

LUCRAREA I

I.1. Obiectivele lucrării

Această lucrare își propune prezentarea mediului de lucru din programul MATLAB, efectuarea de calcule aritmetice, prezentarea și utilizarea funcțiilor elementare predefinite.

I.2. Considerente generale

Necesitatea utilizării de programe matematice a apărut datorită volumului tot mai mare de date obținut în urma diverselor experimente. Aceste programe pot reduce semnificativ timpul de rezolvare a problemelor datorită ușurinței de implementare a algoritmilor utilizați prin simplificarea sintaxei și a modului prin care utilizăm variabilele.

MATLAB (MATrix LABoratory) este un program destinat calculului numeric precum și a reprezentărilor grafice în domeniul ingineresc. Acesta are la bază calculul matriceal și cel vectorial. Alocarea memoriei este dinamică, fiecărui vector sau matrice i se va aloca un număr de locații de memorie corespunzător dimensiunii efective și a tipului de dată utilizată.

Sistemul MATLAB include cinci părți principale:

1. **Interfața**, prezentată în Figura 1 ce cuprinde o zonă de control a fișierelor „*Current folder*” notată cu 1; o zonă de scriere a comenzilor „*Command Window*” notată cu 2; o fereastră ce permite vizualizarea variabilelor definite „*Workspace*” notată cu 3; o fereastră ce conține istoricul comenzilor executate „*Command History*” notată cu 4; un Meniu ce conține diverse unelte ce facilitează utilizarea funcțiilor predefinite sau a celor realizate de către utilizator, diverse opțiuni pentru grafice precum și unelte utilizate pentru a obține alte aplicații realizate cu MATLAB, notat cu 5.
2. **Biblioteca cu funcții matematice**: conține algoritmi pentru calculul funcțiilor matematice elementare, funcții destinate calculului matriceal, funcții Bessel, transformata Fourier rapidă.
3. **Limbajul**: toate variabilele sunt considerate vectori sau matrice; include comenzi de programare, controlul buclelor, funcții, structuri de date, comenzi intrare/ieșire, inclusiv posibilitatea programării orientate pe obiecte.

4. **Grafice:** se pot reprezenta vectori și matrice în grafice bi sau tridimensionale, procesare de imagini, posibilitatea creării unor interfețe grafice pentru aplicațiile MATLAB.
5. **Interfețe externe:** permite dezvoltarea unor programe în C și FORTRAN care să ruleze în MATLAB.

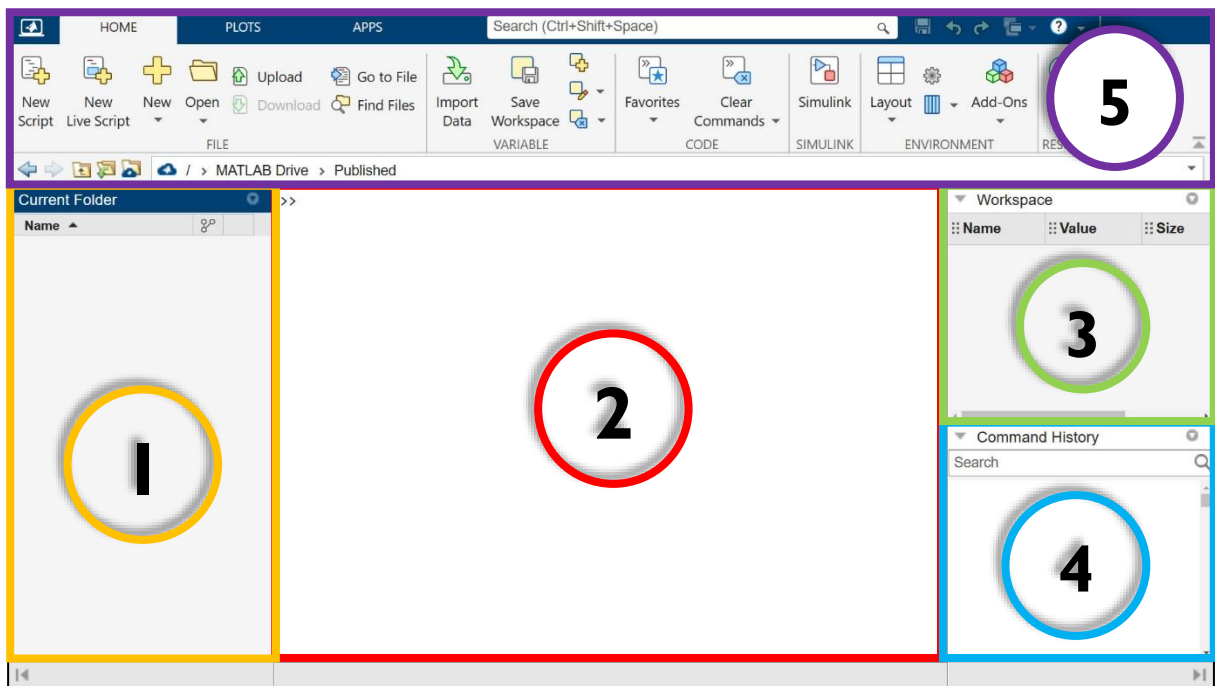


Figura 1. Interfața programului MATLAB. 1) zonă de control a fișierelor, 2) zonă de scriere a comenzilor, 3) fereastră pentru vizualizarea variabilelor definite, 4) fereastră ce conține istoricul comenzilor executate și 5) meniu

I.3. Variabile în MATLAB

În cadrul programului putem realiza calcule matematice complexe, utilizând funcții matematice predefinite în MATLAB, cu ajutorul variabilelor. Declararea acestora se realizează prin operația de atribuire „=”, potrivit căreia programul îi va atribui o anumită expresie:

$$\text{NumeVariabila} = \text{expr}$$

Conform declarației de mai sus, variabilei “NumeVariabila” îi va fi atribuită valoarea expresiei “expr” care poate fi de forma: $a+b$ sau a/b .

Numele variabilelor trebuie să înceapă cu **o literă** și pot conține mai multe caractere, însă primele **31 de caractere** sunt semnificative. În numele variabilelor pot apărea litere, cifre și caracterul `_` dar nu sunt admise alte semne de punctuație sau spații libere. Matlab poate face diferență între variabila `a` și `A`, pentru program sunt variabile diferite astfel $a \neq A$. Utilizarea diacriticelor nu este permisă deoarece nu sunt caractere ASCII recunoscute de către program însă pot fi utilizate în cadrul unei variabile de tip caracter.

```
>> arie_pătrat=20
```

```
arie_pătrat=20
```

↑

Error: Invalid text character. Check for unsupported symbol, invisible character, or pasting of non-ASCII characters.

```
>> mesaj="împărțirea la zero este imposibilă"
```

```
mesaj =
```

```
"împărțirea la zero este imposibilă"
```

Comentariile se pot introduce în program cu ajutorul caracterului `„ % ”` și sunt ignorate de calculator, dar de obicei sunt utile pentru cel care lucrează cu aplicația respectivă pentru că pot oferi detalii despre blocul respectiv de instrucțiuni sau despre program în general.

```
% Acesta este un comentariu
```

```
% Comentariile se introduc pentru a adăuga o descriere blocului respectiv de cod
```

```
disp ('Hello, world')
```

Introducerea unei instrucțiuni, fără a fi urmată de caracterul `„ ; ”` în fereastra de comandă determină un răspuns din partea calculatorului:

```
>> x=10/2
```

```
x =
```

```
5
```

```
>> x=10/2; % dacă folosim „ ; ” Matlab va face împărțirea “10:2” însă nu va afișa rezultatul  
% în consolă
```

Există mai multe comenzi ce pot fi utilizate la managementul variabilelor:

- `clear x y` – șterge variabilele `x` și `y` din memorie;
- `clear` – șterge toate variabilele din memorie;
- `who` – afișează toate variabilele definite la un moment dat;
- `whos` – afișează o listă detaliată cu variabilele definite;
- `save` – salvează spațiul de lucru curent în fișierul cu numele `matlab.mat`;
- `save nume_fișier` - salvează spațiul de lucru curent în fișierul cu numele specificat de utilizator, fișier cu extensia `.mat`;
- `load` – încarcă în spațiul de lucru variabilele salvate în fișierul `matlab.mat`,
- `load nume_fișier` - încarcă în spațiul de lucru variabilele salvate în fișierul specificat.

În cadrul programului MATLAB se regăsesc mai multe variabile predefinite, cum ar fi:

- `ans` – numele variabilei răspuns (*eng*: *answer*). La efectuarea unei operații fără alocarea unui nume, se afișează:

```
>> 10/2
ans =
     5
```

- `pi` = 3.1416
- `inf` sau `Inf` – infinit, adică nedeterminare (poate apare la împărțirea cu 0);
- `nan` sau `NaN` – not-a-number, nedeterminare (poate apare la 0/0);
- `i` și `j` – sunt numerele complexe;
- `date` – data curentă sub forma unei variabile tip șir de caractere. Ex.: 10-July-2023.

Formatul implicit de afișare este cel cu 4 zecimale desemnat ca fiind formatul **short**.

De exemplu numărul 758/15 va fi afișat sub forma 50.5333. Mai există și alte tipuri de formate:

- `long` – cu 16 cifre semnificative pentru partea întreagă și partea fracționară, de exemplu 50.53333333333333;
- `short e` – cu 5 cifre semnificative plus exponentul, de exemplu 5.0533e+001;
- `long e` – 16 cifre semnificative plus exponentul, ex. 5.053333333333333e+001;
- `bank` – cu 2 cifre semnificative pentru partea fracționară, de exemplu 50.53.

Formatul de afișare nu modifică reprezentarea internă a numărului. Pentru a modifica formatul de afișare se tastează în fereastra de comandă formatul dorit. De exemplu:

```
>> format long e
```

O altă posibilitate de afișare este folosind comanda **disp** (display). Ea se poate aplica în două variante:

- disp (variabila) – care afișează valoarea variabilei fără numele acesteia;
- disp ('text') – afișează textul scris între caracterele apostrof.

Cu ajutorul lui **disp** se poate controla modul de afișare al rezultatelor, adică se poate asocia un text la valorile numerice prezentate. Pentru a face acest lucru, rezultatul trebuie convertit din valoare numerică în șir de caractere. Comanda utilizată pentru conversie este:

- num2str (nume_variabila) – convertește variabila din valoare numerică în variabilă de tip caracter, iar concatenarea șirurilor de caractere se realizează cu ajutorul [].

De exemplu:

```
>> disp (['Suma celor două numere este: ', num2str(suma)])  
Suma celor două numere este: 256
```

I.4. Operatori și funcții matematice de bază

Programul utilizează trei tipuri de operatori:

aritmetici relaționali logici

Operatorii aritmetici utilizați la definirea expresiilor sunt:

adunare	$a + b$
scădere	$a - b$
înmulțire	$a \cdot b = a * b$
împărțire	$\frac{a}{b} = a / b$
împărțire la stânga	$\frac{b}{a} = a \backslash b$
ridicare la putere	$a^b = a ^ b$
transpunere	a'

Operatori relaționali utilizați la compararea variabilelor sunt:

mai mic	$<$
mai mare	$>$
mai mic sau egal	$<=$

mai mare sau egal	>=
egal	==
diferit	~=

Operatori logici utilizați pentru obținerea expresiilor logice sunt:

AND	&
OR	
NOT	~

Ordinea de precedență a operatorilor este următoarea:

1.	paranteze	()
2.	transpunere și ridicare la putere	' ^
3.	NOT	~
4.	înmulțire și împărțire	* / \
5.	adunare și scădere	+ -
6.	operatorul :	:
7.	operatorii relaționali	< <= > >= == ~=
8.	AND	&
9.	OR	

Principalele funcții matematice pe care le regăsim în Matlab sunt:

abs(x)	Determină modulul unui număr real sau complex x
sin(x)	Calculează sinus de x
asin(x)	Calculează arc sinus de x
cos(x)	Calculează cosinus de x
acos(x)	Calculează arc cosinus de x
tan(x)	Calculează tangentă de x
atan(x)	Calculează arc tangentă de x
cot(x)	Calculează cotangentă de x
acot(x)	Calculează arc cotangentă de x
exp(x)	Funcția exponențială, e^x
log(x)	Logaritm natural din x, $\ln x$

log2(x)	Logaritm în baza 2 din x, $\log_2 x$
log10(x)	Logaritm în baza 10 din x, $\log_{10} x$
pow2(x)	Calculează 2^x
sqrt(x)	Calculează $\sqrt{x}$
dms2deg(d,m,s)	Convertește în grade un unghi dat în grade(degree), minute(m), secunde(second)
dms2rad(d,m,s)	Convertește în radiani un unghi dat în grade(degree), minute(m), secunde(second)
[th,rho]=cart2pol(x,y)	Transformă coordonatele carteziene (x,y) în coordonate polare (unghiul th este returnat în radiani)
[x,y]=pol2cart(th,rho)	Transformă coordonatele polare (th,rho) în coordonate carteziene (unghiul th este dat în radiani)
[th,phi,r]=cart2sph(x,y,z)	Transformă coordonatele carteziene (x,y,z) în coordonate sferice (unghiurile th și phi sunt returnate în radiani)
[x,y,z]=sph2cart(th,phi,r)	Transformă coordonatele sferice (th,phi,r) în coordonate carteziene (unghiurile th și phi sunt date în radiani)

Pentru a obține o listă cu toate funcțiile matematice elementare predefinite în Matlab trebuie să tastați în zona de scriere a comenzilor „**help elfun**”. Pentru a obține o listă a funcțiilor mai avansate de matematică și de manipulare a matricelor tastați „**help specfun**”, „**help elmat**”.

Numele funcțiilor nu sunt rezervate, deci putem suprascrie peste oricare dintre ele o nouă variabilă, de exemplu **eps = 1.e-6** și apoi să utilizăm acea valoare în calcule ulterioare. Funcția originală poate fi restaurată cu instrucțiunea **clear eps**.

I.5. Aplicații

PI. Se consideră variabilele: $a = 4$, $b = 5$, $c = 6$.

Să se calculeze:

$$x = \frac{a+b}{c}$$

$$y = \frac{a+b}{c-3}$$

$$z = \frac{a+b}{c} - 1$$

$$k = \frac{1}{a} + \frac{1}{b} + \frac{1}{c+2}$$

$$g = a^{\frac{b-1}{b}}$$

$$w = e^{b+c}$$

$$u = 2^a + \frac{a+b}{c^2} \quad e = k^a + (k+1)^b + (k-1)^c \quad h = \frac{a}{1+\frac{b}{1+\frac{1}{c}}}$$

P2. Se consideră variabilele: $x = 4.3$, $y = 5.213$, $z = 6.015$

Să se calculeze:

$$a = \sqrt{x + \frac{y}{z}} \quad b = \sqrt[3]{x + y - z}$$

$$c = \pi^{x+1} - \frac{y+z}{\pi} \quad d = \sqrt{a^2 + b^2}$$

P3. Se consideră variabilele: $a = e$, $b = 2.21$, $c = 5.9$

Să se calculeze:

$$x = \log_2 a \quad y = \log_{10} a$$

$$z = \ln a \quad g = \frac{\log_2 b}{\log_{10} c} \cdot \ln\left(\frac{a}{b+c}\right)$$

$$h = \frac{1+\ln b}{c^3+\log_2 e} \quad i = \frac{\ln a \cdot \ln b \cdot \ln c}{1+\log_2(a \cdot b \cdot c)}$$

P4. Se consideră variabilele: $a = 2.7$, $b = e$

Să se calculeze:

$$c = \log_2 a + \log_{10} a - \frac{\log_2(a+b)}{\ln b} \quad d = e^{a+1} - \frac{e^b}{\ln b}$$

$$e = e^{a+b} + b^{\frac{b+1}{b}} \quad f = \frac{1}{a} \ln(b+1) - \frac{1}{b} \ln(a+1)$$

P5. Se consideră variabila: $\alpha = 35^\circ$

Să se verifice următoarele relații:

$$\sin 3\alpha = 3 \sin \alpha - 4 \sin^3 \alpha$$

$$\cos 3\alpha = -3 \cos \alpha + 4 \cos^3 \alpha$$

$$\sin 2\alpha = 2 \sin \alpha \cos \alpha = \frac{2 \tan \alpha}{1 + \tan^2 \alpha}$$

$$\cos 2\alpha = 2 \cos^2 \alpha - 1 = 1 - 2 \sin^2 \alpha = \cos^2 \alpha - \sin^2 \alpha = \frac{1 - \tan^2 \alpha}{1 + \tan^2 \alpha}$$

$$\sin 4\alpha = 8 \cos^3 \alpha \cdot \sin \alpha - 4 \cos \alpha \sin \alpha$$

$$\cos 4\alpha = 8 \cos^4 \alpha - 8 \cos^2 \alpha + 1$$

P5. Se consideră variabilele: $a = 2$, $b = 5$

Să se calculeze:

$$c = \frac{a^3 \cdot b}{a-b}$$

$$d = \frac{2 \cdot a^3}{a^3-1} + \frac{2 \cdot b^3}{b^3-1} + \frac{a \cdot b}{a \cdot b-1}$$

$$e = \frac{3}{2} \cdot a \cdot b + \frac{1}{5} \cdot \frac{a}{b} - \frac{8}{3} \frac{b}{a}$$

$$f = 3 \cdot a^{\frac{1}{3}} + 5 \cdot b^{\frac{1}{5}} + \frac{1}{4} \cdot a^{0.25} - b^{1.4}$$

$$g = \left(\frac{1}{\frac{1}{a}+1} \right)^2 - \left(\frac{1}{\frac{1}{b}+1} \right)^{\frac{1}{2}}$$

$$h = \left(\frac{3}{4} \cdot a - \frac{2}{5} \cdot b \right)^{-3/4} \cdot \left(\frac{a}{b+1} \right)^{2/5} + \frac{1}{a \cdot b}$$

P5. Se consideră variabilele: $\alpha = 30^\circ$, $\beta = \pi/2$

Să se calculeze:

$$a = \cos \frac{\beta}{5} \left(1 - \sin \frac{\alpha}{2} \right)$$

$$b = \frac{\pi}{3} \cdot \sin(\alpha + \beta)$$

$$c = \frac{1 - \sin \alpha}{1 + \cos \alpha}$$

$$d = \left(\frac{1}{2} + \tan \alpha \right) \sin \left(2 \cdot \pi \cdot \frac{\alpha}{\beta} \right)$$

$$e = \frac{\sin(2 \cdot \pi \cdot \alpha + \beta)}{\cos(\pi \cdot \alpha - \beta)}$$

$$f = \sin(\alpha + \pi) \cdot \cos \left(\beta / 3 \right)$$

P6. Se consideră variabilele: $a = 3$, $b = 4$, $c = 5$

Să se calculeze:

$$d_1 = \sqrt{1 + 2\pi} + \frac{a + \pi}{2 \cdot c^2 + 1} - e^{\frac{a+1}{a}} + \sqrt[3]{2 \cdot a + b + \frac{1}{c}} + \ln \frac{2 + a}{b + c}$$

$$d_2 = \sqrt{b^{a+1}} + \frac{a + 1}{1 + b \cdot c^a} + e^{\frac{a+1}{b}} + \sqrt[3]{2 \cdot \pi + \frac{b + 1}{c}} - 1 + \ln \frac{a + b}{b + c}$$

$$\begin{aligned}
d_3 &= \sqrt{\pi^{a+1}} + \frac{b+a^c}{b+c^a} + e^{\frac{a+1}{b}} + \sqrt[3]{2 \cdot a + \frac{b}{c+1} + 1 + \ln \frac{a+2 \cdot b}{2 \cdot b+c}} \\
d_4 &= \sqrt{\pi^{a+b}} + \frac{1+a^c}{\pi-c^a} + e^{\frac{\pi+1}{b}} + \sqrt[3]{2 \cdot \pi + \frac{b}{c-1} + 1 + \ln \frac{2 \cdot a+b}{b+2 \cdot c}} \\
d_5 &= \sqrt{\pi^{1+b}} + \frac{1+a^c}{a(\pi-c^a)} + e^{\frac{\pi+1}{b-1}} + \sqrt[3]{\pi^2 + \frac{a+b}{c-1} + 1 + \ln \frac{a(a+b)}{b+c}} \\
d_6 &= \sqrt{\pi^{1+\pi}} + \frac{1+a^\pi}{a(b+c^a)} + e^{\frac{1}{\pi-1}} + \sqrt[3]{\pi^3 + \frac{a+1}{b+1} + c + \ln \frac{a(a+b)}{b(b+c)}} \\
d_7 &= \sqrt{a^{\pi-1}} + \frac{a+b^\pi}{\pi(b-c^\pi)} + e^{1+\frac{1}{\pi}} + \sqrt[3]{\pi^{1/a} + \frac{a+1/b}{b+1/c} + 1 + \ln \frac{2(a^c+b)}{3(b+c^a)}} \\
d_8 &= \sqrt{(a+1)^\pi} + \frac{c+a^\pi}{b+a^\pi} + e^{\frac{2 \cdot a+1}{b}} + \sqrt[3]{\frac{1}{\pi} + \frac{2 \cdot a+1}{2 \cdot b+1} + 1 + \ln \frac{3(a^\pi+1)}{7(c^\pi+1)}} \\
d_9 &= \sqrt{(1+a^2)^\pi} + \frac{1+a+b}{c(a+b^\pi)} + e^{\frac{\pi+1}{\pi^2}} + \sqrt[3]{\frac{\pi}{a} + \frac{3 \cdot \pi+1}{\pi(a+1)} + 1 + \ln \frac{c+\pi}{\pi^2}} \\
d_{10} &= \sqrt{1+\frac{a}{b+a}} + \frac{a}{\pi(1+\pi^{0.5})} + e^{\frac{a+\pi}{a-\pi}} + \sqrt[3]{\frac{1}{\pi} + \frac{\pi+a}{b(c-1)} + \log_{10} \frac{2 \cdot a+3 \cdot b}{c-1}}
\end{aligned}$$

LUCRAREA 2

2.1. Obiectivul lucrării

În cadrul acestei lucrări este prezentat modul de interacțiune cu matrice, și anume: cum se pot crea, modalitatea de accesare a elementelor sau a grupurilor de elemente, calcule elementare cu matrice precum și funcții utilizate la generarea de matrice.

2.2. Considerente generale

Matricele sunt elementele de bază ale mediului MATLAB. O matrice este un tablou bidimensional ce conține **m** rânduri și **n** coloane. Există și cazuri speciale în care putem întâlni **vectori linie** (**m**, numărul de linii, este egal cu 1) și **vectorii coloană** (**n**, numărul de coloane, este egal cu 1).

Pentru a defini un vector în MATLAB, trebuie să încadrăm între paranteze pătrate elementele din vector, elemente ce vor fi separate între ele prin virgulă sau spațiu. De exemplu, pentru a crea vectorul **v**, trebuie să scriem:

```
>> v = [1 4 7 10 13]
v =
     1     4     7    10    13
```

Vectorii coloană sunt creați în mod similar cu vectorii linie, diferența o constituie utilizarea (;) pentru a separa elementele din cadrul vectorului:

```
>> w = [1; 4; 7; 10; 13]
w =
     1
     4
     7
    10
    13
```

Putem face trecerea de la un vector linie la unul coloană prin utilizarea operatorului de transpunere care este simbolizat printr-un apostrof (').

```
>> w = v'
```

```
w =
```

```
1  
4  
7  
10  
13
```

Astfel, $v(1)$ este primul element din vectorul v , $v(2)$ este cel de-al doilea element și așa mai departe.

Pentru a putea accesa un grup de elemente din cadrul unui vector sau matrice, în MATLAB se folosește notația (:). De exemplu, pentru a accesa primele 3 elemente din vectorul v , va trebui să scriem:

```
>> v (1 : 3)
```

```
ans =
```

```
1 4 7
```

Pentru a vizualiza toate elementele, de la cel de-al treilea până la ultimul element:

```
>> v (3 : end)
```

```
ans =
```

```
7 10 13
```

în acest caz *end* semnifică ultimul element din vector.

1.3. Definirea matricelor

Pentru a defini o matrice în MATLAB trebuie respectate următoarele reguli:

- începem cu o paranteză pătrată, [
- separarea elementelor într-un rând se face utilizând spațiu sau virgula (,)
- folosim (;) pentru a separa rândurile între ele
- închidem matricea cu o altă paranteză pătrată,].

Mai jos puteți observa o matrice A:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

care a fost obținută astfel,

```
>> A = [1 2 3 ; 4 5 6 ; 7 8 9]
```

```
A =
```

```
1 2 3
4 5 6
7 8 9
```

Putem selecta mai multe elemente din cadrul unei matrice în mod similar cu vectorii însă de această dată avem nevoie de doi indici. Elementul de pe rândul i și coloana j dintr-o matrice A este notat cu A (i , j). Primul indice este atribuit întotdeauna liniei pe care se află elementul iar cel de-al doilea indice determină coloana. De exemplu, A (2, 3) este un element din matricea A care se află linia 2 și coloana 3. Elementul care se află la această adresă este cifra 6.

Cu ajutorul acestei indexări putem modifica valoarea unui element din cadrul matricei. Dacă luăm, de exemplu, elementul din A(3,3), care este 9, putem să-i modificăm valoarea scriind:

```
>> A (3, 3) = 0
```

```
A =
```

```
1 2 3
4 5 6
7 8 0
```

Operatorul (:) este foarte utilizat în MATLAB deoarece de cele mai multe ori avem de introdus foarte multe elemente în matrice sau în vectori. De exemplu, pentru a introduce în MATLAB vectorul x ce conține următoarele puncte: 0, 0.1, 0.2, 0.3,..., 5 este suficient să scriem:

```
>> x = 0 : 0.1 : 5;
```

Cu această scriere vom obține un vector x ce va avea 51 de elemente, ce va începe din 0, va avea ca increment 0.1 (asta înseamnă că pentru a obține, de exemplu, al doilea număr din acest șir va trebui să adăugăm incrementul (0.1) la valoarea anterioară (0) și vom obține 0.1) iar capătul intervalului va fi 5.

În MATLAB avem la dispoziție o comandă în urma căreia se va genera un vector liniar distanțat: *linspace*. Este asemănător cu operatorul ($:$) însă acesta ne conferă control asupra numărului de puncte pe care îl obținem. De exemplu:

$$y = \text{linspace}(a, b)$$

va genera un vector linie y cu 100 de puncte liniar distanțate ce va conține și capetele a și b .

$$y = \text{linspace}(a, b, n)$$

va genera un vector y cu n puncte liniar distanțate ce va conține și capetele a și b . Acest mod de scriere este util pentru cazul în care vrem să împărțim un interval într-un număr de subintervale de aceeași lungime. De exemplu:

$$\alpha = \text{linspace}(0, 2\pi, 101)$$

va împărți intervalul $[0, 2\pi]$ în 100 de segmente egale, după care va crea un vector cu 101 elemente.

Operatorul ($:$) în cadrul matricelor este utilizat pentru a putea selecta fie o anumită linie fie o anumită coloană. $A(m : n, k : l)$ specifică rândurile de la m la n și coloanele de la k la l .

