



# CodeCamp JavaScript

Algorithmics Brasov-Smart Kids

# Ziua 2

**Evenimente, culori și  
animații**

---

# Introducere

## Obiective

- ✓ Înțelegem ce sunt **evenimentele** în JavaScript (click, tastă, mouse).
- ✓ Legăm **interacțiuni** de elemente (buton, div, canvas).
- ✓ Creăm **animații simple** (mișcare, schimbare de culoare, efecte).

# Ce este un eveniment?

Exemple din viața reală:

- Apeși un buton → becul se aprinde.
- Muți mouse-ul → săgeata se mișcă.
- Apeși o tastă → apare o literă pe ecran.



În JavaScript, putem spune calculatorului:



„Dacă se întâmplă X, fă Y.”

# Exemple de evenimente

- Click → când apeși pe un buton.
- Mouseover → când treci cu mouse-ul peste ceva.
- Keydown → când apeși o tastă.

Diferența funcție directă vs. la eveniment:

**Funcție directă:** se execută imediat

```
6 alert("Salut!");
```

**Funcție la eveniment:** se execută doar când se întâmplă ceva

```
8 b.onclick = function() {  
9     alert("Salut, ai dat click!");  
10 };
```

# Ce este o animație?

👉 O animație = ceva care se schimbă pas cu pas.

Exemple:

- O minge care se mișcă.
- O culoare care se transformă.
- Un pătrat care crește și se micșorează.

# Cum facem o animație?

Folosești timpul → calculatorul repetă o acțiune din nou și din nou.



Metode:

- *setInterval* → repetă la fiecare X milisecunde.
- *setTimeout* → așteaptă X timp și apoi rulează.
- *requestAnimationFrame* → (extra, pentru jocuri și animații mai fluide).

# Exemplu animație simplă

```
HTML
1 <div id="box"></div>

CSS
1 #box {
2   width:50px;
3   height:50px;
4   background:red;
5   position:absolute;
6 }

JS
1 let box = document.getElementById("box");
2 let x = 0;
3
4 function move() {
5   x += 5;
6   box.style.left = x + "px";
7 }
8
9 setInterval(move, 100);
```

💡 Rezultat: pătratul roșu se mișcă spre dreapta!



# Joacă-te cu animațiile!

💡 Idei de modificări:

- Schimbă **viteza** (cât adaugi la x).
- Schimbă **culoarea**.
- Mută pătratul **în sus și în jos**.
- Fă-l să se **oprească la margine**.

# Joacă-te cu animațiile!

- Schimbă **viteza** (cât adaugi la x).

JS

```
2
3 • function move() {
4   x += 20; // dacă pui 1 e foarte încet, dacă pui 50 e foarte repede
5   box.style.left = x + "px";
6 }
7
```

# Joacă-te cu animațiile!

- Schimbă **culoarea**.

JS

```
1 box.style.background = "blue"; // devine albastru
2
3 if (x > 100) {
4     box.style.background = "green";
5 }
6
```

# Joacă-te cu animațiile!

- Mută pătratul **în sus și în jos**.

JS

```
1 let box = document.getElementById("box");
2 let y = 0;
3
4 function move() {
5     y += 5;
6     box.style.top = y + "px";
7 }
8
9 setInterval(move, 100);
```

# Joacă-te cu animațiile!

- Fă-l să se **oprească la margine**.

```
JS
1  let viteza = 5;
2
3
4  function move() {
5      x += viteza;
6      box.style.left = x + "px";
7
8      if (x > 300 || x < 0) {
9          viteza = -viteza; // schimbă direcția
10     }
11 }
12 setInterval(move, 50);
```

