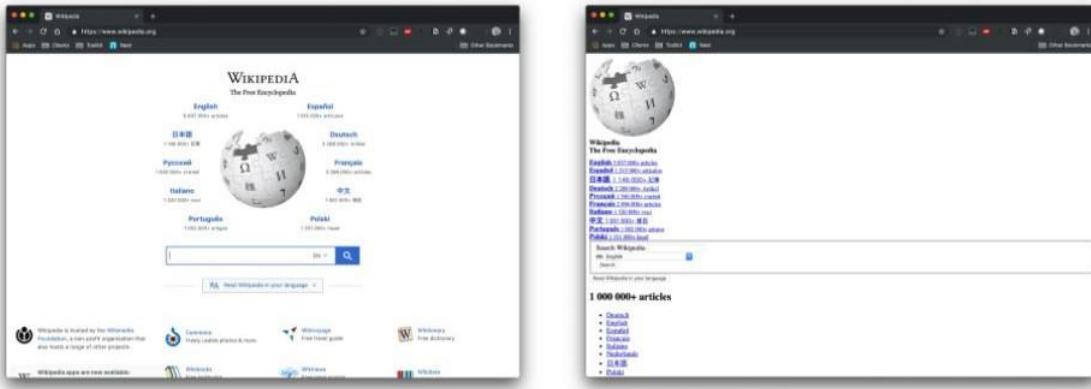


## Introducere în CSS



CSS (*Cascading Style Sheets*) este un limbaj de scriptare utilizat pentru crearea paginilor web, în special pentru **stilizarea elementelor și tag-urilor HTML**. Acesta este direct legat și se utilizează împreună cu scriptul HTML, **rolul CSS-ului fiind de a oferi conținutului design-ul dorit**. Astfel, se va realiza o interfață completă a paginii web.

[Cum arată o pagină web fără CSS:](#)



### La ce poate fi folosit limbajul CSS

Limbajul CSS poate fi folosit pentru:

- ✓ Crearea unei pagini web simple/complexă folosind diverse efecte
- ✓ Setarea culorii, mărimei și fontului textului din pagina web
- ✓ Crearea unui layout (suport pentru elementele HTML), adăugând margine, culoare, imagine de fond și multe altele

### Inserarea limbajului CSS într-un fișier HTML

Codul CSS poate fi inserat în 3 moduri:

- ◆ **Extern** – ca fișier .css

```
1 <html>
2   <head>
3     <link rel="stylesheet" type="text/css" href="fisier.css" />
4   </head>
5   <body>
6
7   </body>
8 </html>
```

**Obs:** un fișier .css se poate scrie/edita în orice editor de text (Notepad, Notepad++, Sublime, Visual Studio Code etc.)

◆ **Intern** – în head-ul documentului *.html*

```
1  <html>
2   <head>
3     <style type="text/css">
4       p { color: black; }
5     </style>
6   </head>
7   <body>
8     <p>Primul meu paragraf</p>
9   </body>
10  </html>
```

În acest caz, se va folosi tag-ul special **<style> </style>**, împreună cu atributul **type = "text/css"**.

◆ **Inline** – în tag-ul care se dorește a fi personalizat

```
1 <html>
2   <head>
3
4   </head>
5   <body>
6     <p style="text-align:right; color:green; font-family:'times new roman'; margin-left:15px;">Primul meu paragraf</p>
7   </body>
8 </html>
```

**Atenție** la modul de scriere al elementelor în atributul **style**:

```
<p style="text-align:right; color:green; font-family:'times new roman'; margin-left:15px;">
```

Valoarea elementului *font-family* este compus din 3 cuvinte. Astfel, se vor folosi ghilimele simple pentru valorile cu multiple cuvinte, iar ghilimele duble se vor folosi pentru a încapsula întreaga valoare a atributului **style**.