De exemplu:

```
>> A(2, :)
```

```
ans =
```

```
4 5 6
```

Ne va arăta toate elementele de pe linia 2 din matricea A .

Operatorul (:) poate fi utilizat și la crearea de sub-matrici:

```
>> B = A (:, 2 : 3)
```

B =

```
2 3
5 6
8 0
```

Astfel, matricea B va conține toate elementele din coloanele 2 și 3.

Pentru a șterge un rând sau o coloană dintr-o matrice vom folosi operatorul [].

```
>> A (3, :) = []
```

A =

```
1 2 3
4 5 6
```

Utilizând această scriere am șters cel de-al treilea rând din matricea A. Pentru a reface cel de-al treilea rând din matrice putem utiliza următoarea instrucțiune:

```
>> A = [ A(1,:); A(2,:); [7 8 0]]
```

A =

```
1 2 3
4 5 6
7 8 0
```

Pentru a determina dimensiunea unei matrice sau al unui vector putem folosi comanda *size*.

```
>> size (A)
```

ans =

```
3 3
```

Asta înseamnă că matricea are 3 rânduri și 3 coloane.

Operatorul de transpunere este notat cu apostrof ('). Acesta întoarce o matrice în jurul diagonalei principale și transformă un vector rând într-un vector coloană.

```
>> A'

ans =

     1     4     7
     2     5     8
     3     6     0
```

Utilizând notația din algebra liniară, transpusa matricei reale A , $m \times n$, este matricea $n \times m$. Matricea transpusă este notată cu A^T .

Concatenarea matricelor are ca scop crearea unei matrice mai mari utilizând ca elemente alte matrice. De exemplu, vom crea matricea B utilizând matricea A.

```
>> B = [A 10*A; -A [1 0 0; 0 1 0; 0 0 1]]

B =

     1     2     3    10    20    30
     4     5     6    40    50    60
     7     8     0    70    80     0
    -1    -2    -3     1     0     0
    -4    -5    -6     0     1     0
    -7    -8     0     0     0     1
```

MATLAB conține o serie de funcții de generare a unor matrice elementare, cum ar fi:

eye(m,n)	Returnează o matrice cu m linii și n coloane cu 1 pe diagonala principală
eye(n)	Returnează matrice pătratică de n lini și coloane cu 1 pe diagonala principală
zeros(m,n)	Returnează o matrice cu m linii și n coloane cu toate elementele zero
ones(m,n)	Returnează o matrice cu m linii și n coloane cu toate elementele unu
diag(A)	Extrage diagonala matricei A

rand(m,n)	Returnează o matrice cu m linii și n coloane cu numere aleatorii, pozitive și subunitare
randn(m,n)	Returnează o matrice cu m linii și n coloane cu orice număr aleatoriu din mulțimea numerelor reale

Pentru a obține o listă completă a matricelor elementare precum și a modurilor de manipulare a matricelor puteți tasta *help elmat* sau *doc elmat*.

<pre>>> b = ones (3,1) b = 1 1 1</pre>	<pre>>> eye (3) ans = 1 0 0 0 1 0 0 0 1</pre>	<pre>>> c = zeros (2,3) c = 0 0 0 0 0 0</pre>
---	--	---

În cadrul MATLAB regăsim și o serie de matrice speciale. Aceste matrice, prezentate în tabelul următor, au proprietăți interesante care le fac utile la construcția de exemple și la testarea algoritmilor.

hilb	Matricea Hilbert
invhilb	Matricea Hilbert inversă
magic	Matricea pătratică magică
pascal	Matricea Pascal
toeplitz	Matricea Toeplitz
vander	Matricea Vandermonde
wilkinson	Matricea testului cu valori proprii a lui Wilkinson

Principalele funcții definite pentru lucrul cu matrice sunt prezentate în tabelul următor:

fliplr(A)	întoarce matricea A de la stânga la dreapta, practic inversează coloanele
flipud(A)	întoarce matricea A de sus în jos, practic inversează liniile

rotește matricea A la 90°, în sens
rot 90(A) trigonometric (linia 1 devine coloana 1;
linia 2 devine coloana 2; etc)

rot 90(A, k) rotește matricea A cu un unghi de k *
90°

2.3 Matrice speciale în MATLAB

a) Matricea diagonală

Funcția „**diag**” creează matricea diagonală sau operează cu diagonalele unei matrice. Dacă aplicăm funcția unei matrice A, atunci vom obține un vector coloană ale cărui elemente sunt chiar elementele aflate pe diagonală principală a matricei A.

Funcția „**diag (A, k)**” generează un vector coloană, cu elementele diagonalei k. Dacă $k > 0$ înseamnă că diagonală k va fi situată deasupra diagonalei principale, $k < 0$ înseamnă că diagonală k va fi sub cea principală.

Exemplu:

>> A = [1 2 3 4; 5 6 7 8; 9 10 11 12; 13 14 15 16]		
>> diag (A) ans = 1 6 11 16	>> diag (A, 2) % a doua diagonală de deasupra celei principale ans = 3 8	>> diag (A, -1) % prima diagonală de sub cea principală ans = 5 10 15

În cazul unui vector **x**, funcția „**diag(x)**” va genera o matrice pătrată cu elementele vectorului pe diagonală principală, iar funcția „**diag(x, k)**” va genera o matrice pătrată cu diagonală k conținând elementele vectorului x.

>> x = [50 100 150 200] % vector linie x = 50 100 150 200	>> diag (x) % generează o matrice pătrată de ordinul 4 ans =	>> diag (x, -2) % generează o matrice pătrată de ordinul 6 ans =
---	---	---

	50 0 0 0 0 100 0 0 0 0 150 0 0 0 0 200	0 0 0 0 0 0 0 0 0 0 0 0 50 0 0 0 0 0 0 100 0 0 0 0 0 0 150 0 0 0 0 0 0 200 0 0
<pre>>> diag (ones (1, 3), 1) ans = 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0</pre>	<pre>>> diag (ones (1, 3), 1) + diag (ones (1, 3), -1) ans = 0 1 0 0 1 0 1 0 0 1 0 1 0 0 1 0</pre>	<pre>>>diag (ones (1, 3), 1) * 10 + diag (ones (1, 3), -1) * (-1) ans = 0 10 0 0 -1 0 10 0 0 -1 0 10 0 0 -1 0</pre>

Cu ajutorul funcției `diag(v,n)`, unde v este un vector iar n este distanța de la diagonală principală se pot obține matricele prezentate în exemplele de mai sus. În cadrul acestor exemple `ones(1,3)` va genera un vector linie cu trei elemente de 1 iar n în acest exemplu este 1 ceea ce înseamnă că vectorul definit anterior va fi poziționat pe prima diagonală **de deasupra** diagonalei principale. În al doilea exemplu, când n are valoarea -1, vectorul va fi poziționat pe prima diagonală **de sub** diagonală principală. Dacă se vrea ca vectorul să aibă 3 elemente de 10 atunci "`diag (ones (1, 3), 1)`" va fi înmulțit cu 10.

b) Matricea pătratul magic

Este o matrice pătrată de ordinul n , construită din numere întregi de la 1 la n^2 în care sumele elementelor pe linii, coloane și diagonale sunt egale între ele. Trebuie ca n să fie mai mare decât 2.

<pre>>> A = magic (6) A = 35 1 6 26 19 24 3 32 7 21 23 25 31 9 2 22 27 20 8 28 33 17 10 15 30 5 34 12 14 16 4 36 29 13 18 11</pre>

<pre>>> sum (A, 1) % suma valorilor pe fiecare coloană</pre> <pre>ans =</pre> <pre>111 111 111 111 111 111</pre>	<pre>>> sum (A, 2) % suma valorilor pe fiecare linie</pre> <pre>ans =</pre> <pre>111 111 111 111 111 111</pre>	<pre>sum (diag (A)) % suma valorilor pe diagonală</pre> <pre>ans =</pre> <pre>111</pre>
---	--	---

c) Matricea Pascal

Este o matrice pătrată simetrică, pozitiv definită, cu elemente numere întregi obținută din *triunghiul lui Pascal*. Triunghiul lui Pascal conține coeficienții binomiali ai descompunerii $(a+b)^n$:

$$(a + b)^n = a^n + C_n^1 \cdot a^{n-1} \cdot b + C_n^2 \cdot a^{n-2} \cdot b^2 + \dots + C_n^{n-1} \cdot a \cdot b^{n-1} + b^n$$

Pentru a obține un binom de ordinul 3 avem dezvoltarea:

$$(a + b)^3 = a^3 + 3 \cdot a^2 \cdot b + 3 \cdot a \cdot b^2 + b^3$$

Triunghiul lui Pascal ce conține coeficienții dezvoltării binomului de ordinul 3 are forma:

```

0           1
1          1  1
2         1  2  1
3        1  3  3  1
```

Numerotarea liniilor se face începând cu 0, deci fiecare linie din triunghi conține descompunerea lui $(a+b)^n$, $n=0, 1, 2, 3$. Sintaxa funcției Pascal este „*pascal(n)*”.

<pre>>> P = pascal (4)</pre> <pre>P =</pre> <pre> 1 1 1 1 1 2 3 4 1 3 6 10 1 4 10 20</pre>

Se observă că elementele din partea stângă, inclusiv diagonală secundară sunt elementele *triunghiului lui Pascal*.

d) Matricea Toeplitz

O matrice Toeplitz conține ca elemente ale primei linii și primei coloane un vector definit anterior. Matricea Toeplitz simetrică este definită doar de un vector. Dacă matricea Toeplitz este definită de doi vectori, de dimensiuni m și n , se va genera o matrice de dimensiune $(m * n)$ a cărei primă coloană este definită de elementele primului vector și prima linie este definită de elementele celui de-al doilea vector.

<pre>>> a = [2 4 6 8 10]; >> t1 = toeplitz (a)</pre> t1 = <table><tr><td>2</td><td>4</td><td>6</td><td>8</td><td>10</td></tr><tr><td>4</td><td>2</td><td>4</td><td>6</td><td>8</td></tr><tr><td>6</td><td>4</td><td>2</td><td>4</td><td>6</td></tr><tr><td>8</td><td>6</td><td>4</td><td>2</td><td>4</td></tr><tr><td>10</td><td>8</td><td>6</td><td>4</td><td>2</td></tr></table>	2	4	6	8	10	4	2	4	6	8	6	4	2	4	6	8	6	4	2	4	10	8	6	4	2	<pre>>> b = [1 3 5]; c = [1 9 11 13]; >> t2 = toeplitz (b, c)</pre> t2 = <table><tr><td>1</td><td>9</td><td>11</td><td>13</td></tr><tr><td>3</td><td>1</td><td>9</td><td>11</td></tr><tr><td>5</td><td>3</td><td>1</td><td>9</td></tr></table>	1	9	11	13	3	1	9	11	5	3	1	9
2	4	6	8	10																																		
4	2	4	6	8																																		
6	4	2	4	6																																		
8	6	4	2	4																																		
10	8	6	4	2																																		
1	9	11	13																																			
3	1	9	11																																			
5	3	1	9																																			

e) Matricea Vandermonde

Este o matrice pătrată ale cărei coloane sunt puterile elementelor vectorului v . Se poate scrie: „ $A=vander(v)$ ”, în care $A(i,j) = v(i)^{(n-j)}$.

Exemplu: să se genereze o matrice Vandermonde pe baza vectorului $v = [1 \ 3 \ 6 \ 9]$.

```
>> v = [1 3 6 9];
>> V = vander (v)
```

V =

1	1	1	1
27	9	3	1
216	36	6	1
729	81	9	1

f) Matricea Hankel

O matrice Hankel se generează pe baza unui vector și reprezintă o matrice anti-simetrică și anti-diagonală. În acest caz matricea este pătrată și prima linie și prima coloană sunt luate elementele vectorului respectiv. Elementele de sub diagonala secundară sunt nule.

Dacă se folosesc doi vectori la generarea matricii Hankel, se obține o matrice cu dimensiunile celor doi vectori, cu prima coloană dată de elementele primului vector și ultima linie cu elementele celui de-al doilea vector.

<pre>>> a = [1 2 3 4]; b = [5 6 7 9]; c = [9 10 11]; >> H1 = hankel (a)</pre> H1 = <table><tr><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>2</td><td>3</td><td>4</td><td>0</td></tr><tr><td>3</td><td>4</td><td>0</td><td>0</td></tr><tr><td>4</td><td>0</td><td>0</td><td>0</td></tr></table>	1	2	3	4	2	3	4	0	3	4	0	0	4	0	0	0	<pre>>>H2 = hankel (b, c)</pre> H2 = <table><tr><td>5</td><td>6</td><td>7</td></tr><tr><td>6</td><td>7</td><td>8</td></tr><tr><td>7</td><td>8</td><td>10</td></tr><tr><td>8</td><td>9</td><td>11</td></tr></table>	5	6	7	6	7	8	7	8	10	8	9	11
1	2	3	4																										
2	3	4	0																										
3	4	0	0																										
4	0	0	0																										
5	6	7																											
6	7	8																											
7	8	10																											
8	9	11																											

g) Matricea Hilbert

Este o matrice pătrată de ordinul n, ale cărei elemente se obțin pe baza relației:

$$H(i,j) = \frac{1}{i+j-1}$$

Matricea Hilbert de ordinul 4 se obține cu sintaxa următoare (dacă se folosește formatarea **format rat**):

```
>> H4=hilb(4)
```

H4 =

1	1/2	1/3	1/4
1/2	1/3	1/4	1/5
1/3	1/4	1/5	1/6
1/4	1/5	1/6	1/7

2.4 Aplicații

PI. Definiți în MATLAB următorii vectori:

$$a = (1, 2, 3);$$

$$b = (4, 5, 6);$$

$$c = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix};$$

$$d = \begin{pmatrix} 2 \\ 0 \\ -1 \end{pmatrix};$$

$$e = (1, 2, 3, 4, 5, 6);$$

$$f = \begin{pmatrix} 2 \\ 0 \\ -1 \\ -1 \\ 0 \\ 1 \end{pmatrix}.$$

P2. Creați vectorii a , b , c și d care să aibă următoarele elemente:

- a) $1, 2, 3, \dots, 100$;
- b) $50, 49, 48, \dots, -50$;
- c) $1, 1/2, 1/3, \dots, 1/100$;
- d) $0, 1/2, 2/3, 3/4, \dots, 99/100$.

Obs. Pentru a obține vectorii c și d va trebui să utilizați operatorii vectoriali, care sunt precedați de $(.)$. Excepție o fac adunarea și scăderea.

P3. Identificarea elementelor din vectori construiți cu operatorul coloană (definiți la P2):

- al cincilea element din vectorul a ;
- al douăzecilea element din vectorul b ;
- primul și ultimul element din vectorul c ;
- de la mijloc până la capătul din dreapta a intervalului din vectorul d .
- elementele pare din vectorul a ;
- elementele impare din vectorul b ;
- modificați valoarea primului termen din vectorul a cu valoarea 700 .

P4. Construiți următoarele matrice utilizând funcțiile de generare a matricelor elementare:

- o matrice A de 2 linii și 4 coloane cu toate elementele zero;
- o matrice B de 3 linii și 5 coloane cu toate elementele unu;
- o matrice C de 4 linii și 6 coloane cu elementele de pe diagonala principală unu;
- o matrice D de 5 linii și 7 coloane cu elemente aleatorii, pozitive și subunitare;
- o matrice E de 6 linii și 8 coloane cu elemente aleatorii, pozitive și negative;
- extrageți diagonalele fiecărei matrice definite la P4 și folosiți numele a , b , c , d și e pentru vectorii obținuți.

P5. Cu ajutorul funcției $\text{diag}(v,n)$, (unde v este un vector iar n este distanța de la diagonala principală), definiți următoarele matrice:

$$- \begin{pmatrix} -2 & 0 & 7 & 0 & 0 \\ 4 & -2 & 0 & 7 & 0 \\ 0 & 4 & -2 & 0 & 7 \\ 0 & 0 & 4 & -2 & 0 \\ 0 & 0 & 0 & 4 & -2 \end{pmatrix};$$

$$- \begin{pmatrix} 3 & 0 & -6 & 0 & 0 & 0 \\ 0 & 3 & 0 & -6 & 0 & 0 \\ 2 & 0 & 3 & 0 & -6 & 0 \\ 0 & 2 & 0 & 3 & 0 & -6 \\ 0 & 0 & 2 & 0 & 3 & 0 \\ 0 & 0 & 0 & 2 & 0 & 3 \end{pmatrix};$$

$$- \begin{pmatrix} 5 & 0 & -3 & 0 & 0 \\ 0 & 5 & 0 & -3 & 0 \\ 0 & 0 & 5 & 0 & -3 \\ 8 & 0 & 0 & 5 & 0 \\ 0 & 8 & 0 & 0 & 5 \end{pmatrix};$$

$$- \begin{pmatrix} -4 & 2 & 0 & 0 & 0 \\ 0 & -4 & 2 & 0 & 0 \\ 1 & 0 & -4 & 2 & 0 \\ 0 & 1 & 0 & -4 & 2 \\ 0 & 0 & 1 & 0 & -4 \end{pmatrix};$$

$$- \begin{pmatrix} -1 & 5 & 0 & 0 & 0 \\ 0 & -1 & 5 & 0 & 0 \\ 3 & 0 & -1 & 5 & 0 \\ 0 & 3 & 0 & -1 & 5 \\ 0 & 0 & 3 & 0 & -1 \end{pmatrix};$$

$$- \begin{pmatrix} 3 & 0 & 6 & 0 & 0 & -4 \\ 0 & 3 & 0 & 6 & 0 & 0 \\ -2 & 0 & 3 & 0 & 6 & 0 \\ 0 & -2 & 0 & 3 & 0 & 6 \\ 0 & 0 & -2 & 0 & 3 & 0 \\ 0 & 0 & 0 & -2 & 0 & 3 \end{pmatrix};$$

$$- \begin{pmatrix} 0 & 6 & 0 & 0 & -2 & 0 \\ 0 & 0 & 6 & 0 & 0 & -2 \\ -2 & 0 & 0 & 6 & 0 & 0 \\ 0 & -2 & 0 & 0 & 6 & 0 \\ 1 & 0 & -2 & 0 & 0 & 6 \\ 0 & 1 & 0 & -2 & 0 & 0 \end{pmatrix};$$

$$- \begin{pmatrix} 1 & -3 & 0 & 4 & 0 & 5 \\ 0 & 1 & -3 & 0 & 4 & 0 \\ 0 & 0 & 1 & -3 & 0 & 4 \\ 0 & 0 & 0 & 1 & -3 & 0 \\ 7 & 0 & 0 & 0 & 1 & -3 \\ 0 & 7 & 0 & 0 & 0 & 1 \end{pmatrix};$$

$$- \begin{pmatrix} 5 & 0 & 0 & -6 & 0 & 0 \\ 0 & 5 & 0 & 0 & -6 & 0 \\ 1 & 0 & 5 & 0 & 0 & -6 \\ 0 & 1 & 0 & 5 & 0 & 0 \\ -2 & 0 & 1 & 0 & 5 & 0 \\ 0 & -2 & 0 & 1 & 0 & 5 \end{pmatrix};$$

$$- \begin{pmatrix} -1 & 0 & 2 & 0 & 3 & 0 \\ 0 & -1 & 0 & 2 & 0 & 3 \\ 2 & 0 & -1 & 0 & 2 & 0 \\ 0 & 2 & 0 & -1 & 0 & 2 \\ 3 & 0 & 2 & 0 & -1 & 0 \\ 0 & 3 & 0 & 2 & 0 & -1 \end{pmatrix};$$

$$- \begin{pmatrix} 0 & 2 & 0 & 4 & 0 & 5 \\ 3 & 0 & 2 & 0 & 4 & 0 \\ 0 & 3 & 0 & 2 & 0 & 4 \\ 4 & 0 & 3 & 0 & 2 & 0 \\ 0 & 4 & 0 & 3 & 0 & 2 \\ 5 & 0 & 4 & 0 & 3 & 0 \end{pmatrix};$$

$$- \begin{pmatrix} 1 & 0 & -2 & 0 & -3 & 0 \\ 0 & 1 & 0 & -2 & 0 & -3 \\ 2 & 0 & 1 & 0 & -2 & 0 \\ 0 & 2 & 0 & 1 & 0 & -2 \\ 3 & 0 & 2 & 0 & 1 & 0 \\ 0 & 3 & 0 & 2 & 0 & 1 \end{pmatrix};$$

$$- \begin{pmatrix} 0 & 0 & 5 & 0 & 0 & 7 \\ 0 & 5 & 0 & 0 & 7 & 0 \\ 5 & 0 & 0 & 7 & 0 & 0 \\ 0 & 0 & 7 & 0 & 0 & 3 \\ 0 & 7 & 0 & 0 & 3 & 0 \\ 7 & 0 & 0 & 3 & 0 & 0 \end{pmatrix};$$

$$- \begin{pmatrix} -1 & 0 & 2 & 0 & 0 & 6 \\ 0 & 2 & 0 & 0 & 6 & 0 \\ 2 & 0 & 0 & 6 & 0 & 0 \\ 0 & 0 & 6 & 0 & 0 & 4 \\ 0 & 6 & 0 & 0 & 4 & 0 \\ 6 & 0 & 0 & 4 & 0 & -1 \end{pmatrix}.$$

LUCRAREA 3

3.1. Obiectivul lucrării

Lucrare își propune prezentarea principalelor probleme ce apar în algebră matriceală: ecuații matriciale, vectori și valori proprii.

3.2 Elemente de analiză matriceală

Rezolvarea sistemelor de ecuații liniare de forma $A \cdot x = B$ (A numită matricea coeficienților, matrice pătrată; x este vectorul necunoscutelor iar B este vectorul termenilor liberi) presupune calculul soluției sub forma $x = A^{-1} \cdot B$.

Matricea A^{-1} este, prin definiție, inversa matricei pătrate A și satisface relația:

$$A * A^{-1} = A^{-1} * A = I$$

în care I este matricea identitate ($A * I = I * A = A$).

Exemplu:

```
>> A=[1 2 3 4; 5 6 7 8; 9 10 11 12; 13 14 15 16]
```

```
A =
```

```
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
```

```
>> A*eye(4)
```

```
ans =
```

```
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
```

```
>> eye(4)*A
```

```
ans =
```

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Calculul inversei unei matrici este posibil dacă și numai dacă determinantul matricii A este nenul (matricea A este nesingulară).

Determinantul unei matrici se apelează cu funcția „**det(A)**” iar inversa unei matrici cu funcția „**inv(A)**”, echivalentă cu A^{-1} .

Exemplu: pentru matricea A cu următoarele valori, calculați determinantul și inversa ei.

```
>> A = [1 7 -3; 4 2 8; 6 -1 5]
A =
     1     7    -3
     4     2     8
     6    -1     5

>> B1 = inv(A)
B1 =
    0.0687   -0.1221    0.2366
    0.1069    0.0878   -0.0763
   -0.0611    0.1641   -0.0992

>> B2 = A^-1
B2 =
    0.0687   -0.1221    0.2366
    0.1069    0.0878   -0.0763
   -0.0611    0.1641   -0.0992
```

Se verifică imediat că $B1$ și $B2$ reprezintă inversa matricii A :

```
>> A * B1
ans =
    1.0000    0.0000    0.0000
    0.0000    1.0000    0.0000
    0.0000    0.0000    1.0000

>> A*B2
```

1.0000	0.0000	0.0000
0.0000	1.0000	0.0000
0.0000	0.0000	1.0000

În ambele cazuri am obținut matricea identitate. Dacă determinantul unei matrici este zero, atunci matricea respectivă nu admite inversă.

Norma unei matrici sau a unui vector este o valoare care dă măsura elementelor unei matrici sau vector.

Funcțiile MATLAB pentru matrici sunt:

norm(A) sau norm(A, 2) dă cea mai mare valoare singulară a lui A

norm(A, 1) calculează suma elementelor pe fiecare coloană a matricei A și returnează valoarea maximă

norm(A, inf) calculează suma elementelor pe fiecare linie a matricei A și returnează valoarea maximă

```
>> norm(A)
ans =
    11.7797
>> norm(A,2)
ans =
    11.7797
>> norm(A,1)
ans =
     16
>> norm(A,inf)
ans =
     14
```

Pentru un vector funcțiile MATLAB sunt:

norm(x, p) calculează radicalul indice p din suma elementelor în valoare absolută a vectorului x , ridicate la puterea p :

$$\sqrt[p]{|x_1|^p + |x_2|^p + \dots + |x_n|^p}$$

norm(x, 2) calculează radicalul sumei pătratelor elementelor vectorului x

norm(x, inf) returnează valoarea absolută maximă a elementelor vectorului x

norm(x, -inf) returnează valoarea absolută minimă a elementelor vectorului x

Pentru vectorul x definit în continuare se obțin următoarele rezultate:

```
>> x=[5 -4 3 2 7 -10]
x =
     5 -4 3 2 7 -10
>> norm(x,3)
ans =
    11.6151
>> norm(x,2)
ans =
    14.2478
>> norm(x,inf)
ans =
     10
>> norm(x,-inf)
ans =
     2
```

3.3 Valori și vectori proprii

Pentru matricele pătrate de ordin n , găsirea soluțiilor nebanale ale ecuației $A*x=\lambda*x$ înseamnă obținerea unui vector coloană de mărime n și a unor valori scalare λ . Vectorul x se numește vector propriu, iar λ se numește valoare proprie.

Rezolvarea ecuației presupune rezolvarea următorului sistem de ecuații omogene: $A*x - \lambda*x = 0$, adică $(A - \lambda*I)*x = 0$, unde I este matricea identitate de ordinul n .

Matricea $(A - \lambda*I)$ se numește matrice caracteristică, iar valorile proprii λ se obțin din rezolvarea ecuației caracteristice $\det(A - \lambda*I) = 0$, care este o ecuație algebrică de ordin n în λ .

În unele probleme ale fizicii matematice apar ecuații de vectori și valori proprii generalizate: $A*x = \lambda*B*x$, A și B fiind matrici pătrate. În acest caz valorile lui λ se numesc valori proprii generalizate, iar vectorii x se numesc vectori proprii generalizați.

Rezolvarea unor astfel de probleme se face aplicând metoda Gauss de rezolvare a sistemelor de ecuații liniare. Programul MATLAB are definite funcții care calculează vectorii proprii (*eigen vectors*) și valorile proprii (*eigen values*) pentru matrici pătrate:

V=eig(A)	determină vectorul ce conține valorile proprii ale matricei A
[V,D]=eig(A)	returnează o matrice diagonală D ce conține valorile proprii ale matricei A și o matrice V ce are pe coloane vectorii proprii corespunzători valorilor proprii
[V,D]=eig(A,'nobalance')	face calculul fără a efectua o balansare. În cazul unor elemente cu valori mici, comparabile cu erorile de rotunjire, operația de balansare le face la fel de semnificative cu restul elementelor, putând duce la rezultate incorecte
V=eig(A,B)	determină vectorul ce conține valorile proprii generalizate ale matricelor A și B
[V,D]=eig(A,B)	rezultă o matrice diagonală D ce conține valorile proprii generalizate și o matrice V ce conține vectorii proprii generalizați: $A*V = B*V*D$

Exemplu: valorile și vectorii proprii ai matricei A, matrice Pascal de ordinul 3:

```
>> [V,D]=eig(A)
V =
    -0.5438    -0.8165    0.1938
     0.7812    -0.4082    0.4722
    -0.3065     0.4082    0.8599
D =
```

0.1270	0	0
0	1.0000	0
0	0	7.8730

3.4. Rezolvarea sistemelor de ecuații liniare

Una dintre problemele întâlnite cel mai frecvent în calculul științific este rezolvarea unor sisteme de ecuații liniare. Utilizând notația matriceală, un sistem de ecuații liniare simultane poate fi scris

$$A \cdot x = b$$

unde există atât de multe ecuații cât sunt necunoscute. A este o matrice pătratică cunoscută de ordinul n , b este un vector coloană cu n componente iar x este un vector coloană necunoscut cu n componente.