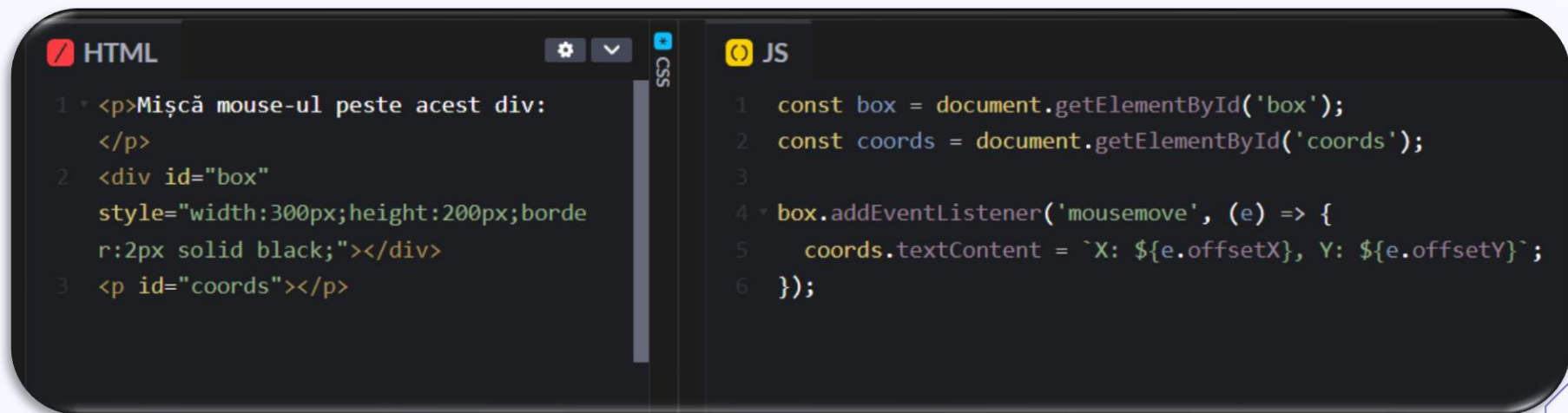


# Interacțiuni în JavaScript

## Exersare și înțelegere:

- **mousemove** – urmărirea cursorului

**Obiectiv:** să arătăm cum coordonatele mouse-ului se pot folosi pentru a interacționa.



```
HTML
1 <p>Mișcă mouse-ul peste acest div:</p>
2 <div id="box"
  style="width:300px;height:200px;border:2px solid black;"></div>
3 <p id="coords"></p>

CSS

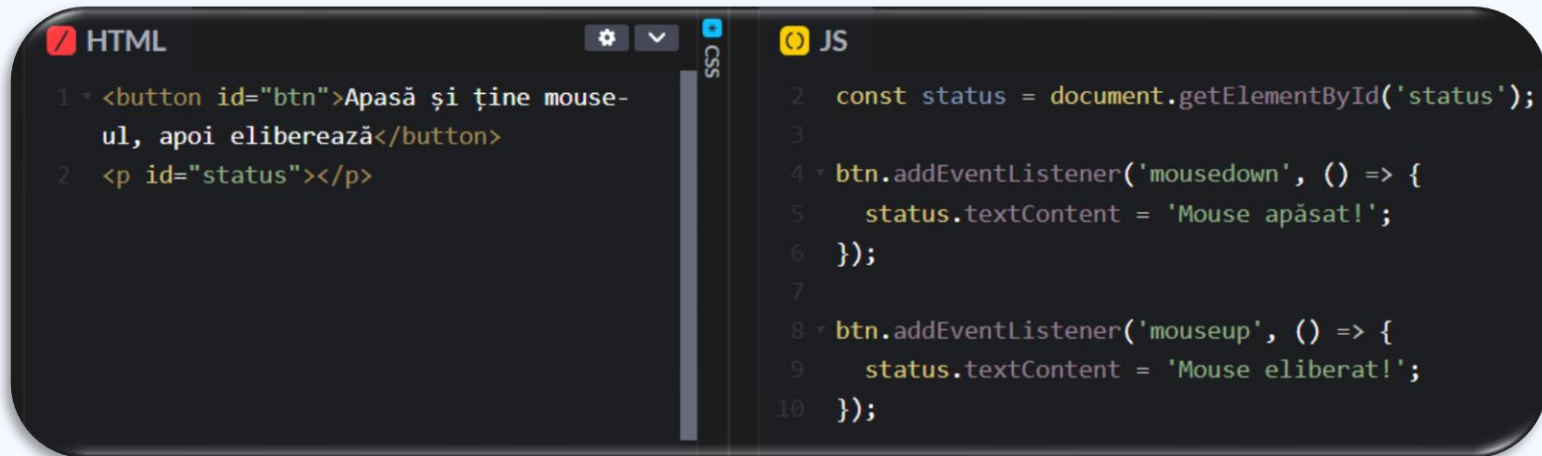
JS
1 const box = document.getElementById('box');
2 const coords = document.getElementById('coords');
3
4 box.addEventListener('mousemove', (e) => {
5   coords.textContent = `X: ${e.offsetX}, Y: ${e.offsetY}`;
6 });
```

- **mousemove** se declanșează la fiecare mișcare a mouse-ului peste div.
- offsetX și offsetY arată poziția cursorului **în interiorul div-ului**.
- Textul se actualizează în timp real, vizualizând interactiv coordonatele.

