

Tehnici Web

CURSUL 7

Semestrul I, 2018-2019
Carmen Chirita

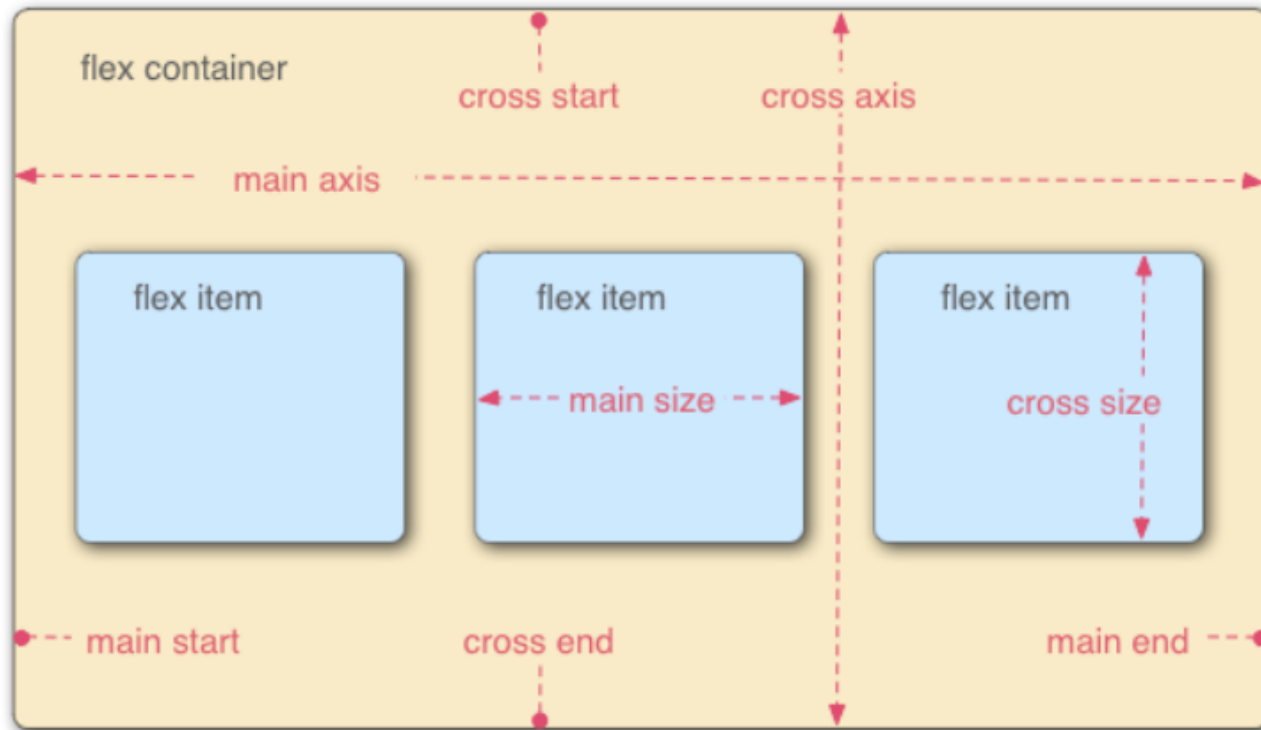
<https://sites.google.com/site/fmitehniciweb/>

CSS layout - flexbox

- Layoutul flexibil ofera o metoda eficienta de a aseza, alinia și a distribui spațiul între elementele din document chiar și atunci când dimensiunea viewportului și a elementelor este necunoscuta sau dinamica.

Layoutul flexibil presupune definirea a doua tipuri de elemente:

- Element părinte - **flex container**
- Elemente copii - **flex items**



Proprietăți pentru părinte (flex container)

display
flex-direction
flex-wrap
flex-flow
justify-content
align-items
align-content

Proprietăți pentru copii (flex items)

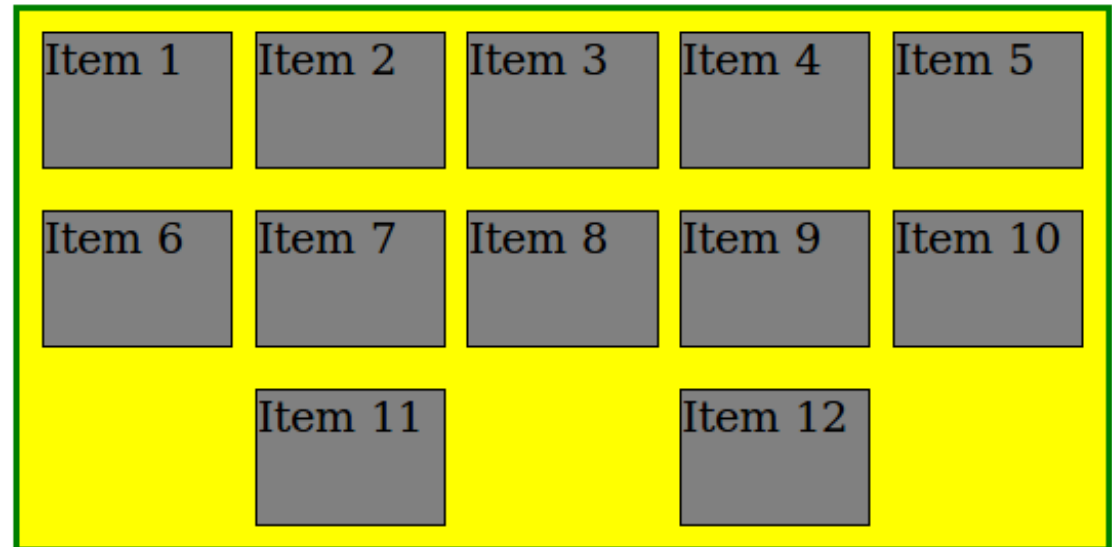
order
flex-grow
flex-shrink
flex-basis
flex
align-self

Exemplu

```
<style>
.container {
    display:flex;
    flex-direction: row;
    flex-wrap: wrap;
    justify-content:space-evenly;
    width:50%; height:200px;
    border:3px solid green;
    background-color:yellow;
    padding:0;
    list-style-type:none;
}

.item {
    align-self:center;
    width:70px;
    height:50px;
    border:1px solid black;
    background-color:grey;
}
</style>
```

```
<body>
<ul class="container">
<li class="item">Item 1</li>
<li class="item">Item 2</li>
<li class="item">Item 3</li>
.....
<li class="item">Item 12</li>
</ul>
</body>
```



Proprietăți pentru părinte (flex container)

`display: flex /* inline-flex */`

- definește un container flexibil; inline sau bloc în funcție de valoarea dată.
- permite un context flexibil pentru toți copiii săi direcți.

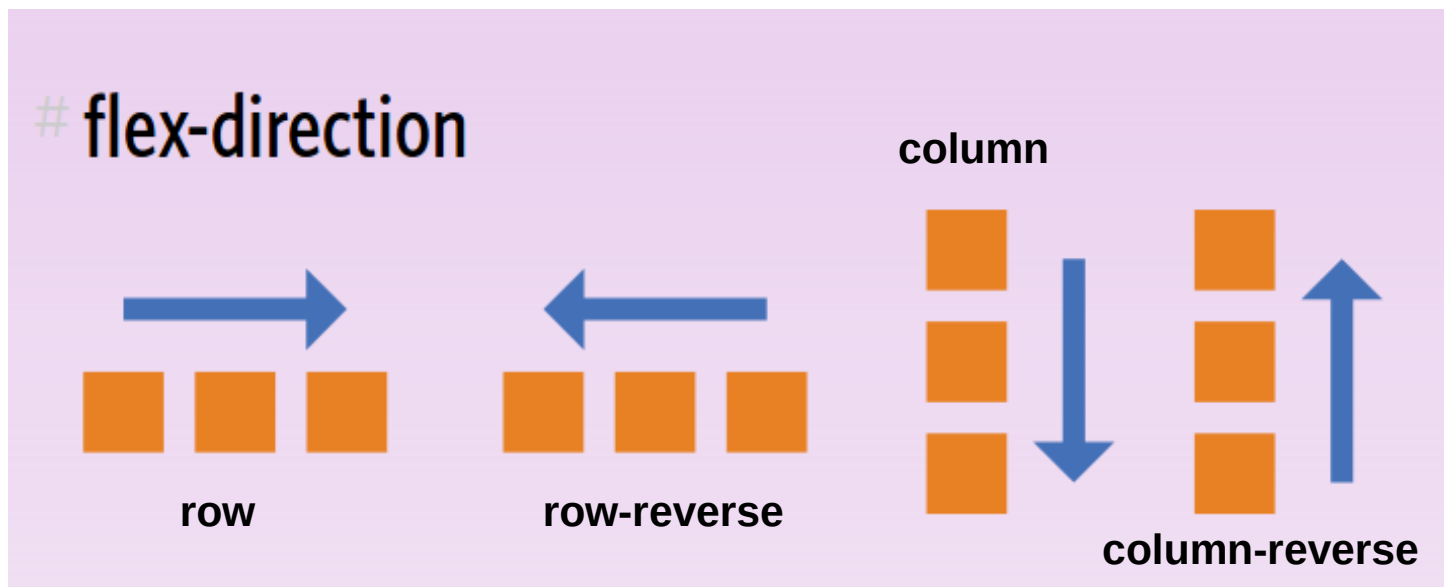
Exemplu:

```
ul {display: flex;}  
body {display: flex;}
```