◆ **Toate cele de mai sus**

### Ordinea de anulare a metodelor de inserare CSS

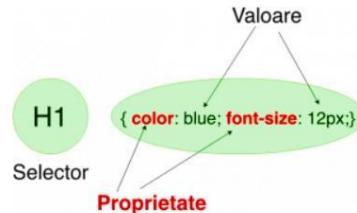
Ordinea de reproducere în browser a comenziilor CSS va fi următoarea:

1. Formatarea CSS in-line (în interiorul tag-ului HTML)
2. Formatarea CSS internă (în secțiunea head a documentului)
3. Formatarea CSS externă (fișier .css extern)
4. Standardul browser-ului (valori CSS atribuite standard)

### Construcția de bază în CSS

Elementele principale ale unei construcții CSS:

- **Selector** – folosit pentru a identifica elementul HTML pentru care se aplică stilul
- **Acolade** – folosite pentru a defini și delimita proprietățile și valorile care se aplică elementului HTML



- **Proprietăți** – elementele de aspect care se doresc a fi modificate
- **Valori ale proprietăților** – noile valori care le atribuim proprietăților

## Tipuri de selectori

Pentru a stabili o formatare personalizată în CSS, putem folosi mai multe tipuri de selectori:

### ◆ Tag-ul simplu

```
1 p {  
2     color: green;  
3     text-align: center;  
4 }
```

Acum lucru se traduce prin:

*Textul scris în tag-ul <p> va avea culoarea verde și va fi aliniat în centru.*

## Gruparea selectorilor

O regulă CSS spune că mai multe elemente cu aceleași proprietăți se pot grupa:

```
1 p, h1, h2, h3 {  
2     color: green;  
3     text-align: center;  
4 }
```

Acum lucru se traduce prin:

*Textele scrise în tagurile <p>, <h1>, <h2>, <h3> vor avea culoarea verde și vor fi aliniate în centru.*

### ◆ Atributul **class**

Clasele de elemente se pot stabili pentru un singur tag HTML sau pentru orice alt tag care are specificat la atributul **class** valoarea aleasă.

```
1 .center {  
2     text-align: center;  
3 }  
  
5 .left {  
6     text-align: left;  
7 }
```

```
1 <html>  
2   <head>  
3     <link rel="stylesheet" type="text/css" href="fisier.css" />  
4   </head>  
5   <body>  
6     <p class="center"> Primul paragraf</p>  
7     <p class="left"> Al doilea paragraf</p>  
8   </body>  
9 </html>
```

În cazul în care se dorește o clasă care să poată fi folosită în mai multe tag-uri diferite, va trebui să se eliminate numele tag-ului din declararea clasei:

```
1 .center {  
2     text-align: center;  
3 }  
  
5 .left {  
6     text-align: left;  
7 }
```

```
1 <html>  
2   <head>  
3     <link rel="stylesheet" type="text/css" href="fisier.css" />  
4   </head>  
5   <body>  
6     <p class="center"> Primul paragraf</p>  
7     <h1 class="left"> Primul titlu</h1>  
8   </body>  
9 </html>
```

**Obs:** NU este recomandat ca numele claselor să înceapă cu un număr (ex: *.67clasamea*)

### ◆ ID-ul

ID-ul este folosit exact ca o clasă, doar că diferă sintaxa de definire.

