune fixa, indiferent de sistem si imbricare.  
Exemplu:

<!DOCTYPE html>

<html>

<head>

<title>Title</title>

<style>

h3 {

font-family:Calibri;

font-size:18px;

}

</style>

</head>

<body>

<h3>Tutorial CSS - font-size</h3>

</body>

</html>

Incercati codul

### Text inclinat

Atributul **font-style** permite scrierea textelor inclinate in doua moduri: "cursiv" si "oblic". Cursivul se refera la versiunea unui font a carui caractere au o inclinare spre dreapta, iar oblicul este un font inclinat fortat spre dreapta.  
Sintaxa:

**selector { font-style:valoare; }**

Unde "valoare" poate fi unul din urmatoarele cuvinte: **normal** (revine la text normal), **italic** (pentru cursiv) sau **oblique** (oblic).  
Exemplu:

<!DOCTYPE html>

<html>

<head>

<title>Title</title>

<style>

h3 {

font-family:Calibri;

font-size:18px;

font-style:italic;

}

</style>

</head>

<body>

<h3>Tutorial CSS - font-style</h3>

</body>

</html>

Incercati codul

- Valoarea **normal** se poate folosi pentru a sterge formatarea cu caractere inclinate atunci cand aceasta este mostenita.

### Grosimea fontului

Cu valorile proprietatii **font-weight** se pot defini diferite dimensiuni de ingrosare a textului, dupa cum urmeaza:

* **bold** – scrie fontul folosind caractere ingrosate
* **bolder** - mareste grosimea fontului relativ la grosimea curenta
* **lighter** – micsoreaza grosimea fontului fata de grosimea curenta
* - o valoare cuprinsa intre **100** si **900** (multiplu de 100) – reprezinta valoarea ingrosarii, bazata pe versiunile disponibile ale fontului.  
  400 ese normal, 700 e similar cu "bold", 800 ese maxim ingrosat.
* **normal** – elimina formatarea de ingrosare.

Sintaxa:

**selector { font-weight:valoare; }**

Exemplu:

<!DOCTYPE html>

<html>

<head>

<title>Title</title>

<style>

.cls\_bold {

font-family:Calibri;

font-size:16px;

font-weight:700;

}

</style>

</head>

<body>

<p class='cls\_bold'>Paragraf cu text ingrosat, font-weight:700;</p>

</body>

</html>

Incercati codul

### Text cu Majuscule

Proprietatea **font-variant:small-caps** afisaza textul cu MAJUSCULE.  
Valoarea **normal** anuleaza valoarea mostenita din parinte sau din alti selectori.

Exemplu:

<!DOCTYPE html>

<html>

<head>

<title>Title</title>

<style>

h3 {

font-family:Calibri;

font-size:18px;

font-variant:small-caps;

}

</style>

</head>

<body>

<h3>Tutorial CSS - font-variant</h3>

</body>

</html>

Incercati codul

### Valori multiple pentru font

Se pot adauga toate elementele fontului intr-o singura definitie.  
Pentru aceasta se foloseste proprietatea "**font**" urmata de valorile pentru celelalte proprietati, separate prin spatiu.  
Sintaxa:

**selector { font:font-style font-variant font-weight font-size font-family; }**

Daca o anumita valoare din lista este omisa, navigatorul va folosi valoarea prestabilita.  
- Iata si un exemplu practic de folosire a mai multor valori in aceeasi definitie:

<!DOCTYPE html>

<html>

<head>

<title>Title</title>

<style>

#dv1 {

font:italic small-caps bold 17px 'Calibri';

}

</style>

</head>

<body>

<div id='dv1'>Tutorial CSS - definire stiluri pentru font</div>

</body>

</html>

Incercati codul

# Configurare text

## Curs Css

[**Home**](https://marplo.net/)[**Engleza**](https://marplo.net/engleza)[**Spaniola**](https://marplo.net/spaniola)[**Html**](https://marplo.net/html)[**JavaScript**](https://marplo.net/javascript)[**PHP-MySQL**](https://marplo.net/php-mysql)[**Ajax**](https://marplo.net/ajax)[**Blog**](https://marplo.net/blog)[**Forum**](https://marplo.net/forum/)[**Games**](https://gamv.eu/)

Top of Form



Bottom of Form

Metodele de afisare a textelor prin controlul nu numai a fontului, dimensiunii si culorilor, ci si prin alte elemente (spatii, aliniere), pot imbunatati aspectul pagini si pot atrage atentia asupra anumitor elemente din text.

### Spatiul intre litere

Se poate aduaga sau reduce spatiul dintre litere folosind proprietatea **letter-spacing**, urmata de o valoare exprimata intr-o anumita unitate de masura, ce poate lua si valori negative.  
Sintaxa:

**selector { letter-spacing:valoare }**

Daca valoarea proprietatii este **normal** sau **0** atunci spatierea va fi fixata la valorile implicite (fara modificari suplimentare).  
Daca se utilizeaza o valoare exprimata in **em** atunci dimensiunea rezultata va fi calculata in raport cu cea a elementului parinte.  
Exemplu:

<!DOCTYPE html>

<html>

<head>

<title>Title</title>

<style>

h1 { letter-spacing:4px; }

</style>

</head>

<body>

<h1>Tutorial CSS - letter-spacing</h1>

</body>

</html>

Incercati codul

- Puteti folosi si pixeli (px).

### Spatiul dintre cuvinte

Spatiul dintre cuvinte poate fi modificat folosind proprietatea **word-spacing**, urmata de o valoare exprimata intr-o anumita unitate, fiind de obicei stabilita in pixeli.  
Pentru anularea spatierii intre cuvinte, valoarea proprietatii trebuie sa fie: **normal**.  
Sintaxa:

**selector { word-spacing:valoare; }**

O valoare pozitiva semnifica un plus de spatiu adaugat la valoarea implicita, iar o valoare negativa reduce acest spatiu. Daca valoarea este egala cu 0 atunci spatiul dintre cuvinte este fixat la valoarea prestabilita, neproducand de fapt nici un efect, fiind similar cu valoarea "normal".  
Iata un exemplu in care marim spatiul intre cuvintele din eticheta "h2" cu 8 pixeli:

<!DOCTYPE html>

<html>

<head>

<title>Title</title>

<style>

h2 { word-spacing:9px; }

</style>

</head>

<body>

<h2>Tutorial CSS - word-spacing</h2>

</body>

</html>

Incercati codul

### Spatiul intre linii

Inaltimea randurilor se refera la spatiul dintre liniile paragrafului.  
Modificarea spatiului intre linii poate avea ca efect o mai buna si usoara citire a textului in cazul in care aceasta este mai mare (creaza o regiune cu spatiu intre liniile de text). Alteori, daca inaltimea este mai mica (folosita la randurile de titlu), poate conferi un aspect mai stilat.  
Pentru a modifiica inaltimea randurilor se foloseste proprietatea **line-height**, urmata de o valoare care poate fi exprimata in trei moduri:

* - un **numar** care inseamna multiplicarea dimensiunii fontului cu numarul respectiv pentru a obtine valoarea spatierii;
* - o **valoare de tip absolut**, exprimata in pixeli, puncte sau orice alta unitate de masura folosita in CSS, care indiferent de dimensiunea stabilita a fontului, inaltimea liniei va avea o valoare exacta;
* - un **procent**, care stabileste inaltimea liniei ca fiind egala cu un anumit procent din dimensiunea fontului folosit pentru scrierea textului.

Sintaxa:

**selector { line-height:valoare; }**

Iata un exemplu in care inaltimea liniei se dubleaza (200% din dimensiunea fontului):  
Exemplu:

<!DOCTYPE html>

<html>

<head>

<title>Title</title>

<style>

.lh\_180 {

font-size:18px;

line-height:200%;

}

</style>

</head>

<body>

<div class='lh\_180'>Tutorial CSS - line-height.<br>

Un al rand din cest Div.</div>

</body>

</html>

Incercati codul

### Marimea (tipul) literelor

Folosind CSS, se poate defini ca textul sa apara cu majuscule initiale, majuscule, minuscule, combinat sau asa cum a fost el scris.  
Cu ajutorul proprietatii **text-transform** se poate controla marimea literelor din text, indiferent de modul cum a fost el scris initial.  
- Valori:

* **capitalize** – pentru ca primul caracter din fiecare cuvant sa fie scris cu majuscula;
* **uppercase** – toate caracterele sa fie scrise cu majuscule;
* **lowercase** - toate caracterele sa fie scrise cu litere mici;
* **none** – textul sa fie afisat asa cum este.

Sintaxa generala:

**selector { text-transform:valoare; }**

Iata un exemplu in care toate literele vor fi afisate cu majuscule:  
Exemplu:

<!DOCTYPE html>

<html>

<head>

<title>Title</title>

<style>

h2 { text-transform:uppercase; }

</style>

</head>

<body>

<h2>Tutorial CSS - text-transform</h2>

</body>

</html>

Incercati codul

### Alinierea textului

Proprietatea **text-align** ofera posibilitatea controlului asupra alinierii textului.  
Definirea alinierii se face specificand una din valorile: **left**, **right**, **center**sau **justify** (aliniat stanga-dreapta).  
Forma generala:

**selector { text-align:valoare; }**

*Proprietatea poate fi aplicata numai elementelor la nivel de bloc (DIV, UL, Hx, p), valoarea sa implicita fiind in majoritatea cazurilor stabilita la valoarea "left".*

Iata un exemplu in care textul din eticheta "h2" este centrat:

<!DOCTYPE html>

<html>

<head>

<title>Title</title>

<style>

h2 { text-align:center; }

</style>

</head>

<body>

<h2>Exemplu cu text-align:center;</h2>

</body>

</html>

Incercati codul

### Alinierea pe verticala a textului

Cu ajutorul proprietatii **vertical-align** se poate preciza pozitia unor elemente (in linie) in raport cu textul din jurul lor. Asta inseamna ca proprietatea poate fi folosita doar cu etichete in linie care nu determina un salt la linie noua, de obicei utilizandu-se cu elementul <span>.  
Sintaxa:

**selector { vertical-align:valoare; }**

- "valoare" poate fi una din urmatoarele optiuni:

* **super** – pentru scrierea textului in stil exponent, deasupra liniei de baza;
* **sub** - pentru scrierea textului in stil indice, sub linia de baza;
* **baseline** – pentru scrierea textului pe linia de baza;
* - una din valorile: "**top**", "**middle**", "**bottom**", "**text-top**", "**text-bottom**" – pentru a alinia textul relativ la alinierea parintelui acestuia;
* - valoare **procentuala** – care ridica sau coboara linia de baza a elementului proportional cu dimensiunea fontului elementului parinte.

Iata un exemplu in care textul din eticheta <span> cu clasa ".va\_sup" va aparea ca exponent fata de textul din stanga-dreapta lui:

<!DOCTYPE html>

<html>

<head>

<title>Title</title>

<style>

span.va\_sup { vertical-align:super; }

</style>

</head>

<body>

<div>Text in Div, <span class='va\_sup'>acesta in span cu class va\_sup</span>, se vede alinierea.</div>

</body>

</html>

Incercati codul

### Proprietatea text-indent

Cu proprietatea **text-indent** se poate stabili un spatiu suplimentar, gol, la inceputul primei linii de text dintr-un paragraf.  
Sintaxa generala este:

**selector { text-indent:valoare; }**

- valoare poate fi in pixeli (px) sau em, ori in procente (proportional cu latimea paragrafului).

*Valorile pozitive determina o indentare tipica (spre dreapta), in timp ce valorile negative determina o indentare inspre margine, fiind necesar sau marirea umplerii (padding) sau marirea marginilor.*

Iata un exemplu de cod CSS in care va fi definit un spatiu de 30 pixeli la inceputul fiecarui paragraf <p>:

<!DOCTYPE html>

<html>

<head>

<title>Title</title>

<style>

p {

text-indent:30px;

}

</style>

</head>

<body>

<p>

A fost odata ca niciodata,<br>

A doua linie continua si incheie povestea.

</p>

<p>

Alt paragraf, nu spune nimic.<br>

Hai si pe alte randuri,<br>

Doar primul e cu 'indent'.

</p>

</body>

</html>

Incercati codul

Daca se utilizeaza o valoare exprimata in unitati "em", atunci rezultatul va fi relativ la dimensiunea fontului elementului asupra caruia se aplica.  
Pentru eliminarea indentarii se aplica valoarea 0.

### Proprietatea text-decoration

Cu proprietatea CSS **text-decoration**: textul poate fi afisat cu: subliniere, supraliniere, si taierea textului cu o linie pe mijloc.  
- Valori

* **underline** – pentru a sublinia textul
* **overline** – pentru a trasa o linie deasupra textului
* **line-through** – taie textul cu o linie

Sintaxa:

**selector { text-decoration:valoare }**

- Valoarea "none" anuleaza efectele "text-decoration" mostenite. Prin folosirea acestei valori se elimina si sublinierile link-urilor, chiar daca navigatorul are prestabilit sa le arate subliniat.  
  
Iata un exemplu de cod in urma caruia link-urile cu clasa '.no\_line' nu vor mai fi afisate subliniat:

<!DOCTYPE html>

<html>

<head>

<title>Title</title>

<style>

a.no\_line:link {

text-decoration:none;

}

</style>

</head>

<body>

<a href='//marplo.net' class='no\_line'>MarPlo.net</a><br>

<a href='//gamv.eu'>Gamv.eu</a><br>

<a href='//coursesweb.net' class='no\_line'>CoursesWeb.net</a>

</body>

</html>

Incercati codul

### Afisare Spatii goale

Liniile goale multiple si spatiile goale din text sunt tratate ca un singur spatiu sau sunt ignorate, daca nu este utilizata eticheta <pre>.  
CSS permite afisarea acestor spatii, precum si a liniilor goale, cu proprietatea **white-space:pre**.  
Sintaxa:

**selector { white-space:valoare; }**

- "valoare" poate fi: "***pre***", "***nowrap***" (impiedica trecerea la linia urmatoare) sau "***normal***".  
  
Iata un exemplu in care in elementele cu clasa '.cls\_pre' spatiile goale din continut si liniile goale vor fi afisate asa cum au fost scrise:

<!DOCTYPE html>

<html>

<head>

<title>Title</title>

<style>

.cls\_pre {

white-space:pre;

}

</style>

</head>

<body>

Util pt. a afisa linii de cod:

<div class='cls\_pre'>

.cls\_cntr {

color: #0000ef; /\* like blue \*/

margin: 2px auto;

text-align:center;

}

/\* Dupa 2 linii \*/

// Exemplu cu: white-space: pre;

</div>

</body>

</html>

Incercati codul

# Configurari pentru culori si fundal

## Curs Css

[**Home**](https://marplo.net/)[**Engleza**](https://marplo.net/engleza)[**Spaniola**](https://marplo.net/spaniola)[**Html**](https://marplo.net/html)[**JavaScript**](https://marplo.net/javascript)[**PHP-MySQL**](https://marplo.net/php-mysql)[**Ajax**](https://marplo.net/ajax)[**Blog**](https://marplo.net/blog)[**Forum**](https://marplo.net/forum/)[**Games**](https://gamv.eu/)

Top of Form



Bottom of Form

### Configurare culoare fundal - background

Cu proprietatile CSS "***background***" se poate schimba fundalul intregii pagini, sau pentru un anumit element din pagina, fie el de tip DIV, paragraf, tabel sau formular.

#### Culoare fundal - background-color

Pentru a schimba culoarea fundalului se utilizeaza proprietatea **background-color**, urmata de o valoare care reprezinta numele culorii sau valoarea RGB, exprimata in cod hexa.  
Sintaxa generala:

**selector {background-color:valoare }**

Daca se foloseste valoarea "***transparent***" navigatorul va afisa culoarea prestabilita sau cea a elementului parinte.  
Pentru schimbarea fundalului intregii pagini se aplica aceasta proprietate selectorului "body".  
Iata un exemplu in care culoarea de fundal a paginii este schimbata:

<!DOCTYPE html>

<html>

<head>

<title>Title</title>

<style>

body {

background-color:#ededfe;

}

</style>

</head>

<body>

Tutorial CSS - background.