Proprietăți pentru părinte (flex container)

`flex-direction: row` (implicit) /* `row-reverse`, `column`, `column-reverse` */

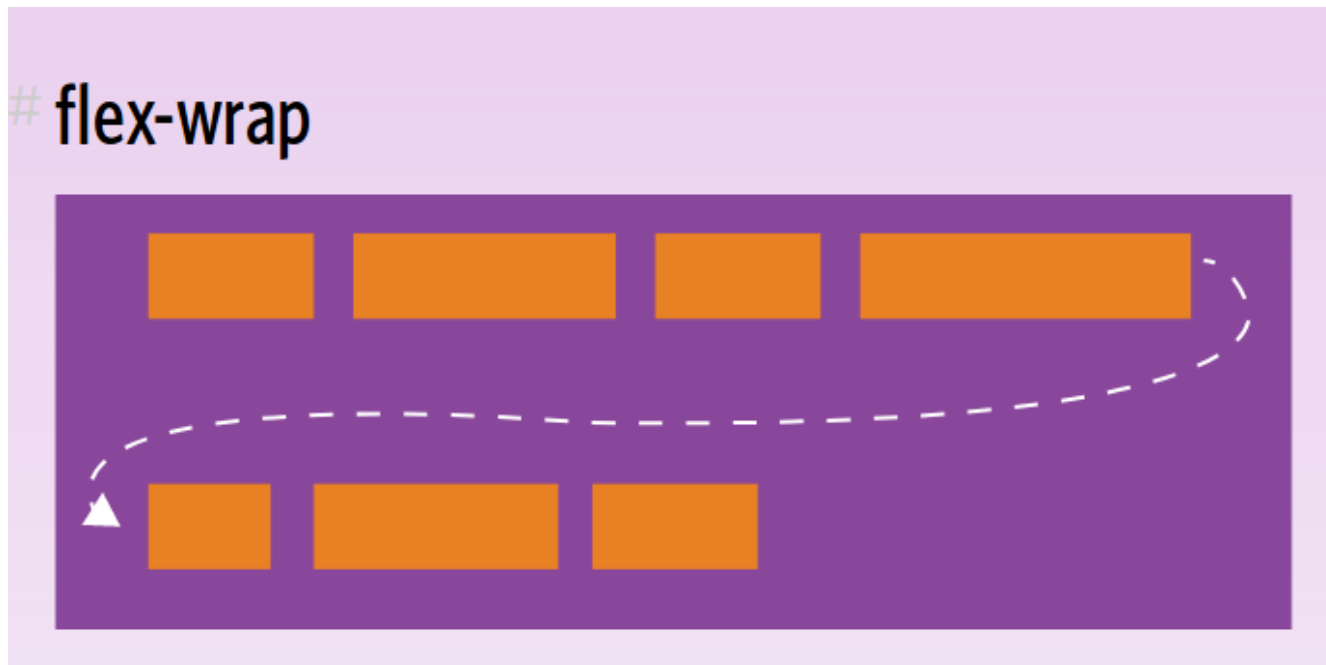
- stabilește direcția în care sunt plasate itemurile în container: pe orizontală (stânga-dreapta sau dreapta-stanga) sau pe verticală (sus-jos sau jos-sus).



Proprietăți pentru părinte (flex container)

flex-wrap: wrap /* nowrap (implicit) , wrap-reverse */

- stabileste dacă itemurile vor fi aranjate pe o singură linie (coloana) sau pe mai multe; implicit ele se vor aranja toate pe o linie (coloana).



Proprietăți pentru părinte (flex container)

flex-flow: flex-direction flex-wrap

- prescurtare pentru proprietatile flex-direction și flex-wrap;
- implicit: **flex-flow: row nowrap**

```
.container {
  display: flex;
  flex: row nowrap;
}
```

Item 1	Item 2	Item 3	Item 4	Item 5	Item 6	Item 7	Item 8	Item 9	Item 10	Item 11	Item 12

Proprietăți pentru părinte (flex container)

justify-content: flex-start (implicit)
/* flex-end,
center,
space-between,
space-around,
space-evenly */

- specifica modul cum sunt aliniate itemurile de-a lungul axei principale

justify-content

flex-start



flex-end



center



space-between



space-around



space-evenly

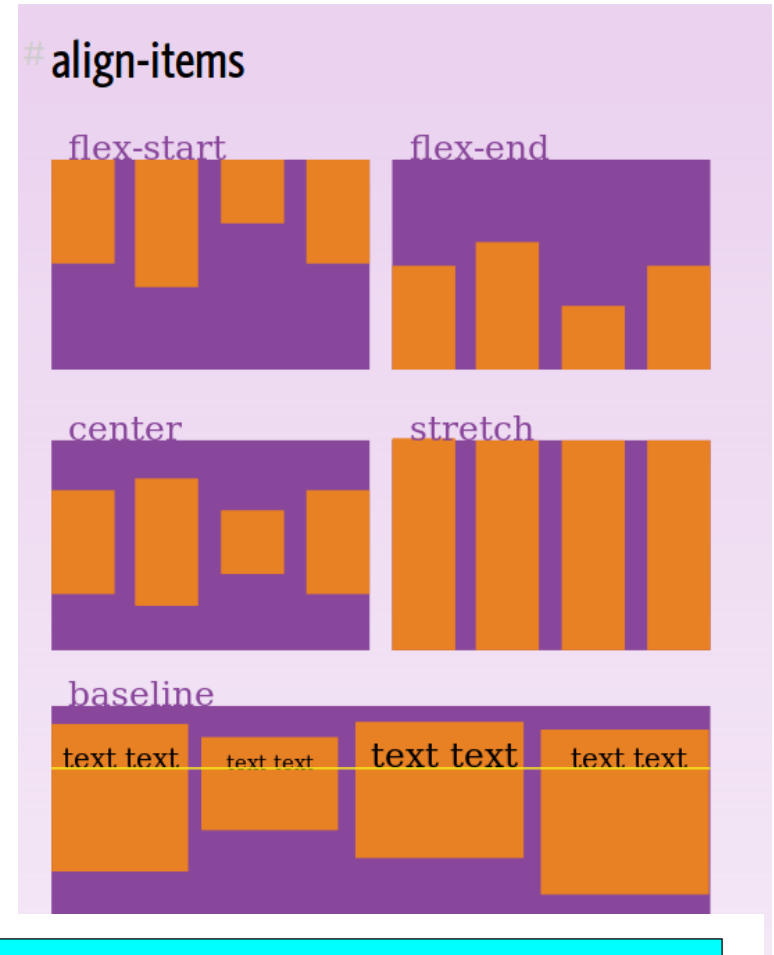


Proprietăți pentru părinte (flex container)

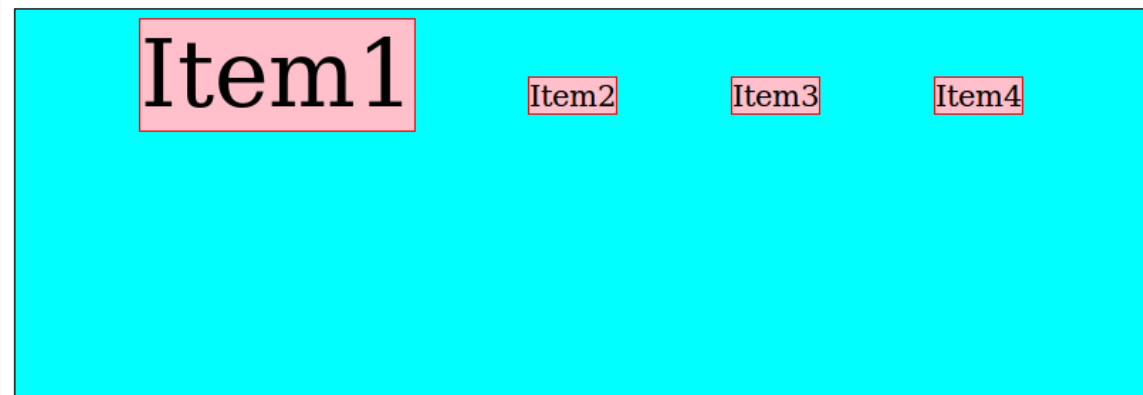
`align-items: flex-start`

`/* flex-end,
center,
stretch (implicit),
baseline */`

- specifica modul cum sunt aliniate itemurile pe axa secundara (axa perpendiculara pe axa principala)