În algebra liniară am învățat că soluția ecuației $A \cdot x = b$ poate fi scrisă ca $x = A^{-1} \cdot b$, unde A^{-1} este inversa matricei A .

De exemplu, dacă se dă următorul sistem de ecuații liniare:

$$\begin{cases} x + 2y + 3z = 1 \\ 4x + 5y + 6z = 1 \\ 7x + 8y = 1 \end{cases}$$

Matricea coeficienților A va fi:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 0 \end{bmatrix} \text{ și vectorul } b = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

De obicei există trei variante prin care putem afla valoarea lui x în Matlab:

1. Prima variantă include utilizarea matricei inverse, **inv**.

```
>> A = [1 2 3; 4 5 6; 7 8 0];
>> b = [1; 1; 1];
>> x = inv(A)*b
x =
    -1.0000
     1.0000
    -0.0000
```

2. Cea de-a doua variantă folosește operatorul ($\backslash$). Algoritmul numeric din spatele acestui operator este eficient din punct de vedere computațional. Acesta este o

modalitate fiabilă din punct de vedere numeric de rezolvare a unui sistem de ecuații liniare prin utilizarea unui proces bine cunoscut de eliminare de tip Gauss.

```
>> A = [1 2 3; 4 5 6; 7 8 0];  
>> b = [1; 1; 1];  
>> x = A\b  
x =  
    -1.0000  
     1.0000  
    -0.0000
```

3. Utilizând scrierea $x=A^{-1}*B$

```
>> A = [1 2 3; 4 5 6; 7 8 0];  
>> b = [1; 1; 1];  
>> x = A^-1 * b  
x =  
    -1.0000  
     1.0000  
    -0.0000
```

Se observă că cele trei metode au produs rezultate identice.

Sistemele de ecuații liniare de forma $y * C=D$ au matricea coeficienților pătrată, vectorul necunoscutelor și cel al termenilor liberi sunt vectori linie. Sistemul de ecuații se înmulțește la dreapta cu C^{-1} , rezultând în final: $y=D*C^{-1}$.

Soluția sistemului în MATLAB este:

- a). $y= D \cdot C^{-1}$
- b). $y=D*\text{inv}(C)$
- c). $y=D\backslash C$

Operațiile de mai sus sunt valabile pentru sisteme determinate și matrici ale coeficienților nesingulare. Pentru sisteme supradeterminate, mai multe ecuații decât necunoscute, se folosește pseudo inversa.

Pentru sisteme nedeterminate (mai puține ecuații decât necunoscute) teoretic **nu există soluție**. Se poate încerca rezolvarea sistemului în MATLAB, prin folosirea operației de împărțire a vectorului B la stânga cu matricea A.

Exemplu: sistemul nedeterminat de 2 ecuații și 3 necunoscute are matricea A și vectorul B:

```
>> A=[1 2 3;4 5 6];
>> B=[7;8];
>> x=A\B % încercăm o soluție posibilă
x =
    -3.0000
         0
     3.3333
```

Soluția obținută a considerat o necunoscută nulă ($x_2=0$) și celelalte două au rezultat dintr-un sistem de 2 ecuații cu 2 necunoscute.

3.4.1 Pseudo inversa unei matrice

Este cazul sistemelor supradeterminate, când numărul de ecuații este mai mare decât cel al necunoscutelor. În acest caz rezolvarea sistemului $A*x=B$ se face aplicând următoarea transformare: $x=C*B$, unde matricea C se numește pseudo inversa matricei A.

Funcția „**pinv(A)**” calculează pseudo inversa lui A, fiind de aceeași dimensiune cu a matricei A transpuse.

$$A*C*A=A, C*A*A=C, (A*C)^T=A*C, (C*A)^T=C*A$$

Exemplu: fie matricea $A=[1 \ 2; 3 \ 4; 5 \ 6]$ și $B=[7;8;9]$. Rezolvați sistemul supradeterminat $A*x = B$ cu 3 ecuații și 2 necunoscute.

```
>> A=[1 2; 3 4; 5 6];
>> B=[7; 8; 9];
>> C=pinv(A) % pseudo inversa
C =
    -1.3333    -0.3333     0.6667
     1.0833     0.3333    -0.4167
>> x=C*B % soluția sistemului
x =
```

–6.0000
6.5000

3.5 Aplicații

a) Fie matricele A, B, C:

$$A = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$B = \begin{bmatrix} 2 & 3 & 2 & 0 & 0 & 0 & 0 \\ 1 & 3 & 4 & 3 & 1 & 0 & 0 \\ 0 & 1 & 3 & 4 & 3 & 1 & 0 \\ 0 & 0 & 1 & 3 & 4 & 3 & 1 \\ 0 & 0 & 0 & 1 & 3 & 4 & 3 \\ 0 & 0 & 0 & 0 & 1 & 3 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

$$C = \begin{bmatrix} 2 & 0 & 2 & 0 & 0 & 0 & 0 \\ 1 & 2 & 1 & 2 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 & 2 & 0 & 0 \\ 0 & 0 & 1 & 2 & 1 & 2 & 1 \\ 0 & 0 & 0 & 0 & 2 & 0 & 2 \\ 0 & 0 & 0 & 0 & 1 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

Să se rezolve ecuațiile matriceale $A * X = B$ și $Y * A = C$.

b) Fie matricele D, E, F:

$$D = \begin{bmatrix} 0 & 0 & 0 & 0 & 4 & 2 & 3 \\ 0 & 0 & 0 & 4 & 2 & 3 & 1 \\ 0 & 0 & 4 & 2 & 3 & 1 & -3 \\ 0 & 4 & 2 & 3 & 1 & -3 & 0 \\ 4 & 2 & 3 & 1 & -3 & 0 & 0 \\ 2 & 3 & 1 & -3 & 0 & 0 & 0 \\ 3 & 1 & -3 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$E = \begin{bmatrix} 0 & 0 & 0 & 0 & 3 & 0 & 10 \\ 0 & 0 & 0 & 3 & 4 & 10 & 0 \\ 0 & 0 & 0 & 4 & 10 & 1 & 2 \\ 0 & 3 & 4 & 10 & 1 & 2 & 0 \\ 3 & 4 & 10 & 1 & 0 & 0 & 0 \\ 0 & 10 & 1 & 2 & 0 & 0 & 0 \\ 10 & 0 & 2 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$F = \begin{bmatrix} \sqrt{2} & 0 & \sqrt{2} & 0 & \sqrt{2} & 0 & 1/5 \\ 0 & \sqrt{2} & 0 & \sqrt{2} & 0 & 1/5 & 0 \\ \sqrt{2} & 0 & \sqrt{2} & 0 & 1/5 & 0 & \sqrt{2} \\ 0 & \sqrt{2} & 0 & 1/5 & 0 & \sqrt{2} & 0 \\ \sqrt{2} & 0 & 1/5 & 0 & \sqrt{2} & 0 & \sqrt{2} \\ 0 & 1/5 & 0 & \sqrt{2} & 0 & \sqrt{2} & 0 \\ 1/5 & 0 & \sqrt{2} & 0 & \sqrt{2} & 0 & \sqrt{2} \end{bmatrix}$$

Să se rezolve ecuațiile matriceale $D * V = E$ și $W * D = F$.

c) Să se rezolve următoarele sistemele de ecuații:

$$- \begin{cases} x_1 + x_2 + x_3 + x_4 = 0 \\ 2 * x_1 - x_2 - x_3 - 2 * x_4 = 1 \\ -x_1 + 3 * x_2 + 2 * x_3 + 3 * x_4 = -1 \\ x_1 - 2 * x_2 + x_3 - 2 * x_4 = -6 \end{cases}$$

$$- \begin{cases} x_1 - x_2 + 2 * x_3 + 3 * x_4 + 5 * x_5 = 2 \\ 2 * x_1 - 2 * x_2 + x_3 \quad \square \quad \square - 4 * x_5 = 0 \\ -3 * x_1 + x_2 - 2 * x_3 - 2 * x_4 \quad \square \quad \square = 1 \\ -x_1 + 2 * x_2 - 2 * x_3 + 3 * x_4 + 3 * x_5 = 3 \\ 5 * x_1 - 2 * x_2 + x_3 + x_4 + x_5 = -2 \end{cases}$$

$$- \begin{cases} 2 * x_1 + 3 * x_2 - x_3 + 4 * x_4 - 5 * x_5 = 10 \\ 5 * x_1 - x_2 + 2 * x_3 - 3 * x_4 + 4 * x_5 = 20 \\ x_1 + 4 * x_2 - 2 * x_3 + 5 * x_4 + 6 * x_5 = 15 \\ 3 * x_1 - 2 * x_2 + x_3 + 4 * x_4 - x_5 = 5 \\ x_1 + 2 * x_2 + 3 * x_3 - 4 * x_4 + 5 * x_5 = 25 \end{cases}$$

$$- \begin{cases} 4 * x_1 - x_2 + 3 * x_3 + 2 * x_4 - 5 * x_5 = 30 \\ 2 * x_1 + 4 * x_2 - 5 * x_3 + x_4 + 3 * x_5 = 12 \\ 3 * x_1 - 2 * x_2 + 2 * x_3 + 5 * x_4 - 4 * x_5 = 8 \\ 5 * x_1 + 2 * x_2 - 3 * x_3 + 4 * x_4 + x_5 = 18 \\ x_1 - 3 * x_2 + 2 * x_3 - x_4 + 5 * x_5 = 22 \end{cases}$$

$$- \begin{cases} 3 * x_1 & + & 2 * x_2 & - & 4 * x_3 & + & x_4 & - & 2 * x_5 & = & 14 \\ x_1 & + & 4 * x_2 & + & 2 * x_3 & - & 3 * x_4 & + & 5 * x_5 & = & 20 \\ 2 * x_1 & - & 3 * x_2 & + & 5 * x_3 & + & x_4 & - & 4 * x_5 & = & 15 \\ 4 * x_1 & + & x_2 & - & 2 * x_3 & + & 3 * x_4 & - & 5 * x_5 & = & 5 \\ 5 * x_1 & - & 2 * x_2 & + & 3 * x_3 & + & 4 * x_4 & - & x_5 & = & 10 \end{cases}$$

$$- \begin{cases} 2 * x_1 & + & 3 * x_2 & - & x_3 & = & 10 \\ 5 * x_1 & - & x_2 & + & 2 * x_3 & = & 20 \\ x_1 & + & 4 * x_2 & - & 2 * x_3 & = & 15 \\ 3 * x_1 & - & 2 * x_2 & + & x_3 & = & 5 \end{cases}$$

LUCRAREA 4

4.1. Obiectivul lucrării

În cadrul acestui laborator, vom explora realizarea de grafice 2D și 3D în Matlab. Graficele constituie un instrument esențial în analiza datelor și în prezentarea rezultatelor într-o formă vizuală și ușor de înțeles.

Scopul acestei lucrări este de a învăța studenții cum să construiască și să interpreteze graficele 2D și 3D folosind Matlab. Vom explora funcțiile de bază necesare pentru crearea graficelor, vom învăța cum să personalizăm aspectul graficelor și cum să adăugăm etichete, titluri și legende pentru o prezentare mai profesională.

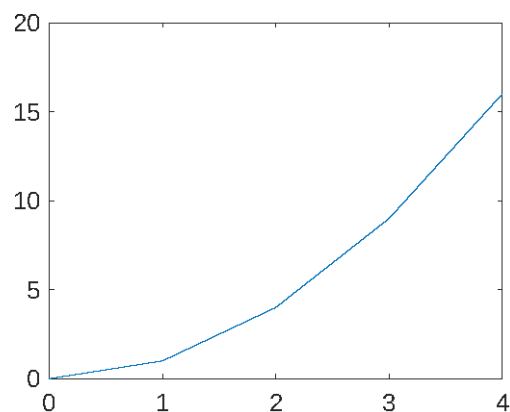
4.2 Grafice 2D

Graficele 2D sunt utilizate pentru reprezentarea relațiilor dintre două variabile. Matlab oferă un set bogat de funcții pentru a realiza diverse tipuri de grafice 2D, cum ar fi linii, bare, puncte, și multe altele.

Pentru a crea un grafic 2D în Matlab, vom folosi funcția „**plot()**”. Această funcție primește ca argumente valorile pe axa x și valorile corespunzătoare de pe axa y, permițându-ne să reprezentăm relația dintre cele două seturi de date.

Exemplu:

```
x = [0, 1, 2, 3, 4]
y = [0, 1, 4, 9, 16]
plot(x, y)
```



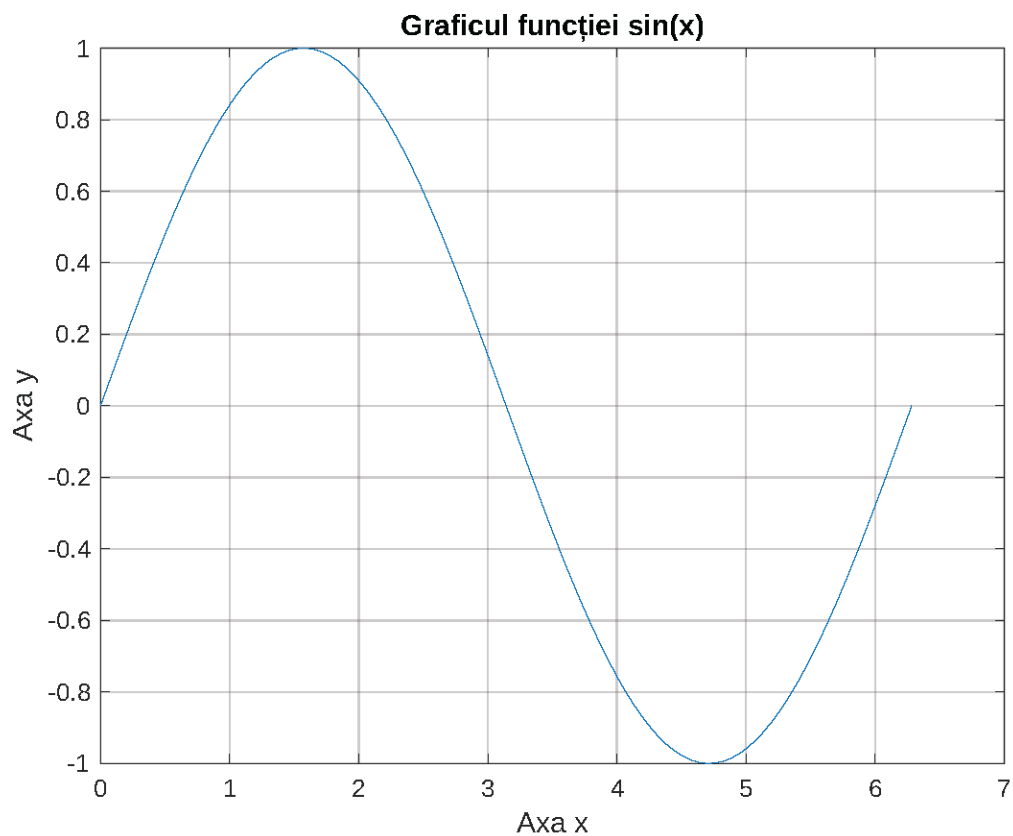
4.2.1. Setarea axelor și etichetelor graficelor

Pentru o prezentare clară a datelor, este important să personalizăm axele și etichetele graficelor. Putem utiliza funcțiile „**xlabel()**”, „**ylabel()**”, „**title()**” și „**grid()**” pentru a adăuga

etichete corespunzătoare pe axele x și y, precum și un titlu graficului și o rețea pentru o mai bună orientare.

Exemplu:

```
x = linspace(0, 2*pi, 100);  
y = sin(x);  
plot(x, y);  
xlabel('Axa x');  
ylabel('Axa y');  
title('Graficul funcției sin(x)');  
grid on;
```



Funcția plot are următoarea formă sintactică:

plot (x, y, S)

x și **y** sunt vectorii ce definesc punctele ce vor fi reprezentate;

S este un șir de caractere cuprins între , ' sau „ ” și reprezintă *tipul liniei* ce unește punctele definite de x și y, *simbolul* cu care se reprezintă punctele definite de x și y și *culoarea* liniei utilizate.

Tipul liniei poate fi:

-	linie continuă (implicit)
:	puncte
-.	linie-punct
--	liniute
(none	fără linie

Simbolurile utilizate

)

sunt:

.	punct
o	cerc
x	cruce
+	semnul +
*	asterisc
s	pătrat
d	romb
v	triunghi cu vârful în jos
^	triunghi cu vârful în sus
>	triunghi cu vârful la dreapta
<	triunghi cu vârful la stânga
p	pentagrama stelată
h	hexagrama stelată

Culorile utilizate sunt:

b	negru
g	verde
r	roșu
c	cyan
m	magenta
y	galben
k	negru
w	alb

De exemplu, **plot(x, y, 'c+:')** va reprezenta un grafic unde punctele vor fi simbolizate cu semnul + , linia ce va uni aceste puncte va fi reprezentată prin puncte de culoarea, **plot(x, y, 'bd')** va reprezenta un grafic unde punctele vor fi simbolizate cu niște romburi de culoare albastră fără să fie unite printr-o linie.

4.2.2. Diverse tipuri de grafice 2D

În funcție de datele pe care dorim să le reprezentăm și de obiectivele noastre, putem alege diferite tipuri de grafice 2D. Printre cele mai comune se numără:

- Graficele de linii „**plot()**”
- Histograma „**histogram()**”
- Graficele de bare „**bar()**”
- Graficele de puncte „**scatter()**”
- Graficele de funcții trigonometrice „**sin()**”, „**cos()**”, „**tan()**”

Fiecare tip de grafic are particularitățile sale și este util în anumite contexte.

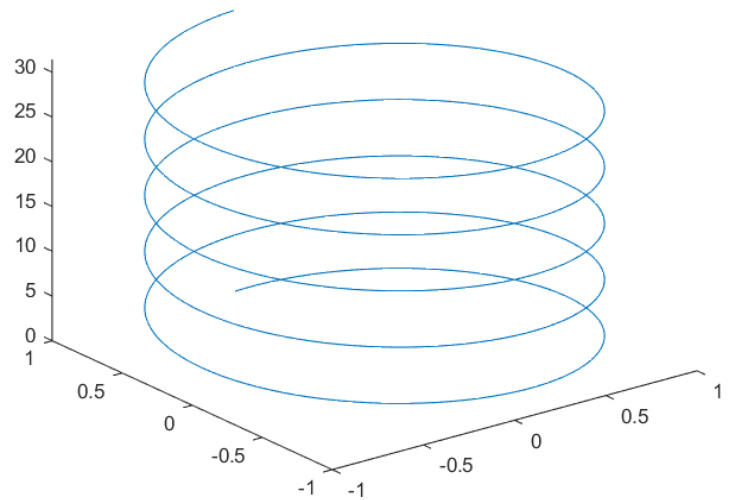
4.3. Grafice 3D

Graficele 3D permit reprezentarea datelor într-un spațiu tridimensional, furnizând o perspectivă suplimentară asupra informațiilor. În Matlab, putem crea diverse tipuri de grafice 3D, inclusiv suprafețe și grafice de tip mesh.

Pentru a realiza un grafic 3D în Matlab, vom folosi funcția „**plot3()**”. Această funcție primește ca argumente valorile pe axa x, valorile pe axa y și valorile corespunzătoare de pe axa z, permițându-ne să reprezentăm datele într-un spațiu tridimensional.

Exemplu:

```
t = 0:pi/50:10*pi;  
x = sin(t);  
y = cos(t);  
z = t;  
plot3(x, y, z);
```

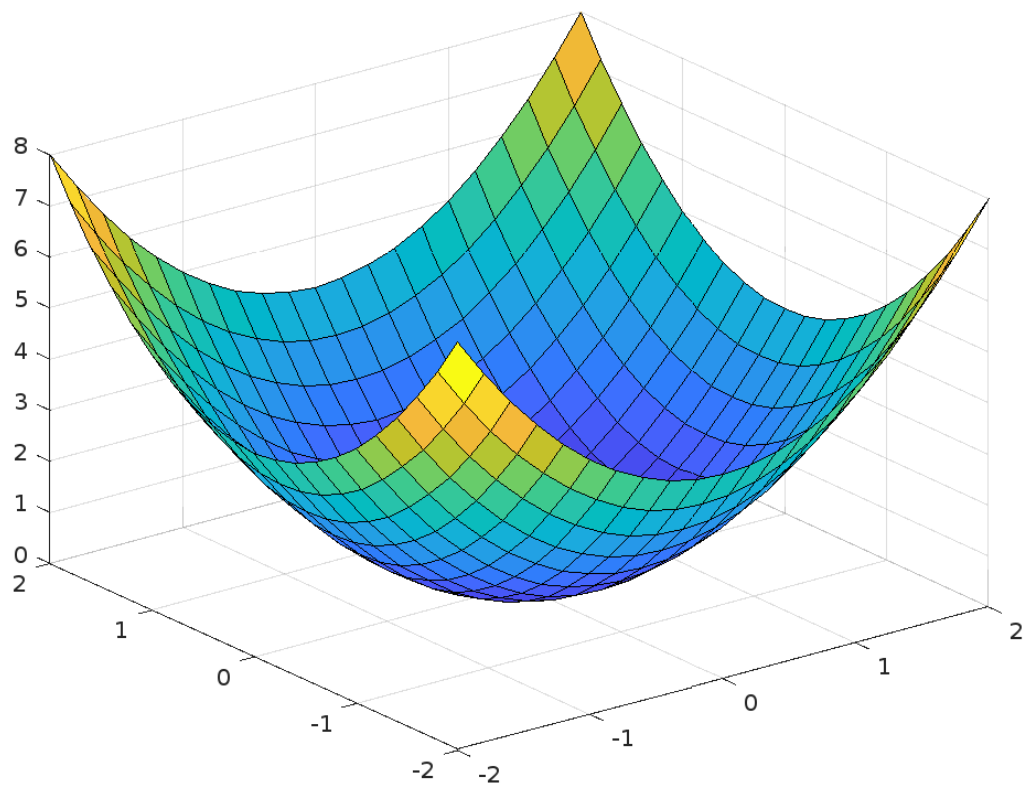


4.3.1. Grafice de suprafețe și grafice de tip mesh

Pentru a crea grafice de suprafețe sau grafice de tip mesh în Matlab, vom utiliza funcțiile „**surf()**” și „**mesh()**”. Aceste tipuri de grafice sunt ideale pentru vizualizarea datelor tridimensionale, precum funcții de două variabile.

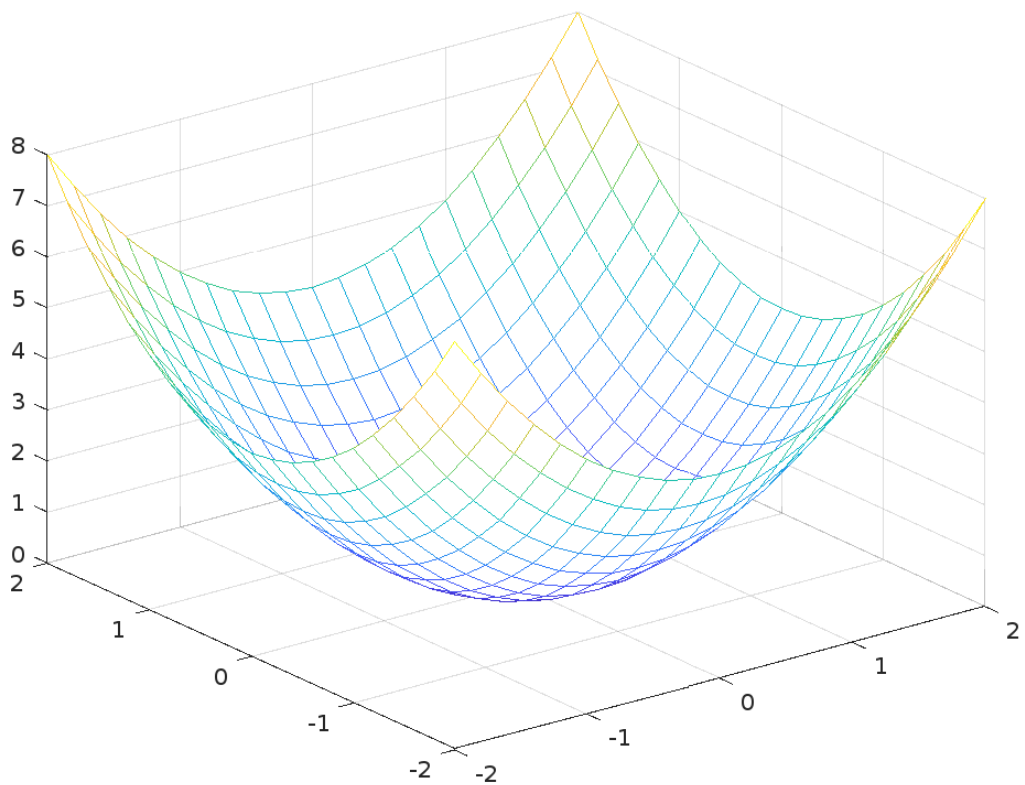
Exemplu - Grafic de suprafață:

```
[X, Y] = meshgrid(-2:0.2:2);  
Z = X.^2 + Y.^2;  
surf(X, Y, Z);
```



Exemplu - Grafic de tip mesh:

```
[X, Y] = meshgrid(-2:0.2:2);  
Z = X.^2 + Y.^2;  
mesh(X, Y, Z);
```

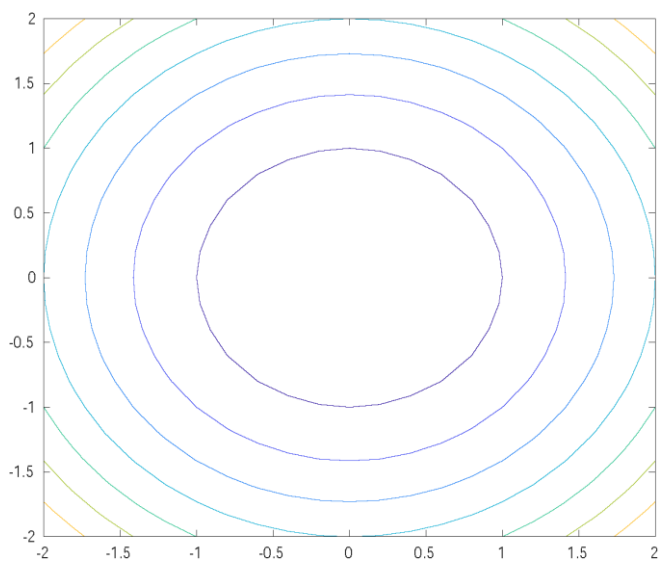


4.3.2. Liniile de contur

Pentru a evidenția curbele de nivel ale unei funcții într-un grafic de suprafață, putem utiliza funcția „**contour()**”. Aceasta ne permite să identificăm mai ușor zonele cu valori similare în datele tridimensionale.