</body>

</html>

Incercati codul

#### Imagine de fundal - background-image

Pentru adaugarea unei imagini la fundalul unui element HTML se foloseste proprietatea **background-image**, urmata de adresa URL a locatiei imaginii.  
Sintaxa generala:

**selector { background-image:url('adresa\_URL'); }**

- Unde "adresa\_URL" este calea si numele imaginii care va fi folosita.  
Daca in locul adresei URL se foloseste valoarea "none", navigatorul nu va folosi nici o imagine.  
Iata un exemplu in care pentru fundalul paginii este folosita o imagine:

<!DOCTYPE html>

<html>

<head>

<title>Title</title>

<style>

body {

background-color:#f8f8fe;

background-image:url('../imgs/smile\_gift.png');

}

</style>

</head>

<body>

Tutorial CSS - background.

</body>

</html>

Incercati codul

#### Repetarea imaginii de fundal

In mod initial imaginea de fundal se repeta. Pentru a controla repetarea se aplica proprietatea **background-repeat**, urmata de o valoare care poate avea una din urmatoarele optiuni:

* **repeat** – repeta imaginea pe toata suprafata elementului respectiv;
* **repeat-x** – repeta imaginea numai pe orizontala;
* **repeat-y** – repeta imaginea numai pe verticala;
* **no-repeat** – afisaza imaginea o singura data, fara repetare.

Forma generala este:

**selector { background-repeat:valoare; }**

Iata un exemplu de cod in care imaginea folosita pentru fundalul paginii se va repeta pe verticala pe toata inaltimea documentului:

<!DOCTYPE html>

<html>

<head>

<title>Title</title>

<style>

body {

background-color:#f8f8fe;

background-image:url('../imgs/smile\_gift.png');

background-repeat:repeat-y;

}

</style>

</head>

<body>

Tutorial CSS - background cu imagine.

</body>

</html>

Incercati codul

#### Proprietatea background-attachment

Cu proprietatea **background-attachment** se poate defini modul de tratare a imaginii de fundal la derularea paginii, cu valorile: "**fixed**", pentru a lipi imaginea de fundal de fereastra navigatorului sau "**scroll**", pentru a permite derularea imaginii de fundal.  
Sintaxa generala:

**selector { background-attachment:valoare }**

Exemplu:

<!DOCTYPE html>

<html>

<head>

<title>Title</title>

<style>

body {

background-attachment:fixed;

background-image:url('../imgs/smile\_gift.png');

background-repeat:no-repeat;

height:1200px;

}

</style>

</head>

<body>

Tutorial CSS - background cu imagine.<br>

Derulati in jos, imaginea va ramane fixata.

</body>

</html>

Incercati codul

#### Pozitia imaginii de fundal

Pentru a pozitiona imaginea de fundal in functie de coltul din stanga-sus al elementului se utilizeaza proprietatea **background-position** urmata de doua valori (coordonate X si Y) separate prin spatiu, 'X' pentru distanta orizontala si 'Y' pentru cea verticala.  
Forma generala este:

**selector { background-position:X Y; }**

Coordonatele X si Y pot fi exprimate ca valori absolute (px, em) sau ca procente (%). Se pot folosi si cuvintele cheie: "**left**", "**center**" sau "**right**" pentru coordonata X; respectiv "**top**", "**center**" sau "**bottom**" pentru coordonata Y.  
  
-Iata un exemplu in care imaginea de fundal intr-un Div este pozitionata centrat pe orizontala, la 20 pixeli fata de marginea de sus:

<!DOCTYPE html>

<html>

<head>

<title>Title</title>

<style>

body {

background-image:url('../imgs/smile\_gift.png');

background-position:center 20px;

background-repeat:no-repeat;

}

</style>

</head>

<body>

<div id='dv\_ex'>Div cu background-image.</div>

</body>

</html>

Incercati codul

#### Proprietatea background

Toate proprietatile fundalului pot fi definite in proprietatea **background**. Se poat stabili simultan printr-o lista de valori proprietatile fundalului, enumerate in orice ordine si separate prin spatiu.  
Cand se utilizeaza proprietatea **background**, nu este necesar sa fie specificate toate proprietatile, cele nespecificate fiind stabilite la valoare lor implicita.  
Sintaxa generala este:

**selector {background: background-color background-image background-repeat background-attachment background-position; }**

Valoarea implicita a proprietatii **background-color** este "transparent".  
Valoarea implicita a proprietatii **background-image** este "none".  
Valoarea implicita a proprietatii **background-attachment** este "scroll".  
Valoarea implicita a proprietatii **background-position** este "top left".  
  
Exemplu:

div {

background: #e8e8fe url('imagine.jpg') center top no-repeat;

}

- DIV-ul va avea culoarea de fundal "#e8e8fe", cu o imagine pozitionata "center top" si care nu se repeta.

### Proprietatea color

Proprietatea **color** poate schimba culoare continutului din elemente HTML: text, linie orizontala sau element de tip formular.  
Sintaxa generala:

**selector { color:valoare; }**

Valoarea culorii poate fi:

* **numele culorii** - una din cele 16 culori predefinite (red, green, blue, white, silver, ...)
* **valoare in hexa** - sub forma #rrggbb
* **valoare RGB** - sub forma rgb(r, g, b), unde r, g, b pot fi reprezentate prin numere intregi cuprinse intre 0 si 255 sau procente de rosu, verde si albastru cu valori intre 0% si 100%.

Iata un exemplu de cod CSS in care se definesc culori pentru fundalul si text la elemente "h3":

<!DOCTYPE html>

<html>

<head>

<title>Title</title>

<style>

h3 {

background:#0000bb;

color:#fefefe;

}

</style>

</head>

<body>

<h3>Si Pacea este Buna</h3>

<h3>E Bun, suntem Fericiti</h3>

</body>

</html>

Incercati codul

# Afisare si chenare din CSS la elemente HTML

## Curs Css

[**Home**](https://marplo.net/)[**Engleza**](https://marplo.net/engleza)[**Spaniola**](https://marplo.net/spaniola)[**Html**](https://marplo.net/html)[**JavaScript**](https://marplo.net/javascript)[**PHP-MySQL**](https://marplo.net/php-mysql)[**Ajax**](https://marplo.net/ajax)[**Blog**](https://marplo.net/blog)[**Forum**](https://marplo.net/forum/)[**Games**](https://gamv.eu/)

Top of Form



Bottom of Form

Indiferent daca un element este de sine statatr sau imbricat in alte elemente, acesta poate fi tratata ca un element distinct pe ecran si poate fi controlata prin intermediul CSS.  
Notiunea de element se refera la componentele unui document HTML, configurate prin intermediul etichetelor HTML.  
Caseta (cadrul elementului) dispune de numeroase proprietati, intre care se numara: marginile (***margin***), chenarul (***border***), completarea (umplerea, cunoscuta ca "padding"), latimea (***width***) si inaltimea (***height***), care pot fi modificate cu ajutorul CSS.  
  
Elementele HTML au patru laturi (sus, dreapta, jos si stanga) la care se pot aplica proprietatile CSS specifice, restul proprietatilor CSS (font, text, color, background) se aplica la continutului casetei. Continutul poate include texte, liste, formulare, imagini.

### Latimea si inaltimea unui element

Latimea si inaltimea elementelor pot fi specificate cu ajutorul proprietatilor **width**, respectiv **height**.  
In mod prestabilit (default) acestea sunt setate automat de catre navigator. Pentru definirea latimii si inaltimii se folosesc urmatoarele tipuri de valori:

* - o valoare de tip **numeric**, de obicei in pixeli (px), sau em;
* - un **procent**, care stabileste o valoare proportionala in functie de cea a elementului parinte;
* - valoarea **auto**, care foloseste latimea si inaltimea calculata de navigator, si care reprezinta cantitatea de spatiu maxim pe care o poate ocupa elementul pentru afisarea continutului.

Sintaxa generala pentru configurarea latimii si inaltimii este:

**selector {**

**width:valoare;**

**height:valoare;**

**}**

Iata un exemplu in care elementul cu id="un\_id" va avea latimea de 300 pixeli si inaltimea de 500 pixeli:

#un\_id { width:300px; height:500px; }

### Marginile unui element

Proprietatea **margin** permite stabilirea distantei dintre un element si alte elemente alaturate, prin specificarea unei valori pentru marginea din fiecare latura (sus, dreapta, jos, stanga).  
Daca specificati cele 4 valori in aceeasi definitie, acestea reprezinta marginile elementului in urmatoarea ordine: sus (top), dreapta (right), jos (bottom), stanga (left).  
Daca specificati doar o valoare, aceasta va fi aplicata tuturor marginilor.  
- In cazu in care specificati doua sau trei valori, atunci valorile care lipsesc sunt copiate dupa valorile marginilor opuse.  
Sintaxa generala pentru configurarea marginilor unui selector este:

**selector { margin: valoare/valori; }**

Tipul de valoare folosit poate avea urmatoarele optiuni:

* - o valoare de tip **lungime (numerica)** – px, em; care poate fi si negativa;
* - o valoare **procentuala (procent)** – creaza o margine proportionala cu latimea elementului parinte;
* - valoarea **auto** – lasa controlul marginilor la decizia navigatorului.

Iata un exemplu in care etichetele DIV vor avea distanta pentru marginea de sus 10 pixeli, dreapta 5 pixeli, jos 8 pixeli si in stanga 15 pixeli:

div { margin:10px 5px 8px 15px; }

Exista posibilitatea de a stabili valoarea marginii doar pentru o singura latura, fara a tine cont de celelalte margini. Pentru aceasta sunt folosite proprietatile **margin-top, margin-bottom, margin-left** si **margin-right** cu aceleasi valori ca si in cazul proprietatii **margin**, dupa cum puteti vedea mai jos.

**selector { margin-top: valoare; }**

**selector { margin-bottom: valoare; }**

**selector { margin-left: valoare; }**

**selector { margin-right: valoare; }**

### Chenarul unui element - border

Proprietatea **border** defineste o bordura in jurul elementului, stabilind grosimea, stilul si culoarea bordurii.  
Sintaxa generala pentru definirea chenarului este:

**selector { border:val\_grosime val\_stil val\_culoare; }**

Unde

* - prima valoare (val\_grosime) reprezinta grosimea chenarului si poate fi una din urmatoarele tipuri:
  + - o valoare de tip lungime (valoarea 0 determina anularea afisarii chenarului);  
    - un cuvant cheie (**thin, medium, thick**) care caracterizeaza o dimensiune relativa.
* - a doua valoare (val\_stil) reprezinta numele stilului atribuit chenarului (***solid, dotted, dashed***). Valoarea "none" anuleaza afisarea chenarului.
* - ultima valoare (val\_culoare) reprezinta culoarea, exprimata in cod hexa sau valoare RGB.

Iata un exemplu in care div-urile cu cu clasa '.bor\_blue' vor avea o bordura groasa de 2 pixeli, solid si culoare albastra:

<!DOCTYPE html>

<html>

<head>

<title>Title</title>

<style>

div.bor\_blue {

border:2px solid blue;

}

</style>

</head>

<body>

<div class='bor\_blue'>Div cu clasa .bor\_blue</div>

<div>Alt DIV, fara class.<br>

Un alt rand in acest Div.</div>

<div class='bor\_blue'>Asta deduceti singur.</div>

</body>

</html>

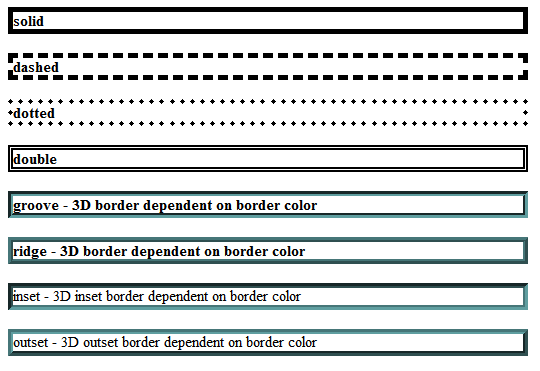
Incercati codul

- Pot fi folosite si proprietati separate pentru fiecare valoare: "border-width" (grosimea), "border-style" (stilul) si "border-color" (culoarea).  
  
Un atribut care poate fi folosit in configurarea chenarului este **border-width**, acesta furnizeaza o metoda rapida pentru stabilirea latimii celor patru laturi din jurul unei casete.  
Sintaxa generala este:

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

Daca specificati toate cele patru valori, ele sunt aplicate in ordinea: sus, dreapta, jos, stanga.  
Daca specificati doar o valoare, ea va fi aplicata tuturor marginilor.  
Daca folositi doua sau trei valori, valorile care lipsesc sunt copiate din valorile marginilor opuse lor in caseta.  
  
Aspectul bordurii poate fi stabilit prin atributul **border-style**; stilul este aplicat celor patru margini si este definit astfel:

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

Pentru specificarea valorii, pot fi folosite urmatoarele tipuri:  
  
  
Un alt atribut care poate fi utilizat in configurarea chenarului este **border-color**, acesta stabileste culoarea pentru toate cele patru laturi ale chenarului si foloseste pentru valoarea ei un singur cuvant cheie exprimat in cod hexa, valoare RGB sau numele culorii.  
Sintaxa generala este:

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

Puteti stabili culoarea fiecarei margini si separat, folosind proprietatile:

**border-top-color, border-right-color, border-bottom-color, border-left-color**

Fiecare latura a chenarului poate avea valori configurate in mod separat (prin care se stabilesc stilul si culoarea fiecarei borduri din jurul elementului) dupa cum este prezentat mai jos:

**selector { border-top: grosime stil culoare; }**

**selector { border-bottom: grosime stil culoare; }**

**selector { border-left: grosime stil culoare; }**

**selector { border-right: grosime stil culoare; }**

### Adaugarea spatiului in interior - padding

Proprietatea **padding** adauga o cantitate de spatiu intre margine (bordura) si continutului elementului, in interiorul chenarului.  
Forma generala pentru utilizarea acestei proprietati este:

**selector { padding: valoare/valori; }**

Valoarea pentru completarea spatiului poate fi una din urmatoarele:

* - o valoare de tip **lungime**
* - o valoare **procentuala** – creaza umplerea in raport cu latimea elementului parinte.

Cand specificati toate cele patru valori, ele sunt aplicate in ordinea: sus, dreapta, jos, stanga.  
Daca specificati doar o valoare, aceasta va fi aplicata tuturor celor patru directii.  
Daca specificati doua sau trei valori, atunci valorile care lipsesc sunt copiate dupa valorile laturilor opuse.  
  
Iata un exemplu in care etichetele DIV vor avea distanta padding: sus 4 pixeli, dreapta 2 pixeli, jos 3 pixeli si in stanga 2 pixeli:

div { padding:4px 2px 3px 2px; }

Fiecare latura a chenarului poate avea proprietatea **padding** configurata separat:

**selector { padding-top:valoare; }**

**selector { padding-bottom:valoare; }**

**selector { padding-left:valoare; }**

**selector { pading-right:valoare; }**

### Elemente float

CSS ofera posibilitatea de a "infasura" anumite elemente in jurul altora. Acest lucru se face prin intermediul proprietatii **float**, care determina o mutare fortata a elementului in directia data de valoarea mentionata, lasand loc liber in spatiul opus directiei, care va fi ocupat de urmatorul element din codul HTML.  
Sintaxa generala:

**selector { float:valoare; }**

Unde "vaoare" poate fi" **left, right** sau **none**  
Valoarea "none" anuleaza valorile "float" la acel element.  
Iata un exemplu in care elementul cu id="un\_id" va fi fortat sa fie pozitionat in dreapta celorlaltor elemente:

<!DOCTYPE html>

<html>

<head>

<title>Title</title>

<style>

#un\_id {

background:#ddddfe;

float:right;

height:80px;

width:180px

}

</style>

</head>

<body>

<div id='un\_id'>Div cu float:right;</div>

<p>Paragraf dupa acel Div.</p>

</body>

</html>

Incercati codul

### Anularea plasarii in spatiul eliberat de "float" - clear

Cu proprietatea **clear** se poate anula ocuparea unui spatiu care a fost lasat liber de un element care utilizeaza proprietatea "float".  
Sintaxa:

**selector { clear:valoare; }**

Unde "valoare" specifica latura in jurul careia este oprita infasurarea textului si poate lua valoarea: **left, right** sau **both**. Se poate folosi si valoarea "none", ceea ce are ca efect anularea altor atribute ale proprietatii **clear**.  
  