`align-items: baseline`



Proprietăți pentru părinte (flex container)

align-content: flex-start

**/* flex-end,
center,
stretch (implicit),
space-between,
space-around */**

- aliniază liniile containerului dacă există un spațiu suplimentar pe axa transversală;
- nu are efect dacă există o singură linie în container.

align-content

flex-start



flex-end



center



stretch



space-between



space-around



Proprietăți pentru copii (flex items)

order || flex-grow || flex-shrink || flex-basis || flex || align-self

order: <intreg> (implicit este 0)

- stabilește ordinea în care itemurile apar în containerul flexibil;
- implicit ele apar în ordinea în care sunt scrise în documentul HTML.

```

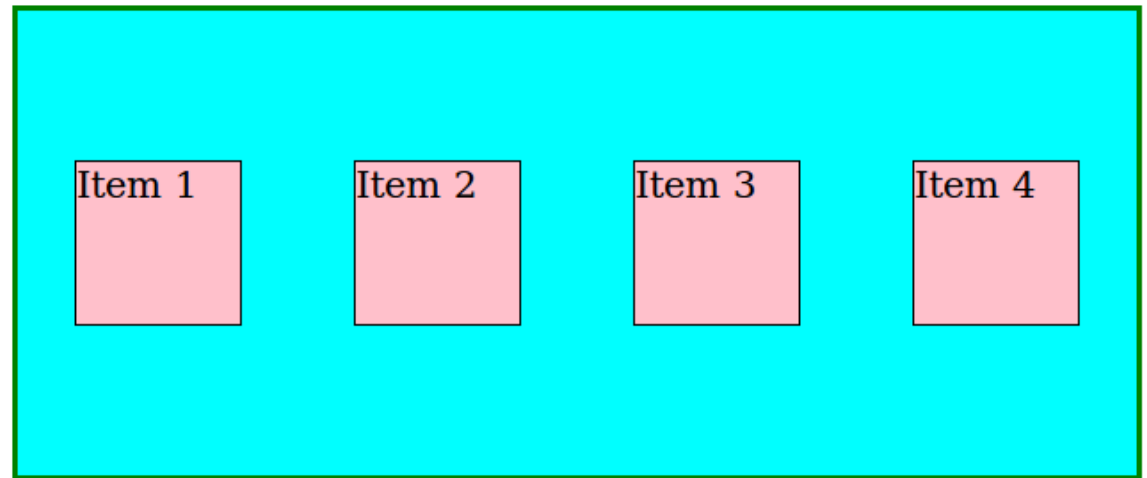
ul {
  display : flex;
  flex-flow: row nowrap;
  justify-content:space-around;
  align-items: center;
  width:50%;
  height:200px;
  border:3px solid green;
  background-color:cyan;
  padding:0;
  list-style-type:none;
}
li {
  width:70px;
  height:70px;
  border:1px solid black;
  background-color:pink;
}

```

```

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

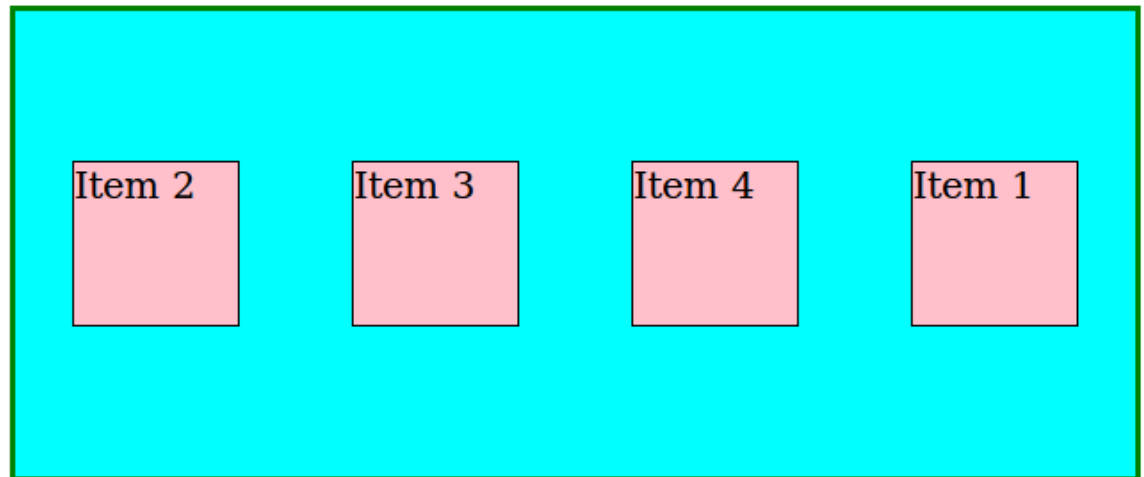
```



Ex.1

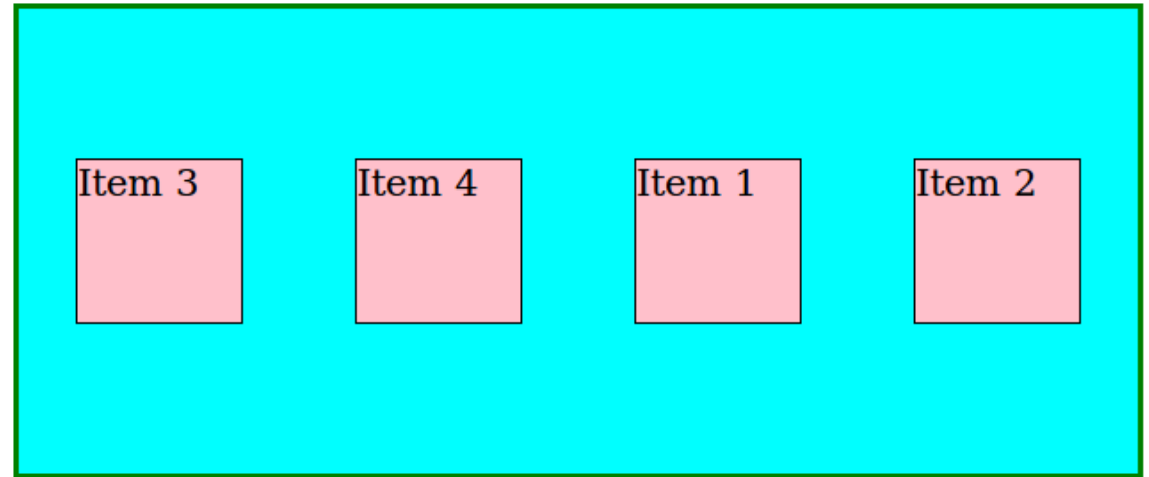
Ex.1

```
li:nth-child(1) { order:1; }
```



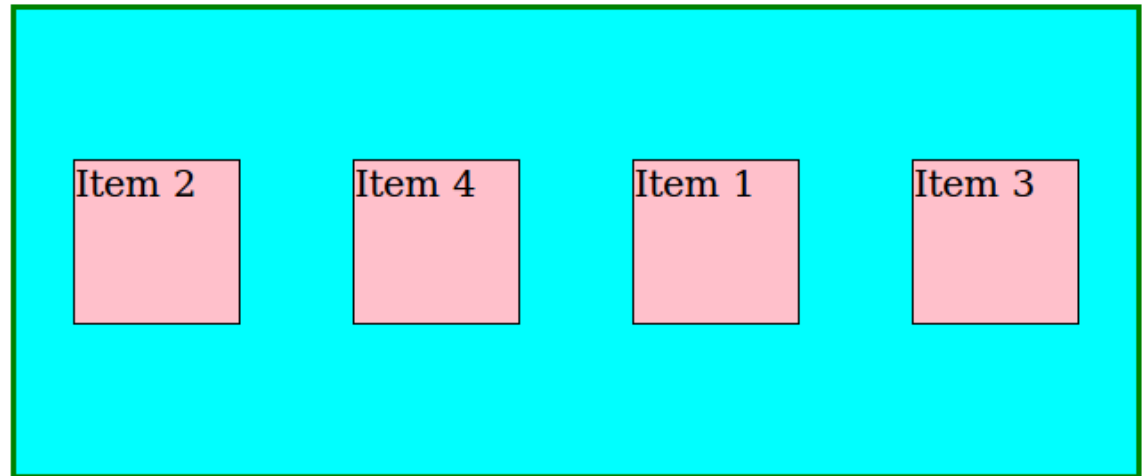
Ex.2

```
li:nth-child(1) {order:1;}  
li:nth-child(2) { order:2;}
```