Exersare și înțelegere:

- **mouseup** – detectarea eliberării mouse-ului

**Obiectiv:** să arătăm când utilizatorul eliberează butonul mouse-ului.



```
HTML
1 <button id="btn">Apasă și ține mouse-
  ul, apoi eliberează</button>
2 <p id="status"></p>

JS
2 const status = document.getElementById('status');
3
4 btn.addEventListener('mousedown', () => {
5   status.textContent = 'Mouse apăsat!';
6 });
7
8 btn.addEventListener('mouseup', () => {
9   status.textContent = 'Mouse eliberat!';
10 });
```

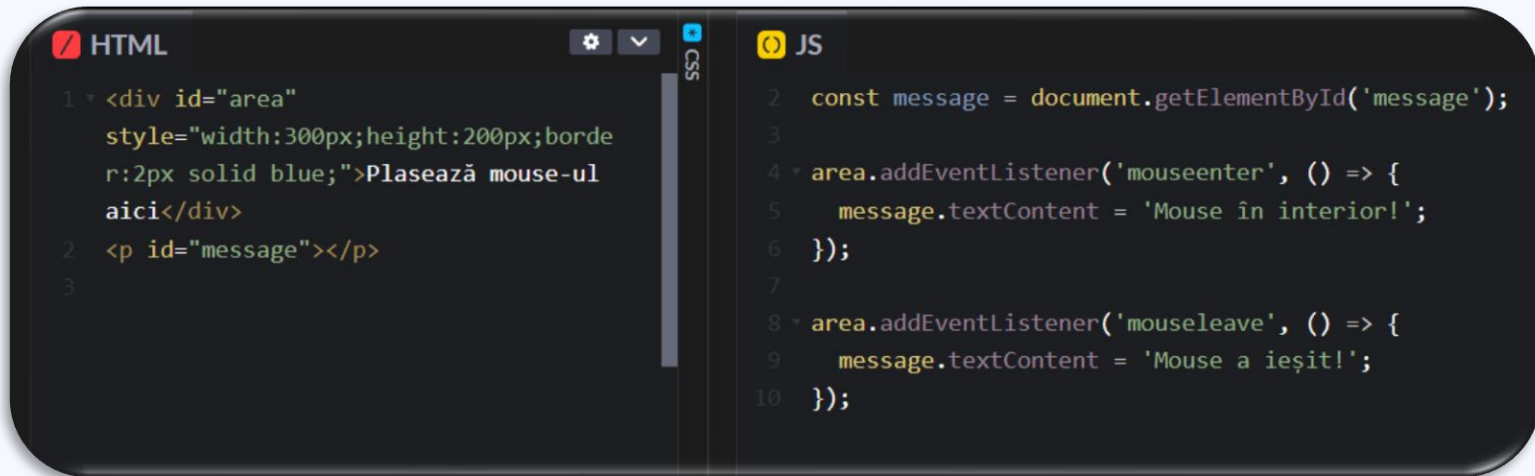
- **mousedown** detectează apăsarea butonului.
- **mouseup** detectează momentul în care utilizatorul eliberează mouse-ul.
- Textul se schimbă dinamic, vizualizând acțiunea.



## Exersare și înțelegere:

- **mouseleave** – detectarea ieșirii cursorului

**Obiectiv:** să demonstrăm cum putem reacționa dacă mouse-ul părăsește zona unui element.



```
HTML
1 <div id="area"
  style="width:300px;height:200px;border:2px solid blue;">Plasează mouse-ul aici</div>
2 <p id="message"></p>
3

JS
2 const message = document.getElementById('message');
3
4 area.addEventListener('mouseenter', () => {
5   message.textContent = 'Mouse în interior!';
6 });
7
8 area.addEventListener('mouseleave', () => {
9   message.textContent = 'Mouse a ieșit!';
10 });
```

- **mouseenter** – eveniment care apare când mouse-ul intră în div.
- **mouseleave** – eveniment care apare când mouse-ul părăsește div-ul.
- Textul se schimbă în funcție de poziția mouse-ului, demonstrând interactivitatea



# Exerciții cu animații



# Schimbă culoarea unui pătrat la click

**Cerinta:** Când apeși pe pătrat, acesta își schimbă culoarea.

HTML

```
1 <body>
2   <div id="box"></div>
3   <script
4     src="script.js">
5   </script>
6 </body>
```