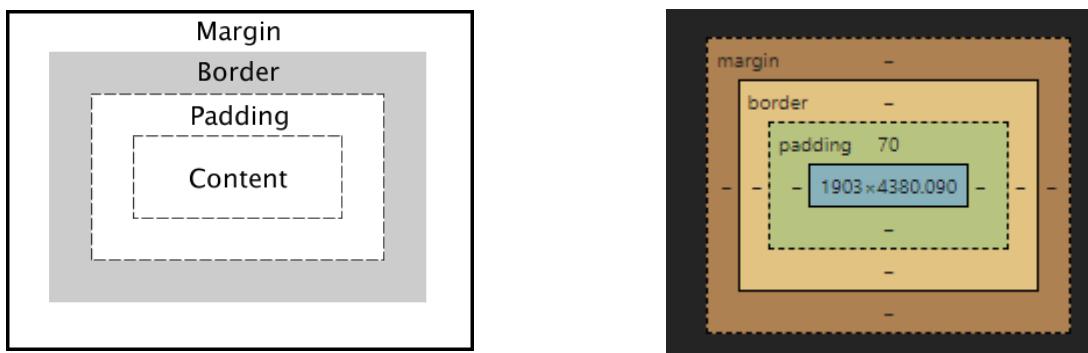
```
1 #center {  
2     text-align: center;  
3 }  
4  
5 #left {  
6     text-align: left;  
7 }
```

```
1 <html>  
2     <head>  
3         <link rel="stylesheet" type="text/css" href="fisier.css" />  
4     </head>  
5     <body>  
6         <p id="center"> Primul paragraf</p>  
7         <h1 id="left"> Primul titlu</h1>  
8     </body>  
9 </html>
```

**Obs:** în general, ID-urile sunt folosite în JavaScript, pentru a localiza un anumit element HTML și a executa acțiuni asupra acestuia. 🤖

### Margin, Border, Padding

În CSS se poate încadra conținutul paginii, stabilind mai multe distanțe:



**Content** – conținutul propriu-zis al paginii (text, imagini, formulare, tabele etc.)

**Border** – marginea și cutia care încadrează conținutul

**Padding** – distanța dintre border și conținut

**Margin** – distanța dintre border și marginea paginii

```
1 div {  
2     width:200px;  
3     padding:15px;  
4     border:5px solid;  
5     margin:15px;  
6 }
```

În funcție de numărul de valori ale proprietății, avem:

sus – dreapta – stânga – jos → **margin: 30px 20px 30px 50px;**  
sus – dreapta & stânga – jos → **margin: 30px 20px 50px;**  
sus & jos – dreapta & stânga → **margin: 30px 20px;**  
sus & jos & dreapta & stânga → **margin: 30px;**

Sau specifice individual:

```
margin-top: 100px;  
margin-bottom: 100px;  
margin-right: 150px;  
margin-left: 80px;
```

## CSS Layout: Flexbox și Grid

FLEXBOX (Flexible Box Layout)



Ce este

Flexbox este un sistem de layout unidimensional — elementele se aranjează pe un singur ax (orizontal sau vertical).

Este excelent pentru aliniere, centrare și distribuire de spațiu între elemente.

### Proprietăți de bază

► Pe container:

```
display: flex;          /* activează Flexbox */
flex-direction: row;    /* direcția principală: row | column */
justify-content: center; /* aliniere pe axa principală */
align-items: center;   /* aliniere pe axa secundară */
gap: 10px;             /* spațiu între elemente */
flex-wrap: wrap;       /* permite trecerea pe rândul următor */
```

► Pe elementele copil:

```
flex: 1;                /* ocupă spațiu egal cu celelalte */
align-self: flex-start; /* aliniere individuală */
order: 2;               /* schimbă ordinea elementelor */
```

### Avantaje Flexbox

- Ușor de centrat elemente (orizontal și vertical)
  - Spațiere uniformă (gap, justify-content)
  - Ordine ușor de schimbat (order)
  - Perfect pentru layout-uri simple, meniuri, carduri, formulare
- ⚠ Limitat la **un singur ax** — nu face layout 2D (rânduri + coloane simultan)

## CSS GRID Layout



CSS Grid este un sistem de layout bidimensional — elementele se aranjează pe rânduri și coloane. Ideal pentru layout-uri complexe: pagini, galerii, structuri principale.