Ex.3

```
li:nth-child(1) {order:1;}  
li:nth-child(3) { order:1;}
```



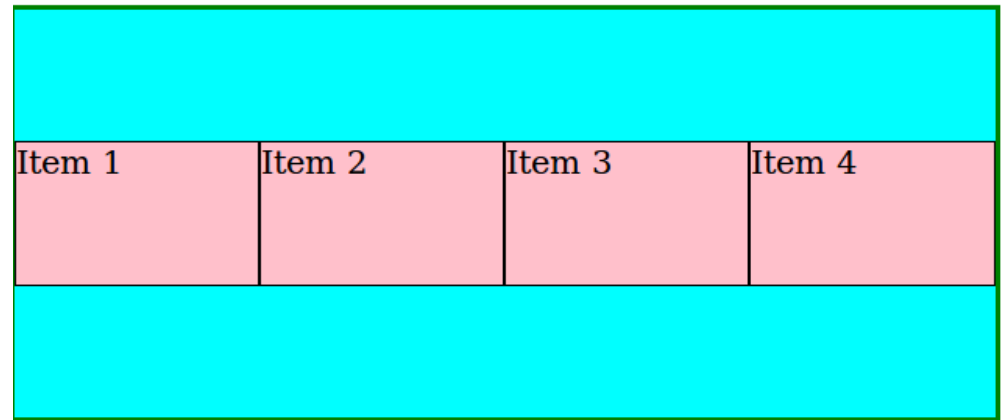
Proprietăți pentru copii (flex items)

order || **flex-grow** || flex-shrink || flex-basis || flex || align-self

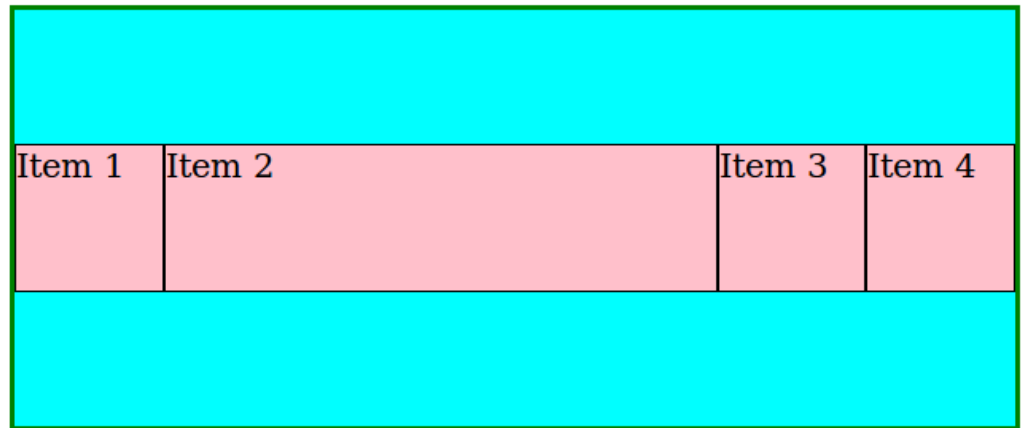
flex-grow: <number> (implicit este 0)

- stabilește capacitatea itemurilor de a-si mari dimensiunea dacă este spațiu disponibil în interiorul containerului;
- dacă toate itemurile au flex-grow egal cu 1, ele vor împărți spațiul disponibil în mod egal.

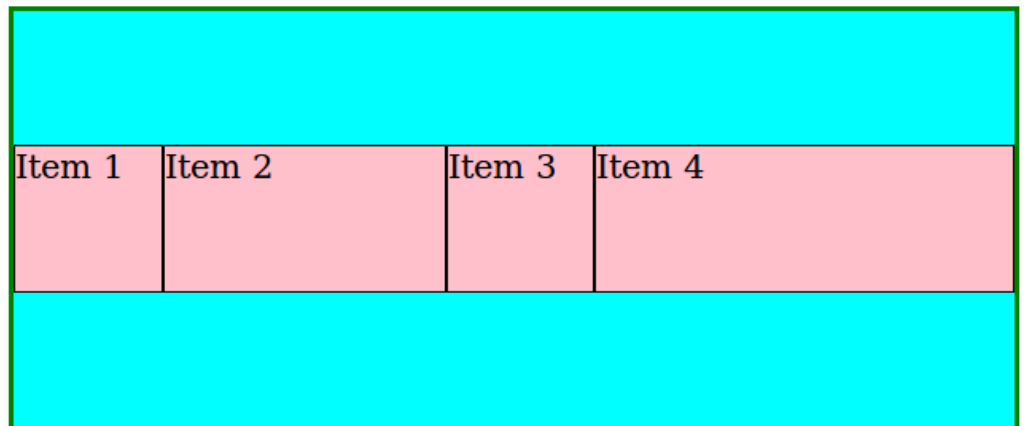

```
li { flex-grow:1; }
```



```
li:nth-child(2) { flex-grow:1; }
```



```
li:nth-child(2) { flex-grow:1; }  
li:nth-child(4) { flex-grow:2; }
```



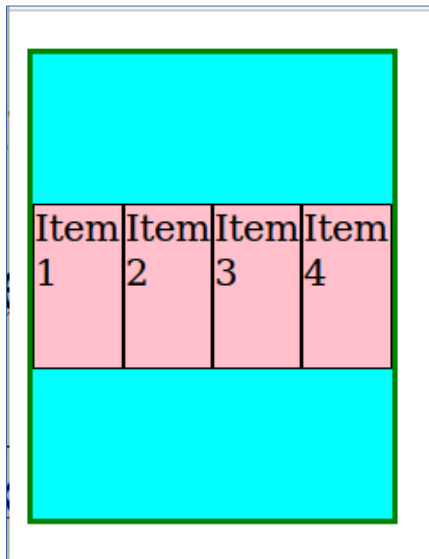
Proprietăți pentru copii (flex items)

order || flex-grow || **flex-shrink** || flex-basis || flex || align-self

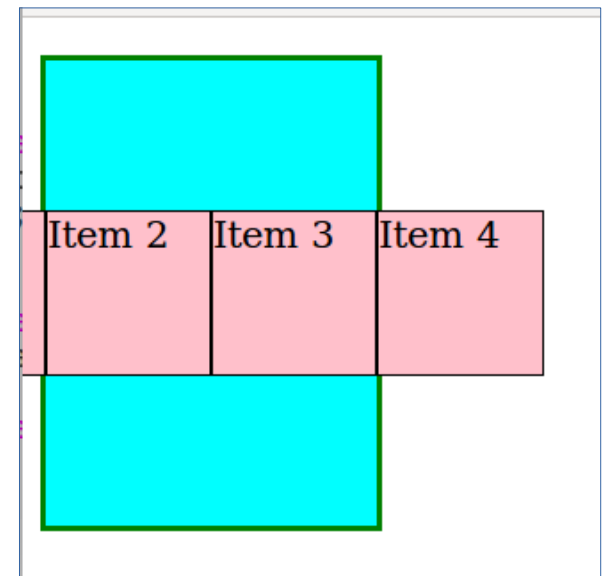
flex-shrink: <number> (implicit este 1)

- definește capacitatea itemurilor de a se micșora dacă este necesar

Implicit (flex-shrink:1)



li {flex-shrink:0;}

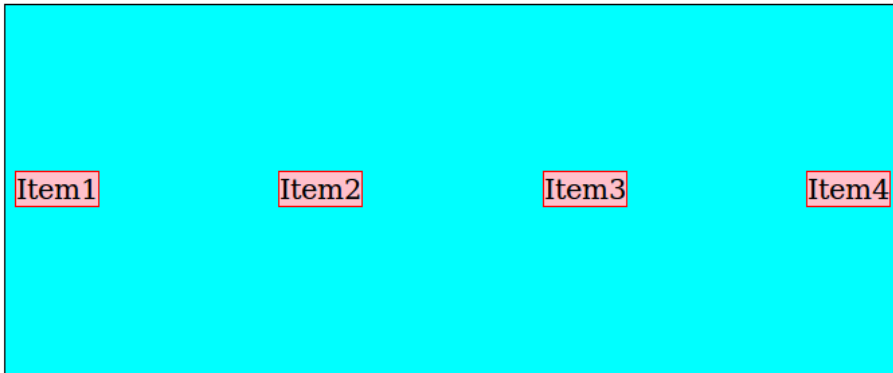


Proprietăți pentru copii (flex items)