CSS

```
1 #box {
2   width: 100px;
3   height: 100px;
4   background: red;
5   margin: 50px auto;
6   cursor: pointer;
7 }
8
```

JS

```
1
2
3 box.addEventListener("click", function() {
4   box.style.background = "blue";
5 });
6
```

# Creează o minge care sare

**Cerinta:** O minge se mișcă singură și ricoșează din pereții ferestrei.

HTML

```
1 <body>
2   <div id="ball"></div>
3   <script src="script.js"></script>
4 </body>
5
```

CSS

```
1 #ball {
2   width: 50px;
3   height: 50px;
4   background: orange;
5   border-radius: 50%;
6   position: absolute;
7   top: 50px;
8   left: 50px;
9 }
10
```

JS

```
1 let ball = document.getElementById("ball");
2 let x = 50, y = 50;
3 let vx = 4, vy = 3;
4
5 function move() {
6   x += vx;
7   y += vy;
8
9   // schimbă direcția dacă atinge marginea
10  if (x <= 0 || x + 50 >= window.innerWidth) vx = -vx;
11  if (y <= 0 || y + 50 >= window.innerHeight) vy = -vy;
12
13  ball.style.left = x + "px";
14  ball.style.top = y + "px";
15 }
16
17 setInterval(move, 20);
18
```



# Listă de mini-exerciții extra

## ■ Ușoare (1–2 linii de cod schimbate)

1. Un pătrat care își schimbă culoarea la **click**.
2. Un cerc care devine mai mare când treci cu **mouse-ul** peste el.
3. Un buton care schimbă textul unei etichete („Salut!” → „Bravo!”).
4. O imagine care dispare și reapare când apeși pe ea.

## ■ Medii (mică logică)

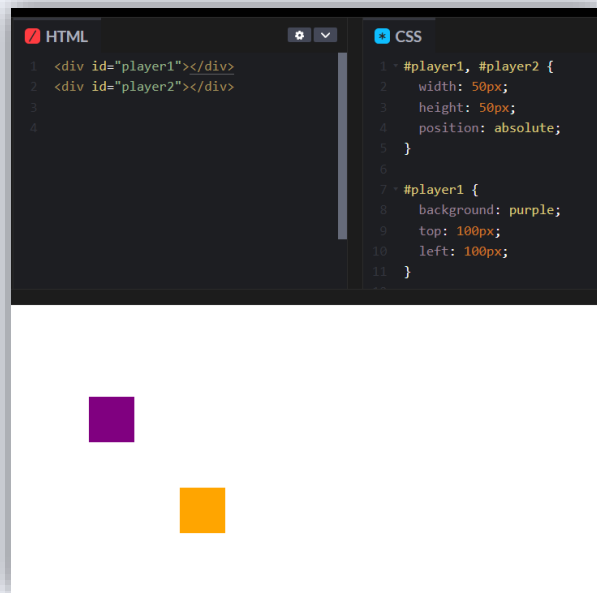
1. O minge care sare stânga–dreapta și nu iese din ecran.
2. O figură (pătrat sau cerc) care se mișcă cu **tastele săgeți**.
3. Un pătrat care își schimbă culoarea **aleator** de fiecare dată când dai click.
4. Un buton care, atunci când e apăsat, creează **o nouă figură** pe ecran.

## ■ Avansate (pentru cei mai curajoși 😊)

1. O minge care ricoșează în toate direcțiile (sus/jos/stânga/dreapta).
2. O figură care „urmează mouse-ul” pe ecran.
3. O mini „bombă cu ceas” → un cerc care își schimbă culoarea la fiecare secundă și dispare după 5 secunde.
4. Două figuri care se mișcă diferit (una cu săgeți, alta cu tastele W/A/S/D).



# Două figuri care se mișcă diferit (una cu săgeți, alta cu tastele W/A/S/D).



Cerință extra:

Adăugați un al treilea player în proiect.

*Link-ul proiectului CodePen:*

<https://codepen.io/Violeta-the-sans/pen/ra0KagK>

# Ce am învățat azi?

## Recapitulare



- Ce este un **eveniment** în JavaScript.
- Cum folosim **onclick**, **onkeydown**.
- Cum atașăm evenimente cu **addEventListener**.
- Cum facem o **animație simplă** cu **setInterval** și **setTimeout**.
- Cum facem obiecte care **se mișcă pe ecran**.
- Cum detectăm **marginea ecranului** și schimbăm direcția.
- Cum să controlăm figuri cu **tastele săgeți** sau **W/A/S/D**.