Exemplu:

```
[X, Y] = meshgrid(-2:0.2:2);
Z = X.^2 + Y.^2;
contour(X, Y, Z);
```



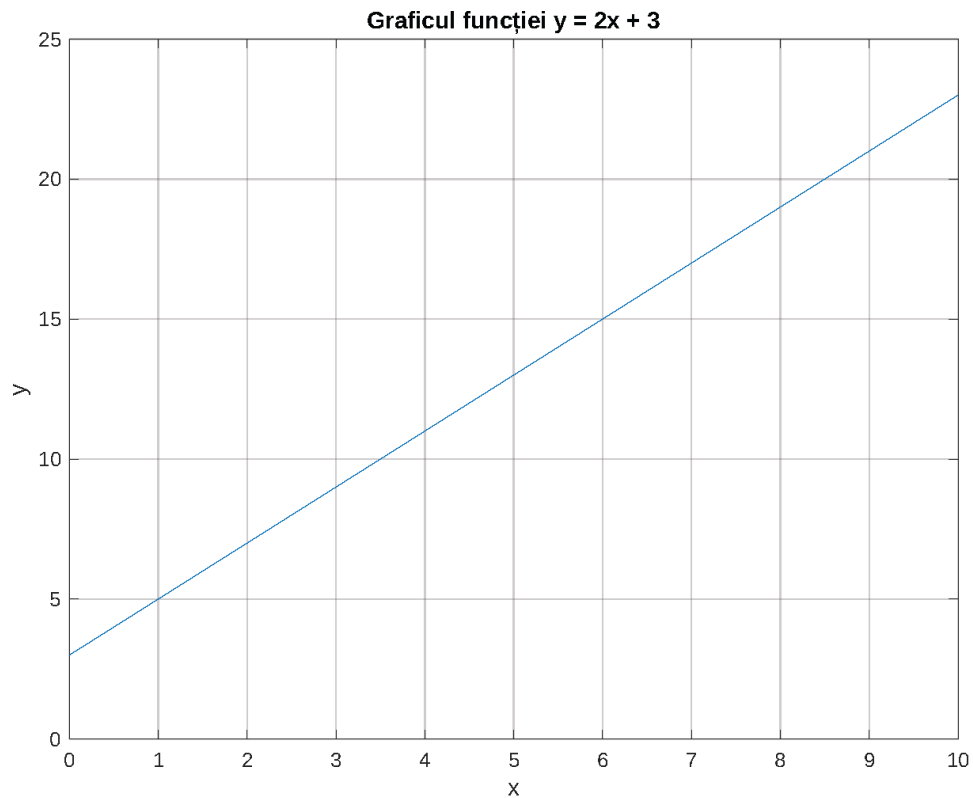
4.4. Exemple de grafice 2D și 3D

În această secțiune, vom explora diverse exemple practice pentru realizarea de grafice 2D și 3D în Matlab. Fiecare exemplu va fi însoțit de codul corespunzător, astfel încât veți putea reproduce graficele și le veți putea modifica după nevoile voastre.

4.4.1. Exemple de grafice 2D

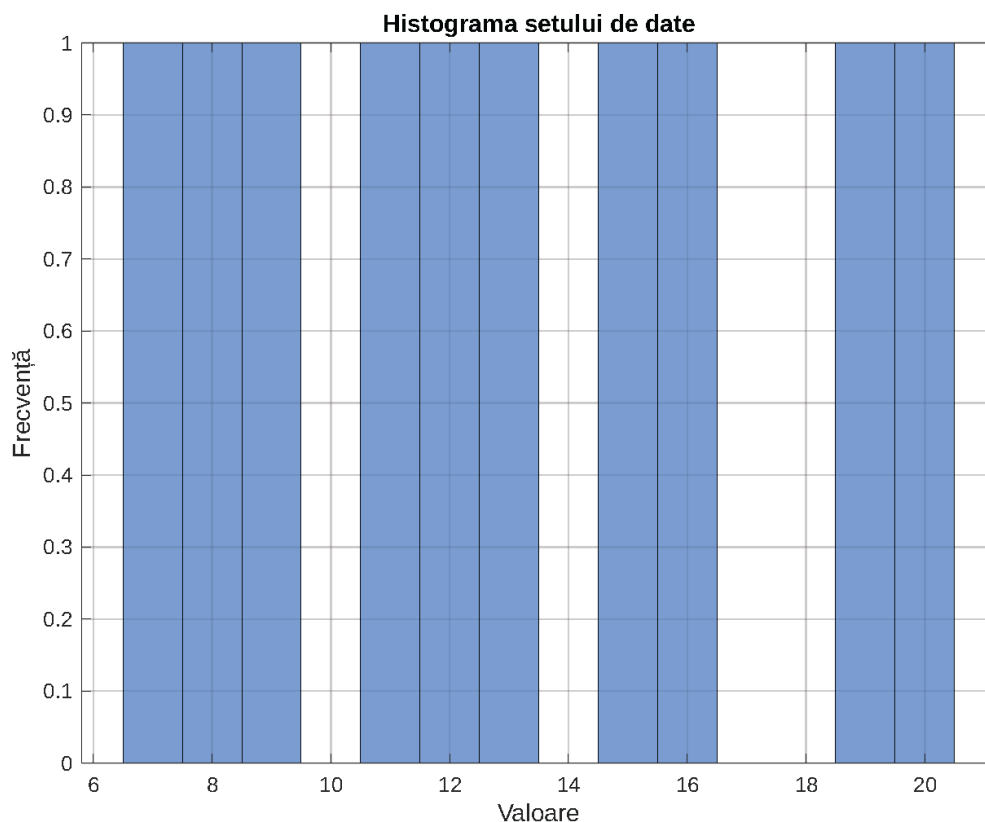
Exemplu 1: Graficul unei funcții liniare

```
x = linspace(0, 10, 100);  
y = 2*x + 3;  
plot(x, y);  
xlabel('x');  
ylabel('y');  
title('Graficul funcției y = 2x + 3');  
grid on;
```



Exemplu 2: Histograma unui set de date

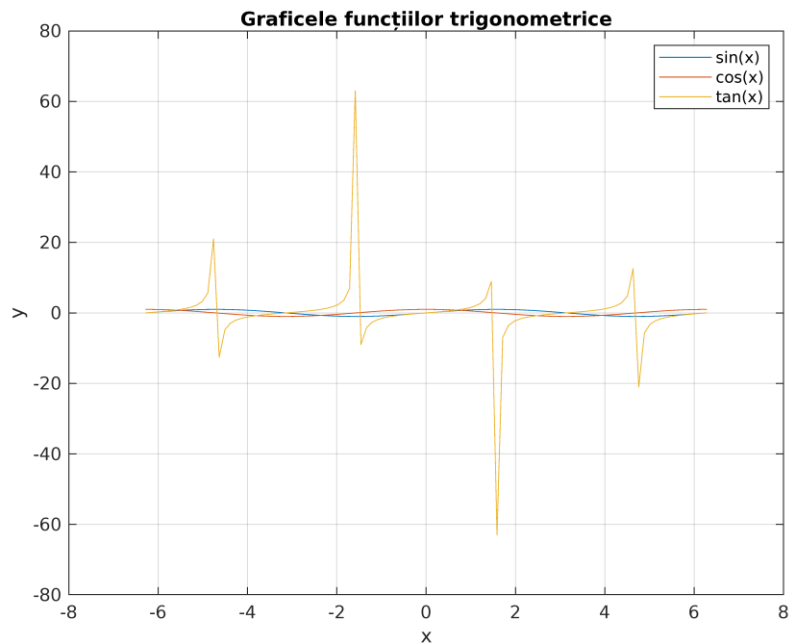
```
data = [12, 7, 15, 9, 20, 11, 8, 13, 16, 19];  
histogram(data);  
xlabel('Valoare');  
ylabel('Frecvență');  
title('Histograma setului de date');  
grid on;
```



Exemplu 3: Graficul mai multor funcții pe aceeași figură

```
x = linspace(-2*pi, 2*pi, 100);  
y1 = sin(x);  
y2 = cos(x);  
y3 = tan(x);  
plot(x, y1, x, y2, x, y3);  
legend('sin(x)', 'cos(x)', 'tan(x)');  
xlabel('x');
```

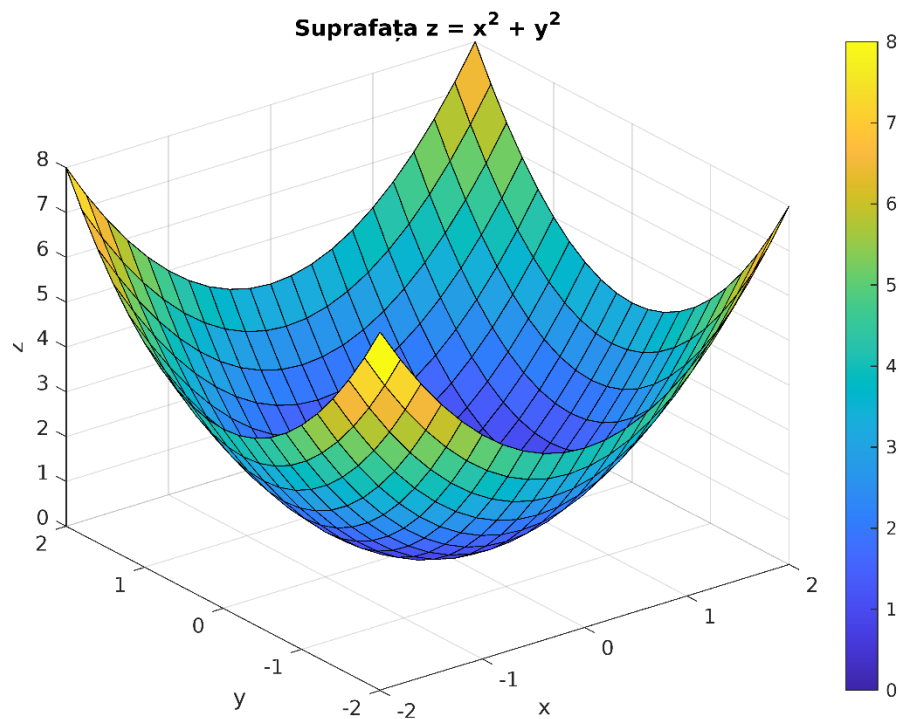
```
ylabel('y');  
title('Graficele funcțiilor trigonometrice');  
grid on;
```



4.4.2. Exemple de grafice 3D

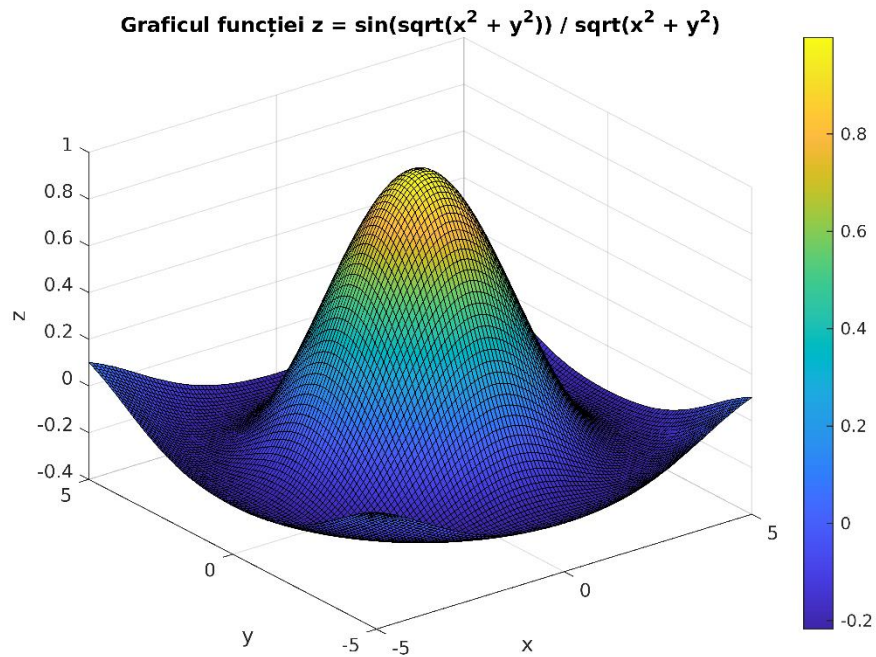
Exemplu 1: Reprezentarea unei suprafețe

```
>> [X, Y] = meshgrid(-2:0.2:2);  
>> Z = X.^2 + Y.^2;  
>> surf(X, Y, Z);  
>> xlabel('x');  
>> ylabel('y');  
>> zlabel('z');  
>> title('Suprafața  $z = x^2 + y^2$ ');  
>> colorbar;
```



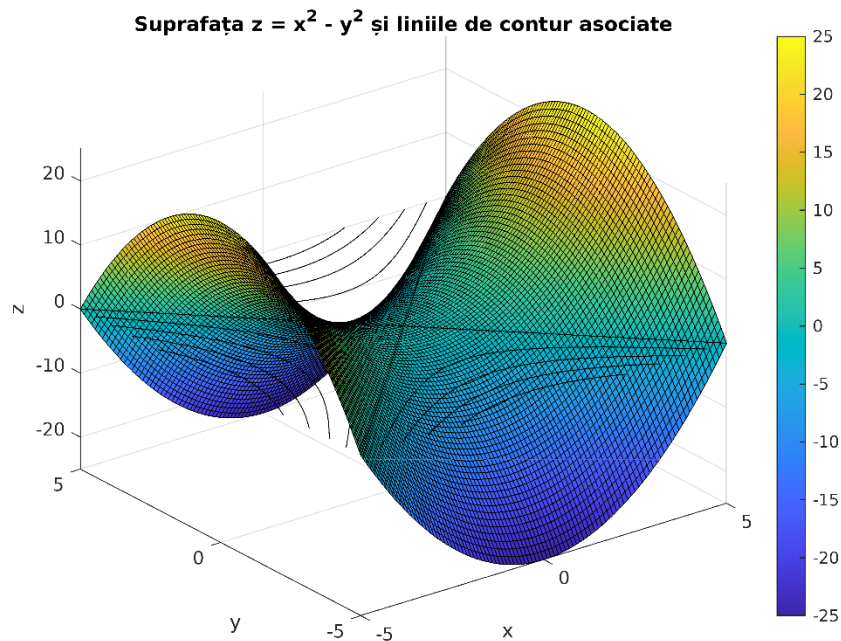
Exemplu 2: Graficul unei funcții de două variabile

```
>> [X, Y] = meshgrid(-5:0.1:5);
>> Z = sin(sqrt(X.^2 + Y.^2)) ./ (sqrt(X.^2 + Y.^2));
>> surf(X, Y, Z);
>> xlabel('x');
>> ylabel('y');
>> zlabel('z');
>> title('Graficul funcției  $z = \sin(\sqrt{x^2 + y^2}) / \sqrt{x^2 + y^2}$ ');
>> colorbar;
```



Exemplu 3: Suprafața și liniile de contur asociate

```
[X, Y] = meshgrid(-5:0.1:5);
Z = X.^2 - Y.^2;
surf(X, Y, Z);
hold on;
contour(X, Y, Z, 'k');
xlabel('x');
ylabel('y');
zlabel('z');
title('Suprafața  $z = x^2 - y^2$  și liniile de contur asociate');
colorbar;
```



4.5. Aplicații

- 1) Realizează un grafic 2D al unei funcții la alegere.
- 2) Creează un grafic 3D pentru o suprafață la alegere.
- 3) Generează un grafic 2D cu două sau mai multe funcții pe același set de axe.
- 4) Realizează un grafic de bare pentru un set de date furnizat. Setul de date este numărul de vânzări realizate în fiecare lună de o firmă: `vanzari = [150, 200, 180, 250, 300]`;
- 5) Reprezintă mai multe seturi de date folosind subplot-uri. Setul de date 1 și 2, evoluția temperaturii în două locații: `temperaturi_locatie1 = [20, 22, 25, 27, 30]` și `temperaturi_locatie2 = [18, 20, 23, 25, 28]`;
- 6) Generează un grafic polar pentru o funcție circulară, $r = 2 \cdot \cos(3 \cdot \theta)$. Unghiul θ este `theta = linspace(0, 2*pi, 100)`.
- 7) Realizează un grafic 3D cu date generate aleatoriu utilizând `scatter3()`.
- 8) Creează un grafic 3D de tip mesh pentru o funcție de două variabile, $z = \sin(\sqrt{X.^2 + Y.^2}) ./ (\sqrt{X.^2 + Y.^2})$.
- 9) Reprezintă un set de date în spațiul tridimensional folosind `scatter3()`.
- 10) Desenează un grafic 2D cu două curbe și personalizează-l cu stiluri și culori diferite. Prima curbă are ecuația $y1 = \sin(x)$ iar a doua $y2 = \cos(x)$.
- 11) Realizează un grafic de tip `bar3` pentru a compara datele în trei dimensiuni. Setul de date este reprezentat de numărul de vânzări pentru fiecare lună și fiecare locație:

vanzari_locatie1 = [150, 200, 180, 250, 300] și vanzari_locatie2 = [180, 190, 220, 240, 260].

- 12) Creează un grafic 3D cu liniile de contur asociate pentru o funcție tridimensională, $Z = X.^2 + Y.^2$.
- 13) Generează un grafic de tip “plăcintă” pentru o distribuție de date (200, 300, 150, 250).
- 14) Reprezintă un set de date într-un grafic 2D cu etichete personalizate pe axele x și y, cum ar fi numărul de vânzări pe zilele unei săptămâni.
- 15) Realizează un grafic 3D interactiv cu date obținute cu funcția rand(). Pentru a fi interactiv trebuie să fie permisă rotirea graficului cu mouse-ul.
- 16) Generează un grafic 2D cu două subgrafice alăturate pentru compararea a două funcții, $\sin(x)$ și $\cos(x)$.
- 17) Reprezintă un set de date într-un grafic de tip boxplot pentru a evidenția distribuția și valorile extreme.
- 18) Creează un grafic 3D de tip surface cu o iluminare personalizată pentru efecte vizuale mai plăcute pentru funcția $Z = \sin(\sqrt{X.^2 + Y.^2}) ./ (\sqrt{X.^2 + Y.^2})$.

Bibliografie

- 1. MATLAB Documentation. MathWorks. (<https://www.mathworks.com/help/matlab/>)
- 2. MATLAB Plot Gallery. (<https://www.mathworks.com/products/matlab/plot-gallery.html>)
- 3. Obe, R., & Bausch, P. (2015). *MATLAB For Dummies*. John Wiley & Sons.
- 4. Gilat, A. (2014). *MATLAB: An Introduction with Applications*. John Wiley & Sons.
- 5. Attaway, S. (2016). *Matlab: A Practical Introduction to Programming and Problem Solving*. Butterworth-Heinemann.
- 6. Tisdall, J. (2017). *Mastering MATLAB*. Pearson.
- 7. Duane C. Hanselman, Bruce L. Littlefield. (2016). *Mastering MATLAB 7*. Pearson.
- 8. Higham, D. J. (2017). *MATLAB Guide*. SIAM.
- 9. Pratap, R. (2006). *Getting Started with MATLAB: A Quick Introduction for Scientists and Engineers*. Oxford University Press.

10. Stein, M. L. (2019). *Interpolation of Spatial Data: Some Theory for Kriging*. Springer.
11. Cleve Moler. (2019). *Numerical Computing with MATLAB*. SIAM.
12. Unde, T., & Ziemer, R. E. (2017). *Matlab Programming for Engineers*. Cengage Learning.
13. Stack Overflow. (<https://stackoverflow.com/>)
14. <https://www.mathworks.com/matlabcentral/fileexchange/>
15. OpenAI. (<https://openai.com/>)

LUCRAREA 5

5.1. Obiectivul lucrării

În această lucrare de laborator, ne propunem să explorăm două concepte fundamentale în programarea Matlab - Scripturile și Funcțiile. Acestea reprezintă elemente esențiale în dezvoltarea de programe și aplicații Matlab mai complexe și mai organizate. Obiectivele specifice ale laboratorului includ:

1. Înțelegerea conceptului de script și utilizarea sa pentru executarea de comenzi Matlab.
2. Definirea și utilizarea funcțiilor în Matlab, inclusiv transmiterea de argumente și returnarea valorilor.
3. Explorarea beneficiilor utilizării scripturilor și funcțiilor în organizarea și gestionarea codului Matlab.
4. Studiarea aplicațiilor practice ale scripturilor și funcțiilor în rezolvarea problemelor de calcul și analiză a datelor.

5.2 Scripturi în Matlab

Scripturile și funcțiile joacă un rol crucial în dezvoltarea de cod Matlab eficient și reutilizabil. Scripturile sunt fișiere care conțin o serie de comenzi Matlab, iar utilizarea lor permite executarea unei serii de acțiuni într-un mod consecvent și automatizat. De asemenea, ele facilitează organizarea logică a codului și permit o mai bună gestionare a instrucțiunilor.

Un script Matlab reprezintă un fișier care conține o serie de comenzi Matlab. Aceste comenzi sunt executate secvențial, așa cum sunt scrise în fișier, fără a fi necesară definirea unei funcții. Scripturile Matlab sunt utile pentru a efectua o serie de acțiuni repetitive sau pentru a rezolva probleme simple care nu necesită modularizarea codului în funcții separate. Pentru a utiliza editorul de text creat pentru scrierea fișierelor script, se vor urma pașii prezentați în Figura 1. În fereastra prezentată în Figura 1 b) se va scrie următorul exemplu:

Exemplu 1: Un simplu script pentru afișarea unui mesaj în Matlab.

```
% Exemplu 1: Afișarea unui mesaj în Matlab
```

```
disp('Bun venit la orele de PCLP 3 unde vom învăța despre software-ul Matlab!');
```

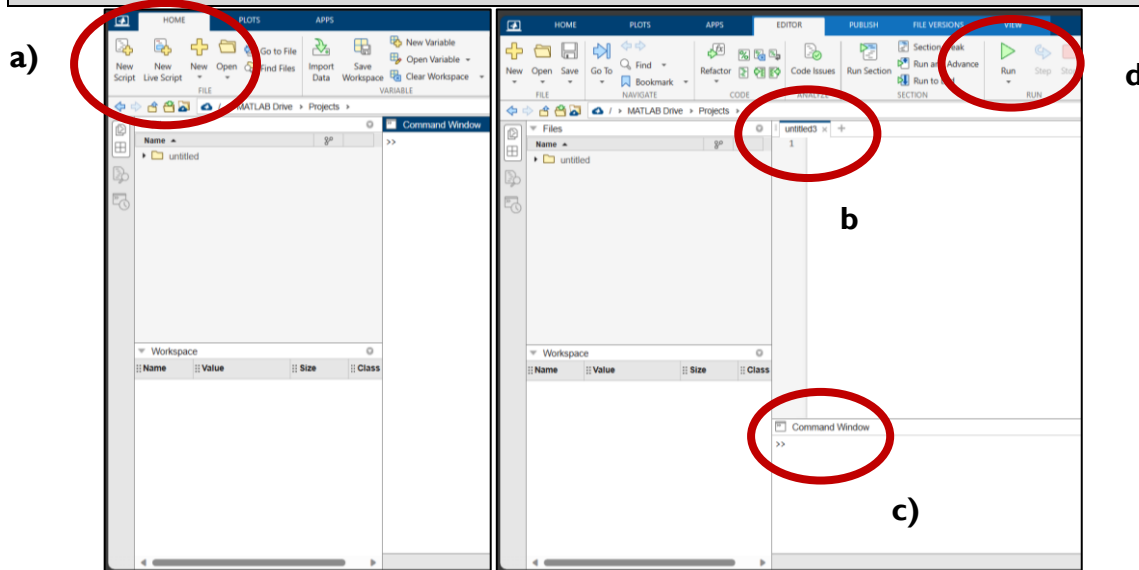


Figura 1. Utilizarea interfeței pentru a deschide un fișier script: a) se apasă butonul “New Script”; după acționarea butonului, fereastra “Command Window” va fi mutată în partea de jos a ecranului, c), și va apare o nouă fereastră, b), care va fi utilizată pentru scrierea scriptului; pentru a rula scriptul creat, va fi acționat butonul “Run”, d), iar rezultatul va putea fi vizualizat în consola c).

În exemplul 1 avem două linii de cod. Prima, de culoare verde, este o linie de comentariu ce poate fi obținută utilizând semnul “ % ”. Aceste linii de comentariu nu vor fi executate de program la apăsarea butonului “Run” ci sunt acolo pentru a ajuta programatorul cât și utilizatorul (dacă este cazul) la structurarea programului în blocuri și etichetarea lor pentru a le înțelege rolul în cadrul programului. În a doua linie se folosește funcția “disp()” care are următoarea sintaxă:

$$\text{disp}(X)$$

afișează matricea X fără a tipări numele matricei sau informații suplimentare de descriere, cum ar fi dimensiunea și numele clasei. Dacă X este un șir sau o matrice de caractere, atunci se va afișa textul respectiv.

Cu ajutorul celei de-a doua linie se va afișa textul “Bun venit la orele de PCLP 3 unde vom învăța despre software-ul Matlab!” iar în consolă, după executare, va fi afișată denumirea fișierului care rulează precum și rezultatul fișierului script.

```
>> untitled3
```

Bun venit la orele de PCLP 3 unde vom învăța despre software-ul Matlab!

Rularea unui fișier script salvat în memoria calculatorului se face fie tastând numele fișierului (fără extensie) în linia de comandă, fie apelând comanda “*Run M-File*” a meniului File, fie apăsând tasta F5 atunci când fișierul este deschis în editor și editorul este fereastra activă.

5.4. Funcții în Matlab

Funcțiile reprezintă un element esențial în programarea Matlab, oferind o modalitate eficientă de organizare a codului și reutilizare a acestuia. O funcție Matlab este o secvență de instrucțiuni care primește unul sau mai multe argumente, efectuează operații specifice și poate returna un rezultat.

Fișierele funcție diferă de fișierele script prin faptul că primele pot lucra cu argumente (pot primi parametri și returna valori), iar celelalte operează doar asupra variabilelor din mediul de lucru (Workspace). Fișierele script se utilizează de obicei pentru rezolvarea unor probleme care necesită executarea unui grup mare de comenzi, pentru care utilizarea modului linie de comandă ar fi greoaie.