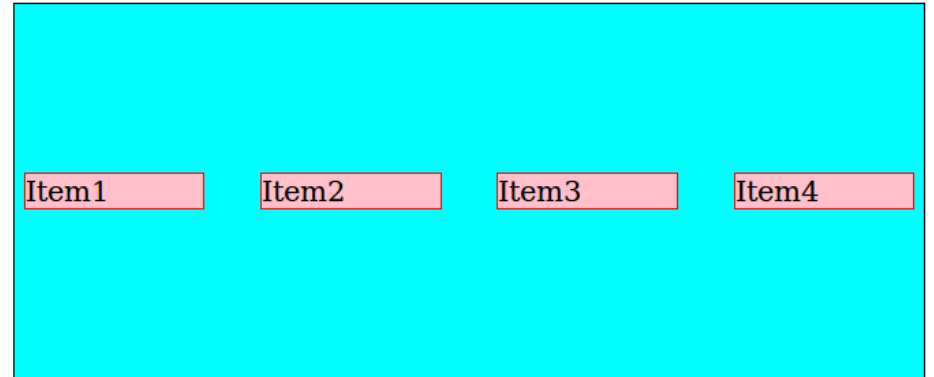
order || flex-grow || flex-shrink || **flex-basis** || flex || align-self

flex-basis: <length> /* auto (implicit)*/

- definește dimensiunea implicită a unui element înainte de distribuirea spațiului rămas;



flex-basis: auto



flex-basis: 100px

Proprietăți pentru copii (flex items)

order || flex-grow || flex-shrink || flex-basis || **flex** || align-self

flex: flex-grow flex-shrink flex-basis

- prescurtare pentru proprietatile flex-grow, flex-shrink si flex-basis;
- Implicit este **flex: 0 1 auto**

```

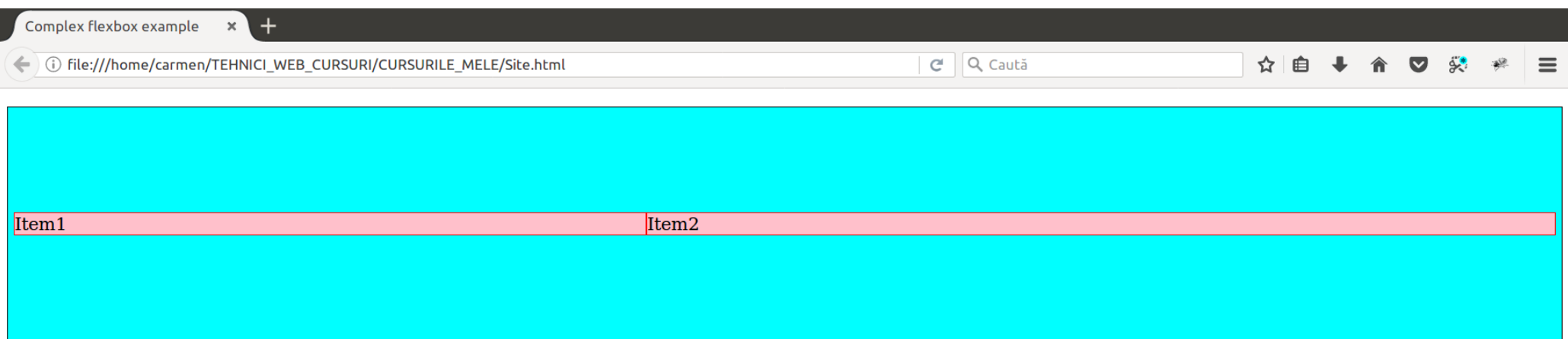
<style>
ul { display:flex;
      flex-flow: row nowrap;
      align-items: center;
      justify-content:space-between;
      border:1px solid black;
      padding:5px;
      background-color:cyan;
      height:200px;
      list-style-type:none;
    }
li { background-color:pink;
     border:1px solid red;
    }
li:nth-child(1)
  { flex:1 1 20em;}
li:nth-child(2)
  { flex:2 2 20em;}
</style>

```

```

<body>
<ul>
<li>Item1</li>
<li>Item2</li>
</ul>
</body>

```



Proprietăți pentru copii (flex items)

order || flex-grow || flex-shrink || flex-basis || flex || align-self

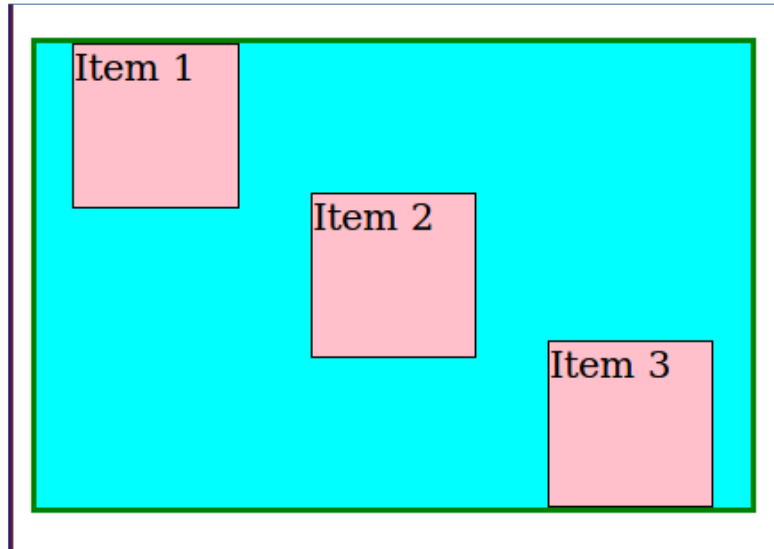
align-self: flex-start

/* flex-end,
center,
stretch,
baseline,
auto*/

- Permite alinierea individuala a itemurilor pe axa secundara (inlocuieste alinierea specificata cu align-items)

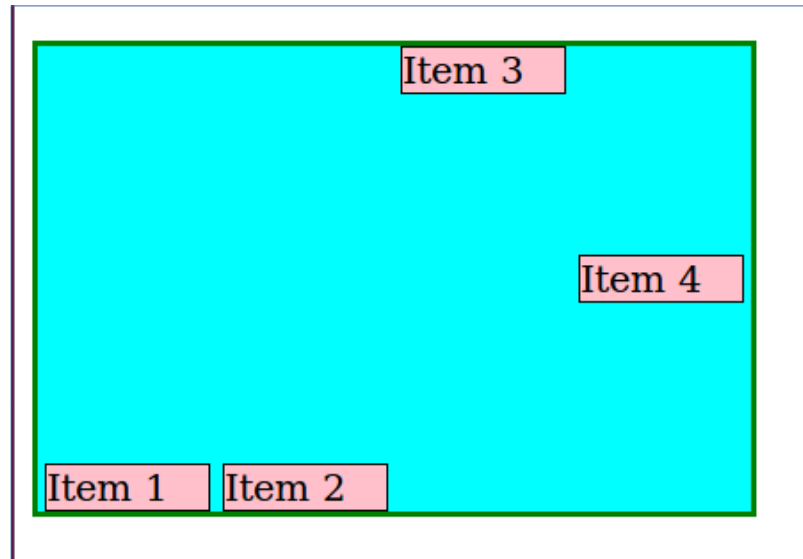
```
ul{display:flex;
    flex-flow: row wrap;
    justify-content:space-around;
    align-items:center; }
```

```
li:nth-child(1){align-self: flex-start;}
li:nth-child(3){align-self: flex-end;}
```



```
ul {display:flex;
    flex-flow: row wrap;
    justify-content:space-around;
    align-items:flex-end;}
```

```
li:nth-child(1){align-self: auto;}
li:nth-child(3){align-self: baseline;}
li:nth-child(4){align-self: center;}
```



Galerie de imagini folosind display:flex

```
<section id="container">
<figure class="item">

<figcaption> flori de primavara
</figcaption>

</figure>
<figure class="item">

<figcaption> peisaj de toamna
</figcaption>
</figure>
.....
```

```
#container {display: flex;
            flex-flow: row wrap;
            width:70%;
            border:1px solid black;
            }
```

```
.item { flex: 1 1 auto;
        margin: 1em;}
```

```
.imgf {width: 100%; height:auto;}
```