In exemplu de sus, cu DIV-ul pozitionat cu "***float:right;***", paragraful care e dupa el va fi afisat in stanga DIV-ului. Pentru a anula acest lucru, ca paragraful sa fie afisat sub DIV, se aplica proprietatea "***clear***", precum in exemplul urmator:

<!DOCTYPE html>

<html>

<head>

<title>Title</title>

<style>

#un\_id {

background:#ddddfe;

float:right;

height:80px;

width:180px

}

.clrb { clear:both; }

</style>

</head>

<body>

<div id='un\_id'>Div cu float:right;</div>

<p class='clrb'>Paragraf cu o clasa ce seteaza 'clear:both;'.</p>

Alt continut dupa Div.

</body>

</html>

Incercati codul

### Afisarea si ascunderea elementelor

Proprietatea **display** defineste modul de afisare a elementului.  
- Sintaxa generala:

**.element { display: valoare; }**

Cele mai utilizate valori:

* **block** – defineste elementul ca fiind de tip bloc, precum <p> (incepe pe un rand nou si are alt rand dupa el).
* **inline** – defineste elementul ca tip in linie (precum <span>). Proprietatile "width" si "height" nu au efect.
* **inline-block** – defineste elementul ca tip in linie, dar pot fi aplicate valori pentru "width" si "height".
* **none** – elementul nu mai este afisat; e complet ascuns din pagina.

Iata un exemplu in care elementele <li> sunt asezate in linie, iar un element cu id="un\_id" va fi ascuns in pagina:

<!DOCTYPE html>

<html>

<head>

<title>Title</title>

<style>

.ds\_li li { display:inline; }

#un\_id { display:none; }

</style>

</head>

<body>

<div id='un\_id'>Div care nu va fi afisat.</div>

<ul class='ds\_li'>

<li>MarPlo.net</li>

<li>GamV.eu</li>

<li>CoursesWeb.net</li>

</ul>

</body>

</html>

Incercati codul

# ozitionare elemente HTML cu CSS

## Curs Css

[**Home**](https://marplo.net/)[**Engleza**](https://marplo.net/engleza)[**Spaniola**](https://marplo.net/spaniola)[**Html**](https://marplo.net/html)[**JavaScript**](https://marplo.net/javascript)[**PHP-MySQL**](https://marplo.net/php-mysql)[**Ajax**](https://marplo.net/ajax)[**Blog**](https://marplo.net/blog)[**Forum**](https://marplo.net/forum/)[**Games**](https://gamv.eu/)

Top of Form



Bottom of Form

Pozitionarea elementelor folosind CSS este mai precisa decat prin intermediul obiectelor grafice HTML sau a tabelelor, afisarea facandu-se mult mai rapid.  
Prin intermediul CSS este permisa pozitionarea exacta sau relativa a elementelor intr-o fereastra sau in raport cu alte elemente.  
Toatele elementele amplasate in fereastra sunt pozitionate relativ la coltul din stanga-sus.

### Proprietatea position

Cu proprietatea **position** se poate specifica pozitia elementului in pagina web.  
Un element poate avea una din urmatoarele valori de pozitionare: **static, relative, absolute** si **fixed**.

#### Pozitionarea statica

Valoarea initiala, prestabilita, a pozitionarii elementelor in fereastra este "static".  
Cand nu este specificata o pozitionare "relativa", "absoluta" sau "fixa"; elementele sunt dispuse unul dupa altul in interiorul documentului.  
Sintaxa:

**selector { position:static; }**

- Un element static nu poate fi repozitionat in mod explicit.

#### Pozitionarea relativa

Un element cu pozitionare "relativa" este amplasat la locul sau in cadrul elementului parinte. Apare dupa toate elementele dinaintea sa, respectiv inaintea tuturor elementelor amplasate dupa el.  
Sintaxa:

**selector { position:relative; }**

- Elementele pozitionate relativ pot fi mutate din locatia lor folosind proprietatile "top" si "left" sau "bottom" si "right".

#### Pozitionarea absoluta

Pozitionarea absoluta are ca efect crearea unui element neafectat de restul documentului, plasarea lui in fereastra fiind facuta intr-o locatie precisa, definita prin intermediul coordonatelor X si Y, indiferent de pozitiile altor elemente.  
Sintaxa:

**selector { position:absolute; }**

- Originea (punctul de coordonate 0,0) este coltul din stanga-sus al obiectului (care are "position: relative") in care este inclus elementul pozitionat absolut.

#### Pozitionarea fixa

Pozitionarea fixa se face in raport cu fereastra browser-ului, iar la derularea paginii elementul fixat ramane pe pozitia lui initiala, fara a se derula.  
Sintaxa:

**selector { position:fixed; }**

### Pozitionarea in raport cu latura de sus, respectiv stanga

Dupa stabilirea tipului de pozitionare, se poate determina distanta intre punctul de origine si laturile de sus si din stanga ale elementului parinte, folosind proprietatile: **top** si **left**  
Forma generala este:

**selector {**

**position: valoare;**

**top: valoare;**

**left: valoare;**

**}**

- La 'top' si 'left', "valoare" poate fi:

* - o valoare de tip numeric: px, em, cm
* - o valoare procentuala (%);
* - valoarea ***auto***, lasa la modul de afisare a browser-ului.

De asemenea, pot lua si valori negative.  
  
Iata un exemplu in care tag-ul <h3> va fi pozitionat la o distanta de 25 pixeli fata de marginile sus si stanga ale unui DIV parinte:

<!DOCTYPE html>

<html>

<head>

<title>Title</title>

<style>

div {

border:1px solid #0000d0;

position:relative;

height:150px;

width:250px;

}

div h3 {

margin:0;

position:relative;

top:25px;

left:25px;

}

</style>

</head>

<body>

<div>

<h3>Tutorial css - position</h3>

</div>

</body>

</html>

Incercati codul

### Pozitionarea in raport cu latura de jos, respectiv dreapta

Pozitionarea in raport cu laturile de jos, respectiv dreapta se poate aplica in cazul pozitionarilor "***absolute***" sau "***fixed***". Originea de referinta va fi coltul din dreapta-jos al elementului parinte.  
Definirea distantei fata de latura de jos, respectiv din dreapta se face cu proprietatile: **bottom** si **right**; ele pot lua aceleasi valori ca si "left" si "top", de asemenea pot fi combinate cu acestea.  
Forma generala este:

**selector {**

**position: absolute | fixed;**

**bottom:valoare;**

**right:valoare;**

**}**

- Exemplu: tag-ul <h3> va fi pozitionat la o distanta de 25 pixeli fata de marginile de jos si dreapta ale unui DIV parinte:

<!DOCTYPE html>

<html>

<head>

<title>Title</title>

<style>

div {

border:1px solid #0000d0;

position:relative;

height:150px;

width:250px;

}

div h3 {

margin:0;

position:absolute;

bottom:25px;

right:25px;

}

</style>

</head>

<body>

<div>

<h3>Tutorial css - position</h3>

</div>

</body>

</html>

Incercati codul

*In cazul in care pentru acelasi element sunt stabilite atat marginile "top / left" cat si "bottom / right", rezultatul afisat depinde de navigator, dar in mod prestabilit se folosesc pozitiile****top / left****.*

### Includerea unui element absolut in unul relativ

In cazul includerii unui element pozitionat absolut intr-un element pozitionat relativ, elementul absolut este pozitionat folosind ca origine coltul din stanga-sus al elementului relativ.  
- Iata un exemplu in care o eticheta <div> (cu class="absolut") este inclusa in alta eticheta <div> (cu class="relativ"):

<!DOCTYPE html>

<html>

<head>

<title>Title</title>

<style>

.relativ {

border:1px solid #0000d0;

position:relative;

height:150px;

width:250px;

}

.absolut {

background:#bdbdfb;

position:absolute;

margin:0;

top:30px;

left:30px;

height:90px;

width:150px;

}

</style>

</head>

<body>

<div class='relativ'>

Inainte de Div.

<div class='absolut'>Div-ul .absolut</div>

Dupa Div.

</div>

</body>

</html>

Incercati codul

### Includerea unui element relativ in unul absolut

In cazul in care un element pozitionat relativ este plasat in interiorul unui element pozitionat absolut, acesta se va deplasa o data cu elementul absolut.  
Iata un exemplu de cod HTML in care o eticheta <div> (cu class="relativ") este inclusa in alta eticheta <div> (cu class="absolut"):

<!DOCTYPE html>

<html>

<head>

<title>Title</title>

<style>

.absolut {

border:1px solid #0000d0;

position:absolute;

margin:0;

top:40px;

left:50px;

height:150px;

width:250px;

}

.relativ {

background:#bdbdfb;

position:relative;

height:90px;

width:150px;

}

</style>

</head>

<body>

<div class='absolut'>

Div-ul .absolut.

<div class='relativ'>Div-ul .relativ</div>

</div>

</body>

</html>

Incercati codul

### Pozitionarea in spatiu 3D

Elementele pot primi o a treia dimensiune, si anume asezarea lor in stiva, unele in raport cu altele.  
Amplasarea se face in mod automat, incepand cu valoarea 0 si continuand prin incrementare cu o unitate, in ordinea aparitiei lor in documentul HTML si relativ la elementele parinte.  
Pentru pozitionarea elementelor in stiva unele peste altele se foloseste proprietatea **z-index**.  
Valoarea acestei proprietati aduce elementul in fata sau in sptele altor obiecte HTML din pagina.  
Sintaxa generala:

**selector { z-index:valoare; }**

Unde "valoare" este un numar intreg, pozitiv, 0 sau negativ.  
In cazul in care continutul elementelor se suprapune, elementul cu numarul de ordine mai mare apare deasupra elementului cu numar mai mic.  
Utilizarea unei valori negative determina amplasarea elementului dedesubtul parintelui.  
  
- Exemplu, Div-ul #dv1 e pozitionat absolut si, datorita proprietatii "z-index" apare deasupra Div-ului #dv2 care e dupa el.

<!DOCTYPE html>

<html>

<head>

<title>Title</title>

<style>

.parent {

border:1px solid #0000d0;

position:relative;

height:180px;

width:280px;

}

#dv1 {

background:#bdbdfb;

position:absolute;

top:40px;

left:50px;

height:100px;

width:150px;

z-index:9999;

}

#dv2 {

background:#ebeb00;

height:100px;

width:150px;

}

</style>

</head>

<body>

<div class='parent'>

<div id='dv1'>Div-ul #dv1</div>

<div id='dv2'>Div-ul #dv2</div>

</div>

</body>

</html>

Incercati codul

# Configurari pentru vizibilitate si mouse

## Curs Css

[**Home**](https://marplo.net/)[**Engleza**](https://marplo.net/engleza)[**Spaniola**](https://marplo.net/spaniola)[**Html**](https://marplo.net/html)[**JavaScript**](https://marplo.net/javascript)[**PHP-MySQL**](https://marplo.net/php-mysql)[**Ajax**](https://marplo.net/ajax)[**Blog**](https://marplo.net/blog)[**Forum**](https://marplo.net/forum/)[**Games**](https://gamv.eu/)

Top of Form



Bottom of Form

### Stabilirea vizibilitatii unui element

Proprietatea **visibility** poate controla faptul ca un element sa fie vizibil sau nu. Daca elementul este ascuns (hidden), in locul suprafetei pe care o ocupa va fi afisat un spatiu gol.  
Sintaxa:

**selector { visibility:valoare }**

Unde "valoare" poate fi:

* **hidden** – ascunde elementul de la afisarea pe ecran;
* **visible** – afiseaza elementul;
* **inherit** – vizibilitatea elementului este mostenita de la elementul parinte.

Exemplu, elementul cu clasa "vi\_hid" va fi ascuns:

<!DOCTYPE html>

<html>

<head>

<title>Title</title>

<style>

p {

border:1px solid #333;

}

.vi\_hid {

visibility:hidden;

}

</style>

</head>

<body>

<p class='vi\_hid'>Oricum nu se vede.</p>

<p>Paragraf dupa cel ascuns.</p>

</body>

</html>

Incercati codul

*Pentru a elimina complet afisarea unui element in pagina, se poate utiliza formula****display:none;***

### Stabilirea suprafetei vizibile a unui element

La elementele care au ***position:absolute;*** (sau "***fixed***") se poate defini suprafata vizibila aplicand proprietatea:

**clip: rect(v1 v2 v3 v4);**

Aceasta stabileste portiunea din elementul respectiv care este vizibila in fereastra navigatorului. Restul continutului acelui element nu dispare, ci este invizibil pentru vizitator.

- Valorile de la ***rect()*** construesc un patrulater, definesc distanta dintre coltul din stanga-sus al elementului si laturile de sus, dreapta, jos si stanga ale regiunii vizibile.  
Daca se foloseste valoarea "auto", navigatorul calculeaza dimensiunea regiunii vizibile la 100%.  
  
Exemplu, suprafata vizibila a unui element cu clasa '.viz\_clip' va fi cea incadrata de valorile functiei **rect()**, dar devine complet vizibila cand mouse-ul e deasupra acelui element:

<!DOCTYPE html>

<html>

<head>

<title>Title</title>

<style>

.viz\_clip {

background:#eeee11;

clip: rect(10px,150px, 45px, 1px);

position:absolute;

}

.viz\_clip:hover {

clip:auto;

}

</style>

</head>

<body>

<p class='viz\_clip'>Exemplu cu proprietatea css clip.<br>

Pozitionati mouse-ul peste acest continut.</p>

</body>

</html>

Incercati codul

### Afisare bara pentru derulare

Cu proprietatea **overflow** se poate stabili tratarea continutului care depaseste dimensiunile elementului in care se afla, controland astfel afisarea (cu o bara de "scroll", derulare) sau nu al acestuia.  
Forma generala:

**selector { overflow:valoare; }**

- Unde "valoare" stabileste unde va fi plasata bara de derulare, folosind una din valorile:

* **visible** – extinde caseta elementului astfel incat sa incapa tot continutul sau, ignorand delimitarea suprafetei vizibile. Este optiunea prestabilita.
* **hidden** – ascunde continutul care nu incape in caseta elementului, si impiedica aparitia barei de derulare.
* **scroll** – adauga intotdeauna bare de derulare elementului, pentru ca sa se poata accesa tot continutul.
* **auto** – barele de derulare apar doar atunci cand este necesar.

Iata un exemplu in care toate elementele cu clasa '.extradim' vor avea bare de scroll cand este necesar:

<!DOCTYPE html>

<html>

<head>

<title>Title</title>

<style>

.extradim {

background:#e0e000;

overflow:auto;

height:80px;

width:250px;

}

</style>

</head>

<body>

<div class='extradim'>

<h2>Exemplu cu proprietatea css overflow</h2>

Afisare bara de scroll,<br>

Daca este continut care depaseste<br>

Dimensiunile elementului in care se afla.

</div>

</body>

</html>

Incercati codul

Pentru definirea proprietatii "overflow" doar la una din directii: orizontala sau verticala; adica bara de derulare sa fie disponibila sau nu doar pentru una din aceste directii, se poate folosi o alta varianta a acestei proprietati, si anume:

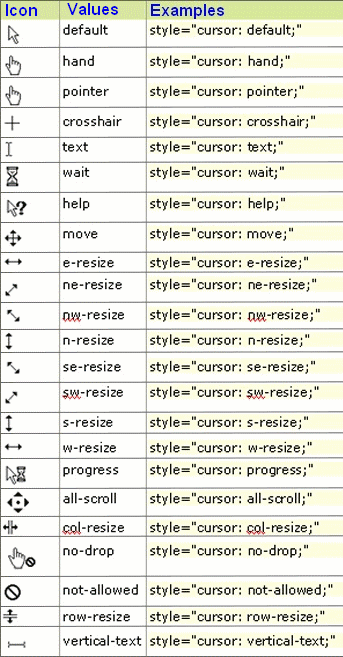
* **overflow-x:valoare** - pentru orizontala
* **overflow-y:valoare** - pentru verticala

- Unde "valoare" poate fi una din valorile prezentate la "overflow".