Obligatoriu, prima linie a unui fișier funcție este de forma:

function [parametri_ieșire]=nume_funcție(parametri_intrare)

unde:

- *function* este cuvântul-cheie care declară fișierul ca fișier funcție;
- *nume_funcție* reprezintă numele funcției, este același cu numele sub care se salvează fișierul;
- *parametri_ieșire* reprezintă lista parametrilor de ieșire separați cu virgulă, cuprinsă între paranteze pătrate; dacă funcția nu are parametri de ieșire, atunci parantezele pătrate și semnul egal se omit;
- *parametri_intrare* reprezintă lista parametrilor de intrare separați cu virgulă, cuprinsă între paranteze rotunde; dacă funcția nu are parametri de intrare, atunci parantezele rotunde se omit.

Rularea unui fișier funcție se face în linia de comandă, folosind sintaxa:

$[parametri_ieşire]=nume_funcţie(parametri_intrare)$

unde:

- *nume_funcție* este numele funcției;
- *parametri_ieșire* lista variabilelor destinate memorării valorilor parametrilor de ieșire, separate cu virgulă, cuprinsă între paranteze pătrate; dacă funcția nu are parametri de ieșire, atunci parantezele pătrate și semnul egal se omit; dacă funcția are un singur parametru de ieșire, parantezele drepte se omit;
- *parametri_intrare* lista valorilor parametrilor de intrare separate cu virgulă, cuprinsă între paranteze rotunde; dacă funcția nu are parametri de intrare, atunci parantezele rotunde se omit.

Dacă un comentariu apare imediat sub linia de declarare a unui fișier funcție, el va constitui help-ul fișierului funcție respectiv, adică textul care va fi afișat în linia de comandă în urma apelului “help nume_funcție” unde nume_funcție este numele (fără extensie) a fișierului funcție.

5.5. Exemple

- Script pentru calculul mediei aritmetice a două numere

Să se creeze un script Matlab care să calculeze media aritmetică a două numere date și să afișeze rezultatul.

```
% Calculul mediei a două numere
a = 5;
b = 8;
media = (a + b) / 2;
disp(['Media numerelor ', num2str(a), ' și ', num2str(b), ' este: ', num2str(media)]);
```

- Script pentru generarea unui vector de numere aleatorii

Să se creeze un script Matlab care să genereze un vector de numere aleatorii întregi în intervalul [1, 100] și să-l afișeze.

```
% Generarea unui vector de numere aleatorii
vector_aleator = randi([1, 100], 1, 10);
```

```
disp('Vectorul generat este:');  
disp(vector_aleator);
```

- Script pentru ridicarea la putere a unui număr

Să se creeze un script Matlab care să permită utilizatorului să introducă un număr și să calculeze valoarea sa ridicată la puterea k, unde k este un exponent dat.

```
% Ridicarea la putere a unui număr  
numar = input('Introduceți un număr: ');  
exponent = input('Introduceți exponentul: ');  
rezultat = numar^exponent;  
disp(['Rezultatul ridicării la putere este: ', num2str(rezultat)]);
```

- Script pentru afișarea inversului unui șir de caractere

Să se creeze un script Matlab care să permită utilizatorului să introducă un șir de caractere și să afișeze inversul acestuia.

```
% Afișarea inversului unui șir de caractere  
sir_caractere = input('Introduceți un șir de caractere: ', 's');  
invers = flip(sir_caractere);  
disp(['Inversul șirului de caractere este: ', invers]);
```

- Script pentru determinarea celei mai mari cifre dintr-un număr

Să se creeze un script Matlab care să permită utilizatorului să introducă un număr și să determine cea mai mare cifră din acest număr.

```
% Determinarea celei mai mari cifre dintr-un număr  
numar = input('Introduceți un număr: ');  
cifre = num2str(numar) - '0';  
  
% Transformare număr în vector de cifre  
cifra_maxima = max(cifre);  
disp(['Cea mai mare cifră din număr este: ', num2str(cifra_maxima)]);
```

- Script pentru desenarea unui cerc în Matlab

Să se creeze un script Matlab care să deseneze un cerc cu un anumit centru și rază dată.

```
% Desenarea unui cerc în Matlab
centru_x = 2;
centru_y = 3;
raza = 5;
theta = linspace(0, 2*pi, 100);
x = centru_x + raza * cos(theta);
y = centru_y + raza * sin(theta);
plot(x, y);
axis equal;
title('Cerc desenat în Matlab');
```

- Script pentru reprezentarea grafică a unei funcții

Să se creeze un script Matlab care să reprezinte grafic o funcție dată într-un interval dat.

```
% Reprezentarea grafică a unei funcții
x = linspace(-5, 5, 100);
y = 2 * x.^2 + 3 * x - 4;
plot(x, y);
title('Graficul funcției y = 2x^2 + 3x - 4');
xlabel('x');
ylabel('y');
grid on;
```

- Script pentru generarea unei matrice de numere aleatorii

Să se creeze un script Matlab care să genereze o matrice de dimensiuni date, în care elementele sunt numere aleatorii între 0 și 1.

```
% Generarea unei matrice de numere aleatorii
nr_linii = input('Introduceți numărul de linii al matricei: ');
```

```
nr_coloane = input('Introduceți numărul de coloane al matricei: ');
matrice_aleatoare = rand(nr_linii, nr_coloane);
disp('Matricea generată este:');
disp(matrice_aleatoare);
```

- Funcție pentru calculul sumei a două numere

Să se definească o funcție Matlab care primește doi parametri (a și b) și returnează suma acestora.

```
% Funcție pentru calculul sumei a două numere
function suma = calculeaza_suma(a, b)
suma = a + b;
end
```

```
% Utilizarea funcției în consolă
rezultat = calculeaza_suma(5, 7);
disp(['Suma celor două numere este: ', num2str(rezultat)]);
```

- Funcție pentru determinarea celei mai mari cifre dintr-un număr

Să se creeze o funcție Matlab care primește un număr ca argument și returnează cea mai mare cifră din acesta.

```
% Funcție pentru determinarea celei mai mari cifre dintr-un număr
function cifra_maxima = determina_cifra_maxima(numar)
cifre = num2str(numar) - '0';
cifra_maxima = max(cifre);
end
```

```
% Utilizarea funcției
numar = 25437;
rezultat = determina_cifra_maxima(numar);
disp(['Cea mai mare cifră din număr este: ', num2str(rezultat)]);
```

- Funcție pentru găsirea maximului și minimului dintr-un vector

Să se creeze o funcție Matlab care primește un vector ca argument și returnează valorile maximă și minimă din acesta.

```
% Funcție pentru găsirea maximului și minimului dintr-un vector
function [maxim, minim] = gaseste_maxim_minim(vector)
maxim = max(vector);
minim = min(vector);
end
```

```
% Utilizarea funcției
vector = [2, 7, 1, 8, 4, 5];
[maximul, minimul] = gaseste_maxim_minim(vector);
disp(['Valoarea maximă din vector este: ', num2str(maximul)]);
disp(['Valoarea minimă din vector este: ', num2str(minimul)]);
```

- Funcție pentru generarea unei matrice diagonale

Să se definească o funcție Matlab care primește un vector ca argument și returnează o matrice diagonală cu elementele vectorului pe diagonala principală.

```
% Funcție pentru generarea unei matrice diagonale
function matrice_diag = genereaza_matrice_diagonala(vector)
matrice_diag = diag(vector);
end
```

```
% Utilizarea funcției
vector = [1, 2, 3, 4];
rezultat = genereaza_matrice_diagonala(vector);
disp('Matricea diagonală generată este:');
disp(rezultat);
```

- Funcție pentru calculul mediei aritmetice a elementelor unui vector

Să se creeze o funcție Matlab care primește un vector ca argument și returnează media aritmetică a elementelor sale.

```
% Funcție pentru calculul mediei aritmetice a elementelor unui vector  
function medie = calculeaza_medie(vector)  
medie = mean(vector);  
end
```

```
% Utilizarea funcției  
vector = [3, 6, 9, 12];  
rezultat = calculeaza_medie(vector);  
disp(['Media aritmetică a elementelor vectorului este: ', num2str(rezultat)]);
```

- Funcție pentru sortarea unui vector în ordine descendentă

Să se definească o funcție Matlab care primește un vector ca argument și returnează vectorul sortat în ordine descendentă.

```
% Funcție pentru sortarea unui vector în ordine descendentă  
function vector_sortat = sorteaza_descendent(vector)  
vector_sortat = sort(vector, 'descend');  
end
```

```
% Utilizarea funcției  
vector = [5, 1, 8, 3, 2, 7];  
rezultat = sorteaza_descendent(vector);  
disp('Vectorul sortat în ordine descendentă este:');  
disp(rezultat);
```

- Funcție pentru ridicarea la putere a elementelor unui vector

Să se definească o funcție Matlab care primește un vector și o putere ca argumente și returnează vectorul rezultat din ridicarea la putere a elementelor.

```
% Funcție pentru ridicarea la putere a elementelor unui vector
```

```
function vector_putere = ridicare_la_putere(vector, putere)
vector_putere = vector .^ putere;
end
```

% Utilizarea funcției

```
vector = [1, 2, 3, 4];
putere = 3;
rezultat = ridicare_la_putere(vector, putere);
disp(['Vectorul ridicat la puterea ', num2str(putere), ' este:']);
disp(rezultat);
```

5.6. Aplicații

1. Scrieți un script Matlab care să calculeze și să afișeze media a două numere date.
2. Definiți o funcție Matlab care să primească un vector de numere și să returneze suma lor.
3. Creați un script care să genereze un vector cu toate numerele întregi de la 1 la 100.
4. Implementați o funcție Matlab care să determine cel mai mare număr dintr-un vector dat.
5. Creați un script care să determine și să afișeze cel mai mic multiplu comun (CMMD) a două numere.
6. Implementați o funcție Matlab care să inverseze un vector dat.
7. Definiți o funcție Matlab care să returneze suma elementelor diagonalei principale dintr-o matrice pătratică.
8. Scrieți un script care să genereze și să afișeze o matrice cu numere aleatorii între 1 și 100.
9. Definiți o funcție Matlab pentru a căuta și returna valoarea maximă dintr-o matrice dată.
10. Creați un script care să rezolve o ecuație de gradul 2 ($ax^2 + bx + c = 0$) date coeficienții a, b și c.
11. Implementați o funcție Matlab pentru a verifica dacă un număr dat este palindrom (*Un palindrom este un cuvânt, un număr, o frază sau o altă secvență de simboluri care se citește*

la fel și de la început la sfârșit și de la sfârșit la început, cum ar fi cuvântul “madam” din limba engleză).

12. Scrieți un script pentru a crea un vector de numere naturale și a calcula suma lor până la un anumit index.
13. Definiți o funcție Matlab pentru a genera o matrice diagonală cu elemente date pe diagonala principală.
14. Creați un script care să calculeze și să afișeze media și deviația standard a unui set de date dat.
15. Implementați o funcție Matlab pentru a calcula produsul scalar al a două vectori.
16. Scrieți un script care să genereze și să afișeze un vector cu numerele puterilor lui 2 până la 2^{10} .
17. Scrieți un script care să genereze și să afișeze o matrice cu numere aleatorii între 1 și 100.

Bibliografie

1. Matlab Documentation: MathWorks oferă o documentație cuprinzătoare a limbajului Matlab, care include ghiduri de utilizare, exemple și informații despre funcții și comenzi specifice. Disponibil la: <https://www.mathworks.com/help/matlab/>
2. Stein, M. L. (2018). "Interpolation of Spatial Data: Some Theory for Kriging." Springer Science & Business Media. ISBN: 978-1441917466.
3. Attaway, S. (2016). "Matlab: A Practical Introduction to Programming and Problem Solving." Elsevier. ISBN: 978-0128045251.
4. Gilat, A. (2017). "Matlab: An Introduction with Applications." Wiley. ISBN: 978-1119385127.
5. Higham, D. J. (2019). "Numerical Methods and Analysis." Society for Industrial and Applied Mathematics. ISBN: 978-1611975591.
6. Luenberger, D. G. (2008). "Linear and Nonlinear Programming." Springer Science & Business Media. ISBN: 978-0387745027.

7. Hahn, J., & Valentine, A. (2014). "Essential MATLAB for Engineers and Scientists." Academic Press. ISBN: 978-0128052716.
8. Chapman, S. J. (2017). "Matlab Programming for Engineers." Cengage Learning. ISBN: 978-1305637100.
9. Pratap, R. (2017). "Getting Started with MATLAB: A Quick Introduction for Scientists and Engineers." Oxford University Press. ISBN: 978-0190649135.
10. Szeliski, R. (2010). "Computer Vision: Algorithms and Applications." Springer Science & Business Media. ISBN: 978-1848829343.

Anexă

În această anexă, sunt prezentate câteva exemple de scripturi Matlab care abordează diverse sarcini și probleme din domeniul Știința și Ingineria Materialelor. Aceste scripturi pot fi utilizate ca resurse de învățare și inspirație pentru studenții interesați să exploreze potențialul programării în Matlab în contextul materiei lor de studiu.

Exemplu 1: Calculul Densității

Scopul acestui script este să calculeze densitatea unui material pe baza masei și volumului său cunoscute. Utilizatorul va introduce valorile corespunzătoare, iar scriptul va returna densitatea materialului.

```
% Exemplu 1: Calculul Densității
```

```
% Introducerea valorilor masei și volumului
```

```
masa = input('Introduceți masa materialului (kg): ');
```

```

volum = input('Introduceți volumul materialului (m^3): ');

% Calculul densității
densitate = masa / volum;

% Afișarea rezultatului
fprintf('Densitatea materialului este %.2f kg/m^3.\n', densitate);

```

Exemplu 2: Determinarea Durității

Acest script determină duritatea materialului pe baza valorii forței de indentare și a ariei secțiunii transversale a urmei. Valorile sunt introduse de către utilizator, iar scriptul va afișa duritatea în scala Vickers.

```

% Exemplu 2: Determinarea Durității

% Introducerea valorilor forței de indentare și a ariei secțiunii transversale
forta_indentare = input('Introduceți forța de indentare (N): ');
arie_imprimare = input('Introduceți aria secțiunii transversale a impresiunii (mm^2): ');

% Calculul durității Vickers
duritate_vickers = forta_indentare / arie_imprimare;

% Afișarea rezultatului
fprintf('Duritatea Vickers este %.2f HV.\n', duritate_vickers);

```

Exemplu 3: Calculul Modulului de Elasticitate

Acest script calculează modulul de elasticitate al unui material utilizând legea lui Hooke. Utilizatorul va introduce valorile corespunzătoare pentru forța aplicată și deformarea rezultată, iar scriptul va returna modulul de elasticitate.

```

% Exemplu 3: Calculul Modulului de Elasticitate

% Introducerea valorilor forței aplicate și a deformării rezultate

```

```

forta = input('Introduceți forța aplicată (N): ');
deformare = input('Introduceți deformarea rezultată (m): ');

% Calculul modului de elasticitate
mod_elasticitate = forta / deformare;

% Afișarea rezultatului
fprintf('Modulul de elasticitate este %.2f Pa.\n', mod_elasticitate);

```

Exemplu 4: Analiza Datelor Experimentale

Acest script permite analiza unui set de date experimental prin calcularea mediei, deviației standard și reprezentarea grafică a datelor utilizând o histogramă.

```

% Exemplu 4: Analiza Datelor Experimentale

% Introducerea setului de date experimental
date_experimentale = [12.5, 15.2, 11.8, 14.6, 13.2, 12.9, 15.7, 13.8, 14.5, 13.6];

% Calculul mediei
media = mean(date_experimentale);

% Calculul deviației standard
deviatie_standard = std(date_experimentale);

% Afișarea rezultatelor
fprintf('Media datelor experimentale este %.2f.\n', media);
fprintf('Deviația standard a datelor experimentale este %.2f.\n', deviatie_standard);

% Reprezentarea grafică a datelor utilizând un histogramă
figure;
histogram(date_experimentale, 10);
title('Histograma Datelor Experimentale');
xlabel('Valoare');

```

```
ylabel('Număr de Observații');
```

Exemplu 5: Fitare Liniară

Acest script demonstrează fitarea liniară a unui set de date prin determinarea pantei și a interceptului liniei de regresie.

```
% Exemplu 5: Fitare Liniară

% Datele experimentale
x = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];
y = [2.1, 4.3, 6.4, 8.2, 9.9, 12.1, 14.2, 15.9, 18.2, 20.3];

% Fitare liniară
p = polyfit(x, y, 1);
a = p(1); % Panta
b = p(2); % Interceptul

% Afișarea rezultatelor
fprintf('Ecuația liniei de regresie este y = %.2fx + %.2f.\n', a, b);

% Reprezentarea grafică a datelor și liniei de regresie
figure;
plot(x, y, 'o', 'MarkerFaceColor', 'b');
hold on;
plot(x, polyval(p, x), 'r');
title('Fitare Liniară a Datelor Experimentale');
xlabel('x');
ylabel('y');
legend('Date Experimentale', 'Linie de Regresie');
```

Exemplu 6: Analiza Compoziției Materialelor

Acest script calculează proporția procentuală a elementelor chimice într-o compoziție materială dată și reprezintă rezultatele utilizând un grafic de tip “pie”.

% Exemplu 6: Analiza Compoziției Materialelor

% Compoziția materială (elementele chimice și proporția lor)

```
elemente = {'Fe', 'C', 'Si', 'Mn', 'P', 'S'};
```

```
proportie = [78, 0.3, 2, 0.5, 0.1, 0.1];
```

% Calculul sumei proporțiilor

```
suma_proportii = sum(proportie);
```

% Calculul proporției procentuale

```
proportie_procentuala = (proportie / suma_proportii) * 100;
```

% Afișarea rezultatelor

```
fprintf('Compoziția materială:\n');
```

```
for i = 1:length(elemente)
```

```
    fprintf('%s: %.2f%%\n', elemente{i}, proportie_procentuala(i));
```

```
end
```

% Reprezentarea grafică a compoziției utilizând un grafic de tip pie

```
figure;
```

```
pie(proportie_procentuala, elemente);
```

```
title('Compoziția Materială');
```

Exemplu 7: Analiza Spectrului Materialelor

Acest script utilizează transformata Fourier pentru a analiza spectrul unei semnale provenite din caracterizarea unui material și reprezintă rezultatele utilizând un grafic de tip spectrogramă.

% Exemplu 7: Analiza Spectrului Materialelor

% Semnalul de caracterizare a materialului

```
t = 0:0.001:1; % Vectorul de timp
```

```

frecventa1 = 10; % Frecvența componentei 1 a semnalului
frecventa2 = 100; % Frecvența componentei 2 a semnalului
semnal = 2*sin(2*pi*frecventa1*t) + 1.5*sin(2*pi*frecventa2*t);

% Calculul spectrului semnalului utilizând transformata Fourier
spectru = abs(fft(semnal));

% Afișarea rezultatelor
figure;
subplot(2, 1, 1);
plot(t, semnal);
title('Semnalul de Caracterizare a Materialului');
xlabel('Timp (s)');
ylabel('Amplitudine');
subplot(2, 1, 2);
frecventa = 0:length(spectru)-1;
plot(frecventa, spectru);
title('Spectrul Semnalului');
xlabel('Frecvență (Hz)');
ylabel('Amplitudine');

```

Exemplu 8: Simulare Mecanică a Materialelor

Acest script simulează comportamentul mecanic al unui material într-un test de încovoiere și reprezintă curba de încărcare și deplasare utilizând un grafic de tip curba de încovoiere.

```

% Exemplu 8: Simulare Mecanică a Materialelor

% Parametrii materialului
modul_elasticitate = 2.1e5; % Modulul de elasticitate (Pa)
moment_inertie = 4.5e-4; % Momentul de inerție (m^4)
lungime_bara = 2; % Lungimea barei (m)

```

```

% Încărcare aplicată asupra barei
incarcare = 1000:100:5000; % Încărcare (N)

% Calculul deplasării
deplasare = (incarcare .* lungime_bara.^3) ./ (3 * modul_elasticitate * moment_inertie);

% Afișarea curbei de încovoiere
figure;
plot(incarcare, deplasare);
title('Curba de Încovoiere a Materialului');
xlabel('Încărcare (N)');
ylabel('Deplasare (m)');

```

Exemplu de Script Matlab pentru Calculul Proprietăților Materialelor

```

% Exemplu de Script Matlab pentru Calculul Proprietăților Materialelor

% Acest script Matlab demonstrează cum pot fi calculate anumite proprietăți ale
materialelor, folosind formule matematice și metode numerice specifice. Vom presupune
că avem datele de intrare pentru un material și dorim să obținem valori pentru
proprietățile sale.

% Date de intrare pentru exemplu:
densitate_material = 7850; % kg/m^3 (densitatea materialului)
viteza_sunet = 5000; % m/s (viteza sunetului în material)
modul_Young = 2e11; % Pa (modulul Young)
coeficient_Poisson = 0.3; % (coeficientul Poisson)

% Calculul proprietăților materialelor

% I. Viteza de propagare a undelor ultrasunete
viteza_ultrasunete = sqrt(modul_Young / densitate_material);

```

% 2. Modulul de rigiditate

$\text{modul_rigiditate} = \text{modul_Young} / (2 * (1 + \text{coeficient_Poisson}));$

% 3. Viteza de propagare a undelor de transversale

$\text{viteza_transversala} = \text{sqrt}(\text{modul_rigiditate} / \text{densitate_material});$

% 4. Coeficientul de comprimare

$\text{coeficient_comprimare} = 1 - (\text{viteza_transversala} / \text{viteza_sunet})^2;$

% 5. Modulul de elasticitate longitudinal

$\text{modul_elasticitate_longitudinal} = \text{densitate_material} * \text{viteza_ultrasunete}^2;$

% 6. Modulul de elasticitate transversal

$\text{modul_elasticitate_transversal} = \text{densitate_material} * \text{viteza_transversala}^2;$

% 7. Modulul de rezistență la încovoiere

$\text{modul_rezistenta_incovoiere} = (\text{modul_Young} * \text{densitate_material}) / (3 * (1 - 2 * \text{coeficient_Poisson}));$

% 8. Modulul de compresibilitate

$\text{modul_compresibilitate} = 1 / \text{viteza_ultrasunete}^2;$

% 9. Viteza de propagare a undelor de dilatare

$\text{viteza_dilatare} = \text{sqrt}((\text{modul_elasticitate_longitudinal} * (1 - \text{coeficient_Poisson})) / \text{densitate_material});$

% 10. Viteza de propagare a undelor de torsiune

$\text{viteza_torsiune} = \text{sqrt}(\text{modul_rigiditate} / \text{densitate_material});$

% 11. Coeficientul Poisson invers

$\text{coeficient_Poisson_invers} = (3 * \text{viteza_ultrasunete}^2 - 2 * \text{viteza_transversala}^2) / (2 * \text{viteza_ultrasunete}^2 - \text{viteza_transversala}^2);$

% 12. Viteza de propagare a undelor de compresiune

viteza_compresiune = sqrt(modul_compresibilitate / densitate_material);

% 13. Viteza de propagare a undelor de torsiune deosebită

viteza_torsiune_deosebita = sqrt(modul_rigiditate / (2 * densitate_material));

% 14. Viteza de propagare a undelor de dilatare radială

viteza_dilatare_radiala = sqrt((2 * modul_elasticitate_transversal) / densitate_material);

% 15. Modulul de rezistență la forfecare

modul_rezistenta_forfecare = viteza_torsiune * densitate_material;

% 16. Modulul de rezistență la comprimare

modul_rezistenta_comprimare = viteza_compresiune * densitate_material;

% 17. Coeficientul de Poisson majorat

coeficient_Poisson_majorat = (viteza_ultrasunete^2 - viteza_transversala^2) / (2 *
(viteza_ultrasunete^2 - viteza_transversala^2));

% 18. Viteza de propagare a undelor de torsiune radiale

viteza_torsiune_radiale = sqrt((modul_elasticitate_transversal) / densitate_material);

% 19. Coeficientul de dilatare termică

coeficient_dilatare_termica = (viteza_dilatare_radiala^2 - viteza_dilatare^2) / (2 *
viteza_dilatare_radiala^2);

% 20. Viteza de propagare a undelor de torsiune radiale deosebită

viteza_torsiune_radiale_deosebita = sqrt(modul_elasticitate_transversal /
densitate_material);

% Afișarea rezultatelor

```

disp('Proprietățile materialelor:');
disp(['Viteza de propagare a undelor ultrasunete: ', num2str(viteza_ultrasunete), ' m/s']);
disp(['Modulul de rigiditate: ', num2str(modul_rigiditate), ' Pa']);
disp(['Viteza de propagare a undelor de transversale: ', num2str(viteza_transversala), ' m/s']);
disp(['Coeficientul de comprimare: ', num2str(coeficient_comprimare)]);
disp(['Modulul de elasticitate longitudinal: ', num2str(modul_elasticitate_longitudinal), ' Pa']);
disp(['Modulul de elasticitate transversal: ', num2str(modul_elasticitate_transversal), ' Pa']);
disp(['Modulul de rezistență la încovoiere: ', num2str(modul_rezistenta_incovoiere), ' Pa']);
disp(['Modulul de compresibilitate: ', num2str(modul_compresibilitate), ' m^2/N']);
disp(['Viteza de propagare a undelor de dilatare: ', num2str(viteza_dilatare), ' m/s']);
disp(['Viteza de propagare a undelor de torsiune: ', num2str(viteza_torsiune), ' m/s']);
disp(['Coeficientul Poisson invers: ', num2str(coeficient_Poisson_invers)]);
disp(['Viteza de propagare a undelor de compresiune: ', num2str(viteza_compresiune), ' m/s']);
disp(['Viteza de propagare a undelor de torsiune deosebită: ',
num2str(viteza_torsiune_deosebita), ' m/s']);
disp(['Viteza de propagare a undelor de dilatare radială: ', num2str(viteza_dilatare_radiala), ' m/s']);
disp(['Modulul de rezistență la forfecare: ', num2str(modul_rezistenta_forfecare), ' Pa']);
disp(['Modulul de rezistență la comprimare: ', num2str(modul_rezistenta_comprimare), ' Pa']);
disp(['Coeficientul de Poisson majorat: ', num2str(coeficient_Poisson_majorat)]);
disp(['Viteza de propagare a undelor de torsiune radiale: ',
num2str(viteza_torsiune_radiale), ' m/s']);
disp(['Coeficientul de dilatare termică: ', num2str(coeficient_dilatare_termica)]);
disp(['Viteza de propagare a undelor de torsiune radiale deosebită: ',
num2str(viteza_torsiune_radiale_deosebita), ' m/s']);

% Sfârșitul scriptului

```

% Acest exemplu calculează o serie de proprietăți materiale pentru un material dat, folosind formule specifice și metode numerice. Valorile date de intrare sunt doar exemple și trebuie înlocuite cu date reale pentru un material specific. Formulele și metodele utilizate pot varia în funcție de proprietățile materialelor și de cerințele specifice ale aplicației. Rezultatele sunt afișate în consolă pentru a fi vizualizate.

LUCRAREA 6

6.1. Obiectivul lucrării

Această lucrare de laborator este concepută pentru a introduce conceptele de bază ale programării în Matlab. Programarea este un instrument esențial în înțelegerea și soluționarea problemelor practice din domeniul Științei și Ingineriei Materialelor, ajutând la analizarea datelor experimentale, modelarea comportamentului materialului și automatizarea proceselor complexe.