flori de primavara



peisaj de toamna



flori de vara



peisaj de iarna



flori de primavara



peisaj de toamna



flori de vara



flori de primavara



peisaj de toamna



flori de vara



peisaj de iarna



flori de primavara

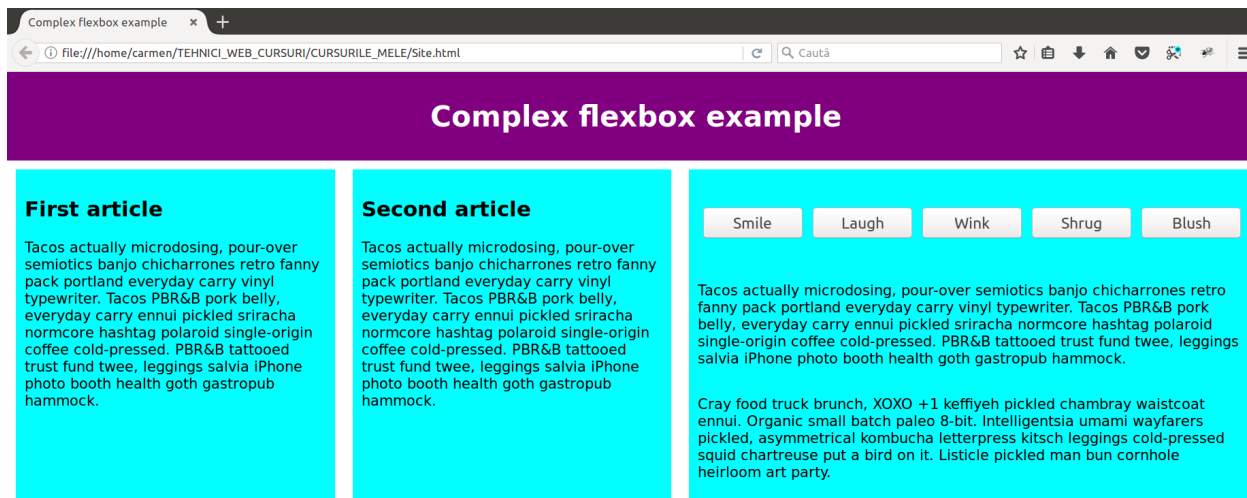


peisaj de toamna


```

section - article
      article
    article - div - button
                div    button
                div    button
                button
                button

```



```

section {
  display: flex;
}
article {
  flex: 1 200px;
}
article:nth-of-type(3) {
  flex: 3 200px;
  display: flex;
  flex-flow: column;
}
article:nth-of-type(3) div:first-
child {
  flex: 1 100px;
  display: flex;
  flex-flow: row wrap;
  align-items: center;
  justify-content: space-around;
}
button {
  flex: 1;
  margin: 5px;
  font-size: 18px;
  line-height: 1.5;
}

```

Formulare

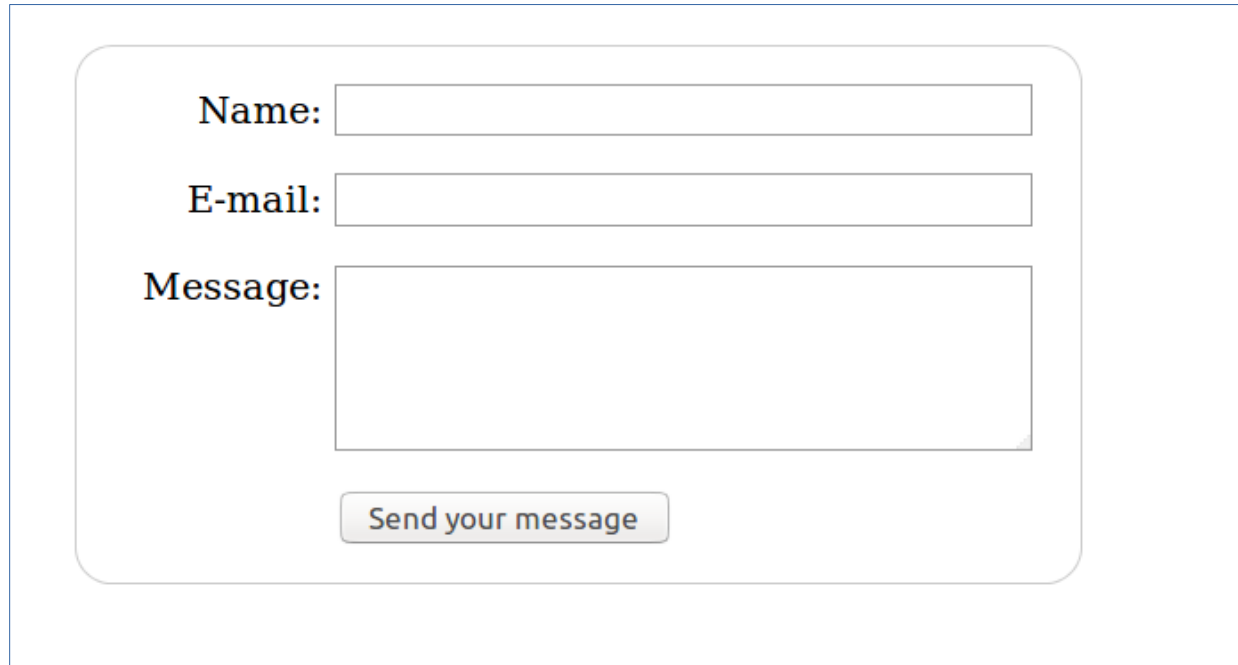
Elementul HTML `<form>` reprezintă o secțiune a documentului care conține controale interactive pentru a trimite informații către un server web.

```
<form>
```

```
<!-- se poate pune orice element HTML, dar importante  
sunt elementele speciale care creaza controale in  
browser -->
```

```
</form>
```

Formulare



A web form with a light gray border and rounded corners. Inside, there are three input fields and a button. The first field is labeled 'Name:' and is a single-line text input. The second field is labeled 'E-mail:' and is a single-line text input. The third field is labeled 'Message:' and is a multi-line text area. Below the text area is a button labeled 'Send your message'.

Name:

E-mail:

Message:

Dupa ce forma este completata, datele sunt trimise serverului ca un **query string**.

Forms

```
<form id="myform">
```

```
<!-- lista de elemente asociate formelor -->
```

```
<input>
```

```
<textarea>
```

```
<fieldset>
```

```
<select>
```

```
<button type="submit"> Submit </button>
```

```
</form>
```

```
<input type="submit" form="myform" value="Submit">
```

```
<button type="submit" form="myform"> Submit </button>
```

Elemente specifice formelor: tagul **input**


Elementul HTML **<input>** poate fi afișat în mai multe moduri, în funcție de valorile atributului **type**.

Input Type Text

(câmp pentru introducerea unui text pe o singura linie)

Varianta 1 (eticheta label simpla)

```
<label><input type="text" name="client">  
Numele clientului </label>
```



A screenshot of a web form element. It consists of a rectangular text input field with a light gray border. To the right of the input field, the text 'Numele clientului' is displayed in a dark gray font.

Varianta 2 (eticheta label cu atributul for)

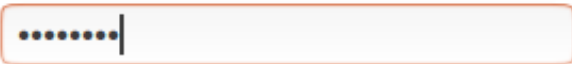
```
<label for ="user">Numele clientului </label>  
<input type="text" name="client" id="user">
```

Atribute specifice: size (width pentru câmp), maxlength, required

Elemente specifice formelor: tagul **input**

Input Type password

```
<label><input type="password" name="parola" maxlength="8">  
Parola</label>
```



Parola

Input Type text cu pattern

```
<label><input type="text" id="cnp" pattern="[0-9]{13}"> CNP  
</label>
```

```
<label><input type="text" id="ci" pattern="[A-Z]{2}[0-9]{4}">C.I  
</label>
```

Elemente specifice formelor: tagul **input**

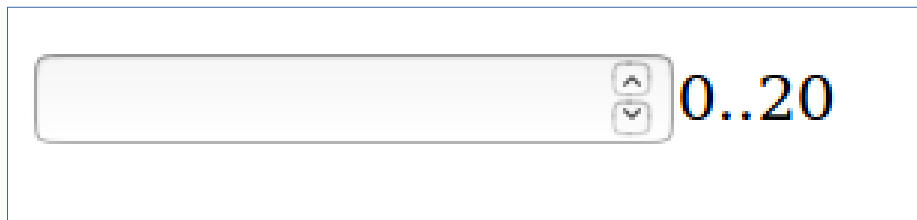
Input Type file

`<label>Ataseaza fisier <input type="file" name="fileField"> </label>`

A screenshot of a web form element. It consists of the text "Ataseaza fisier" followed by a button labeled "Răsfoiește..." and then the text "Niciun fișier selectat." The button has a light gray background and rounded corners.

Input Type number

`<label><input type="number" min="0" max="20" name="numar">
0..20 </label>`

A screenshot of a web form element. It shows a number input field with a light gray background and rounded corners. To the right of the input field is a small vertical spinner control with up and down arrows. To the right of the spinner, the text "0..20" is displayed in a bold, black font.