### Aspectul indicatorului de mouse

In mod implicit, aspectul indicatorului de mouse este determinat de browser. Navigatorul modifica indicatorul de mouse in functie de continutul deasupra caruia se afla acesta.  
Proprietatea **cursor** ajuta la stabilirea aspectului unui indicator de mouse.  
Sintaxa:

**selector { cursor:valoare; }**

Unde "valoare" poate avea urmatoarele nume pentru indicatoarele de mouse:  
  
  
Daca se foloseste valorea "auto", navigatorul decide asupra tipului de indicator folosit.  
  
- Exemplu, cand mouse-ul este peste elementul cu clasa '.cs\_point' va avea aspectul "pointer".

<!DOCTYPE html>

<html>

<head>

<title>Title</title>

<style>

div {

background:#dbdbfb;

height:80px;

width:150px;

}

.cs\_point {

cursor:pointer;

}

</style>

</head>

<body>

<div class='cs\_point'>Pozitionati mouse-ul aici.</div>

</body>

</html>

Incercati codul

# Pseudo-clase CSS

## Curs Css

[**Home**](https://marplo.net/)[**Engleza**](https://marplo.net/engleza)[**Spaniola**](https://marplo.net/spaniola)[**Html**](https://marplo.net/html)[**JavaScript**](https://marplo.net/javascript)[**PHP-MySQL**](https://marplo.net/php-mysql)[**Ajax**](https://marplo.net/ajax)[**Blog**](https://marplo.net/blog)[**Forum**](https://marplo.net/forum/)[**Games**](https://gamv.eu/)

Top of Form



Bottom of Form

**Pseudo-clasele CSS** permit adaugarea de stiluri CSS doar la anumite elemente ale aceluiasi selector, id sau clasa. Acestea se aplica dupa caracterul doua-puncte (:) dupa selectorul CSS.  
Sintaxa pentru utilizare pseudo-clase:

**element:pseudo-clasa { proprietate:valoare; }**

- unde "element" este un selector, id sau clasa, iar "pseudo-clasa" poate fi una din expresiile urmatoare:

* **active** - Adauga un stil unui element cand acesta este activat (*actionat prin click pe el*)
* **first-child** - Adauga un stil unui element care este primul din acel tip de elemente
* **focus** - Folosit pentru input-urile de formulare, le adauga un stil cand acestea sunt active (*dupa click si cursorul de text in ele*)
* **hover** - Adauga un stil unui element cand mouse-ul se afla deasupra lui
* **lang(cuvant)** - Adauga un stil unui element care are atributul: *lang="cuvant"*
* **last-child** - Adauga un stil unui element care este ultimul din tipul specificat
* **link** - Adauga un stil unui link nevizitat
* **visited** - Adauga un stil unui link vizitat

*Sunt mai multe feluri de pseudo-clase, o lista cu acestea gasiti la:*[*Pseudo-classes*](https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-classes)*.*

Ca sa intelegeti mai bine cum functioneaza si ce fac pseudo-clasele, studiati exemplele care sunt prezentate in continuare.

### Pseudo-clase cu selector

Selectoarele fac referire la tag-urile HTML pe care le denumesc (de ex.: **p** pentru <p>, **li** pentru <li>, **div** pentru <div>, etc.).  
- In exemplul urmator este folosit "first-child" pentru paragraf:

<!DOCTYPE html>

<html>

<head>

<title>Title</title>

<style>

p:first-child {

color:blue;

}

</style>

</head>

<body>

<p>Un text din primul paragraf</p>

<p>Continut din al doilea paragraf</p>

<p>Textul din al treilea paragraf</p>

</body>

</html>

Incercati codul

- Rezultatul va fi urmatorul, primul paragraf va avea textul albastru.

Un text din primul paragraf

Continut din al doilea paragraf

Textul din al treilea paragraf

Iata si un exemplu cu "hover" pentru <li>:

<!DOCTYPE html>

<html>

<head>

<title>Title</title>

<style>

li:hover {

background-color:#88fe88;

}

</style>

</head>

<body>

<ul>

<li>MarPlo.net</li>

<li>GamV.eu</li>

<li>CoursesWeb.net</li>

</ul>

</body>

</html>

Incercati codul

- Cand pozitionati mouse-ul deasupra fiecarui LI, acesta va avea background verde, testati mai jos

* MarPlo.net
* GamV.eu
* CoursesWeb.net

### Pseudo-clase si class

Trebuie stiut ca pseudo-clasele nu sunt acelasi lucru ca si clasele, acestea fiind cele care fac referire la valoarea atributului "class" si in CSS se adauga dupa un caracter punct (**.nume\_class**).  
- Iata un exemplu in care sunt folosite pseudo-clasele "lang(cuvant)" si "last-child" la o clasa ".test'.

<!DOCTYPE html>

<html>

<head>

<title>Title</title>

<style>

.test:last-child {

background-color:#88fe88;

}

.test:lang(da) {

color:blue;

}

</style>

</head>

<body>

<ul>

<li class="test">Text in list cu class .test ...</li>

<li class="test" lang="da">Alt LI cu class .test si lang='da' ...</li>

<li class="test">Ultimul LI cu class .test ...</li>

</ul>

</body>

</html>

Incercati codul

- Observati ca stilul definit pentru ".test:last-child" va fi aplicat doar ultimului element care are clasa "test", iar "lang(da)" se aplica numai elementului care pe langa *class="test"* are si *lang="da"*.  
- Rezultatul este acesta:

* Text in list cu class .test ...
* Alt LI cu class .test si lang='da' ...
* Ultimul LI cu class .test ...

Iata un alt exemplu in care primul tag <em> (first-child) din fiecare element cu clasa '.test' este ingrosat:

<!DOCTYPE html>

<html>

<head>

<title>Title</title>

<style>

.test em:first-child {

font-weight:bold

}

</style>

</head>

<body>

<p class="test"><em>Paragraf</em> cu mai multe tag-uri EM: <em>text oblic de la EM</em></p>

<p class="test">Alt paragraf: <em>sir inclinat si bold</em>, alt sir <em>italic de la EM</em></p>

</body>

</html>

Incercati codul

- Rezultat:

***Paragraf*** cu mai multe tag-uri EM: *text oblic de la EM*

Alt paragraf: ***sir inclinat si bold***, alt sir *italic de la EM*

- Daca doriti ca stilul sa fie aplicat tuturor etichetelor <em> din prima clasa, scrieti:   **.clasa:first-child em**  
- Daca vreti sa fie aplicat numai primului <em> din prima clasa, scrieti:   **.clasa:first-child em:first-child**

### Pseudo-clase cu id-uri si elemente de formular

In CSS, id-urile fac referire la valoarea atributului ID si se scriu dupa caracterul **#**.  
- Iata un exemplu in care este folosita pseudo-clasa "hover" impreuna cu un ID, iar "focus" impreuna cu o clasa.

<!DOCTYPE html>

<html>

<head>

<title>Title</title>

<style>

#unid:hover {

background-color:#cbcbfe;

}

.test:focus {

background-color:#ededfe;

}

</style>

</head>

<body>

<form action="#" method="post">

<input type="text" class="test" /><br />

<textarea cols="20" rows="5" class="test"></textarea><br />

<input type="button" value="Buton" id="unid" />

</form>

</body>

</html>

Incercati codul

-Rezultat: cand mouse-ul se afla deasupra butonului (care are id="unid") acesta isi schimba culoarea de fundal, iar cand se da click pe zonele de text din formular (care au class="test"), se modifica culoarea background.

# Pseudo-elemente CSS

## Curs Css

[**Home**](https://marplo.net/)[**Engleza**](https://marplo.net/engleza)[**Spaniola**](https://marplo.net/spaniola)[**Html**](https://marplo.net/html)[**JavaScript**](https://marplo.net/javascript)[**PHP-MySQL**](https://marplo.net/php-mysql)[**Ajax**](https://marplo.net/ajax)[**Blog**](https://marplo.net/blog)[**Forum**](https://marplo.net/forum/)[**Games**](https://gamv.eu/)

Top of Form



Bottom of Form

**Pseudo-elementele CSS** permit adaugarea de stiluri CSS anumitor parti din continutul unui element HTML.  
De exemplu, cand definiti un stil pentru un anumit tag HTML (sau pentru o clasa), tot continutul incadrat de acel tag va avea acelasi stil, iar daca vreti sa adaugati un stil CSS diferit primei litere sau primului rand dintr-un paragraf, se folosesc pseudo-elemente.  
Sintaxa pentru utilizare pseudo-elemente:

**obiect\_css:pseudo-element { proprietate:valoare; }**

- unde "obiect\_css" este un selector, id sau clasa, iar "pseudo-element" este una din expresiile urmatoare:

* **after**   - Adauga continut dupa un element HTML
* **before**   - Adauga continut inaintea unui element HTML)
* **first-letter**   - Adauga un stil css primului caracter dintr-un text
* **first-line**   - Adauga un stil primei linii dintr-un text

Ca sa intelegeti mai bine cum functioneaza si ce fac pseudo-elementele, iata cateva exemple cu fiecare in parte.

### :after

Pentru a adauga un anumit continut (text, imagine, sunet .wav) in HTML prin CSS, se foloseste proprietatea **content** si valoarea ce reprezinta continutul respectiv: 'textul' se adauga intre ghilimele, iar imagine sau sunet .wav se adauga folosind ca valoare **url(adresa\_fisier)**  
- In urmatorul exemplu va fi adaugat un continut text dupa fiecare element <h4>.

<!DOCTYPE html>

<html>

<head>

<title>Title</title>

<style>

h4:after {

content:'MarPlo';

color:blue;

}

</style>

</head>

<body>

<h4>Continutul din primul tag H4 ... </h4>

<h4>Un alt text din alt element H4</h4>

</body>

</html>

Incercati codul

- Rezultatul va fi cel de jos.  
- Observati cum este adaugat cuvantul "MarPlo", imediat dupa ultimul caracter al fiecarui element <h4>, ca si cum ar face parte din textul initial, dar putand avea stil propriu.

#### Continutul din primul tag H4 ... MarPlo

#### Un alt text din alt element H4MarPlo

- In loc de selector puteti folosi si ID sau clasa.

### :before

Pseudo-elementul **Before** este similar cu "after", continutul se adauga la fel, dar la inceputul elementului.  
Pentru o mai mare difersitate, puteti combina pseudo-clasele cu pseudo-elemente folosind sintaxa:

**obiect\_css:pseudo-clasa:pseudo-element { proprietate:valoare; }**

Ca sa vedeti efectul, iata un exemplu aplicat cu "before" acelorasi tag-uri H4 din exemplul precedent.

<!DOCTYPE html>

<html>

<head>

<title>Title</title>

<style>

h4:first-child:before {

content:'MarPlo- ';

color:blue;

}

</style>

</head>

<body>

<h4>Continutul din primul tag H4 ... </h4>

<h4>Un alt text din alt element H4</h4>

</body>

</html>

Incercati codul

- De data aceasta continutul "MarPlo- " este adaugat imediat in fata, doar primului element <h4> (*precizat prin 'first-child'*), dupa cum puteti vedea mai jos:

#### MarPlo- Continutul din primul tag H4 ...

#### Un alt text din alt element H4

- Daca vreti sa adaugati in loc de continut text o imagine, de exemplu .gif, scrieti codul CSS astfel:

h4:first-child:before { content:url(imagine.gif); }

### :first-letter

Pseudo-elementul **first-letter** aplica un stil CSS primului caracter dintr-un text.  
- In urmatorul exemplu este aplicat un stil CSS primului caracter din continutul fiecarui element cu clasa ".test".

<!DOCTYPE html>

<html>

<head>

<title>Title</title>

<style>

.test:first-letter {

color:red;

font-size:25px;

}

</style>

</head>

<body>

<p class="test">Textul din paragraf ...</p>

<div class="test">Alt continut int-un tag DIV, dar cu aceeasi clasa.</div>

</body>

</html>

Incercati codul

- Dupa cum puteti vedea in rezultatul de mai jos, prima litera din continutul fiecarui tag cu class="test" este de culoare rosie si marime 25px.

Textul din paragraf ...

Alt continut int-un tag DIV, dar cu aceeasi clasa.

### :first-line

Pseudo-elementul **First-line** aplica un stil CSS primei linii din continutul elementului HTML.  
- Exemplul:

<!DOCTYPE html>

<html>

<head>

<title>Title</title>

<style>

.test:first-line {

color:blue;

font-weight:bold;

}

</style>

</head>

<body>

<p class="test">Paragraf cu mai multe linii<br />

A doua linie ...<br />

Alta linie din acelasi paragraf.</p>

<div class="test">Continut pe doua linii intr-un tag DIV<br />

A doua linie din DIV.</div>

</body>

</html>

Incercati codul

- Rezultat, prima linie din fiecare element cu class="test" are textul albastru si ingrosat.

**Paragraf cu mai multe linii**  
A doua linie ...  
Alta linie din acelasi paragraf.

**Continut pe doua linii intr-un tag DIV**  
A doua linie din DIV.

# Butoane pentru link-uri folosind o singura imagine si CSS

## Curs Css

[**Home**](https://marplo.net/)[**Engleza**](https://marplo.net/engleza)[**Spaniola**](https://marplo.net/spaniola)[**Html**](https://marplo.net/html)[**JavaScript**](https://marplo.net/javascript)[**PHP-MySQL**](https://marplo.net/php-mysql)[**Ajax**](https://marplo.net/ajax)[**Blog**](https://marplo.net/blog)[**Forum**](https://marplo.net/forum/)[**Games**](https://gamv.eu/)

Top of Form



Bottom of Form

    • [O singura imagine pt meniu cu mai multe link-uri](https://marplo.net/css/butoane-link-folosind-singura-imagine-css#menulinksimg)

Fiecare imagine care trebuie incarcata in pagina necesita o cerere de la browser catre server si asteptarea raspunsului, astfel, daca aveti un meniu cu 5 link-uri cu imagine, pentru a face ca pagina sa se incarce mai repede, e bine sa folositi un singur fisier cu imagini /butoane , nu 15 fisiere (5 \* 3).  
Acest tutorial arata cum se poate utiliza **o singura imagine si CSS** pentru toate cele trei stari ale unui buton /link: **normal, hover (mouse-ul deasupra), si active (cand se apasa clic)**. De asemenea, cum se poate folosi o singura imagine pentru mai multe link-uri intr-un meniu.  
  
Este simplu, ideea este ca desenele butonului pt cele 3 stari sa fie simetric asezate in imagine, vertical: ***top, center, bottom*** (sus, centru, jos); sau orizontal: ***left, center, right*** (stanga, centru, dreapta). Este indicat ca inaltimea imaginii (sau lungimea, in functie de cum sunt pozitionate butoanele in imagine) sa se imparta exact la 3.  
Apoi, in CSS setati proprietatile **width** si **height** ale link-ului cu aceeasi valoare ca lungimea si inaltimea butonului desenat in imagine.