Importanța programării în Matlab în domeniul Științei și Ingineriei Materialelor constă în:

- **Analiza Datelor:** Matlab permite importul, manipularea și analiza datelor experimentale. Acest lucru este esențial pentru înțelegerea proprietăților materialelor și pentru luarea deciziilor informate în dezvoltarea și optimizarea materialelor.
- **Modelarea și Simularea:** Cu Matlab, puteți crea modele matematice pentru a simula comportamentul materialelor în diferite condiții. Acest lucru este valoros în proiectarea materialelor și în evaluarea performanței lor.
- **Automatizarea Proceselor:** Matlab poate fi folosit pentru a automatiza procesele de colectare și analiză a datelor, ceea ce economisește timp și reduce erorile umane.
- **Vizualizare și Comunicare:** Generarea de grafice și vizualizări este ușoară în Matlab, permițând comunicarea clară și eficientă a rezultatelor și a conceptelor.

Scopul acestui laborator este de a familiariza studenții cu conceptele de bază ale programării în Matlab și de a le oferi instrumentele necesare pentru a aborda probleme specifice din domeniul Științei și Ingineriei Materialelor.

Obiectivele specifice ale laboratorului includ:

- Înțelegerea sintaxei de bază a limbajului Matlab.
- Deprinderea conceptelor de structuri de control, cum ar fi construcții if, bucle for și while precum și construcții switch.
- Aplicarea acestor structuri de control pentru rezolvarea problemelor specifice din domeniul materialelor.
- Crearea de scripturi și funcții Matlab pentru automatizarea sarcinilor și proceselor.

6.2 Instrucțiunea "if"

Instrucțiunea "if" este una dintre cele mai fundamentale structuri de control în Matlab. Ea permite programatorului să creeze condiții și să ia decizii bazate pe aceste condiții. Cu ajutorul acestei instrucțiuni, puteți controla fluxul programului dvs., să executați anumite blocuri de cod numai dacă o anumită condiție este îndeplinită.

Structura de bază a unei instrucțiuni "if" arată astfel:

```
if (condiție)
    % Blocul de cod care se execută dacă condiția este adevărată
else
    % Blocul de cod care se execută dacă condiția este falsă
end
```

- "condiție" este o expresie sau o variabilă care poate fi evaluată ca adevărată (non-zero) sau falsă (zero). De exemplu, puteți verifica dacă două variabile sunt egale, dacă o variabilă este mai mică sau mai mare decât alta, sau oricare altă condiție logică.
- Blocul de cod dintre "if" și "else" este executat doar dacă condiția este adevărată.
- Blocul de cod dintre "else" și "end" este executat doar dacă condiția este falsă.

Puteți folosi "if" și "else" pentru a gestiona cazurile în care trebuie să luați decizii în codul dvs. de Matlab.

Exemple Practice

Verificarea unei condiții simple:

```
nota = 8;

if nota >= 5
    disp('Studentul a trecut examenul.');
```

else

```
    disp('Studentul a picat examenul.');
```

end

În acest exemplu, se verifică dacă nota unui student este mai mare sau egală cu 5. Dacă condiția este adevărată, se afișează că studentul a trecut examenul, altfel se afișează că a picat.

Verificarea mai multor condiții:

```
temperatura = 28;

if temperatura <= 0
    disp('Este foarte frig.');
```

elseif temperatura <= 15

```
    disp('Este moderat de frig.');
```

elseif temperatura <= 25

```
    disp('Este temperat.');
```

else

```
    disp('Este cald.');
```

end

În acest exemplu, se verifică temperatura și se afișează un mesaj corespunzător în funcție de intervalul în care se încadrează.

Verificarea unui șir de caractere:

```
culoare_preferata = 'albastru';

if strcmp(culoare_preferata, 'rosu')
    disp('Culoarea preferată este roșu.');
```

elseif strcmp(culoare_preferata, 'verde')

```
    disp('Culoarea preferată este verde.');
```

elseif strcmp(culoare_preferata, 'albastru')

```
    disp('Culoarea preferată este albastru.');
```

else

```
    disp('Culoarea preferată nu este recunoscută.');
```

end

În acest exemplu, se verifică culoarea preferată și se afișează un mesaj corespunzător.

Instrucțiunea "if" vă permite să creați fluxuri de lucru complexe în Matlab, în funcție de condițiile specificate. Acesta este un instrument fundamental în programarea în Matlab și este

utilizat pe scară largă pentru a controla execuția programelor dvs. în funcție de datele de intrare sau alte condiții logice.

6.3 Instrucțiunea "for"

Instrucțiunea "for" este o altă structură de control importantă în Matlab. Aceasta permite repetarea unui bloc de cod de mai multe ori, ceea ce este util atunci când doriți să iterați printr-un set de date sau să efectuați aceeași acțiune în mod repetat.

Structura de bază a unei instrucțiuni "for" arată astfel:

```
for variabila = valoare_iniciala:pas:valoare_finala
    % Blocul de cod care se repetă
end
```

- "variabila" este o variabilă care va fi folosită pentru a itera prin valori. Această variabilă poate fi folosită în blocul de cod pentru a efectua operații care depind de valoarea sa.
- "valoare_iniciala" este valoarea inițială a variabilei.
- "pas" reprezintă valoarea cu care variabila se incrementează la fiecare iterație.
- "valoare_finala" este valoarea la care se oprește iterația.

Structura "for" vă permite să creați bucle care rulează de un număr specific de ori sau până când o condiție specifică este îndeplinită.

Example Practice

Iterarea prin valori de la 1 la 5:

```
for i = 1:5
    disp(i);
end
```

Acest exemplu va afișa numerele de la 1 la 5 în linie.

Iterarea prin elemente dintr-un vector:

```
vector = [2, 4, 6, 8, 10];

for element = vector
```

```
disp(element);  
end
```

Aici, bucla "for" iterează prin elementele vectorului și le afișează.

Suma numerelor de la 1 la 100:

```
suma = 0;  
  
for i = 1:100  
    suma = suma + i;  
end  
disp(['Suma este: ', num2str(suma)]);
```

Acest exemplu utilizează o variabilă pentru a calcula suma numerelor de la 1 la 100.

Instrucțiunea "for" este esențială în programarea Matlab atunci când aveți nevoie să efectuați aceeași acțiune de mai multe ori sau când doriți să iterați prin date. Aceasta vă permite să automatizați sarcini repetitive și să lucrați eficient cu datele din Matlab.

6.4 Instrucțiunea "while" în Matlab

Instrucțiunea "while" este o altă structură de control importantă în Matlab. Această instrucțiune permite repetarea unui bloc de cod atâta timp cât o anumită condiție este adevărată.

Structura de bază a unei instrucțiuni "while" arată astfel:

```
while conditie  
    % Blocul de cod care se repetă  
end
```

- "condiție" este o expresie logică. Atâta timp cât această condiție este adevărată (adică are valoarea 1 sau true), blocul de cod din interiorul buclei "while" se va repeta.

Exemple Practice

Contorizarea până la 5:

```
contor = 1;

while contor <= 5
    disp(contor);
    contor = contor + 1;
end
```

Acest exemplu va afișa numerele de la 1 la 5, folosind o buclă "while" pentru a realiza acest lucru.

Calcularea puterii a 2-a:

```
putere = 1;
exponent = 0;

while exponent < 5
    putere = putere * 2;
    exponent = exponent + 1;
end

disp(['2 la puterea 5 este: ', num2str(putere)]);
```

Aici, bucla "while" este folosită pentru a calcula 2 la puterea 5.

Găsirea celei mai mici puteri a lui 3 mai mari decât 50:

```
putere = 3;
exponent = 1;

while putere <= 50
    putere = 3^exponent;
    exponent = exponent + 1;
end
```

```
disp(['Cea mai mică putere a lui 3 mai mare decât 50 este: 3 la puterea ',  
num2str(exponent - 1)]);
```

Acest exemplu folosește o buclă "while" pentru a găsi cea mai mică putere a lui 3 care este mai mare decât 50.

Instrucțiunea "while" este utilă atunci când nu știți în avans câte iterații sunt necesare și doriți să repetați un bloc de cod până când o anumită condiție este îndeplinită. Trebuie să fiți atenți să evitați buclele infinite, asigurându-vă că condiția din instrucțiunea "while" va deveni falsă la un moment dat.

6.5 Instrucțiunea "switch"

Instrucțiunea "switch" este o altă structură de control care poate fi folosită pentru a face decizii în Matlab. Aceasta este folosită atunci când există mai multe cazuri sau opțiuni, iar programul trebuie să aleagă unul dintre ele în funcție de valoarea unei expresii.

Structura de bază a unei instrucțiuni "switch" arată astfel:

```
switch expresie  
    case valoare1  
        % Blocul de cod pentru prima opțiune  
    case valoare2  
        % Blocul de cod pentru a doua opțiune  
    case valoare3  
        % Blocul de cod pentru a treia opțiune  
    otherwise  
        % Blocul de cod implicit, dacă niciuna dintre opțiuni nu se potrivește  
end
```

- "expresie" este o valoare sau o variabilă pe baza căreia se iau decizii. Aceasta este evaluată, iar programul se va uita la fiecare "case" pentru a vedea dacă se potrivește cu "valoarea1", "valoarea2", etc.
- "case" specifică cazurile posibile, iar programul va executa blocul de cod corespunzător cazului care se potrivește cu "expresia".

- "otherwise" este opțional și reprezintă un caz implicit. Dacă niciuna dintre "case"-urile anterioare nu se potrivește, se va executa acest bloc de cod.

Exemple Practice

Ziua săptămânii:

```
zi = input('Introduceți un număr de la 1 la 7: ');

switch zi
    case 1
        disp('Luni');
    case 2
        disp('Marți');
    case 3
        disp('Miercuri');
    case 4
        disp('Joi');
    case 5
        disp('Vineri');
    case 6
        disp('Sâmbătă');
    case 7
        disp('Duminică');
    otherwise
        disp('Numărul introdus nu este valid pentru o zi a săptămânii.');
```

end

Aici, programul afișează ziua corespunzătoare numărului introdus de la 1 la 7.

Convertirea scorului în calificativ:

```
scor = input('Introduceți un scor între 0 și 100: ');

switch true
    case scor >= 90
```

```

    calificativ = 'A';
case scor >= 80
    calificativ = 'B';
case scor >= 70
    calificativ = 'C';
case scor >= 60
    calificativ = 'D';
otherwise
    calificativ = 'F';
end

disp(['Calificativul pentru scorul ', num2str(scor), ' este: ', calificativ]);

```

Acest exemplu convertește un scor numeric într-un calificativ (A, B, C, D sau F) folosind o instrucțiune "switch".

Operații matematice:

```

a = 5;
b = 10;
operatie = 'adunare';

switch operatie
case 'adunare'
    rezultat = a + b;
case 'scadere'
    rezultat = a - b;
case 'inmultire'
    rezultat = a * b;
case 'impartire'
    rezultat = a / b;
otherwise
    disp('Operație necunoscută.');
```

```
end
```

```
disp(['Rezultatul pentru operația de ', operatie, ' este: ', num2str(rezultat)]);
```

Acest exemplu demonstrează cum se poate utiliza o instrucțiune "switch" pentru a efectua diferite operații matematice în funcție de valoarea variabilei "operație".

Instrucțiunea "switch" este utilă atunci când trebuie să luați decizii bazate pe valoarea unei expresii și să efectuați diferite acțiuni în funcție de aceasta. Poate face codul mai ușor de citit și de înțeles atunci când există multe opțiuni.

6.6. Exemple

- Verificarea rezistenței unui material

Să presupunem că avem un set de date care reprezintă rezistența la tracțiune a unor mostre de material. Vrem să verificăm dacă aceste mostre sunt rezistente și să afișăm un mesaj corespunzător pentru fiecare mostră.

```
% Exemplu de utilizare a instructiunii if pentru verificarea rezistentei unui material
```

```
rezistenta = [350, 410, 390, 420, 380];
```

```
limita_rezistenta = 400;
```

```
for i = 1:length(rezistenta)
```

```
    if rezistenta(i) >= limita_rezistenta
```

```
        disp(['Proba ', num2str(i), ' este rezistenta.']);
```

```
    else
```

```
        disp(['Proba ', num2str(i), ' nu este rezistenta.']);
```

```
    end
```

```
end
```

- Calculul conductivității termice medii

Să presupunem că avem un set de date care reprezintă conductivitatea termică a diferitelor materiale. Dorim să calculăm conductivitatea termică medie a acestor materiale și să o afișăm.

% Exemplu de utilizare a instructiunii for pentru calculul conductivitatii termice medii

```
conductivitate_termica = [0.5, 0.6, 0.7, 0.4, 0.8];
numar_materiale = length(conductivitate_termica);
suma_conductivitati = 0;

for i = 1:numar_materiale
    suma_conductivitati = suma_conductivitati + conductivitate_termica(i);
end

conductivitate_medie = suma_conductivitati / numar_materiale;
disp(['Conductivitatea termica medie a materialelor este: ',
num2str(conductivitate_medie)]);
```

- Calculul vitezei de curgere

În Știința și Ingineria Materialelor, viteza de curgere este o proprietate importantă pentru a evalua comportamentul materialelor sub tensiune. Să presupunem că avem un vector care reprezintă stresul aplicat asupra unui material și un alt vector care reprezintă deformarea sa corespunzătoare. Dorim să calculăm viteza de curgere pentru acest material.

% Exemplu de utilizare a instrucțiunii while pentru calculul vitezei de curgere

```
stres = [50, 70, 80, 90, 100];
deformare = [0.02, 0.03, 0.035, 0.04, 0.045];
viteza_de_curgere = 0;
i = 1;

while deformare(i) < 0.045
    viteza_de_curgere = stres(i);
    i = i + 1;
end

disp(['Viteza de curgere a materialului este: ', num2str(viteza_de_curgere)]);
```

- Determinarea modulului de elasticitate

În acest exemplu, dorim să calculăm modulul de elasticitate pentru un material folosind legea lui Hooke, care stabilește relația dintre tensiune și deformare elastică.

```
% Exemplu de utilizare a instructiunii switch pentru determinarea modulului de elasticitate
stres = [100, 150, 200, 250, 300];
deformare = [0.01, 0.015, 0.02, 0.025, 0.03];

switch_case = 2;
switch switch_case
    case 1
        % Determinarea modulului de elasticitate pentru primul caz
        modul_elasticitate = stres(2) / deformare(2);
    case 2
        % Determinarea modulului de elasticitate pentru al doilea caz
        modul_elasticitate = stres(4) / deformare(4);
    otherwise
        disp('Nu s-a gasit un caz corespunzator.');
```

```
end

disp(['Modulul de elasticitate al materialului este: ', num2str(modul_elasticitate)]);
```

- Analiza durabilității materialelor

Pentru această aplicație, vom considera un set de date care conține date experimentale privind rezistența la coroziune a mai multor materiale expuse la diferite medii. Scopul este să analizăm durabilitatea acestor materiale și să identificăm cel mai rezistent material la coroziune.

```
% Exemplu de analiză a durabilității materialelor
rezistenta_coroziune = [50, 70, 30, 80, 90];
nume_materiale = {'Aluminiu', 'Oțel', 'Cupru', 'Fier', 'Titan'};

[materiale_rezistente, index_materiale_rezistente] = max(rezistenta_coroziune);
```

```
disp(['Materialul cel mai rezistent la coroziune este ', nume_materiale  
{index_materiale_rezistente}, ' cu o rezistență de ', num2str(materiale_rezistente), '  
unități.']);
```

6.7. Aplicații

1. Calculați densitatea unui material cu masa de 150 kg și volumul de 0.05 m^3 .
2. Determinați valoarea medie a proprietăților mecanice ale unui set de 10 mostre de material.
3. Găsiți coeficientul de dilatare termică al unui material, cunoscând lungimea inițială și variația lungimii sub acțiunea temperaturii.
4. Verificați dacă un material îndeplinește criteriul de rezistență la curgere, cunoscând limitele de curgere și tensiunile maxime admise.
5. Calculați energiile potențiale și cinetice ale unui material sub acțiunea unei forțe externe.
6. Modelați comportamentul termic al unui material într-un mediu cu temperatură variabilă.
7. Determinați gradul de cristalinitate al unui material cunoscând datele de difracție de raze X.
8. Găsiți masa atomică medie a unui material cunoscând izotopele sale și proporțiile acestora.
9. Estimați proprietățile termoelectrice ale unui material în funcție de structura sa cristalină.
10. Calculați rezistența termică a unui material într-un sistem multistrat.
11. Verificați dacă un material este feromagnetic sau antiferomagnetic în funcție de comportamentul său la aplicarea unui câmp magnetic.
12. Calculați coeficientul de absorbție a radiației electromagnetice pentru un anumit material.
13. Estimați viteza de propagare a undelor acustice într-un material cunoscând modulul de elasticitate și densitatea sa.
14. Analizați comportamentul termic al unui material sub acțiunea radiației solare.

15. Determinați coeficientul de dilatare termică a unui material în funcție de schimbările de temperatură.
16. Modelați difuzia atomilor într-un material în timpul tratamentelor termice.
17. Estimați conductivitatea electrică a unui material în funcție de densitatea de curent și diferența de potențial aplicate.
18. Verificați dacă un material are proprietăți semiconductoare în funcție de banda sa interzisă.
19. Analizați comportamentul unui material sub acțiunea unui câmp magnetic variabil în timp.

Bibliografie

1. Matlab Documentation, The MathWorks Inc., <https://www.mathworks.com/help/matlab/>
2. Gander, W., Gander, M. J., Kwok, F., "Scientific Computing with MATLAB and Octave," Texts in Computational Science and Engineering, Springer, 2017.
3. Chapra, S. C., Canale, R. P., "Numerical Methods for Engineers," 7th Edition, McGraw-Hill, 2014.
4. Ashby, M. F., Jones, D. R. H., "Engineering Materials I: An Introduction to Properties, Applications, and Design," 4th Edition, Butterworth-Heinemann, 2012.
5. Callister Jr., W. D., Rethwisch, D. G., "Materials Science and Engineering: An Introduction," 10th Edition, Wiley, 2017.

LUCRAREA I

I.1. Obiectivele lucrării

În această lucrare este prezentată structura de bază a unei pagini HTML, modul de formatare a textului într-o pagină web, listele ordonate și neordonate precum și ancorele care se pot utiliza. De asemenea, sunt prezentate și o serie de exemple de utilizare a elementelor prezentate.

I.2. Ce este HTML și de ce este important în dezvoltarea web?

HTML, prescurtare de la HyperText Markup Language (Limbaajul de marcare pentru hipertext), este un limbaj de marcare utilizat pentru crearea și structurarea conținutului paginilor web. A fost inventat în anii '90 de către Tim Berners-Lee, considerat unul dintre părinții World Wide Web-ului, și a devenit temelia pe care este construită întreaga arhitectură a internetului.

Prin intermediul HTML, dezvoltatorii pot crea pagini web care pot conține texte, imagini, link-uri, videoclipuri și alte elemente multimedia, precum și formulare pentru colectarea datelor de la utilizatori. HTML utilizează o sintaxă simplă și bazată pe etichete, cunoscute și ca tag-uri, care definesc structura și semantica conținutului paginii.

Importanța HTML în dezvoltarea web provine din faptul că reprezintă o componentă esențială pentru afișarea și organizarea informațiilor pe internet. Prin intermediul acestui limbaj, paginile web pot fi interpretate și afișate corect pe diverse dispozitive, indiferent de sistemul de operare, dimensiunea ecranului sau browser-ul utilizat.

În contextul evoluției tehnologiei web, HTML a fost mereu adaptat pentru a permite noi funcționalități și caracteristici avansate, îmbogățind experiența utilizatorilor online. Acest limbaj continuă să joace un rol crucial în cadrul dezvoltării web și reprezintă baza pe care se construiesc celelalte tehnologii web, cum ar fi CSS (Cascading Style Sheets) pentru stilizarea paginilor și JavaScript pentru interactivitate.

I.3. Structura de bază a unei pagini HTML

I.3.1. Documentul HTML, tag-urile de bază și structura acestora

Un document HTML reprezintă pagina web însăși și este construit folosind diferite tag-uri (etichete) care definesc structura și semantica conținutului. O pagină HTML începe cu un tag '`<html>`' care marchează începutul documentului și se termină cu tag-ul '`</html>`' care semnalează sfârșitul acestuia. În interiorul acestui tag '`<html>`', există două secțiuni principale: '`<head>`' și '`<body>`'.

- Tag-ul '`<head>`': Acesta conține informații despre documentul HTML, dar nu va fi afișat direct în browser. În schimb, aceasta secțiune este folosită pentru a furniza metadate, stiluri CSS, legături către fișiere externe și alte instrucțiuni speciale pentru pagină.
- Tag-ul '`<body>`': Aici se află conținutul vizibil al paginii, care va fi afișat utilizatorilor. Textul, imagini, videoclipuri și alte elemente media se introduc în această secțiune. Totodată, aici sunt adăugate și structurile de bază ale paginii, cum ar fi paragrafele, listele, link-urile și altele.

Exemplu de structură de bază a unei pagini HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Titlul paginii</title>
</head>
<body>

<!-- Conținutul vizibil al paginii -->
  <h1>Titlul paginii</h1>
  <p>Aceasta este o pagină web simplă.</p>

</body>
</html>
```

I.3.2. Elementele head și body

- Tag-ul '`<head>`'

Secțiunea '`<head>`' este folosită pentru a furniza informații și instrucțiuni pentru pagină, dar conținutul său nu va fi afișat în corpul paginii. Acesta conține, de obicei, următoarele elemente:

- ‘<title>’: Acest tag definește titlul paginii, care va apărea în bara de titlu a browserului și poate fi utilizat de motoarele de căutare pentru indexare.
- ‘<meta>’: Aceste tag-uri sunt utilizate pentru a furniza metadate despre pagină, cum ar fi setarea caracterului de codificare, descrieri, cuvinte cheie și alte informații relevante pentru SEO (Search Engine Optimization).
- ‘<link>’: Acest tag este folosit pentru a lega pagină la alte resurse externe, cum ar fi fișiere CSS sau fonturi. Prin intermediul acestui tag, paginile pot fi stilizate și formate cu ajutorul stilurilor definite în fișiere separate.
- ‘<script>’: Acest tag este folosit pentru a adăuga cod JavaScript sau alte scripturi în pagină.

Exemplu de utilizare a tag-urilor în secțiunea ‘<head>’:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Titlul paginii</title>
  <meta charset="UTF-8">
  <meta name="description" content="Aceasta este o pagină exemplu">
  <meta name="keywords" content="HTML, pagină web, exemplu">
  <link rel="stylesheet" href="style.css">
  <script src="script.js" defer></script>
</head>
<body>
  <!-- Conținutul vizibil al paginii -->
  <h1>Titlul paginii</h1>
  <p>Aceasta este o pagină web simplă.</p>
</body>
</html>
```

- Tag-ul ‘<body>’

Secțiunea ‘<body>’ conține tot conținutul vizibil al paginii web. Aici sunt introduse toate elementele care vor fi afișate utilizatorilor. Aceasta secțiune include text, imagini, videoclipuri, link-uri, tabele, formulare și orice alt conținut interactiv.

Exemplu de utilizare a tag-ului ‘<body>’:

```
<!DOCTYPE html>
<html>
<head>
  <title>Exemplu Pagina</title>
</head>
<body>
  <!-- Conținutul vizibil al paginii -->
```

```
<h1>Titlul paginii</h1>
<p>Aceasta este o pagină web simplă.</p>

</body>
</html>
```

Secțiunile ‘<head>’ și ‘<body>’ reprezintă fundația unei pagini HTML și asigură organizarea corectă a conținutului și a resurselor asociate, contribuind astfel la o experiență coerentă și optimă pentru utilizatori.

I.4. Tag-uri de bază în HTML

I.4.1. Paragrafe și formatarea textului

- Tag-uri de paragraf

Tag-ul ‘<p>’ este utilizat pentru a crea paragrafe în cadrul paginilor web. Paragrafele sunt elementele de bază pentru organizarea și structurarea textului într-o manieră coerentă și ușor de citit.

Exemplu de utilizare a tag-ului ‘<p>’:

```
<!DOCTYPE html>
<html>
<head>
  <title>Exemplu Pagina</title>
</head>
<body>
  <p>Aceasta este o pagină web simplă, creată cu HTML.</p>
  <p>HTML (HyperText Markup Language) este limbajul folosit pentru construirea
  paginilor web.</p>
</body>
</html>
```

- Formatarea textului

Pentru a adăuga formatare textului, HTML utilizează o serie de tag-uri speciale. Printre cele mai frecvent utilizate se numără:

- ‘’ sau ‘’: Textul inclus între aceste tag-uri va fi afișat boldat (cu o greutate mai mare a fontului). De obicei, ‘’ este preferat în locul ‘’ pentru a evidenția importanța semantică a textului, în locul stilului său.
- ‘’ sau ‘<i>’: Textul inclus între aceste tag-uri va fi afișat înclinat (italic). Asemănător cu tag-urile pentru text boldat, ‘’ este preferat în favoarea ‘<i>’ pentru a indica importanța semantică a textului.

- ‘<u>’: Textul inclus între acest tag va fi subliniat.
- ‘<s>’: Textul inclus între acest tag va fi tăiat cu o linie (tăiat).
- ‘<sup>’ și ‘<sub>’: Aceste tag-uri sunt utilizate pentru a crea texte supra și subscrise, respectiv.

Exemplu de formatare a textului:

```
<!DOCTYPE html>
<html>
<head>
  <title>Exemplu Pagina</title>
</head>
<body>
  <p><strong>Acesta este un text important.</strong></p>
  <p><em>Textul acesta este italicizat.</em></p>
  <p><u>Acest text este subliniat.</u></p>
  <p><s>Acest text este tăiat.</s></p>
  <p>Textul poate conține<sup>indice superior</sup> și <sub>indice inferior</sub>.</p>
</body>
</html>
```

1.4.2. Listele neordonate și ordonate

- Listele neordonate

Listele neordonate sunt utilizate pentru a grupa elemente într-o listă fără o ordine specifică. Ele sunt reprezentate prin tag-ul ‘’ și fiecare element al listei este definit prin tag-ul ‘’.

Exemplu de listă neordonată:

```
<!DOCTYPE html>
<html>
<head>
  <title>Exemplu Pagina</title>
</head>
<body>
  <ul>
    <li>Element 1</li>
    <li>Element 2</li>
    <li>Element 3</li>
  </ul>
</body>
</html>
```

- Element 1
- Element 2
- Element 3

- Listele ordonate

Listele ordonate sunt utilizate pentru a grupa elemente într-o listă cu o ordine specifică. Ele sunt reprezentate prin tag-ul '', iar fiecare element al listei este definit prin tag-ul ''.