Elemente specifice formelor: tagul **input**

Input Type range

<label>

```
<input type="range" min="0" max="20" step="2" name="numar">  
0..20 </label>
```



Input Type email

```
<label><input type="email" name="adresa" required multiple>  
Email </label>
```

Atributul **multiple** funcționează doar pentru input cu type = file, email

Elemente specifice formelor: tagul **input**

Input Type radio

☐ iute
☒ dulce

```
<label><input type="radio" name="gust" value="iute"> iute </label><br>
```

```
<label><input type="radio" name="gust" value="dulce" checked> dulce</label>
```

numai unul din butoanele radio poate fi selectat, de aceea atributul **name** trebuie sa aiba aceeaasi valoare pentru toate optiunile

Input Type checkbox

```
<label><input type="checkbox" name="topping1" value="rosii"> rosii</label>
```

```
<label><input type="checkbox" name="topping2" value="branza">branza  
</label>
```

pot fi selectate simultan mai multe butoane checkbox

☒ rosii
☒ branza

În HTML5 au fost introduse și alte valori pentru input type

<input type=

color \\ introducerea unei culori

date \\introducerea unei date în formatul
yyyy-mm-dd

image \\creaza un buton grafic folosind o imagine

tel \\ introducerea unui numar de tel

time \\ data în formatul hh:mm

url \\ introducerea unui URL în formatul
urlscheme://restofurl

week \\data în formatul yyyy-Www

(ex. 2017-W45)

Elemente specifice formelor: tagul **fieldset**

Este folosit pentru gruparea elementelor in interiorul formelor

```
<fieldset>
```

```
<label><input type="radio" name="gust" value="iute"> iute </label><br>
```

```
<label><input type="radio" name="gust" value="dulce"> dulce</label>
```

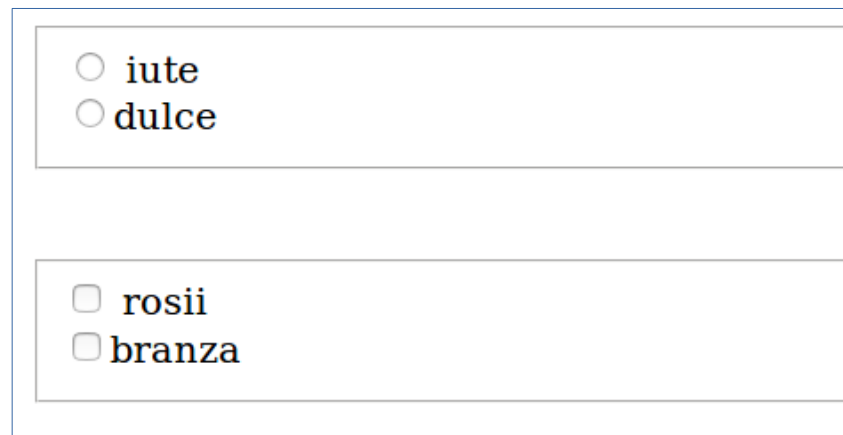
```
</fieldset>
```

```
<fieldset>
```

```
<label><input type="checkbox" name="topping1" value="rosii"> rosii</label><br>
```

```
<label><input type="checkbox" name="topping2" value="branza">branza  
</label>
```

```
</fieldset>
```



<input type="radio"/> iute <input type="radio"/> dulce
<input type="checkbox"/> rosii <input type="checkbox"/> branza

Elemente specifice formelor: **textarea**

```
<textarea name="comments" rows="10" cols="48">  
Text initial </textarea>
```

Reprezintă un camp de editare a textului pe mai multe linii
Implicit cols = 20

Elemente specifice formelor: **select**

Reprezintă un control care ofera un meniu de optiuni

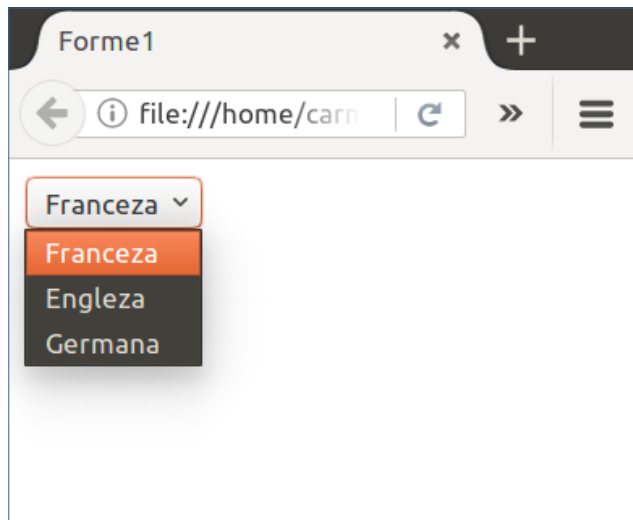
```
<select name="optional">
```

```
<option value="franceza"> Franceza </option>
```

```
<option value="engleza">Engleza </option>
```

```
<option value="germana">Germana</option>
```

```
</select>
```



Submiterea formelor: elementele **input** si **button**

Input Type submit

```
<input type="submit" value="Trimite">
```

Input Type reset

```
<input type="reset" value="Reseteaza" size="12">
```

Button Type submit

```
<button type="submit"> Trimite </button>
```

Button Type reset

```
<button type="reset"> Reseteaza </button>
```

Elementul **button** poate fi folosit in afara formelor cu atributul `type="button"`

Exemplu

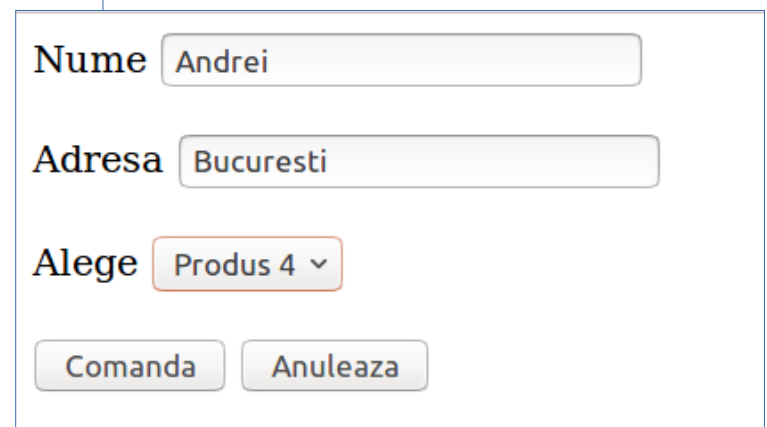
```
<form id="myform">
  <label for="nu">Nume</label>
  <input id="nu" type="text" name="nume"><br><br>

  <label for="adr">Adresa</label>
  <input id="adr" type="text" name="adresa"><br><br>

  <label for="adr">Alege</label>
  <select name="produs">
    <option value="p1">Produs 1</option>
    <option value="p2">Produs 2</option>
    <option value="p3">Produs 3</option>
    <option value="p4">Produs 4</option>
  </select><br><br>
  <button type="submit">Comanda</button>
  <input type="reset" value="Anuleaza">
</form>
```

sau în afara elementului <form>:

```
<button type="submit"
form="myform">Comanda</button>
<input type="reset" form="myform" value="Anuleaza">
```



Nume

Adresa

Alege

URSURI/EXEMPLE/form1.html?nume=Andrei&adresa=Bucuresti&produs=p4



Atributele formelor

```
<form target="_blank" method="post" action="URL">
```

```
target = "_self" / "_blank" / "numeiframe"  
// locul in care este afisat raspunsul dupa  
submiterea formei (implicit: _self)
```

```
method = "post"/"get"  
// metoda folosita pentru comunicarea  
cu serverul (implicit: get)
```

```
action = URL-ul unui program care va procesa forma  
(implicit: URL paginii curente)
```


Exemplu: target cu iframe

```
<form action="http://localhost:8080/cale" method="GET"
target="numeiframe">
  <label>Nume:</label>
  <input type="text" name="name">
<br>
  <label> Varsta:</label>
  <input type="text" name="age">
<br>
  <label>Localitate:</label>
  <select name="city">
    <option value="Bucuresti" selected>Bucuresti</option>
    <option value="Timisoara">Timisoara</option>
  </select>
<br>
<button type="submit" id="buton"> Trimite </button>
</form>
<br>
<iframe name="numeiframe"> <iframe>
</body>
```

Nume:

Varsta:

Localitate:

Buna George din Timisoara

Atributul method = "post"/"get"

```
POST /cale HTTP/1.1
Host: localhost:8080
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:56.0) Gecko/20100101 Firefox/56.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: ro-R0,ro;q=0.8,en-US;q=0.6,en-GB;q=0.4,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 33
Connection: keep-alive
Upgrade-Insecure-Requests: 1

name=George&age=20&city=Timisoara
```

POST
Query string in corpul mesajului

```
GET /cale?name=George&age=20&city=Timisoara HTTP/1.1
Host: localhost:8080
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:56.0) Gecko/20100101 Firefox/56.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: ro-R0,ro;q=0.8,en-US;q=0.6,en-GB;q=0.4,en;q=0.2
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1
```