• Daca butoanele pt starile link-ului (normal, hover, active) sunt desenate vertical, precum acestea:  
  
Setati proprietatea CSS **background-position** cu aceste perechi de valori: 0, 0; 0, 50%; si 0, 100% (sau: top, center, bottom); pt fiecare conditie (normal, hover, active). ***Folosind valori in procente, ofera un control mai bun al pozitionarii***.  
- Iata un exemplu (lungimea imaginii exte 125 pixeli, inaltimea 120 pixeli; cu butoanele desenate vertical. Inaltimea butonului /link-ului este 40px):

<!DOCTYPE html>

<html>

<head>

<title>Title</title>

<style>

#addnew {

width: 125px;

height: 40px;

display: block;

background: url('css/button\_img1.gif');

background-position: 0 0;

background-repeat: no-repeat;

border: none;

text-indent: -9999px; /\* Sa nu se afiseze textul din link \*/

}

/\* Cand cursorul de mouse e deasupra butonului \*/

#addnew:hover {

background-position: 0 50%;

}

/\* Cand butonul /link-ul e apasat \*/

#addnew:active {

background-position: 0 100%;

}

</style>

</head>

<body>

<a href="coursesweb.net/" title="Web Programming Courses" id="addnew">CoursesWeb.net</a>

</body>

</html>

Incercati codul

Rezultat (pozitionati cursorul de mouse pe buton, apoi tineti apasat clic ca sa vedeti starea "active"):

CoursesWeb.net

• Daca butoanele pt starile link-ului (normal, hover, active) sunt desenate orizontal, precum acestea:  
image butoane orizontale  
- Setati proprietatea CSS **background-position** cu aceste perechi de valori: 0, 0; 50% 0; si 100% 0 (sau: left, center, right); pt fiecare conditie (normal, hover, active).  
- Iata un exemplu cu butoanele desenate orizontal:

<!DOCTYPE html>

<html>

<head>

<title>Title</title>

<style>

#addnew {

width: 125px;

height: 40px;

display: block;

background: url('css/button\_img2.gif');

background-position: 0 0;

background-repeat: no-repeat;

text-indent: -9999px; /\* Sa nu se afiseze textul din link \*/

}

/\* Cand cursorul de mouse e deasupra butonului \*/

#addnew:hover {

background-position: 50% 0;

}

/\* Cand butonul /link-ul e apasat \*/

#addnew:active {

background-position: 100% 0;

}

</style>

</head>

<body>

<a href="//marplo.net/" title="Cursuri Jocuri Anime" id="addnew">MarPlo.net</a>

</body>

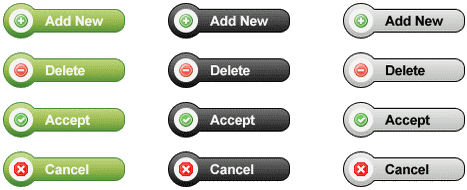
</html>

Incercati codul

Rezultat:

MarPlo.net

### O singura imagine pt meniu cu mai multe link-uri

In mod similar se poate folosi o singura imagine cu butoane pt un meniu cu mai multe link-uri. Trucul este sa desenati butoanele simetric, apoi definiti proprietatea CSS **background-position** in functie de pozitia butoanelor in imagine, folosind procente ce reprezinta distanta fata de coltul stanga-sus, ***perechea de valori: stanga, sus*** (poate fi necesar sa testati mai multe valori pana obtineti afisarea cea mai buna a fiecarui buton).  
- Iata un exemplu, Meniu cu 4 link-uri folosind aceasta imagine:  


<!DOCTYPE html>

<html>

<head>

<title>Title</title>

<style>

/\* Seteaza imaginea pt butoanele din link-uri si proprietatile comune in #menu \*/

#menu a {

width: 125px;

height: 40px;

display: block;

background: url('css/buttons.gif');

background-repeat: no-repeat;

text-indent: -9999px; /\* Sa nu se afiseze textul din link \*/

}

/\* Butonul din imagine pt Primul link \*/

#menu #addnew {

background-position: 0 0;

}

/\* Primul link hover \*/

#menu #addnew:hover {

background-position: 48% 0;

}

/\* Primul link cand e apasat \*/

#menu #addnew:active {

background-position: 100% 0;

}

/\* Butonul din imagine pt al Doilea link \*/

#menu #delete {

background-position: 0 32%;

}

/\* al Doilea link hover \*/

#menu #delete:hover {

background-position: 48% 32%;

}

/\* al Doilea link cand e apasat \*/

#menu #delete:active {

background-position: 100% 32%;

}

/\* Butonul din imagine pt al Treilea link \*/

#menu #accept {

background-position: 0 66%;

}

/\* al Treilea link hover \*/

#menu #accept:hover {

background-position: 48% 66%;

}

/\* al Treilea link cand e apasat \*/

#menu #accept:active {

background-position: 100% 66%;

}

/\* Butonul din imagine pt al patrulea link \*/

#menu #cancel {

background-position: 0 99%;

}

/\* al patrulea link hover \*/

#menu #cancel:hover {

background-position: 48% 99%;

}

/\* al patrulea link cand e apasat \*/

#menu #cancel:active {

background-position: 100% 99%;

}

</style>

</head>

<body>

<nav id="menu">

<a href="#" title="Add New" id="addnew">Add New</a>

<a href="#" title="Delete" id="delete">Delete</a>

<a href="#" title="Accept" id="accept">Accept</a>

<a href="#" title="Cancel" id="cancel">Cancel</a>

</nav>

</body>

</html>

Incercati codul

Rezultat (pozitionati cursorul de mouse pe buton, apoi tineti apasat clic ca sa vedeti starea "active"):

# CSS3 - Background proprietati noi

## Curs Css

[**Home**](https://marplo.net/)[**Engleza**](https://marplo.net/engleza)[**Spaniola**](https://marplo.net/spaniola)[**Html**](https://marplo.net/html)[**JavaScript**](https://marplo.net/javascript)[**PHP-MySQL**](https://marplo.net/php-mysql)[**Ajax**](https://marplo.net/ajax)[**Blog**](https://marplo.net/blog)[**Forum**](https://marplo.net/forum/)[**Games**](https://gamv.eu/)

Top of Form



Bottom of Form

* [Culori Gradient](https://marplo.net/css/css3-background-proprietati-noi#gradient)
* [background-origin](https://marplo.net/css/css3-background-proprietati-noi#origin)
* [background-clip](https://marplo.net/css/css3-background-proprietati-noi#clip)
* [Imagini Multiple la Background](https://marplo.net/css/css3-background-proprietati-noi#multiple)

CSS3 contine trei noi proprietati pentru background: background-size, background-origin si background-clip, in plus, se pot adauga mai multe imagini pt. background la un element HTML si se pot crea culori gradient cu: linear-gradient.

### background-size

Proprietatea CSS **background-size: width height;** permite definirea lungimii si inaltimii (width height) de afisare a imaginii pt. background.  
  
- Valori:

* **auto** - (default) Seteaza dimensiunea laturii astfel incat sa se mentina raportul de aspect al imaginii. Daca ambele valori sunt "auto", imaginea apare la dimensiunile ei.
* **percentage** - Seteaza dimensiunile de background a imaginii in procente raportate la elementul parinte (nu la imagine). Prima valoare pt lungime (width), a doua pt. inaltime (height).  
  Daca se adauga o singura valoare, a doua e considerata "auto".
* **pixels** - Seteaza lungimea si inaltimea de afisare a imaginii pt. background la marimile specificate.  
  Daca se adauga o singura valoare, a doua e considerata "auto".
* **cover** - Imaginea de background e dimensionata astfel incat sa umple complet zona de fundal.
* **contain** - Dimensioneaza imaginea la marimea ce mai mare astfel incat sa fie continuta in suprafata de fundal.

- Exemplu:

<style>

#id1 {

width:200px;

height:100px;

border:1px solid blue;

background:url('css/css3.jpg');

background-size:100px 60px;

background-repeat:no-repeat;

margin:5px auto;

}

#id2 {

width:300px;

height:120px;

border:1px solid #01da02;

background:url('css/css3.jpg');

background-size:contain;

background-repeat:no-repeat;

margin:5px auto;

}

#id3 {

width:300px;

height:120px;

border:1px solid silver;

background:url('css/css3.jpg');

background-size:cover;

margin:5px auto;

}

</style>

<div id="id1">Free CSS Course - size in pixels</div>

<div id="id2">coursesweb.net - size contain</div>

<div id="id3">marplo.net - size cover</div>

Incercati codul

Rezultat:



Free CSS Course - size in pixels



coursesweb.net - size contain

marplo.net - size cover

### Culori Gradient

Pentru adaugare culoare gradient la fundal, se foloseste proprietatea **background: linear-gradient();** cu aceasta sintaxa:

**background: linear-gradient(direction, color-stop1, color-stop2, ...);**

- "direction" reprezinta directia de afisarie a culorilor: "to right", "to bottom right", "top left", "top bottom" (default). Daca nu e specificata, se considera "top bottom".  
- "color-stop" sunt culorile de referinta pt. gradient.  
  
Exemplu:

<style>

#id1 {

background: linear-gradient(to bottom right, #11ee22, #fefe88, #1122fb);

height:120px;

width:300px;

}

</style>

<div id="id1">Free CSS Course<br />

coursesweb.net</div>

Incercati codul

Rezultat:

Free CSS Course  
coursesweb.net

Se poate defini si distanta de referinta a culorilor din gradient, adaugand un procent dupa valoarea culorii.  
- Exemplu:

<style>

#id1 {

background: linear-gradient(to bottom right, #11ee22 0%, #fefe88 40%, #1122fb 75%);

height:120px;

width:300px;

}

</style>

<div id="id1">Free CSS Course<br />

coursesweb.net</div>

Incercati codul

### background-origin

Proprietatea CSS **background-origin** specifica zona de pozitionare a imaginii de fundal.  
Preia una din aceste valori:

* **padding-box** - pozitionarea e relativa la coltul din stanga-sus a marginii de padding.
* **border-box** - pozitionarea e relativa la coltul din stanga-sus a bordurii.
* **content-box** - background-ul incepe din coltul stanga-sus a continutului.

- Exemplu:

<style>

#id1 {

width:350px;

height:120px;

padding:20px;

border:3px solid blue;

background:url('css/css3.jpg');

background-repeat:no-repeat;

background-color:#bcfede;

background-origin:content-box;

}

#id2 {

width:350px;

height:120px;

padding:20px;

border:3px solid blue;

background:url('css/css3.jpg');

background-repeat:no-repeat;

background-color:#bcfede;

background-origin:border-box;

margin-top:5px;

}

</style>

<div id="id1">Position the background image within the content-box</div>

<div id="id2">Position the background image relative to the border</div>

Incercati codul

Rezultat:

Position the background image within the content-box

Position the background image relative to the border

### background-clip

Proprietatea **background-clip** se foloseste pentru a determina suprafata de afisare a culorii pentru fundal ( background-color ).  
Preia una din aceste valori:

* **border-box** - culoare de fundal se extinde pana la bordura.
* **padding-box** - culoarea de fundal se extinde la padding.
* **content-box** - culoarea de fundal apare doar in suprafata ocupata de continut.

Exemplu:

<style>

#id1 {

width:350px;

height:120px;

padding:20px;

border:2px solid blue;

background-color:#cdfeda;

background-clip:padding-box;

}

#id2 {

width:350px;

height:120px;

padding:20px;

border:2px solid blue;

background-color:#cdfeda;

background-clip:content-box;

}

</style>

<div id="id1">background-clip with padding-box</div>

<div id="id2">background-clip with content-box</div>

Incercati codul

Rezultat:

background-clip with padding-box

background-clip with content-box

### Imagini Multiple la Background

Ca sa aplicati mai multe imagini la fundalul unui element HTML, se adauga la proprietatea css **background-image** functii **url()** (separate prin virgula) cu adresele imaginilor.  
- Celelalte proprietati legate de "background-image" (background-position, background-repeat) folosesc valori separate prin virgula, in functie de modul dorit de afisare a imaginilor adaugate. Sau o singura valoare care se aplica la toate.  
  
- Exemplu:

<style>

#id1 {

width:400px;

height:150px;

background-image: url('css/html\_course.jpg'), url('css/css3.jpg');

background-repeat: no-repeat, repeat-x;

background-position: center top, center bottom;

}

</style>

<div id="id1">Multiple background images</div>

Incercati codul

Rezultat:

Multiple background images

# SS3 - Border proprietati noi

## Curs Css

[**Home**](https://marplo.net/)[**Engleza**](https://marplo.net/engleza)[**Spaniola**](https://marplo.net/spaniola)[**Html**](https://marplo.net/html)[**JavaScript**](https://marplo.net/javascript)[**PHP-MySQL**](https://marplo.net/php-mysql)[**Ajax**](https://marplo.net/ajax)[**Blog**](https://marplo.net/blog)[**Forum**](https://marplo.net/forum/)[**Games**](https://gamv.eu/)

Top of Form



Bottom of Form

* [Adaugare umbre la chenare](https://marplo.net/css/css3-border-proprietati-noi#shadow)
* [CSS3 Imagine pentru bordura](https://marplo.net/css/css3-border-proprietati-noi#image)

CSS3 contine caracteristici care permit crearea de colturi rotunjite, umbre la chenare si folosirea unei imagini pentru aspectul bordurii.

### CSS3 Colturi rotunjite

Proprietatea CSS border-radius permite crearea cu usurinta a colturilor rotunjite in designul elementelor, fara a fi nevoie de imagini sau mai multe tag-uri <div>.  
  
Exemplu:

<style>

#id1 {

width:300px;

height:120px;

border:2px solid blue;

border-radius:28px;

text-align:center;

}

</style>

<div id="id1"> Curs CSS - marplo.net</div>

Incercati codul

Rezultat:

Curs CSS - marplo.net

Colturile rotunjite pot fi create si independent, folosind cele patru proprietati individuale: border-top-left-radius, border-top-right-radius, border-bottom-right-radius, si border-bottom-left-radius.  
Exemplu:

<style>

#id1 {

width:300px;

height:120px;

background-color:#bbfeda;

border:2px solid blue;

border-top-left-radius:20px;

border-bottom-right-radius:38px;

text-align:center;

}

</style>

<div id="id1"> Curs CSS - marplo.net</div>

Incercati codul

Rezultat:

Curs CSS - marplo.net

### Adaugare umbre la chenare

Proprietatea box-shadow se foloseste pentru a adauga umbre la chenare. Este recunoscuta in IE9+, Firefox 4, Chrome, si Opera.  
box-shadow are urmatoarea sintaxa:

elm { box-shadow: X\_offset Y\_offset blur marime culoare inset; }

- *elm* - este elementul HTML la care se aplica aceasta proprietate.  
- *X\_offset* - este pozitia /distanta umbrei pe latura orizontala. Sunt permise si valori negative.  
- *Y\_offset* - este pozitia /distanta umbrei pe latura verticala. Sunt permise si valori negative.  
- *blur* - defineste distanta "blur" (optionala, 0 sau nespecificata inseamna fara "blur").  
- *marime* - marimea umbrei (optionala).  
- *culoare* - culoarea umbrei (optionala, negru daca nu e specificata).  
- *inset* - schimba umbra din exterior sa apara in interior.  
  
Primele doua valori trebuie adaugate, restul sunt optionale. Daca "blur" sau marime nu sunt specificate, se considera 0.  
  
Exemplu:

<style>

#id1 {

width:300px;

height:120px;

background-color:#bbfeda;

box-shadow: 11px 11px 5px #7878da;

-webkit-box-shadow: 11px 11px 5px #7878da; /\* Safari si Chrome \*/

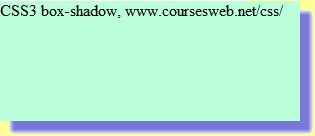
}

</style>

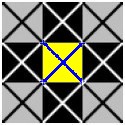
<div id="id1"> CSS3 box-shadow, www.coursesweb.net/css/</div>

Incercati codul

Rezultat:



### CSS3 Imagine pentru bordura

Cu proprietatea CSS border-image se poate folosi o imagine pentru aspectul bordurii.  
Aceasta se obtine prin impartirea zonei bordurii in noua segmente, reprezentand cele patru colturi, cele patru laturi si centrul (dupa cum se vede in imaginea de mai jos). Se foloseste un singur fisier cu imaginea, impartit similar in cele noua segmente care vor reprezenta colturile, laturile si centrul.  
  
box-shadow are urmatoarea sintaxa:

elm { border-image: url\_img slice width outset repeat; }

- *url\_img* - adresa si numele imaginii folosita pt. bordura.  
- *slice* - specifica distanta pentru zona din imagine folosita in cele patru margini.  
- *width* - lungimea bordurii cu imaginea.  
- *outset* - specifica distanta fata de marginea chenarului, de unde incepe bordura cu imagine.  
- *repeat* - specifica daca partea de imagine trebuie repetata, rotunjita ca incadrare, sau intinsa (poate avea aceste valori: *stretch, repeat, round*).