Exemplu de listă ordonată:

```
<!DOCTYPE html>
<html>
<head>
  <title>Exemplu Pagina</title>
</head>
<body>
  <ol>
    <li>Pasul 1</li>
    <li>Pasul 2</li>
    <li>Pasul 3</li>
  </ol>
</body>
</html>
```

1. Pasul 1
2. Pasul 2
3. Pasul 3

1.4.3. Link-uri și ancorări

- Tag-ul '<a>'

Tag-ul '<a>' este utilizat pentru a crea link-uri către alte pagini web, fișiere, adrese de email sau alte resurse online. Atributul 'href' al acestui tag specifică adresa URL către care se face legătura.

Exemplu de utilizare a tag-ului '<a>' pentru un link către o altă pagină web:

```
<!DOCTYPE html>
<html>
<head>
  <title>Exemplu Pagina</title>
</head>
<body>
  <p>Link către <a href="https://www.exemplu.com">Exemplu Website</a></p>
</body>
</html>
```

Link către [Exemplu Website](https://www.exemplu.com)

- Ancorări (Link-uri interne):

Ancorările, cunoscute și ca link-uri interne, permit crearea de legături între secțiuni diferite ale aceleiași pagini web. Pentru a crea o ancoră, se utilizează atributul 'id' pentru a marca elementul către care se va face referință.

Exemplu de ancoră:

```
<!DOCTYPE html>
<html>
<head>
```

```

<title>Exemplu Pagina</title>
</head>
<body>
  <h2 id="sectiune1">Secțiunea 1</h2>
  <p>Aceasta este conținutul secțiunii 1.</p>

  <h2 id="sectiune2">Secțiunea 2</h2>
  <p>Aceasta este conținutul secțiunii 2.</p>

  <p><a href="#sectiune1">Accesați Secțiunea 1</a></p>
  <p><a href="#sectiune2">Accesați Secțiunea 2</a></p>
</body>
</html>

```

Secțiunea 1

Aceasta este conținutul secțiunii 1.

Secțiunea 2

Aceasta este conținutul secțiunii 2.

[Accesați Secțiunea 1](#)

În acest exemplu, link-urile către "Secțiunea 1" și "Secțiunea 2" vor duce utilizatorul la secțiunile corespunzătoare din cadrul aceleiași pagini web.

I.5. Exemple practice

a) Crearea unui document HTML simplu cu text și imagini.

```

<!DOCTYPE html>
<html>
<head>
  <title>Pagina Exemplu</title>
</head>
<body>
  <h1>Bun venit pe Pagina Exemplu!</h1>
  <p>Aceasta este o pagină simplă creată folosind HTML.</p>
  
  <p>Pentru mai multe informații, vizitați <a
href="https://www.exemplu.com">Exemplu Website</a>.</p>
</body>
</html>

```

Bun venit pe Pagina Exemplu!

Aceasta este o pagină simplă creată folosind HTML.



Pentru mai multe informații, vizitați [Exemplu Website](#).

b) Adăugarea unui meniu de navigare cu link-uri interne.

```
<!DOCTYPE html>
<html>
<head>
  <title>Pagina cu Meniu de Navigare</title>
</head>
<body>
  <h1>Meniu</h1>
  <ul>
    <li><a href="#sectiune1">Secțiunea 1</a></li>
    <li><a href="#sectiune2">Secțiunea 2</a></li>
    <li><a href="#sectiune3">Secțiunea 3</a></li>
  </ul>

  <h2 id="sectiune1">Secțiunea 1</h2>
  <p>Aici putem adăuga conținut pentru secțiunea 1.</p>

  <h2 id="sectiune2">Secțiunea 2</h2>
  <p>Conținutul secțiunii 2 va fi adăugat aici.</p>

  <h2 id="sectiune3">Secțiunea 3</h2>
  <p>Conținutul secțiunii 3 va fi inserat în acest loc.</p>
</body>
</html>
```

Meniu

- [Secțiunea 1](#)
- [Secțiunea 2](#)
- [Secțiunea 3](#)

Secțiunea 1

Aici putem adăuga conținut pentru secțiunea 1.

Secțiunea 2

Conținutul secțiunii 2 va fi adăugat aici.

Secțiunea 3

Conținutul secțiunii 3 va fi inserat în acest loc.

c) Structurarea conținutului folosind liste.

```
<!DOCTYPE html>
<html>
<head>
  <title>Structurarea Conținutului cu Liste</title>
</head>
<body>
  <h1>Lista Neordonată</h1>
  <ul>
    <li>Element 1</li>
    <li>Element 2</li>
    <li>Element 3</li>
  </ul>

  <h1>Lista Ordonată</h1>
  <ol>
    <li>Pas 1</li>
    <li>Pas 2</li>
    <li>Pas 3</li>
  </ol>

  <h1>Listă de Definiții</h1>
  <dl>
    <dt>HTML</dt>
    <dd>HyperText Markup Language</dd>
    <dt>CSS</dt>
    <dd>Cascading Style Sheets</dd>
  </dl>
</body>
</html>
```

```
</dl>
</body>
</html>
```

Lista Neordonată

- Element 1
- Element 2
- Element 3

Lista Ordonată

1. Pas 1
2. Pas 2
3. Pas 3

Listă de Definiții

HTML

HyperText Markup Language

CSS

Cascading Style Sheets

d) Realizarea unei pagini web personale pentru a prezenta pasiuni și hobby-uri.

```
<!DOCTYPE html>
<html>
<head>
  <title>Pagina Mea Personală</title>
</head>
<body>
  <h1>Bună ziua!</h1>
  <h2>Despre Mine</h2>
  <p>Nume: John Doe</p>
  <p>Studii: Student la Facultatea de Știința și Ingineria Materialelor</p>
  <p>Email: john.doe@example.com</p>

  <h2>Pasiuni și Hobby-uri</h2>
  <h3>Hobby-ul 1 - Fotografie</h3>
  <p>Îmi place să explorez lumea înconjurătoare și să prind momente frumoase
în fotografii.</p>
  
```

```

<h3>Hobby-ul 2 - Muzică</h3>
<p>Mă bucur să cânt la chitară și să explorez diverse genuri muzicale.</p>


<h3>Hobby-ul 3 - Călătorii</h3>
<p>Îmi place să călătoresc și să descopăr culturi și locuri noi.</p>


<h2>Contact</h2>
<p>Pentru a mă contacta, trimiteți un email la adresa de mai sus.</p>
</body>
</html>

```

e) Crearea unei pagini pentru o prezentare științifică cu imagini și link-uri utile.

```

<!DOCTYPE html>
<html>
<head>
  <title>Prezentare Științifică</title>
</head>
<body>
  <h1>Titlul Prezentării Științifice</h1>
  <h2>Introducere</h2>
  <p>O scurtă introducere despre subiectul prezentat și de ce este important.</p>

  <h2>Explicații și Detalii</h2>
  <p>Explicații detaliate și informații relevante despre subiect.</p>
  
  <p>Pentru o înțelegere mai bună, vedeți și acest <a href="https://www.exemplu.com/videoclip">videoclip</a>.</p>

  <h2>Studii de Caz</h2>
  <p>Prezentarea unor studii de caz relevante din domeniu.</p>
  

  <h2>Concluzii</h2>
  <p>Concluziile finale și importanța subiectului prezentat.</p>

  <h2>Surse Utile</h2>
  <ul>
    <li><a href="https://www.exemplu.com/sursa1">Sursă 1</a></li>
    <li><a href="https://www.exemplu.com/sursa2">Sursă 2</a></li>
    <li><a href="https://www.exemplu.com/sursa3">Sursă 3</a></li>
  </ul>
</body>
</html>

```

Aceste exemple oferă o privire asupra modului de utilizare a diferitelor tag-uri HTML pentru a crea pagini web cu diferite elemente și structuri. În primul exemplu, am creat o pagină simplă cu text și imagini. În al doilea exemplu, am adăugat un meniu de navigare cu link-uri interne care permit utilizatorilor să sari direct la anumite secțiuni ale paginii. În al treilea exemplu, am arătat cum putem structura conținutul folosind liste neordonate, ordonate și de definiții.

În următoarele două exemple sunt prezentate idei pentru dezvoltarea unor pagini web cu conținut relevant și organizat în mod coerent. Aceste pagini pot fi adaptate și extinse în funcție de interesul și subiectul specific al fiecărui student. Astfel, vor avea oportunitatea de a-și pune în valoare datele științifice sau hobby-urile personale, oferind utilizatorilor o experiență informativă și plăcută pe paginile create.

1.6. Aplicații

1. Creați o pagină web simplă cu un titlu și un paragraf de prezentare.
2. Adăugați o imagine reprezentativă în pagină și oferiți un text alternativ pentru aceasta.
3. Realizați un link către o altă pagină web externă.
4. Adăugați un link intern către o secțiune specifică din aceeași pagină.
5. Creați o listă neordonată cu cel puțin 5 elemente relevante.
6. Realizați o listă ordonată cu cel puțin 3 pași de urmat.
7. Folosiți liste de definiții pentru a descrie termeni relevanți dintr-un domeniu de interes.
8. Adăugați o secțiune de prezentare pentru un hobby personal, folosind imagini și text.
9. Structurați informațiile despre o călătorie recentă folosind paragrafe și imagini.
10. Adăugați un antet și un subsol la pagina web cu informații relevante

Bibliografie

1. Mozilla Developer Network (MDN) - HTML: (<https://developer.mozilla.org/ro/docs/Web/HTML>)
2. W3Schools - HTML Tutorial: (<https://www.w3schools.com/html/>)
3. HTML.com - HTML Tutorial: (<https://html.com/>)
4. Codurile HTML - Tutoriale și Resurse: (<https://coduri-html.com/>)

5. "Head First HTML and CSS" de Elisabeth Robson și Eric Freeman, 2012, Publisher(s): O'Reilly Media, Inc., ISBN: 9780596159900

6. "Learning Web Design" de Jennifer Robbins, 2018, Publisher(s): O'Reilly Media, Inc., ISBN: 9781491960202

Rezolvare aplicații

Problemă 1: Creați o pagină web simplă cu un titlu și un paragraf de prezentare.

```
<!DOCTYPE html>
<html>
<head>
  <title>Titlul Paginii</title>
</head>
<body>
  <h1>Bună ziua!</h1>
  <p>Salutare! Bine ai venit pe pagina mea web. Aceasta este o pagină simplă
creată pentru rezolvarea primei probleme din Capitolul VI.</p>
</body>
</html>
```

Problemă 2: Adăugați o imagine reprezentativă în pagină și oferiți un text alternativ pentru aceasta.

```
<!DOCTYPE html>
<html>
<head>
  <title>Titlul Paginii</title>
</head>
<body>
  <h1>Pagina cu Imagine</h1>
  
  <p>Text alternativ pentru imagine: O imagine cu un peisaj frumos.</p>
</body>
</html>
```

Problemă 3: Realizați un link către o altă pagină web externă.

```
<!DOCTYPE html>
<html>
<head>
  <title>Titlul Paginii</title>
</head>
<body>
  <h1>Link către Pagină Externă</h1>
  <p>Următorul link va deschide o pagină web externă într-o fereastră
nouă:</p>
```

```
<a href="https://www.exemplu.com" target="_blank">Vizitați Exemplu.com</a>
</body>
</html>
```

Problemă 4: Adăugați un link intern către o secțiune specifică din aceeași pagină.

```
<!DOCTYPE html>
<html>
<head>
  <title>Titlul Paginii</title>
</head>
<body>
  <h1>Link intern către Secțiune</h1>
  <p>Următorul link vă va duce la o secțiune specifică din aceeași pagină:</p>
  <a href="#despre">Salt la secțiunea Despre</a>

  <h2 id="despre">Despre</h2>
  <p>Aceasta este secțiunea Despre, unde puteți găsi informații relevante
despre subiect.</p>
</body>
</html>
```

Problemă 5: Creați o listă neordonată cu cel puțin 5 elemente relevante.

```
<!DOCTYPE html>
<html>
<head>
  <title>Titlul Paginii</title>
</head>
<body>
  <h1>Listă Neordonată</h1>
  <ul>
    <li>Element 1</li>
    <li>Element 2</li>
    <li>Element 3</li>
    <li>Element 4</li>
    <li>Element 5</li>
  </ul>
</body>
</html>
```

Problemă 6: Realizați o listă ordonată cu cel puțin 3 pași de urmat.

```
<!DOCTYPE html>
<html>
<head>
  <title>Titlul Paginii</title>
</head>
<body>
  <h1>Listă Ordonată</h1>
```

```

<ol>
  <li>Pasul 1: Pregătiți materialele necesare.</li>
  <li>Pasul 2: Urmați instrucțiunile din ghidul de utilizare.</li>
  <li>Pasul 3: Finalizați procesul și verificați rezultatele.</li>
</ol>
</body>
</html>

```

Problemă 7: Folosiți liste de definiții pentru a descrie termeni relevanți dintr-un domeniu de interes.

```

<!DOCTYPE html>
<html>
<head>
  <title>Titlul Paginii</title>
</head>
<body>
  <h1>Listă de Definiții</h1>
  <dl>
    <dt>HTML</dt>
    <dd>Limbaajul de marcare utilizat pentru structurarea și organizarea
    conținutului paginilor web.</dd>

    <dt>CSS</dt>
    <dd>Limbaajul de stilizare folosit pentru a formata și a da stil paginilor
    web.</dd>

    <dt>JavaScript</dt>
    <dd>Limbaajul de programare utilizat pentru a adăuga interactivitate și
    funcționalitate paginilor web.</dd>
  </dl>
</body>
</html>

```

Problemă 8: Adăugați o secțiune de prezentare pentru un hobby personal, folosind imagini și text.

```

<!DOCTYPE html>
<html>
<head>
  <title>Titlul Paginii</title>
</head>
<body>
  <h1>Hobby - Gătitul</h1>
  
  <p>Gătitul este una dintre pasiunile mele. Îmi place să experimentez rețete
  noi și să gătesc mâncăruri delicioase pentru familia și prietenii mei. În
  timpul liber, îmi petrec timpul în bucătărie, încercând să dezvolt noi
  abilități culinare și să descopăr gusturi și arome interesante.</p>

```

```
</body>
</html>
```

Problemă 9: Structurați informațiile despre o călătorie recentă folosind paragrafe și imagini.

```
<!DOCTYPE html>
<html>
<head>
  <title>Titlul Paginii</title>
</head>
<body>
  <h1>Călătorie în Munții Carpați</h1>
  
  <p>În vara aceasta, am avut o călătorie memorabilă în Munții Carpați. Am explorat peisaje frumoase, am urcat pe vârfuri înalte și am petrecut nopțile sub cerul înstelat. Am simțit o conexiune specială cu natura și am creat amintiri de neuitat alături de prieteni.</p>
</body>
</html>
```

Problemă 10: Adăugați un antet și un subsol la pagina web cu informații relevante.

```
<!DOCTYPE html>
<html>
<head>
  <title>Titlul Paginii</title>
</head>
<body>
  <header>
    <h1>Antet - Numele Site-ului</h1>
    <nav>
      <ul>
        <li><a href="/">Acasă</a></li>
        <li><a href="/despre">Despre Noi</a></li>
        <li><a href="/servicii">Servicii</a></li>
        <li><a href="/contact">Contact</a></li>
      </ul>
    </nav>
  </header>

  <h2>Bun venit pe pagina noastră!</h2>
  <p>Aceasta este o pagină web simplă creată pentru rezolvarea problemei 11 din Capitolul VI.</p>

  <footer>
    <p>Contact: contact@example.com</p>
    <p>&copy; 2023 Numele Companiei. Toate drepturile rezervate.</p>
  </footer>
</body>
</html>
```

Probleme suplimentare

Problemă 11: Creați o pagină web cu un formular de contact.

```
<!DOCTYPE html>
<html>
<head>
  <title>Formular de Contact</title>
</head>
<body>
  <h1>Contactați-ne</h1>
  <p>Completează formularul de mai jos pentru a ne trimite un mesaj.</p>

  <form action="/contact" method="post">
    <label for="nume">Nume:</label>
    <input type="text" id="nume" name="nume" required>

    <label for="email">Email:</label>
    <input type="email" id="email" name="email" required>

    <label for="mesaj">Mesaj:</label>
    <textarea id="mesaj" name="mesaj" rows="4" required></textarea>

    <input type="submit" value="Trimite Mesajul">
  </form>
</body>
</html>
```

Problemă 12: Stilizați pagina web folosind CSS pentru a adăuga culori și fonturi.

```
<!DOCTYPE html>
<html>
<head>
  <title>Titlul Paginii</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f0f0f0;
      color: #333;
      margin: 0;
      padding: 0;
    }

    header {
      background-color: #007bff;
      color: #fff;
      padding: 10px;
    }
  </style>
</head>
<body>
  <h1>Titlul Paginii</h1>
</body>
</html>
```

```

header h1 {
    margin: 0;
}

header nav ul {
    list-style: none;
    margin: 0;
    padding: 0;
}

header nav ul li {
    display: inline-block;
    margin-right: 20px;
}

header nav ul li:last-child {
    margin-right: 0;
}

header nav ul li a {
    color: #fff;
    text-decoration: none;
}

h2 {
    color: #007bff;
}

footer {
    background-color: #ccc;
    padding: 10px;
    text-align: center;
}
</style>
</head>
<body>
<header>
<h1>Antet - Numele Site-ului</h1>
<nav>
<ul>
<li><a href="/">Acasă</a></li>
<li><a href="/despre">Despre Noi</a></li>
<li><a href="/servicii">Servicii</a></li>
<li><a href="/contact">Contact</a></li>
</ul>
</nav>
</header>

<h2>Bun venit pe pagina noastră!</h2>

```

```

    <p>Aceasta este o pagină web simplă creată pentru rezolvarea problemei 12
    din Capitolul VI.</p>

    <footer>
      <p>Contact: contact@example.com</p>
      <p>&copy; 2023 Numele Companiei. Toate drepturile rezervate.</p>
    </footer>
  </body>
</html>

```

Problemă 13: Realizați o pagină web cu un meniu de navigare vertical.

```

<!DOCTYPE html>
<html>
<head>
  <title>Meniu Vertical</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
    }

    header {
      background-color: #333;
      color: #fff;
      padding: 10px;
      text-align: center;
    }

    nav {
      background-color: #007bff;
      color: #fff;
      width: 200px;
      padding: 10px;
      position: fixed;
      top: 0;
      bottom: 0;
    }

    nav ul {
      list-style: none;
      padding: 0;
      margin: 0;
    }

    nav ul li {
      margin-bottom: 10px;
    }
  </style>

```

```

nav ul li a {
    color: #fff;
    text-decoration: none;
    display: block;
}

section {
    margin-left: 220px;
    padding: 20px;
}
</style>
</head>
<body>
<header>
    <h1>Antet - Numele Site-ului</h1>
</header>

<nav>
    <ul>
        <li><a href="/">Acasă</a></li>
        <li><a href="/despre">Despre Noi</a></li>
        <li><a href="/servicii">Servicii</a></li>
        <li><a href="/contact">Contact</a></li>
    </ul>
</nav>

<section>
    <h2>Bun venit pe pagina noastră!</h2>
    <p>Aceasta este o pagină web cu un meniu de navigare vertical creată
    pentru rezolvarea problemei 13 din Capitolul VI.</p>
</section>
</body>
</html>

```

Problemă 14: Adăugați o galerie foto cu efect lightbox.

```

<!DOCTYPE html>
<html>
<head>
    <title>Galerie Foto cu Lightbox</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
        }

        .galerie {

```

```

        display: grid;
        grid-template-columns: repeat(3, 1fr);
        gap: 10px;
        padding: 20px;
    }

    .galerie img {
        width: 100%;
        height: auto;
        cursor: pointer;
    }

    .lightbox {
        display: none;
        position: fixed;
        top: 0;
        left: 0;
        width: 100%;
        height: 100%;
        background-color: rgba(0, 0, 0, 0.9);
        justify-content: center;
        align-items: center;
    }

    .lightbox img {
        max-width: 90%;
        max-height: 90%;
    }

    .lightbox.active {
        display: flex;
    }
</style>
</head>
<body>
    <h1>Galerie Foto cu Lightbox</h1>

    <div class="galerie">
        
        
        
        <!-- Adăugați mai multe imagini aici -->
    </div>

    <!-- Lightbox -->
    <div class="lightbox" onclick="closeLightbox()">
        <img id="lightboxImage" src="" alt="Imagine Lightbox">
    </div>

```

```

<script>
  function openLightbox(imageSrc) {
    const lightbox = document.querySelector('.lightbox');
    const lightboxImage = document.getElementById('lightboxImage');

    lightboxImage.src = imageSrc;
    lightbox.classList.add('active');
  }

  function closeLightbox() {
    const lightbox = document.querySelector('.lightbox');

    lightbox.classList.remove('active');
  }
</script>
</body>
</html>

```

Problemă 15: Realizați un formular pentru înregistrarea utilizatorilor.

```

<!DOCTYPE html>
<html>
<head>
  <title>Formular Înregistrare Utilizator</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
    }

    form {
      width: 400px;
      margin: 20px auto;
      padding: 20px;
      border: 1px solid #ccc;
    }

    label {
      display: block;
      margin-bottom: 10px;
    }

    input[type="text"],
    input[type="email"],
    input[type="password"] {
      width: 100%;
      padding: 8px;
      margin-bottom: 15px;
    }
  </style>

```

```

    }

    input[type="submit"] {
        background-color: #007bff;
        color: #fff;
        padding: 10px 20px;
        border: none;
        cursor: pointer;
    }

    input[type="submit"]:hover {
        background-color: #0056b3;
    }
</style>
</head>
<body>
    <h1>Formular Înregistrare Utilizator</h1>

    <form action="/register" method="post">
        <label for="nume">Nume:</label>
        <input type="text" id="nume" name="nume" required>

        <label for="prenume">Prenume:</label>
        <input type="text" id="prenume" name="prenume" required>

        <label for="email">Email:</label>
        <input type="email" id="email" name="email" required>

        <label for="parola">Parolă:</label>
        <input type="password" id="parola" name="parola" required>

        <input type="submit" value="Înregistrare">
    </form>
</body>
</html>

```

Problemă 16: Creați o pagină web cu un tabel pentru afișarea datelor.

```

<!DOCTYPE html>
<html>
<head>
    <title>Tabel cu Date</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
        }
    </style>

```

```

table {
    width: 80%;
    margin: 20px auto;
    border-collapse: collapse;
}

th, td {
    border: 1px solid #ccc;
    padding: 8px;
    text-align: center;
}

th {
    background-color: #f0f0f0;
}

tr:nth-child(even) {
    background-color: #f2f2f2;
}
</style>
</head>
<body>
    <h1>Tabel cu Date</h1>

    <table>
        <tr>
            <th>Nume</th>
            <th>Prenume</th>
            <th>Email</th>
        </tr>
        <tr>
            <td>Popescu</td>
            <td>Ion</td>
            <td>ion.popescu@example.com</td>
        </tr>
        <tr>
            <td>Ionescu</td>
            <td>Maria</td>
            <td>maria.ionescu@example.com</td>
        </tr>
        <!-- Adăugați mai multe rânduri cu date aici -->
    </table>
</body>
</html>

```

Problemă 17: Realizați un meniu dropdown în cadrul meniului de navigare.

```

<!DOCTYPE html>
<html>

```

```

<head>
  <title>Meniu Dropdown</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
    }

    header {
      background-color: #333;
      color: #fff;
      padding: 10px;
      text-align: center;
    }

    nav {
      background-color: #007bff;
      color: #fff;
      padding: 10px;
    }

    nav ul {
      list-style: none;
      margin: 0;
      padding: 0;
      display: flex;
    }

    nav ul li {
      margin-right: 20px;
    }

    nav ul li:last-child {
      margin-right: 0;
    }

    nav ul li a {
      color: #fff;
      text-decoration: none;
    }

    /* Stiluri pentru meniul dropdown */
    .dropdown {
      position: relative;
    }

    .dropdown-content {
      display: none;

```

```

        position: absolute;
        top: 100%;
        left: 0;
        background-color: #007bff;
        min-width: 120px;
        padding: 5px;
    }

    .dropdown:hover .dropdown-content {
        display: block;
    }
</style>
</head>
<body>
    <header>
        <h1>Antet - Numele Site-ului</h1>
    </header>

    <nav>
        <ul>
            <li><a href="/">Acasă</a></li>
            <li><a href="/despre">Despre Noi</a></li>
            <li class="dropdown">
                <a href="#">Servicii</a>
                <div class="dropdown-content">
                    <a href="/servicii/web-design">Web Design</a>
                    <a href="/servicii/dezvoltare-web">Dezvoltare Web</a>
                    <a href="/servicii/maintenance">Maintenance</a>
                </div>
            </li>
            <li><a href="/contact">Contact</a></li>
        </ul>
    </nav>
</body>
</html>

```

Problemă 18: Realizați o pagină web responsivă cu ajutorul media queries.

```

<!DOCTYPE html>
<html>
<head>
    <title>Pagină Responsivă</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
        }
    </style>

```

```

header {
  background-color: #333;
  color: #fff;
  padding: 10px;
  text-align: center;
}

nav {
  background-color: #007bff;
  color: #fff;
  padding: 10px;
  display: flex;
}

nav ul {
  list-style: none;
  margin: 0;
  padding: 0;
  display: flex;
}

nav ul li {
  margin-right: 20px;
}

nav ul li:last-child {
  margin-right: 0;
}

nav ul li a {
  color: #fff;
  text-decoration: none;
}

/* Media queries pentru o pagină responsivă */
@media only screen and (max-width: 600px) {
  nav {
    flex-direction: column;
  }

  nav ul {
    flex-direction: column;
  }

  nav ul li {
    margin-right: 0;
    margin-bottom: 10px;
  }
}

```

```

    </style>
</head>
<body>
    <header>
        <h1>Antet - Numele Site-ului</h1>
    </header>

    <nav>
        <ul>
            <li><a href="/">Acasă</a></li>
            <li><a href="/despre">Despre Noi</a></li>
            <li><a href="/servicii">Servicii</a></li>
            <li><a href="/contact">Contact</a></li>
        </ul>
    </nav>
</body>
</html>

```

Problemă 19: Realizați un slideshow cu imagini automat.

```

<!DOCTYPE html>
<html>
<head>
    <title>Slideshow cu Imagini</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
        }

        #slideshow {
            max-width: 800px;
            margin: 20px auto;
            overflow: hidden;
        }

        #slideshow img {
            width: 100%;
            height: auto;
            opacity: 0;
            transition: opacity 1s ease-in-out;
        }

        #slideshow img.active {
            opacity: 1;
        }
    </style>
</head>

```

```

<body>
  <h1>Slideshow cu Imagini</h1>

  <div id="slideshow">
    
    
    
    <!-- Adăugați mai multe imagini aici -->
  </div>

  <script>
    const slideshow = document.getElementById('slideshow');
    const images = slideshow.getElementsByTagName('img');
    let currentIndex = 0;

    function showNextImage() {
      images[currentIndex].classList.remove('active');
      currentIndex = (currentIndex + 1) % images.length;
      images[currentIndex].classList.add('active');
    }

    setInterval(showNextImage, 3000); // Schimbă imaginea la fiecare 3 secunde
  </script>
</body>
</html>