► Pe container:

```
display: grid;  
grid-template-columns: repeat(3, 1fr); /* 3 coloane egale */  
grid-template-rows: auto;  
gap: 15px;
```

► Pe elemente:

```
grid-column: 1 / 3; /* ocupă coloanele 1 și 2 */  
grid-row: 2; /* se află pe rândul 2 */
```

## Avantaje Grid

- Permite layout **2D** (rânduri + coloane)
  - Control precis asupra dimensiunilor și poziției elementelor
  - Suportă unități moderne (fr, minmax(), auto-fit)
  - Ideal pentru structuri complexe: pagini, galerii, dashboard-uri
- ⚠ Nu e cel mai potrivit pentru alinierea simplă pe un singur rând → acolo e mai bun Flexbox



[CSS Tutorial \(w3schools.com\)](#)

[Tutorial CSS - Introducere în CSS \(tutorialehtml.com\)](#)

[CSS-Tricks - Tips, Tricks, and Techniques on using Cascading Style Sheets.](#)

## Exerciții laborator:

### 1. Realizează o pagină HTML numită **programExcursie.html**

**Scop:** o pagină de prezentare a unui program turistic (de exemplu, o excursie montană).

#### Cerințe:

- a. Un titlu sugestiv – *Programul excursiei la munte*.
- b. O imagine reprezentativă pentru excursie (munți, natură, drumeție).
- c. Un tabel cu programul excursiei: fiecare rând reprezintă o zi.

Coloanele tabelului vor fi:

- **Ziua**
- **Activitate principală**
- **Locație**
- **Durată (ore)**
  - d. Cerințe de stilizare:
- Tabelul să aibă **borduri vizibile și culori alternative pe rânduri**.
- Antetul tabelului (<th>) să aibă fundal diferit.
- La trecerea cu mouse-ul peste un rând, fundalul să se schimbe.
- Titlul și imaginea să fie **centrate**.
- Imaginea să aibă **colțuri rotunjite și umbră discretă**.
- e. Un paragraf scurt la final care descrie scopul excursiei.

### Programul excursiei la munte



Ziua	Activitate principală	Locație	Durată
1	Drumeție spre cabană	Valea Cerbului	4 ore
2	Vizită la lacul glaciar	Lacul Bâlea	5 ore
3	Relaxare și picnic	Poiana Stânei	3 ore

Excursia are scopul de a oferi o experiență montană relaxantă și educativă în mijlocul naturii.

## 2. Realizează o pagină HTML numită **înregistrareAnimale.html**

**Scop:** un formular pentru înregistrarea animalelor la o clinică veterinară.

**Cerințe:**

- a. Un titlu sugestiv – *Formular de înregistrare animale*.
- b. Un scurt paragraf introductiv despre scopul formularului.
- c. Formularul trebuie să conțină:
  - **Nume stăpân** – input text
  - **Telefon** – input text
  - **Nume animal** – input text
  - **Specie** – listă drop-down (Câine, Pisică, Iepure, Altul)
  - **Sexul animalului** – radio (Mascul / Femelă)
  - **Servicii dorite** – checkbox (Vaccinare, Deparazitare, Consult)
  - **Observații** – textarea
- d. Două butoane:
  - Unul pentru trimitere (type="submit")
  - Unul pentru resetare (type="reset")
- e. Cerințe CSS:
  - Formularul să fie centrat pe pagină, cu **fundal alb și umbră ușoară**.
  - Câmpurile să aibă **margin și padding** consistente.
  - Butoanele să fie colorate diferit și să aibă efect de hover.
  - Câmpurile să își schimbe culoarea bordurii când sunt selectate (focus).

**Formular de înregistrare animale**

Completează formularul de mai jos pentru a înregistra animalul tău la clinica veterinară.

Nume stăpân:

Telefon:

Nume animal:

Specie:

**Sexul animalului**

Mascul  
 Femelă

**Servicii dorite**

Vaccinare  
 Deparazitare  
 Consult

Observații:

**Trimite** **Resetează**

**Instrucțiuni laborator:** Rezolvările exercițiilor se vor încărca, sub formă de arhivă (rar/zip) pe Moodle, până la finalul laboratorului. Arhiva se va numi GRUPĂ\_NUME\_PRENUME\_Lab2 (ex: 321AC\_Mihai\_Roxana\_Lab2.zip)