*- Ca sa functioneze, este necesar sa adaugati si proprietatea: border cu valoarea "****transparent****" la culoare. - border-image e recunoscut in Internet Explorer 11+.*

In exemplul urmator se foloseste aceasta imagine: 

<style>

#id1 {

width:200px;

height:120px;

border: 12px solid transparent;

border-image:url('css/border\_image.png') 30 30 round;

}

#id2 {

width:200px;

height:120px;

border: 15px solid transparent;

border-image:url('css/border\_image.png') 30 30 stretch;

}

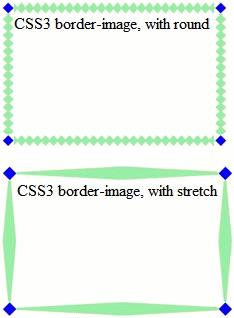
</style>

<div id="id1"> CSS3 border-image, with round</div><br>

<div id="id2"> CSS3 border-image, with stretch</div>

Incercati codul

Rezultat:



# CSS3 - text-shadow, word-wrap, text-overflow

## Curs Css

[**Home**](https://marplo.net/)[**Engleza**](https://marplo.net/engleza)[**Spaniola**](https://marplo.net/spaniola)[**Html**](https://marplo.net/html)[**JavaScript**](https://marplo.net/javascript)[**PHP-MySQL**](https://marplo.net/php-mysql)[**Ajax**](https://marplo.net/ajax)[**Blog**](https://marplo.net/blog)[**Forum**](https://marplo.net/forum/)[**Games**](https://gamv.eu/)

Top of Form



Bottom of Form

CSS3 introduce noi proprietati pentru efecte de text, in aceasta lectie sunt prezentate: text-shadow, word-wrap, si text-overflow.

### CSS3 text-shadow

Proprietatea **text-shadow** introdusa in CSS3 permite adaugarea unui efect de umbra care sa fie aplicat la textul dintr-un element HTML. Umbra apare in jurul literelor.  
Sintaxa:

**text-shadow: offset\_X offset\_Y blur culoare;**

- *offset\_X* - specifica pozitia umbrei pe orizontala. Sunt permise si valori negative.  
- *offset\_Y* - specifica pozitia umbrei pe verticala. Sunt permise si valori negative.  
- *blur* - defineste distanta pentru valoarea "blur" (optional).  
- *culoare* - defineste culoarea umbrei (optional). Daca nu e specificata, se foloseste negru.  
  
Exemplu:

<style>

h2 {

text-shadow: 2px 3px 3px #a0a1fe;

}

</style>

<h2>Text cu text-shadow</h2>

Incercati codul

Rezultat:

**Text cu text-shadow**

Se pot adauga mai multe umbre la acelasi text, prin adaugarea mai multor liste de umbre in proprietatea text-shadow, separate prin virgula.  
Urmatorul cod amesteca o culoare verde cu una albastra ca sa creeze un efect de umbra cu aceste culori:

<style>

h2 {

text-shadow: -1px -2px rgba(10,5,205,0.2), 2px 3px rgba(200,0,1,0.3);

}

</style>

<h2>Text cu doua culori pt. text-shadow</h2>

Incercati codul

- Observati ca s-a folosit formula *rgba(Rosu, Verde, Albastru, Alpha)* la definirea culorii, adaugand si transparenta (Alpha) in acelasi timp.  
Rezultat:

**Text cu doua culori pt. text-shadow**

### CSS3 word-wrap

Cu **word-wrap** puteti determina browser-ul sa rupa cuvintele lungi, cand acestea depasesc lungimea elementului, trecand pe un rand nou.  
Sintaxa:

**word-wrap: valoare;**

"valoare" poate fi:  
  - *normal* - Nu rupe cuvintele intregi (valoarea prestabilita).  
  - *break-word* - Permite ruperea cuvintelor intregi. Cuvintele sunt rupte la nivel de caracter, nu in silabe.  
  
Exemplu:

<style>

#id1 {

width:100px;

border:1px solid blue;

word-wrap:break-word;

}

</style>

<div id="id1">Curs CSS gratuit - word-wrap un\_cuvant\_mai\_lung.</div>

Incercati codul

Rezultat:

Curs CSS gratuit - word-wrap un\_cuvant\_mai\_lung.

### CSS3 text-overflow

Cu proprietatea **text-overflow** puteti determina ce sa se intample cu textul, cand acesta depaseste lungimea elementului.  
Sintaxa:

**text-overflow: valoare;**

"valoare" poate fi:  
  - *clip* - taie textul (valoarea prestabilita).  
  - *ellipsis* - afiseaza trei-puncte ("…") in locul textului taiat.  
  
• In general, proprietatea ***text-overflow*** se foloseste impreuna cu **white-space:nowrap;** si **overflow:hidden;**.  
Exemplu:

<style>

#id1 {

width:230px;

border:1px solid blue;

white-space:nowrap;

overflow:hidden; /\* "overflow" valoarea trebuie sa fie diferita de "visible" \*/

text-overflow:ellipsis;

}

#id2 {

width:230px;

border:1px solid green;

white-space:nowrap;

overflow:hidden;

text-overflow:clip;

}

</style>

<div id="id1">CSS Tutorial - un text lung pe o singura linie, alte cuvinte.</div>

<div id="id2">Site web marplo.net - alt text lung intr-un singur rand.</div>

Incercati codul

Rezultat:

CSS Tutorial - un text lung pe o singura linie, alte cuvinte.

Site web marplo.net - alt text lung intr-un singur rand.

# CSS3 opacity

## Curs Css

[**Home**](https://marplo.net/)[**Engleza**](https://marplo.net/engleza)[**Spaniola**](https://marplo.net/spaniola)[**Html**](https://marplo.net/html)[**JavaScript**](https://marplo.net/javascript)[**PHP-MySQL**](https://marplo.net/php-mysql)[**Ajax**](https://marplo.net/ajax)[**Blog**](https://marplo.net/blog)[**Forum**](https://marplo.net/forum/)[**Games**](https://gamv.eu/)

Top of Form



Bottom of Form

Proprietatea **opacity** seteaza nivelul de opacitate al unui element HTML, **opacity** permite modificarea transparentei elementului.

- Sintaxa:

**opacity: X;**

- X - reprezinta o valoare intre 0 (complet transparent) si 1 (fara transparenta).

## Exemple cu CSS opacity

1) Exemplu, un <div> cu o transparenta CSS de 40%:

<style>

#dv {

background-color:#8899fe;

width:200px;

height:100px;

opacity:0.4;

}

</style>

<div id="dv">Un continut oarecare ...</div>

Incercati codul

Rezultat:

Un continut oarecare ...

2) Exemplu, seteaza opacitate 50% la o imagine:

<style>

#im {

opacity:0.5;

}

</style>

<img src="css/html\_course.jpg" alt="Curs HTML si CSS" width="155" height="160" id="im" />

Incercati codul

Rezultat:



3) Exemplu, efect imagine transparenta 50%, iar la mouseover 100%:

<style>

#im2 {

opacity:0.5;

}

#im2:hover {

opacity:1;

}

</style>

<img src="css/html\_course.jpg" alt="Curs HTML si CSS" width="155" height="160" id="im2" />

Incercati codul

Rezultat (*pozitionati mouse-ul peste imaginea de mai jos*):



4) Continut intr-un DIV care la mouseover este acoperit cu un alt <div> transparent (*DIV-ul transparent trebuie sa fie gol, adaugat in primul, si setat cu:   position:absolute;*):

<style>

#trans {

display:none;

position:absolute;

top:2%;

left:2%;

width:96%;

height:95%;

background-color:#07fe08;

opacity:0.5;

}

#cnt {

position:relative;

background:#e7e8fe;

width:300px;

height:200px;

border:1px solid blue;

padding:20px;

}

#cnt:hover #trans { display:block; }

</style>

<div id="cnt">

<div id="trans"></div>

Plasati mouse-ul peste aceasta caseta pentru a vedea efectul de opacitate.

<img src="css/css3.jpg" alt="CSS3 opacity" width="160" height="98" />

</div>

Incercati codul

Rezultat:

**Plasati mouse-ul peste aceasta caseta pentru a vedea efectul de opacitate.** 

# CSS3 Transformari 2D

## Curs Css

[**Home**](https://marplo.net/)[**Engleza**](https://marplo.net/engleza)[**Spaniola**](https://marplo.net/spaniola)[**Html**](https://marplo.net/html)[**JavaScript**](https://marplo.net/javascript)[**PHP-MySQL**](https://marplo.net/php-mysql)[**Ajax**](https://marplo.net/ajax)[**Blog**](https://marplo.net/blog)[**Forum**](https://marplo.net/forum/)[**Games**](https://gamv.eu/)

Top of Form



Bottom of Form

* [CSS rotate()](https://marplo.net/css/css3-transformari-2d#rotate)
* [Metoda scale()](https://marplo.net/css/css3-transformari-2d#scale)
* [CSS skew()](https://marplo.net/css/css3-transformari-2d#skew)

Cu proprietatea **CSS transform** se poate modifica forma, marimea si pozitia elementelor HTML.  
Se pot transforma elementele HTML in plan 2D sau 3D.  
- Sintaxa:

**transform: metoda(valori);**

- Acest tutorial prezinta metodele de transformare in plan 2D:   **translate(), rotate(), scale(), skew()** .

### Metoda translate()

Metoda **translate()** muta elementul de la pozitia initiala, in functie de parametri dati pentru pozitionare de la Stanga (axa-X) si de Sus (axa-Y) .  
- Sintaxa:

**transform: translate(X, Y);**

Exemplu. Cand utilizatorul plaseaza cursorul mouse-ului pe un anumit Div, ii muta pozitia cu 20 pixeli de la stanga si 15 pixeli din partea de sus.

<style>

#idv {

width:90px;

height:90px;

background:#b0b1fe;

font-size:17px;

}

#idv:hover {

transform: translate(20px, 15px);

}

</style>

<div id="idv">Pozitionati mouse-ul aici.</div>

Incercati codul

Demo:

Pozitionati mouse-ul aici.

### CSS rotate()

Metoda **rotate()** roteste elementul HTML, in sensul orelor de ceas daca valoarea e pozitiva, in sens invers orelor de ceas daca valoarea e negativa.  
- Sintaxa:

**transform: rotate(grade);**

Exemplu. Roteste element cu 60 grade in sensul orelor de ceas.

<style>

#idv2 {

width:90px;

height:90px;

background:#b0b1fe;

font-size:17px;

}

#idv2:hover {

transform: rotate(60deg);

}

</style>

<div id="idv2">Pozitionati mouse-ul aici.</div>

Incercati codul

Demo:

Pozitionati mouse-ul aici.

### Metoda scale()

Metoda **scale()** mareste sau micsoreaza marimea elementului HTML (inclusiv continutul din el), in functie de parametri pentru Lungime (axa-X) si Inaltime (axa-Y).  
- Sintaxa:

**transform: scale(Lungime, Inaltime);**

- Valorile pentru Lungime si Inaltime sunt in procente. De exemplu, 1.5 inseamna 150% din marimea originala.  
  
Exemplu. Transforma lungimea sa fie de 2 ori marimea originala, si inaltimea 1.5 ori inaltimea originala.

<style>

#idv3 {

width:90px;

height:90px;

background:#b0b1fe;

font-size:17px;

margin:40px 0 0 50px;

}

#idv3:hover {

transform: scale(2, 1.5);

}

</style>

<div id="idv3">Pozitionati mouse-ul aici.</div>

Incercati codul

Demo:

Pozitionati mouse-ul aici.

### CSS skew()

Metoda skew() distorsioneaza elementul HTML pe orizontala (axa-X) si verticala (axa-Y), incluzand si continutul din el.  
- Sintaxa:

**transform: skew(Xdeg, Ydeg);**

Exemplu. Distorsioneaza elementul 20 grade pe lungime (axa-X) si 25 grade pe verticala (axa-Y).

<style>

#idv4 {

width:160px;

height:90px;

background:#abcdfe;

font-size:18px;

margin:40px 0 0 30px;

transform: skew(20deg, 25deg);

}

</style>

<div id="idv4">//coursesweb.net</div>

Incercati codul

Demo:

//coursesweb.net

• Cele 4 metode de transformare pot fi utilizate si intr-o singura definitie **transform**, separate prin spatiu.  
- Sintaxa:

**transform: translate(X, Y) rotate(grade) scale(Lungime, Inaltime) skew(Xdeg, Ydeg);**

- Puteti adauga doar acele metode pe care doriti sa le folositi in transformare.  
  
Exemplu, muta elementul cu 50 pixeli de la stanga si 25 pixeli din partea de sus, il roteste cu 20 grade in sens invers orelor de ceas, transforma lungimea de 2 ori cea originala si inaltimea 1.5 ori, distorsioneaza elementul cu 15 grade pe orizontala si 20 grade pe verticala.

<style>

#idv5 {

width:90px;

height:90px;

background:#00da01;

font-size:17px;

margin:40px 0 0 30px;

transform: translate(50px, 25px) rotate(-20deg) scale(2, 1.5) skew(15deg, 20deg);

}

</style>

<div id="idv5">Viata este Fericire.</div>

Incercati codul

Demo:

Viata este Fericire.

# CSS3 transition

## Curs Css

[**Home**](https://marplo.net/)[**Engleza**](https://marplo.net/engleza)[**Spaniola**](https://marplo.net/spaniola)[**Html**](https://marplo.net/html)[**JavaScript**](https://marplo.net/javascript)[**PHP-MySQL**](https://marplo.net/php-mysql)[**Ajax**](https://marplo.net/ajax)[**Blog**](https://marplo.net/blog)[**Forum**](https://marplo.net/forum/)[**Games**](https://gamv.eu/)

Top of Form



Bottom of Form

CSS3 **transition** poate fi utilizat pentru a anima proprietatile CSS, adaugand un efect de animatie cand se modifica proprietatile CSS ale unui element HTML, schimband gradual de la un stil la altut.  
CSS3 transition are 4 componente:

* **transition-property** - Specifica numele proprietatii (sau proprietatilor) la care transition trebuie aplicat (precum: width, color, font-size, etc.).  
        CSS3 transition poate fi utilizat cu proprietatile prezentate aici: [Proprietati CSS ce pot fi animate](https://marplo.net/css/css3-transition#cssap)
* **transition-duration** - Specifica durata tranzitiei (animatiei), in sesunde (s), milisecunde (ms), (implicit 0).
* **transition-timing-function** - Defineste tipul vitezei efectului in timpul transformarii:  
        **ease** (implicit), **linear, ease-in, ease-out, ease-in-out**
* **transition-delay** - Defineste timpul de asteptare pana cand incepe efectul "transition" (implicit 0)

Pentru a folosi CSS3 transition, trebuie specificate aceste doua lucruri:  
      1. Proprietatea CSS la care se aplica efectul ( **transition-property** ).  
      2. Durata efectului ( **transition-duration** ).  
- Ultimele doua componente ( transition-timing-function si transition-delay ) sunt optionale.  
  
Exemplu. Cand mouse-ul e deasupra unui anumit Div, se schimba gradual lungimea.

<style>

#iddv {

width:80px;

height:80px;

background:#b8b9fe;

font-size:17px;

transition-property: width;

transition-duration: 1.4s;

}

#iddv:hover {

width:200px;

}

</style>

<div id="iddv">Pozitionati mouse-ul pe acest patrat.</div>

Incercati codul

Demo (Cand cursorul mouse-ului iese din zona elementului, revine gradual la stilul initial):

Pozitionati mouse-ul pe acest patrat.