GET
Query string in URL

Aplicația server
(Node.js)

```
var http = require('http');
var url = require('url');

http.createServer(function (req, res) {
  var url_parts = url.parse(req.url, true);
  var query = url_parts.query;

  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Buna ' + query.name + ' din ' + query.city);
}).listen(8080); |
```

Atributul action cu mailto

```
<form action="mailto:mymail@yahoo.com" method="post" enctype="text/plain">  
Name:<br>  
<input type="text" name="name"><br>  
E-mail:<br>  
<input type="text" name="mail"><br>  
Comment:<br>  
<input type="text" name="comment" size="50"><br><br>  
<input type="submit" value="Send">  
<input type="reset" value="Reset">  
</form>
```

La submiterea formei
se trimite un email

Name:

E-mail:

Comment:

De la: carmen <carmen_stama...>

Către: mymail@yahoo.com

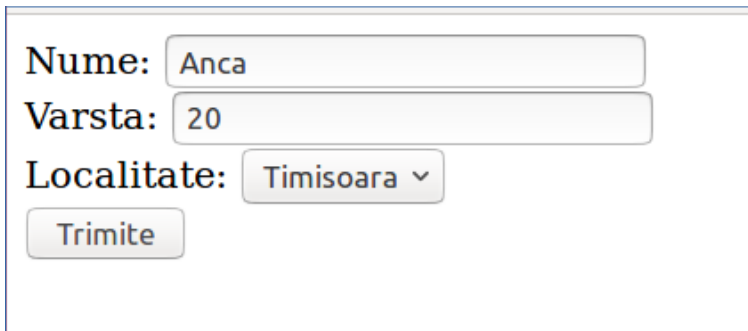
Subiect: Formular postat de Firefox

name=Carmen
mail=carmen_stama@yahoo.com
comment=hello

O forma este reprezentata în JavaScript de un obiect HTMLFormElement

proprietatea **elements** intoarce colectia campurilor formei (array-like object)

proprietatea **length** intoarce numarul de campuri



Nume: Anca
Varsta: 20
Localitate: Timisoara ▼
Trimite

```
<script>
window.onload=function()
{
var f=document.getElementById("forma");
alert(f.elements); //[objectHTMLFormControlsCollection]
alert(f.length); // 4
alert(f.elements[0].value); //Anca
}
</script>
```

```
<form id="forma">
  <label>Nume:</label>
  <input type="text" name="name" value="Anca">
<br>
  <label> Varsta:</label>
  <input type="text" name="age" value="20">
<br>
  <label>Localitate:</label>
  <select name="city">
    <option value="Bucuresti" selected>Bucuresti</option>
    <option value="Timisoara">Timisoara</option>
  </select>
<br>
  <button type="submit" id="buton"> Trimite </button>
</form>
```

Forme: proprietatea value

- pentru `<input type="text">` si `<textarea>` proprietatea value reprezinta valoarea introdusa de utilizator la submiterea formei si este de tipul string
- proprietatea value pentru `<option>` reprezinta valoarea optiunii
- altfel reprezinta valoarea atributului value

```
<input type="text" value="5">
```

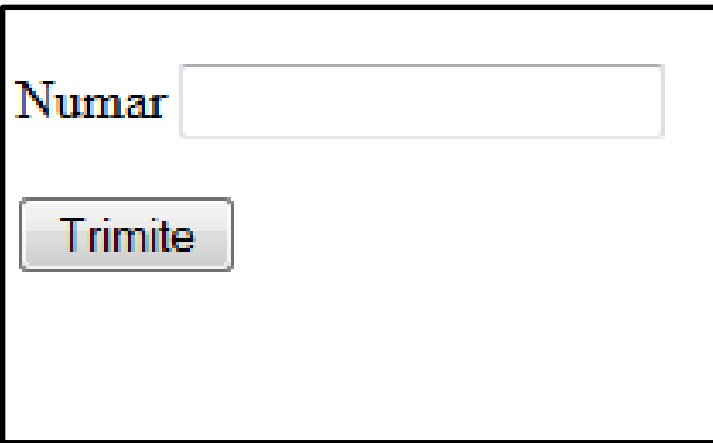
```
<input type="submit" value="Trimite">
```

```
<input type="radio" value="v1">
```

```
<input type="checkbox" value="v1">
```

Exemplu

Citesc un numar folosind
campul input; la click pe
buton afisez suma



The image shows a simple web form. It consists of a text input field with the label 'Numar' to its left. Below the input field is a button with the label 'Trimite'.

```
window.onload = myMain;
```

```
function myMain() {  
  document.getElementById("citire").onsubmit  
    = suma;  
}
```

```
function suma() {  
  var x = document.getElementById("nr").value;  
  var s = 0;  
  for (var i=1; i<= parseInt(x) ; i++) s=s+i;  
  alert('Suma este ' + s);  
}
```

```
<form id="citire" >  
  Numar <input type="text" id="nr" >  
  <button type="submit"> Trimite </button>  
</form>
```

```
<form id="citire" >  
  Numar <input type="text" id="nr" >  
  <button type="submit"> Trimite  
</button>  
</form>
```

```
window.onload = myMain;  
  
function myMain() {  
  document.getElementById("citire").onsubmit  
    = suma;  
}  
  
function suma() {  
  var x = document.getElementById("nr").value;  
  var s = 0;  
  for (var i=1; i<= parseInt(x) ; i++) s=s+i;  
  alert('Suma este ' + s);  
}
```

doua evenimente

Name	Target	Property	Handler
load	window	onload	myMain
submit	form	onsubmit	suma

Forme: proprietati specifice campurilor

- proprietatea checked (boolean) pentru elemente

`<input type="radio" value="v1">`

`<input type="checkbox" value="v1">`

- proprietatile campului `<select>`

options array-like care contine optiunile

selectedIndex pentru determinarea

optiunii selectate

`<select multiple>` atributul multiple permite selectarea
mai multor optiuni

- proprietatile campului `<fieldset>`

elements array-like care contine componentele

Deși formele au modalități specifice de selectare este recomandat ca, atât formele, cât și câmpurile lor să fie accesate folosind metoda `getElementById`

Exemplu: prelucrarea butoanelor radio

```
<input type="radio" name="a" id="a1" value="0">  
<input type="radio" name="a" id="a2" value="1">  
<input type="radio" name="a" id="a3" value="2">
```

```
var sel = 0;  
for (var i = 1; document.getElementById('a'+i); i++) {  
    if (document.getElementById('a'+i).checked) {sel = i; break;}  
}
```

Exemplu - quiz

```
<form name="quiz" id="quiz">
<fieldset><legend> 3+5=? </legend>
<input type="radio" name="q1" value="0">10<br>
<input type="radio" name="q1" value="1">8 <br>
<input type="radio" name="q1" value="0">55<br>
</fieldset>

<fieldset><legend> 3*5=? </legend>
<input type="radio" name="q2" value="0">10<br>
<input type="radio" name="q2" value="1">15 <br>
<input type="radio" name="q2" value="0">55<br>
</fieldset>

<button type="submit" id="buton"> Click me </button>
</form>
```

3+5=?

☐ 10

☐ 8

☐ 55

3*5=?

☐ 10

☐ 15

☐ 55

Click me

Exemplu - quiz

```
window.onload = myMain;
```

```
function myMain() {  
  document.getElementById("quiz").onsubmit = totalQuiz;}
```

```
function totalQuiz() {  
  var fe = document.getElementById("quiz").elements;  
  var q1 = fe[0].elements;  
  var q2 = fe[4].elements;  
  var x=0;  
  for (var i = 0; i<q1.length; i++) if (q1[i].checked) x = x + parseInt(q1[i].value);  
  for (var i = 0; i<q2.length; i++) if (q2[i].checked) x = x + parseInt(q2[i].value);  
  alert(x);  
}
```

Evenimentul **change**

Modificarea unui câmp urmat de Enter, Tab, schimbarea focusului declanșează evenimentul **change**.

```
<head>
<script>
window.onload = function () {
    document.getElementById("tt").onchange = function (){
        alert(document.getElementById("tt").value); }
    }
</script>
</head>
<body>
    <textarea id="tt" width="50" height="20" >!!! </textarea>
</body>
```

Înainte de ev. change

!!!

Astazi invatam despre
formulare

Astazi invatam despre
formulare

Ok

```
<script>
window.onload=main;

function main() {
obp=document.getElementById("scie");
obsel=document.getElementById("sel");
obsel.onchange = function () {
                obp.innerHTML+=obsel.options[obsel.selectedIndex].value;
            }
}
</script>
```

```
<body>
<p id="scie">Ati selectat:</p>
<form>
    <select id="sel">
        <option> val1 </option>
        <option> val2 </option>
        <option> val3 </option>
    </select>
</form>
</body>
```

Înainte de declansarea ev. change

Ati selectat:

val1 ▾

dupa declansarea ev. change

Ati selectat: val3

val3 ▾