### Proprietatea transition prescurtata

Se pot adauga toate cele 4 componente intr-o singura proprietate **transition**.  
Sintaxa:

**transition: property duration timing-function delay;**

Exemplu: Schimba gradual "font-size" (in 0.4 secunde, cu viteza "ease-out") cand mouse-ul e deasupra unui element cu class="clse".

<style>

.clse {

width:200px;

font-size:13px;

transition: font-size 0.4s ease-out;

}

.clse:hover {

font-size: 16px;

}

</style>

<ul>

<li class="clse"><a href="//marplo.net/css" title="Curs CSS gratuit">Curs CSS gratuit</a></li>

<li class="clse"><a href="//marplo.net/html" title="Curs HTML">Curs HTML</a></li>

<li class="clse"><a href="//coursesweb.net/" title="Cursuri Programare Web">Cursuri Programare Web</a></li>

</ul>

Incercati codul

• Se pot adauga mai multe proprietati pt efect "transition" intr-o singura definitie, separate prin virgula.  
Exemplu. Adaugare efect la: background, opacity si transform:

<style>

.dv1 {

width:115px;

height:100px;

position:relative;

font-size:17px;

text-align:center;

padding-top:18px;

}

.clsdv {

width:120px;

height:100px;

position:absolute;

top:0;

left:0;

background:#b8b9fe;

transition: background 1.3s, opacity 1.8s, transform 1.4s;

}

.clsdv:hover {

background: #00da01;

opacity:0.5;

transform:rotate(180deg);

}

</style>

Puneti mouse-ul deasupra acestui dreptunghi.

<div class="dv1">

<div class="clsdv"></div>

Text oarecare, ascuns

</div><style>

.dv1 {

width:115px;

height:100px;

position:relative;

font-size:17px;

text-align:center;

padding-top:18px;

}

.clsdv {

width:120px;

height:100px;

position:absolute;

top:0;

left:0;

background:#b8b9fe;

transition: background 1.3s, opacity 1.8s, transform 1.4s;

}

.clsdv:hover {

background: #00da01;

opacity:0.5;

transform:rotate(180deg);

}

</style>

Puneti mouse-ul deasupra acestui dreptunghi.

<div class="dv1">

<div class="clsdv"></div>

Text oarecare, ascuns

</div>

Incercati codul

Demo:

Puneti mouse-ul deasupra acestui dreptunghi.

Text oarecare, ascuns

### Proprietati CSS ce pot fi animate

Lista cu proprietati CSS care pot fi utilizate in transition:  
  
- **Proprietati pt text**:   color, font-size, font-weight, letter-spacing, line-height, text-indent, text-shadow, vertical-align, word-spacing.  
- **Proprietati pt elemente tip bloc**:   background, background-color, background-image, background-position, border-left-color etc., border-spacing, border-left-width etc., clip, crop, height, min-height, max-height, margin-left etc., opacity, outline-width, outline-offset, outline-color, padding-left etc., width, min-width, max-width.  
- **Proprietati de pozitionare**:   bottom, top, left, right, grid-, visibility, z-index, zoom.

*• O alta metoda de a anima proprietatile unui element HTML cu CSS este utilizand proprietatile "animation". Vedeti tutorialul:*[***Animatie elemente HTML cu CSS***](https://marplo.net/css/animatie-html-css)*.*

# CSS3 Flexbox Container

## Curs Css

[**Home**](https://marplo.net/)[**Engleza**](https://marplo.net/engleza)[**Spaniola**](https://marplo.net/spaniola)[**Html**](https://marplo.net/html)[**JavaScript**](https://marplo.net/javascript)[**PHP-MySQL**](https://marplo.net/php-mysql)[**Ajax**](https://marplo.net/ajax)[**Blog**](https://marplo.net/blog)[**Forum**](https://marplo.net/forum/)[**Games**](https://gamv.eu/)

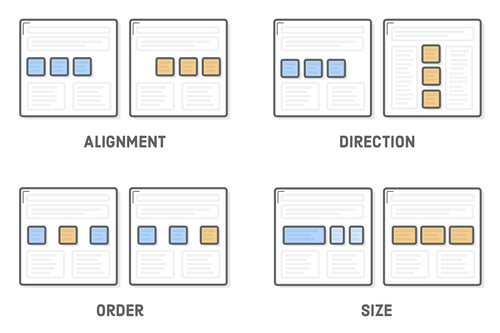
Top of Form



Bottom of Form

- [flex-direction](https://marplo.net/css/css3-flexbox-container#anc_fd)  
- [flex-wrap](https://marplo.net/css/css3-flexbox-container#anc_fw)  
- [flex-flow](https://marplo.net/css/css3-flexbox-container#anc_ff)  
- [justify-content](https://marplo.net/css/css3-flexbox-container#anc_fjc)  
- [align-items](https://marplo.net/css/css3-flexbox-container#anc_fai)  
- [align-content](https://marplo.net/css/css3-flexbox-container#anc_fac)  
- [Testare proprietati Flexbox](https://marplo.net/css/css3-flexbox-container#test_flexbox)

**Flexbox** (sau Flexibil container) este un nou mod in CSS3 de amplasare elemente in pagina, destinate sa se ajusteze la diferite dimensiuni si dispozitive de afisare.  
Acest model de amplasare flexibil nu foloseste 'float', si nici nu face continutul sa depaseasca marginile.  
Flexbox ofera control pentru aliniere, directie, ordonare si marime la elementele din container.

  
- Acest tutorial prezinta **exemple si proprietati css3 pentru containerul Flex**.

## Creare container flexbox

Flexbox consta din container flexibil si elemente interioare flexibile.  
Pentru a crea un flex container, setati proprietatea **display** cu valoarea **flex** (redata ca block) sau **inline-flex** (redata ca liniara).  
Fiecare element dintr-un flex container devine un flex-item. Si textul direct dintr-un container e considerat un flex-item.

<style>

.container {

display: ; /\* or: inline-flex \*/

margin:2px;

padding:3px;

}

.container .item {

background:#ccccfe;

margin:2px;

}

</style>

- Content Before..

<div class='container'>

Parent content.

<div class='item'>Child 1</div>

<div class='item'>Child\_2</div>

<div class='item'>Child-3</div>

</div>

- Content After..

Result:

- Content Before..

Parent content.

Flex item #1

Child 2

Flexbox child 3

- Content After..

## Proprietati ale containerului flex

### flex-direction

Proprietatea **flex-direction** specifica directia elementelor din flex container. Pot fi aranjate orizontal in linie, sau vertical pe coloane.  
- Valori pt. **flex-direction**:

* **row** - (default) flex-items sunt aranjate in linie, de la stanga la dreapta (in context ***ltr***).
* **row-reverse** - flex-items sunt aranjate in linie, in sens invers, de la dreapta la stanga (in context ***ltr***).
* **column** - flex-items sunt aranjate pe coloane, de sus in jos.
* **column-reverse** - flex-items sunt aranjate pe coloane, invers, de jos in sus.

  
Exemplu cu inline-flex si column:

.container {

display: inline-flex;

flex-direction: column;

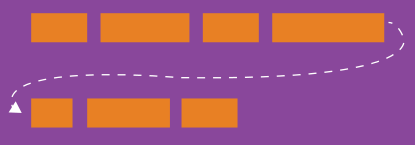
}

Test Flexbox example

### flex-wrap

In mod initial containerul flex pune toate elementele din el pe o singura linie.  
Proprietatea **flex-wrap** seteaza posibilitatea de a afisa elementele pe mai multe linii (daca pe o linie depasesc lungimea containerului), si directia lor.  
- Valori pt. **flex-wrap**:

* **nowrap** - (default) toate elementele din container flex sunt pe o singura linie.
* **wrap** - flex-items vor fi aranjate pe mai multe linii de sus in jos.
* **wrap-reverse** - flex-items vor fi aranjate pe mai multe linii, invers, de jos in sus.



.container {

display: flex;

flex-wrap: wrap;

}

Test Flexbox example

### flex-flow

**flex-wrap** seteaza intr-o singura proprietate valorile pentru flex-direction si flex-wrap. Default: ***row nowrap***  
- Sintaxa **flex-flow**:

**flex-flow: flex-direction flex-wrap;**

.container {

display: flex;

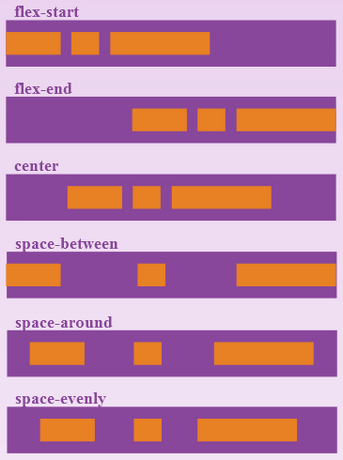
flex-flow: row wrap;

}

### justify-content

**justify-content** seteaza aranjarea pe orizontala a elementelor din flex container cand acestea nu ocupa tot spatiul din el. Distribue spatiul ramas liber.  
- Valori pt. **justify-content**:

* **flex-start** - (default) elementele sunt pozitionate la inceputul containerului.
* **flex-end** - elementele sunt pozitionate la sfarsitul containerului.
* **center** - elementele sunt pozitionate in centrul containerului.
* **space-between** - elementele sunt pozitionate cu spatiu egal intre ele, primul si ultimul flex-item sunt la marginile containerului.
* **space-around** - elementele sunt pozitionate cu spatiu egal in jurul lor, inclusiv primul si ultimul flex-item.
* **space-evenly** - elementele sunt pozitionate cu spatiu egal intre ele si margini.



.container {

display: flex;

justify-content: space-around;

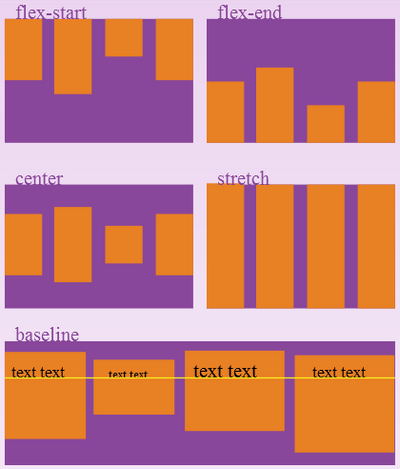
}

Test Flexbox example

### align-items

**align-items** seteaza aranjarea pe verticala a elementelor din flex container cand acestea nu ocupa tot spatiul din el. Este similar cu justify-content, dar pe axa verticala.  
- Valori pt. **align-items**:

* **stretch** - (default) intinde elementele pana umple containerul (respecta min-width/max-width).
* **flex-start** - elementele sunt pozitionate in partea de sus a containerului.
* **flex-end** - elementele sunt pozitionate in partea de jos a containerului.
* **center** - elementele sunt puse in centru, pe axa verticala.
* **baseline** - flex-items sunt aliniate astfel incat linia lor de baza sa fie aliniata.



.container {

display: flex;

align-items: center;

}

Test Flexbox example

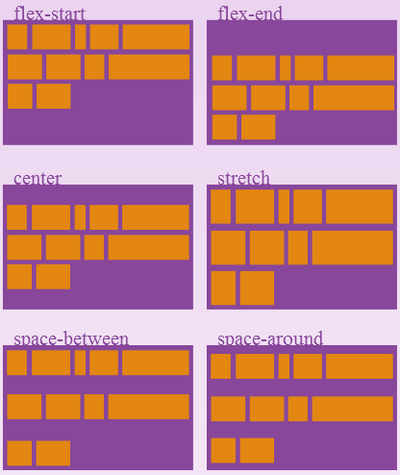
### align-content

**align-content** aliniaza liniile din flex container cand e spatiu ramas liber pe verticala. Similar cum justify-content aliniaza elementele individual pe orizontala.

*Aceasta proprietate nu are efect cand elementele sunt pe o singura linie.*

- Valori pt. **align-content**:

* **stretch** - (default) Liniile se intind pentru a prelua spatiul ramas.
* **flex-start** - Liniile sunt aranjate la inceputul containerului.
* **flex-end** - Liniile sunt aranjate la sfarsitul containerului.
* **center** - Liniile sunt aranjate in centrul containerului.
* **space-between** - Liniile sunt distribuite uniform; prima linie este la inceputul containerului, iar ultima este la sfarsit.
* **space-around** - Liniile sunt distribuite uniform cu spatiu egal in jurul fiecarei linii.



.container {

display: flex;

align-content: space-around;

}

### Testare proprietati Flexbox

Aici puteti testa direct combinatia de diferite proprietati si valori pentru containerul flex.

.container {

display: ;

flex-direction: ;

flex-wrap: ;

justify-content: ;

align-items: ;

align-content: ;

margin:2px;

padding:3px;

height: ;

width: ;

}

.container .item {

background:#ccccfe;

height:50px;

margin:2px;

}

Demo:

- Content Before..

Parent content.

Flex item #1

Child 2

Flexbox child 3

- Content After..

# CSS3 Flexbox Item

## Curs Css

[**Home**](https://marplo.net/)[**Engleza**](https://marplo.net/engleza)[**Spaniola**](https://marplo.net/spaniola)[**Html**](https://marplo.net/html)[**JavaScript**](https://marplo.net/javascript)[**PHP-MySQL**](https://marplo.net/php-mysql)[**Ajax**](https://marplo.net/ajax)[**Blog**](https://marplo.net/blog)[**Forum**](https://marplo.net/forum/)[**Games**](https://gamv.eu/)

Top of Form



Bottom of Form

- [flex-grow](https://marplo.net/css/css3-flexbox-item#anc_fg)  
- [flex-basis](https://marplo.net/css/css3-flexbox-item#anc_fb)  
- [flex-shrink](https://marplo.net/css/css3-flexbox-item#anc_fs)  
- [flex](https://marplo.net/css/css3-flexbox-item#anc_f)  
- [align-self](https://marplo.net/css/css3-flexbox-item#anc_as)  
- [Testare proprietati Flexbox](https://marplo.net/css/css3-flexbox-item#test_flexbox)

## Proprietati ale elementelor din containerul flex

- **Proprietatile CSS3 flexbox** prezentate in continuare se aplica la fiecare element din interiorul containerului flex.

***float, clear****si****vertical-align****nu au efect la flex-item.*

### order

Proprietatea **order** stabileste ordinea unui element flexibil relativ la celelalte elemente in interiorul containerului flex. Valorile sunt numere intregi (-2, -1, 0, 1, 2, ..). Default e 0, cresterea si descresterea valorii muta elementul la dreapta, respectiv la stanga.

*Setand****margin: auto;****va absorbi spatiul suplimentar din jurul elementului.*

.flex-item {

order: -1;

}



- Pentru a schimba locurile intre primul si ultimul element flexibil, folositi un cod ca acesta:

.item:first-child {

order:1;

}

.item:last-child {

order:-1;

}

### flex-grow

Proprietatea **flex-grow** specifica cat va creste elementul in raport cu restul elementelor flexibile din interiorul containerului. Valorile sunt numere pozitive (0, 1, 2, ..). Default e 0.  
- in codul urmator, al doilea element flexibil ocupa de trei ori mai mult spatiu decat celelalte:

.item:nth-child(2) {

flex-grow: 3;

}



### flex-basis

**flex-basis** specifica lungimea initiala a unui element flexibil. Default: auto.  
- In codul urmator, flex-basis e specificat la al patrulea flex-item, si defineste lungimea lui initiala:

.container .item {

flex-basis:auto;

}

.container .item:nth-child(4) {

flex-basis: 350px;

}



### flex-shrink

**flex-shrink** specifica cat se va micsora elementul in raport cu restul elementelor flexibile din interiorul containerului. Valorile sunt numere pozitive (0, 1, 2, ..). Default e 1.  
- in codul urmator, al doilea flex-item se micsoreaza de trei ori mai mult decat restul:

.container .item {

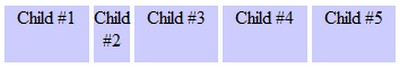
flex-basis: 100px;

}

.container .item:nth-child(2) {

flex-shrink: 3;

}



### flex

**flex-shrink** stabileste in aceeasi proprietate valorile pentru flex-grow, flex-shrink, si flex-basis.

**flex: flex-grow flex-shrink flex-basis;**

Default: **0 1 auto**.  
- In exemplul urmator, al doilea flex-item creste de doua ori mai mult decat restul elementelor, cu un flex-basis de 150px. Valoarea pentru flex-shrink nu e adaugata, deci ramane cea implicita (1).

.container .item:nth-child(2) {

flex: 2 150px;

}

flexbox-flex

### align-self

Proprietatea **align-self** specifica alinierea elementului selectat in interiorul containerului flexibil. Rescrie valoarea proprietatii align-items care se defineste la containerul flex.  
- Valori pt. **align-self**:

* **auto** - (default) Elementul preia valoarea proprietatii ***align-items*** a containerului flex parinte.
* **stretch** - Elementul este pozitionat astfel inat sa se potriveasca cu dimensiunea containerului.
* **flex-start** - Elementul este pozitionat in partea de sus a containerului.
* **flex-end** - Elementul este pozitionat in partea de jos a containerului.
* **center** - Elementul e centrat pe axa verticala.
* **baseline** - Elementul este pozitionat pe linia de baza a containerului.

- Exemplu, al treilea si al patrulea flex-item au rescrisa valoarea pentru proprietatea align-self:

.container {

align-items: flex-start;

display: flex;

height:90px;

width: 400px;

}

.container .item:nth-child(3) {

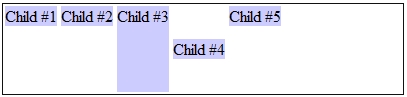
align-self: stretch;

}

.container .item:nth-child(4) {

align-self: center;

}

Resultat:  


### Testare proprietati Flexbox

Aici puteti testa direct combinatia de diferite proprietati si valori pentru containerul flex si individual pentru fiecare flex-item.

/\* Flexbox Container Properties \*/

.container {

display: ;

flex-direction: ;

flex-wrap: ;

justify-content: ;

align-items: ;

align-content: ;

margin:2px;

padding:3px;

height: ;

width: ;

}

/\* Flexbox Item Properties \*/

.container .item:nth-child(1) {

order: ;

flex-grow: ;

flex-basis: ;

flex-shrink: ;

align-self: ;

}

.container .item:nth-child(2) {

order: ;

flex-grow: ;

flex-basis: ;

flex-shrink: ;

align-self: ;

}

.container .item:nth-child(3) {

order: ;

flex-grow: ;

flex-basis: ;

flex-shrink: ;

align-self: ;

}

.container .item:nth-child(4) {

order: ;

flex-grow: ;

flex-basis: ;

flex-shrink: ;

align-self: ;

}

Demo:

- Content Before..

Flex-Item  
#1

Flex-Item  
#2

Flex-Item  
#3

Flex-Item  
#4

# Animatie elemente HTML cu CSS

## Curs Css

[**Home**](https://marplo.net/)[**Engleza**](https://marplo.net/engleza)[**Spaniola**](https://marplo.net/spaniola)[**Html**](https://marplo.net/html)[**JavaScript**](https://marplo.net/javascript)[**PHP-MySQL**](https://marplo.net/php-mysql)[**Ajax**](https://marplo.net/ajax)[**Blog**](https://marplo.net/blog)[**Forum**](https://marplo.net/forum/)[**Games**](https://gamv.eu/)

Top of Form



Bottom of Form

* [Animatie CSS - pasi multipli](https://marplo.net/css/animatie-html-css#hsh_ms)
* [Animatie CSS - sub-proprietati](https://marplo.net/css/animatie-html-css#hsh_sp)
* [Animatie cu CSS la pseudo-clase si pseudo-elemente](https://marplo.net/css/animatie-html-css#hsh_pse)
* [Proprietatea animation](https://marplo.net/css/animatie-html-css#hsh_sap)

Proprietatea "animation", sau sub-proprietatile ei pot fi utilizate pentru a anima alte proprietati CSS aplicate elementelor HTML, cum ar fi: color, background-color, height, sau width.

*• O alta metoda de a anima proprietatile unui element HTML cu CSS este utilizand proprietatea "transition". Vedeti tutorialul:*[***CSS3 transition***](https://marplo.net/css/css3-transition)*.*

*Elementul <style> din aceste exemple se adauga in sectiunea <head> din pagina.*

## Creare Animatie cu CSS

Ca sa faceti animatii cu CSS, mai intai se definesc "keyframes" cu un nume pentru animatie, folosind instructiunea **@keyframes**, in care se definesc stilurile CSS pe care trebuie sa le aibe elementul HTML la un moment dat.  
Apoi, se adauga proprietatea **animation** (sau sub-proprietatile) la elementul HTML, folosind acelasi nume.  
  
- In urmatorul exemplu se defineste si se adauga animatia "ex1\_anim" la elementul "#anim\_ex1". Animatia va dura 4 secunde de la inceput pana la sfarsit (0% la 100%), si va schimba gradual valoarea margin-left pentru #anim\_ex1 de la "0" la "80%". Aceasta va avea ca efect miscarea acelui <div> de la stanga la dreapta.

<style>

/\* The animation code \*/

@keyframes ex1\_anim {

0% { margin-left: 0;}

100% { margin-left: 80%;}

}

/\* The element to apply the animation to \*/

#anim\_ex1 {

animation-name: ex1\_anim;

animation-duration: 4s;

background-color: red;

padding: 20px 0;

width: 90px;

}

</style>

<div id='anim\_ex1'>HTML Div</div>

Incercati codul

- Demo:

- Clic pe acest buton: Click ca sa vedeti rezultatul codului de deasupra.

HTML Div

*Daca proprietatea animation-duration nu este adaugata, animatia nu se va efectua, deoarece valoarea implicita e 0s (0 secunde).*

### Animatie CSS - pasi multipli

In exemplu de sus, animatia "@keyframes ex1\_anim" are definiti doi pasi: 0% este inceputul animatiei, si 100% e sfarsitul.  
Puteti adauga oricat de multi pasi (schimbari de stil) doriti.

- Urmatorul exemplu are patru pasi pentru animatie (0%, 30%, 55, 100%) care seteaza diferite proprietati CSS pentru a transforma, roti, muta si schimba culoarea de fundal in mod gradual.

<style>

@keyframes ex2\_anim {

0% { margin-left: 0;}

30% {

border-radius: 100%;

transform: scale(1.5);

}

55% {

transform: rotate(180deg);

}

100% {

background-color: #0000e0;

margin-left: 85%;

transform: rotate(360deg);

}

}

#anim\_ex2 {

animation-name: ex2\_anim;

animation-duration: 4s;

background-color: red;

padding: 20px 0;

width: 90px;

}

</style>

<div id='anim\_ex2'>HTML Div</div>

Incercati codul

- Demo:

- Clic pe acest buton: Click ca sa vedeti rezultatul codului de deasupra.

HTML Div

### Animatie CSS - sub-proprietati

Sunt mai multe sub-proprietati care se pot utiliza la configurarea animatiei cu CSS. Mai jos este o lista cu aceste sub-proprietati.  
  
- Iata un exemplu cu unele din acestea. Animatia incepe dupa 1 secunde (delay 1) si e rulata de 3 ori, "alternate" (prima rulare in fata, a doua inapoi), si "linear" (cu aceeasi viteza de la inceput pana la sfarsit).  
Cand animatia se termina, elementul ramane in pozitia din fata (forwards, cu stilul definit la ultimul pas, 100%): animation-fill-mode: forwards;.

<style>

@keyframes ex3\_anim {

0% { margin-left: 0;}

50% {

transform: scale(.8);

transform: rotate(225deg);

}

100% {

background-color: #5588ed;

border-radius: 100%;

margin-left: 85%;

transform: rotate(450deg);

}

}

#anim\_ex3 {

animation-name: ex3\_anim;

animation-duration: 2.5s;

animation-delay: 1s;

animation-direction: alternate;

animation-iteration-count: 3;

animation-fill-mode: forwards;

animation-play-state: running;

animation-timing-function: linear;

background-color: red;

padding: 20px 0;

width: 80px;

}

</style>

<div id='anim\_ex3'>HTML Div</div>

Incercati codul

- Demo:

- Clic pe acest buton: Click ca sa vedeti rezultatul.

HTML Div

#### Sub-proprietati pentru animatia CSS

* animation-delay - durata de timp pana cand va incepe animatia; numar de secunde sau milisecunde (***Xs, Xms***).
* animation-direction - configureaza directia si modul animatiei.  
  Valori: ***reverse, alternate, alternate-reverse, normal (default)***.
* animation-duration - durata de timp necesara animatiei pentru o ruta; numar de secunde sau milisecunde (***Xs, Xms***).
* animation-fill-mode - specifica un stil pentru elementul HTML cand animatia nu mai ruleaza (inainte de a incepe, dupa ce se termina, sau ambele).  
  Valori: ***forwards, backwards, both, none (default)***.
* animation-iteration-count - numarul de rulari efectuate de animatie (default 1).
* animation-name - numele animatiei definit la @keyframes.
* animation-play-state - permite intreruperea si reluarea secventei de animatie.  
  Valori: ***paused, running (default)***.
* animation-timing-function - specifica modul de acceleratie al animatiei.
  + linear - animatia are aceeasi viteza de la inceput pana la sfarsit.
  + ease - animatia are un start incet, apoi rapid si iar incetineste spre sfarsit (default).
  + ease-in - start incet.
  + ease-out - incetineste la sfarsit.
  + ease-in-out - incet la start si la sfarsit.
  + cubic-bezier(x1, y1, x2, y2) - puteti defini propriile vlori in functia cubic-bezier.  
    Valorile permise sint numere de la 0 la 1 (ex. cubic-bezier(0.5, 0.2, 0.3, 1.0)).

### Animatie cu CSS la pseudo-clase si pseudo-elemente

Animatia cu CSS se poate crea si la pseudo-elemente (:before, :after), si poate fi aplicata la pseudo-classe (precum :hover).  
- Exemplu, cand mouse-ul este deasupra elementului #anim\_pse, se creaza un pseudo-element cu ":after", care va fi animat.

<style>

@keyframes pse\_anim {

0% {

left: 50%;

height:0;

width: 0;

}

50% {

transform: rotate(180deg);

}

100% {

border-radius: 100%;

transform: rotate(360deg);

left: 250px;

height: 40px;

width: 40px;

}

}

#anim\_pse {

position: relative;

text-align:center;

padding:20px 0;

width: 90px;

background-color: red;

}

#anim\_pse:hover:after {

animation-name: pse\_anim;

animation-duration: 2s;

animation-fill-mode: forwards;

animation-timing-function: linear;

background-color: #0000da;

content: ' ';

display: block;

margin: 0;

position: absolute;

top: 25%;

}

</style>

Pozitionati mouse-ul peste patratul rosu.

<div id="anim\_pse">Hello Sir</div>

Incercati codul

Demo:

Pozitionati mouse-ul peste patratul rosu.

Hello Sir

### Proprietatea animation

Pentru cei mai avansati, se pot adauga valorile mai multor sub-proprietati intr-o singura proprietate CSS: animation.  
- Exemplu urmator foloseste sase din proprietatile CSS de animatie:

#elm\_id {

animation-name: example;

animation-duration: 4s;

animation-timing-function: linear;

animation-delay: 2s;

animation-iteration-count: infinite;

animation-direction: alternate;

}

- Acelasi efect de animatie se poate obtine cu proprietatea animation:

#elm\_id {

animation: example 4s linear 2s infinite alternate;

}

Se pot anima multe proprietati CSS: "background-color, height, margin, opacity, padding, transform, width", si altele.  
La MDN e o lista cu [Animatable CSS properties](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_animated_properties).

# CSS Content

## Curs Css

[**Home**](https://marplo.net/)[**Engleza**](https://marplo.net/engleza)[**Spaniola**](https://marplo.net/spaniola)[**Html**](https://marplo.net/html)[**JavaScript**](https://marplo.net/javascript)[**PHP-MySQL**](https://marplo.net/php-mysql)[**Ajax**](https://marplo.net/ajax)[**Blog**](https://marplo.net/blog)[**Forum**](https://marplo.net/forum/)[**Games**](https://gamv.eu/)

Top of Form



Bottom of Form

Proprietatea CSS **content** se foloseste pentru a genera continut intr-un element HTML direct din CSS. Poate fi utilizata cu pseudo-elementele **:after** si **:before**.

<style>

.cnt\_ex1:before {

content: '~> ';

color: #0000d0;

}

.cnt\_ex1:after {

content: ' <~';

color: #0000d0;

}

</style>

<div class='cnt\_ex1'>CoursesWeb.net</div>

<div class='cnt\_ex1'>MarPlo.net</div>

Incercati codul

*Elementul <style> din aceste exemple se adauga in sectiunea <head> din pagina.*

## Valori pentru CSS content

* none - Pseudo-elementul nu e creat.
* normal - Evalueza ca none (default).
* counter(counter\_index) - Seteaza valoarea ca un contor.  
  - Mai intai se creaza un index pentru contor, cu proprietatea counter-increment.
* <style>
* .cnt\_ex2 {
* counter-increment: ex\_index; /\* Sets a counter to be increased \*/
* }
* .cnt\_ex2:before {
* /\* Space is added to create a space between counter and rest of content \*/
* content: counter(ex\_index) '. ';
* color: #0000d0;
* }
* </style>
* <div class='cnt\_ex2'>CoursesWeb.net</div>
* <div class='cnt\_ex2'>MarPlo.net</div>

Incercati codul

Rezultat:

CoursesWeb.net

MarPlo.net

* attr(attribute) - Un sir cu valoarea din atributul specificat.  
  - Vedeti exemplu de la: [Tooltip Animat cu HTML si CSS](https://marplo.net/css/tooltip-animat-html-css).
* string - Un text adaugat pt. continut. Se pot adauga si caractere speciale, folosind codul Hex, cu back-slash "\" la inceput ( ***content: '\Hex-code';*** ).
* content: '\27A4'; /\* Displays: ➤ \*/
* open-quote | close-quote - Valoarea este ghilimele de inceput | sfarsit.  
  - Pentru a fi generat "close-quote" e necesar si "open-quote".
* no-open-quote | no-close-quote - Elimina ghilimele de inceput | sfarsit, daca au fost adaugate.
* url('url') - Adauga continutul de la adresa "url" specificata (imagine, sunet, video, etc.).  
  - Exemplu:
* <style>
* .cnt\_ex3:before {
* content: url('https://coursesweb.net/imgs/8.gif');
* margin-right: 10px;
* }
* </style>
* <div class='cnt\_ex3'>CoursesWeb.net</div>
* <div class='cnt\_ex3'>MarPlo.net</div>

Incercati codul

Rezultat:

CoursesWeb.net

MarPlo.net

### Alte exemple cu proprietatea content

#### Modificare marcator de la liste

- Urmatorul exemplu afisaza simbolul ( » ) ca marcator la listele din <ul> care are clasa 'li\_mark1':

<style>

ul.li\_mark1 { list-style: none; }

ul.li\_mark1 li:before {

content: '\00BB'; /\* Escaped Hex code notation \*/

padding-right: 8px;

}

</style>

<ul class='li\_mark1'>

<li>Free Courses and Tutorials</li>

<li>HTML-CSS examples</li>

<li>Today is a good day for "Doing" Nothing</li>

</ul>

Incercati codul

Demo:

* Free Courses and Tutorials
* HTML-CSS examples
* Today is a good day for "Doing" Nothing

#### Rand nou in css content

Ca sa adaugati un text cu mai multe linii in proprietatea "content", adaugati in textul respectiv: **\a** unde vreti crearea unui nou rand, si proprietatea white-space: pre;

<style>

#cnt\_ex4:after {

color:#0000e0;

content: '- Text from content \a Another line.';

white-space: pre;

}

</style>

<div id='cnt\_ex4'>HTML Div.</div>

Incercati codul