```

Problemă 20: Realizați un formular de căutare cu o casetă de text și un buton de căutare.

```

<!DOCTYPE html>
<html>
<head>
  <title>Formular de Căutare</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
    }

    form {
      max-width: 400px;
      margin: 20px auto;
      padding: 20px;
      border: 1px solid #ccc;
      display: flex;
    }

    input[type="text"] {
      flex: 1;
    }

```

```

padding: 8px;
}

input[type="submit"] {
background-color: #007bff;
color: #fff;
padding: 8px 20px;
border: none;
cursor: pointer;
}

input[type="submit"]:hover {
background-color: #0056b3;
}
</style>
</head>
<body>
<h1>Formular de Căutare</h1>

<form action="/search" method="get">
<input type="text" id="query" name="q" placeholder="Introduceți termenul
de căutare" required>
<input type="submit" value="Caută">
</form>
</body>
</html>

```

Problemă 21: Adăugați un video în pagina web.

```

<!DOCTYPE html>
<html>
<head>
<title>Video Încorporat</title>
<style>
body {
font-family: Arial, sans-serif;
margin: 0;
padding: 0;
}

video {
max-width: 800px;
margin: 20px auto;
display: block;
}
</style>
</head>
<body>
<h1>Video Încorporat</h1>

```

```

<video controls>
  <source src="video.mp4" type="video/mp4">
  <!-- Adăugați mai multe surse video aici pentru compatibilitate cu mai
multe browsere -->
  Dvs. browserul nu suportă elementul video.
</video>
</body>
</html>

```

Problemă 22: Realizați un footer fixat în partea de jos a paginii.

```

<!DOCTYPE html>
<html>
<head>
  <title>Footer Fixat</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
      height: 100%;
    }

    #content {
      min-height: calc(100% - 50px);
    }

    footer {
      background-color: #333;
      color: #fff;
      padding: 10px;
      text-align: center;
      position: fixed;
      left: 0;
      bottom: 0;
      width: 100%;
    }
  </style>
</head>
<body>
  <div id="content">
    <h1>Conținutul Paginii</h1>
    <p>Adăugați aici conținutul principal al paginii.</p>
  </div>

  <footer>
    <p>Drepturi de autor &copy; 2023 Numele Companiei. Toate drepturile
rezervate.</p>

```

```
</footer>
</body>
</html>
```

Problemă 23: Realizați o pagină web cu un efect de hover la imaginile din galerie.

```
<!DOCTYPE html>
<html>
<head>
  <title>Galerie cu Efect de Hover</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
    }

    .galerie {
      display: grid;
      grid-template-columns: repeat(3, 1fr);
      gap: 10px;
      padding: 20px;
    }

    .galerie img {
      width: 100%;
      height: auto;
      cursor: pointer;
      transition: transform 0.3s ease;
    }

    .galerie img:hover {
      transform: scale(1.1);
    }
  </style>
</head>
<body>
  <h1>Galerie cu Efect de Hover</h1>

  <div class="galerie">
    
    
    
    <!-- Adăugați mai multe imagini aici -->
  </div>
</body>
</html>
```

Problemă 24: Realizați o pagină web cu un meniu fixat în partea de sus.

```
<!DOCTYPE html>
<html>
<head>
  <title>Meniu Fixat</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
    }

    header {
      background-color: #333;
      color: #fff;
      padding: 10px;
      text-align: center;
      position: fixed;
      top: 0;
      left: 0;
      width: 100%;
    }

    nav {
      background-color: #007bff;
      color: #fff;
      padding: 10px;
      display: flex;
    }

    nav ul {
      list-style: none;
      margin: 0;
      padding: 0;
      display: flex;
    }

    nav ul li {
      margin-right: 20px;
    }

    nav ul li:last-child {
      margin-right: 0;
    }

    nav ul li a {
      color: #fff;
      text-decoration: none;
```

```

    }

    #content {
        margin-top: 50px;
        padding: 20px;
    }
</style>
</head>
<body>
    <header>
        <h1>Antet - Numele Site-ului</h1>
    </header>

    <nav>
        <ul>
            <li><a href="/">Acasă</a></li>
            <li><a href="/despre">Despre Noi</a></li>
            <li><a href="/servicii">Servicii</a></li>
            <li><a href="/contact">Contact</a></li>
        </ul>
    </nav>

    <div id="content">
        <h1>Conținutul Paginii</h1>
        <p>Adăugați aici conținutul principal al paginii.</p>
    </div>
</body>
</html>

```

Problemă 25: Adăugați un efect de hover la link-urile din meniu.

```

<!DOCTYPE html>
<html>
<head>
    <title>Meniu cu Efect de Hover</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
        }

        header {
            background-color: #333;
            color: #fff;
            padding: 10px;
            text-align: center;
            position: fixed;
            top: 0;

```

```

    left: 0;
    width: 100%;
}

nav {
    background-color: #007bff;
    color: #fff;
    padding: 10px;
    display: flex;
}

nav ul {
    list-style: none;
    margin: 0;
    padding: 0;
    display: flex;
}

nav ul li {
    margin-right: 20px;
}

nav ul li:last-child {
    margin-right: 0;
}

nav ul li a {
    color: #fff;
    text-decoration: none;
    transition: color 0.3s ease;
}

nav ul li a:hover {
    color: #f0f0f0;
}

#content {
    margin-top: 50px;
    padding: 20px;
}
</style>
</head>
<body>
    <header>
        <h1>Antet - Numele Site-ului</h1>
    </header>

    <nav>
        <ul>

```

```

    <li><a href="/">Acasă</a></li>
    <li><a href="/despre">Despre Noi</a></li>
    <li><a href="/servicii">Servicii</a></li>
    <li><a href="/contact">Contact</a></li>
  </ul>
</nav>

<div id="content">
  <h1>Conținutul Paginii</h1>
  <p>Adăugați aici conținutul principal al paginii.</p>
</div>
</body>
</html>

```

Problemă 26: Realizați un formular cu validare a datelor introduse.

```

<!DOCTYPE html>
<html>
<head>
  <title>Formular cu Validare</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
    }

    form {
      max-width: 400px;
      margin: 20px auto;
      padding: 20px;
      border: 1px solid #ccc;
    }

    label {
      display: block;
      margin-bottom: 10px;
    }

    input[type="text"],
    input[type="email"],
    input[type="password"] {
      width: 100%;
      padding: 8px;
      margin-bottom: 15px;
    }

    input[type="submit"] {
      background-color: #007bff;

```

```

        color: #fff;
        padding: 10px 20px;
        border: none;
        cursor: pointer;
    }

    input[type="submit"]:hover {
        background-color: #0056b3;
    }

    /* Stiluri pentru mesajele de eroare */
    .error {
        color: red;
        margin-top: -10px;
        margin-bottom: 15px;
    }
</style>
</head>
<body>
    <h1>Formular cu Validare</h1>

    <form action="/submit" method="post" onsubmit="return validateForm()">
        <label for="nume">Nume:</label>
        <input type="text" id="nume" name="nume" required>
        <span class="error" id="numeError"></span>

        <label for="email">Email:</label>
        <input type="email" id="email" name="email" required>
        <span class="error" id="emailError"></span>

        <label for="parola">Parolă:</label>
        <input type="password" id="parola" name="parola" required>
        <span class="error" id="parolaError"></span>

        <input type="submit" value="Înregistrare">
    </form>

    <script>
        function validateForm() {
            let nume = document.getElementById('nume').value;
            let email = document.getElementById('email').value;
            let parola = document.getElementById('parola').value;

            let numeError = document.getElementById('numeError');
            let emailError = document.getElementById('emailError');
            let parolaError = document.getElementById('parolaError');

            let valid = true;

```

```

    if (nume === "") {
        numeError.innerHTML = "Introduceți numele.";
        valid = false;
    } else {
        numeError.innerHTML = "";
    }

    if (email === "") {
        emailError.innerHTML = "Introduceți adresa de email.";
        valid = false;
    } else {
        emailError.innerHTML = "";
    }

    if (parola === "") {
        parolaError.innerHTML = "Introduceți parola.";
        valid = false;
    } else {
        parolaError.innerHTML = "";
    }

    return valid;
}
</script>
</body>
</html>

```

Problemă 27: Realizați o pagină web cu un carusel de imagini.

```

<!DOCTYPE html>
<html>
<head>
    <title>Carusel de Imagini</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
        }

        .carousel {
            max-width: 800px;
            margin: 20px auto;
            overflow: hidden;
            position: relative;
        }

        .carousel img {
            width: 100%;

```

```

        height: auto;
        opacity: 0;
        transition: opacity 1s ease-in-out;
    }

    .carusel img.active {
        opacity: 1;
    }

    .prev,
    .next {
        position: absolute;
        top: 50%;
        transform: translateY(-50%);
        font-size: 24px;
        color: #fff;
        background-color: rgba(0, 0, 0, 0.5);
        padding: 10px;
        cursor: pointer;
        transition: background-color 0.3s ease;
    }

    .prev:hover,
    .next:hover {
        background-color: rgba(0, 0, 0, 0.8);
    }

    .prev {
        left: 10px;
    }

    .next {
        right: 10px;
    }
</style>
</head>
<body>
    <h1>Carusel de Imagini</h1>

    <div class="carusel">
        
        
        
        <!-- Adăugați mai multe imagini aici -->
        <span class="prev" onclick="showSlide(-1)">◀</span>
        <span class="next" onclick="showSlide(1)">▶</span>
    </div>

    <script>

```

```

    let currentSlide = 0;
    const slides =
document.getElementsByClassName('carousel')[0].getElementsByTagName('img');

    function showSlide(n) {
        slides[currentSlide].classList.remove('active');
        currentSlide = (currentSlide + n + slides.length) % slides.length;
        slides[currentSlide].classList.add('active');
    }

    setInterval(function() { showSlide(1); }, 3000); // Schimbă imaginea la
fiecare 3 secunde
</script>
</body>
</html>

```

Problemă 28: Realizați o pagină web cu un efect de parallax.

```

<!DOCTYPE html>
<html>
<head>
    <title>Efect de Parallax</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
        }

        .parallax {
            background-image: url("fundal.jpg");
            background-attachment: fixed;
            background-position: center;
            background-repeat: no-repeat;
            background-size: cover;
            height: 500px;
        }

        .content {
            background-color: #fff;
            padding: 50px;
            text-align: center;
        }

        .parallax-text {
            color: #fff;
            font-size: 36px;
            text-align: center;
            padding-top: 200px;
        }
    </style>

```

```

    }

    .content-text {
        font-size: 24px;
        color: #333;
    }
</style>
</head>
<body>
    <div class="parallax">
        <div class="parallax-text">Efect de Parallax</div>
    </div>

    <div class="content">
        <h1>Conținutul Paginii</h1>
        <p class="content-text">Adăugați aici conținutul principal al paginii.
Efectul de parallax creează o iluzie de mișcare pe măsură ce utilizatorul
derulează pagina.</p>
    </div>

    <div class="parallax">
        <div class="parallax-text">Efect de Parallax</div>
    </div>
</body>
</html>

```

Problemă 29: Realizați o pagină web cu un meniu responsiv pentru dispozitive mobile.

```

<!DOCTYPE html>
<html>
<head>
    <title>Meniu Responsiv</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
        }

        header {
            background-color: #333;
            color: #fff;
            padding: 10px;
            text-align: center;
            position: fixed;
            top: 0;
            left: 0;
            width: 100%;

```

```

}

nav {
  background-color: #007bff;
  color: #fff;
  padding: 10px;
  display: flex;
}

nav ul {
  list-style: none;
  margin: 0;
  padding: 0;
  display: flex;
}

nav ul li {
  margin-right: 20px;
}

nav ul li:last-child {
  margin-right: 0;
}

nav ul li a {
  color: #fff;
  text-decoration: none;
  transition: color 0.3s ease;
}

nav ul li a:hover {
  color: #f0f0f0;
}

/* Stiluri pentru meniul responsiv */
.menu-icon {
  display: none;
  cursor: pointer;
  font-size: 24px;
}

@media screen and (max-width: 600px) {
  .menu-icon {
    display: block;
  }

  nav ul {
    display: none;
    flex-direction: column;
  }

```

```

        position: absolute;
        top: 50px;
        left: 0;
        width: 100%;
        background-color: #333;
    }

    nav ul li {
        margin-right: 0;
        margin-bottom: 10px;
    }

    nav ul li:last-child {
        margin-bottom: 0;
    }

    nav ul li a {
        padding: 10px;
        display: block;
    }

    .menu-icon.active {
        color: #007bff;
    }

    .menu-icon.active ~ ul {
        display: flex;
    }
}
</style>
</head>
<body>
    <header>
        <h1>Antet - Numele Site-ului</h1>
    </header>

    <nav>
        <span class="menu-icon" onclick="toggleMenu()">#9776;</span>
        <ul>
            <li><a href="/">Acasă</a></li>
            <li><a href="/despre">Despre Noi</a></li>
            <li><a href="/servicii">Servicii</a></li>
            <li><a href="/contact">Contact</a></li>
        </ul>
    </nav>

    <div style="height: 1000px; background-color: #f0f0f0; padding: 20px;">
        <h1>Conținutul Paginii</h1>
    </div>

```

```

    <p>Adăugați aici conținutul principal al paginii. Meniul se va transforma
    într-un meniu vertical pentru dispozitivele mobile.</p>
  </div>

  <script>
    function toggleMenu() {
      const menu = document.querySelector('nav ul');
      const menuIcon = document.querySelector('.menu-icon');
      menu.classList.toggle('active');
      menuIcon.classList.toggle('active');
    }
  </script>
</body>
</html>

```

Problemă 30: Realizați o pagină web cu o hartă încorporată de la Google Maps.

```

<!DOCTYPE html>
<html>
<head>
  <title>Hartă Google Maps</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
    }

    #map {
      height: 500px;
    }
  </style>
</head>
<body>
  <h1>Hartă Google Maps</h1>

  <div id="map"></div>

  <script>
    function initMap() {
      const location = { lat: 47.1585, lng: 27.6014 }; // Coordonate pentru
      Iași, România
      const map = new google.maps.Map(document.getElementById('map'), {
        center: location,
        zoom: 15
      });

      const marker = new google.maps.Marker({
        position: location,

```

```
        map: map,  
        title: 'Iași, România'  
    });  
    }  
</script>  
<script  
src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&callback=initMap  
" async defer></script>  
</body>  
</html>
```

LUCRAREA 2

2.1. Obiectivele lucrării

În cadrul acestei lucrări de laborator, ne vom concentra pe conceptul de CSS (Cascading Style Sheets) și importanța sa în dezvoltarea web. CSS reprezintă o parte esențială în crearea și stilizarea paginilor web, iar în acest capitol vom explora pe scurt ce este CSS și de ce joacă un rol crucial în dezvoltarea site-urilor web moderne.

2.2. De ce este CSS important în dezvoltarea web?

1. Separarea conținutului de prezentare: O caracteristică cheie a CSS-ului este că permite separarea conținutului (structura HTML) de prezentare (stilurile CSS). Aceasta înseamnă că putem modifica aspectul paginilor web fără a schimba structura conținutului, facilitând astfel menținerea și actualizarea site-urilor.

2. Consistență și coerență: Prin aplicarea stilurilor CSS la întregul site, putem asigura o coerență în design-ul paginilor. Astfel, toate paginile vor avea același aspect, oferind utilizatorilor o experiență mai plăcută și profesională.

3. Flexibilitate și adaptabilitate: CSS ne permite să cream stiluri flexibile și să facem paginile responsive, astfel încât acestea să se adapteze în mod automat la diferite dimensiuni de ecran, precum cele ale dispozitivelor mobile sau tabletelor.

4. Eficiența în codare: CSS ne permite să definim stilurile o singură dată și să le aplicăm la mai multe elemente, reducând astfel repetarea și mărimea fișierelor de stil. Acest lucru duce la o codare mai eficientă și o încărcare mai rapidă a paginilor.

5. Separarea între design și conținut: Prin utilizarea CSS, designerii pot lucra separat de dezvoltatori, făcând mai ușoară colaborarea în cadrul unui proiect web. Designerii pot defini stilurile vizuale, iar dezvoltatorii pot integra aceste stiluri în pagini utilizând CSS.

În concluzie, CSS joacă un rol esențial în dezvoltarea web, oferind control asupra aspectului paginilor și creând o experiență vizuală plăcută pentru utilizatori. Prin utilizarea CSS, putem crea pagini web atractive, responsive și ușor de întreținut. În următoarele capitole, vom explora în detaliu cum funcționează CSS și cum putem aplica stiluri la elementele HTML pentru a crea pagini web moderne și profesionale.

2.3. Cum funcționează CSS și sintaxa

CSS (Cascading Style Sheets) funcționează într-un mod foarte simplu și logic. Atunci când un browser afișează o pagină web, el combină trei elemente cheie: conținutul HTML, stilurile CSS și JavaScript (dacă este cazul). CSS-ul este responsabil pentru aspectul vizual al paginii web.

Cuvântul "cascading" din numele CSS se referă la modul în care stilurile sunt aplicate în pagină. Întrucât o pagină web poate avea multiple fișiere CSS și stiluri definite în interiorul elementelor HTML, CSS folosește o ierarhie pentru a determina care stiluri se aplică la fiecare element.

Sintaxa CSS este relativ simplă și constă dintr-un selector și o declarație de stil. Selectorul indică la ce element HTML se aplică stilul, iar declarația de stil conține proprietăți și valorile lor asociate.

Un exemplu de sintaxă CSS arată astfel:

```
selector {  
    proprietate: valoare;  
}
```

- Selectorul poate fi numele unui element HTML, clasa sau ID. De exemplu:
 - Selector pentru toate elementele `p`: `p`
 - Selector pentru un element cu clasa `titlu`: `.titlu`
 - Selector pentru un element cu ID-ul `logo`: `#logo`
- Proprietatea reprezintă aspectul pe care dorim să-l stilizăm, cum ar fi culoarea textului sau dimensiunea fontului.
- Valoarea este ceea ce atribuim proprietății. De exemplu, pentru a seta culoarea textului la roșu, vom utiliza: `color: red`.

Pentru a aplica stiluri în fișierul CSS extern, vom folosi următoarea sintaxă:

```
<link rel="stylesheet" href="style.css">
```

Aceasta conectează fișierul CSS extern numit `style.css` la pagina HTML, iar toate stilurile definite în acel fișier vor fi aplicate la elementele corespunzătoare din pagină.

Exemple de utilizare a CSS

Exemplu 1: Stilizarea textului

```
p {  
  color: blue;  
  font-size: 16px;  
  font-family: Arial, sans-serif;  
}
```

Exemplu 2: Stilizarea link-urilor

```
a {  
  text-decoration: none;  
  color: #ff6600;  
}  
  
a:hover {  
  text-decoration: underline;  
}
```

2.4. Stilizarea elementelor HTML

Textul reprezintă unul dintre cele mai importante elemente într-o pagină web, iar CSS ne oferă o serie de opțiuni pentru a-l stiliza. Iată câteva exemple de proprietăți CSS utilizate pentru stilizarea textului:

```
/* Schimbarea culorii textului */  
p {  
  color: blue;  
}  
  
/* Mărimea fontului */  
h1 {  
  font-size: 36px;  
}  
  
/* Stilul fontului */  
h2 {  
  font-family: "Helvetica", sans-serif;  
}  
  
/* Decorațiuni pentru text */  
a {  
  text-decoration: none;  
}  
  
/* Stilizarea textului la hover */
```

```
a:hover {
    text-decoration: underline;
}
```

Imaginile sunt un alt element important în designul paginilor web. CSS ne permite să ajustăm dimensiunile imaginilor și să le aranjăm în pagină. Iată câteva exemple de stilizare a imaginilor:

```
/* Mărimea imaginilor */
img {
    width: 300px;
    height: 200px;
}

/* Rotirea imaginilor */
img.rotate {
    transform: rotate(45deg);
}

/* Adăugarea unei umbre la imagini */
img.shadow {
    box-shadow: 3px 3px 5px rgba(0, 0, 0, 0.5);
}
```

Link-urile sunt importante pentru navigarea între pagini și trebuie să fie evidente pentru utilizatori. CSS ne oferă posibilitatea de a stiliza link-urile în diferite moduri. Iată câteva exemple:

```
/* Culoarea link-urilor */
a {
    color: #007bff;
}

/* Stilizarea link-urilor vizitate */
a:visited {
    color: #800080;
}

/* Stilizarea link-urilor la hover */
a:hover {
    text-decoration: underline;
}
```

Fundalurile joacă un rol important în definirea atmosferei paginii web. CSS ne permite să adăugăm fundaluri colorate, imagini sau modele. Iată câteva exemple:

```
/* Fundal cu culoare solidă */
body {
    background-color: #f2f2f2;
}
```

```

}

/* Fundal cu imagine */
div {
    background-image: url("background.jpg");
    background-size: cover;
}

/* Fundal cu model repetat */
section {
    background-image: url("pattern.png");
    background-repeat: repeat;
}

```

Acestea sunt doar câteva exemple de proprietăți CSS pe care le putem utiliza pentru a stiliza diferitele elemente HTML. Combinațiile sunt nelimitate, și CSS ne oferă libertatea de a crea design-uri personalizate și atractive pentru paginile web.

2.5. Selectori și Stilizarea cu Clase și ID-uri

Selectoarele în CSS ne permit să identificăm elementele HTML cărora dorim să le aplicăm stiluri. Există mai multe tipuri de selectoare pe care le putem utiliza pentru a face selecții precise.

Selectori de tip element

Acești selectori identifică toate elementele de același tip din pagină. De exemplu:

```

p {
    color: blue;
}

```

Acest selector aplică stiluri la toate elementele `p` (paragrafe) din pagină, schimbându-le culoarea textului în albastru.

Selectori de clasă

Acești selectori ne permit să aplicăm stiluri doar la elementele care au o clasă specificată. Clasele sunt indicate în HTML utilizând atributul `class`. De exemplu:

```

.titlu {
    font-size: 24px;
}

```

În acest caz, toate elementele care au clasa `titlu` vor avea mărimea fontului de 24 de pixeli.

Selectorii de ID

Selectoarele de ID ne permit să stilizăm un singur element care are un ID unic în pagină. ID-urile sunt indicate în HTML utilizând atributul `id`. De exemplu:

```
#logo {  
  width: 200px;  
  height: 100px;  
}
```

Aici, un singur element care are ID-ul `logo` va avea lățimea de 200 de pixeli și înălțimea de 100 de pixeli.

Stilizarea cu Clase

Un exemplu de utilizare a claselor în CSS arată astfel:

```
<style>  
  .titlu {  
    color: #007bff;  
    font-size: 32px;  
  }  
  
  .descriere {  
    font-style: italic;  
    color: #666;  
  }  
</style>  
  
<h2 class="titlu">Titlul Paginii</h2>  
<p class="descriere">Aceasta este o scurtă descriere a paginii.</p>
```

În acest exemplu, clasa `titlu` este utilizată pentru stilizarea unui titlu cu culoarea albastră și mărimea fontului de 32 de pixeli. Clasa `descriere` este folosită pentru a adăuga o stilizare italică și o culoare mai închisă pentru textul de descriere.

Stilizarea cu ID-uri

Un exemplu de utilizare a ID-urilor în CSS este următorul:

```
<style>  
  #logo {  
    width: 150px;  
    height: 100px;  
  }  
</style>  
  

```

Aici, ID-ul `logo` este utilizat pentru a seta dimensiunile imaginii cu 150 de pixeli lățime și 100 de pixeli înălțime.

Stilizarea cu selecții combinate

Putem combina selectoarele pentru a face selecții mai complexe și precise. De exemplu:

```
.header h1 {  
  color: #ff6600;  
  font-size: 36px;  
}
```

Acest exemplu aplică stiluri la toate elementele `h1` din interiorul elementului cu clasa `header`, schimbându-le culoarea textului în portocaliu și mărimea fontului în 36 de pixeli.

Utilizarea selectoarelor în CSS ne permite să aplicăm stiluri la elementele HTML într-un mod precis și personalizat. Stilizarea cu clase și ID-uri ne oferă flexibilitatea de a crea design-uri unice și profesionale pentru paginile web.

2.6. Modelul de Tip Box și Poziționarea Elementelor

Modelul de Tip Box în CSS

Modelul de Tip Box reprezintă modul în care sunt organizate și afișate elementele HTML în pagină. Fiecare element HTML este reprezentat ca o "cutie" sau un "container" în care putem controla dimensiunile, marginile, padding-ul și bordurile.

Elementele HTML au patru componente principale în cadrul Modelului de Tip Box:

1. **Conținutul (Content):** Este conținutul efectiv al elementului, cum ar fi textul sau imaginea.
2. **Padding-ul:** Reprezintă spațiul între conținut și bordură. Padding-ul adaugă un spațiu gol în jurul conținutului elementului.
3. **Bordura (Border):** Este o linie care înconjoară conținutul și padding-ul elementului. Bordura poate avea o culoare, o grosime și un stil definite.
4. **Marginile:** Reprezintă spațiul dintre elementul curent și alte elemente din jur. Marginile controlează distanța între elementul curent și celelalte elemente din pagină.

Modificarea dimensiunilor și poziționarea elementelor

Pentru a modifica dimensiunile și poziționarea elementelor, putem utiliza diferite proprietăți CSS.

Proprietățile pentru Dimensiuni:

```
/* Lățimea și înălțimea elementului */
div {
  width: 300px;
  height: 200px;
}

/* Modificarea padding-ului */
p {
  padding: 20px;
}

/* Modificarea grosimii bordurii */
h1 {
  border: 2px solid #007bff;
}
```

Aceste exemple arată cum putem modifica lățimea și înălțimea unui element, cum putem adăuga un padding între conținut și bordură și cum putem schimba grosimea și culoarea bordurii.

Proprietățile pentru Poziționare:

```
/* Poziționarea absolută */
#element1 {
  position: absolute;
  top: 50px;
  left: 100px;
}

/* Poziționarea relativă */
#element2 {
  position: relative;
  top: 20px;
  left: 30px;
}

/* Poziționarea fixă */
#element3 {
  position: fixed;
  top: 0;
  right: 0;
}
```

Aceste exemple arată cum putem poziționa elementele în pagină folosind proprietatea `position`. Poziționarea absolută le va scoate din fluxul normal al paginii, iar poziționarea relativă le va menține în flux, permițându-ne să ajustăm poziția relativ față de locul inițial. Poziționarea fixă va face ca elementele să fie poziționate fix în raport cu fereastra de vizualizare, indiferent de scroll-ul paginii.

Exemplu de utilizare a Modelului de Tip Box

```
<style>
  .box {
    width: 300px;
    height: 150px;
    padding: 20px;
    border: 2px solid #ccc;
    margin: 10px;
  }
</style>

<div class="box">
  <p>Acesta este un paragraf într-o cutie.</p>
</div>
```

Acest exemplu creează o cutie cu dimensiunile 300x150 pixeli, adaugă un padding de 20 de pixeli și o bordură de 2 pixeli cu culoarea gri. Marginile de 10 pixeli îl separă de alte elemente din pagină.

Modelul de Tip Box din CSS ne permite să controlăm dimensiunile, padding-ul, bordurile și marginile elementelor HTML, permițându-ne să ajustăm poziționarea și spațiile dintre elemente pentru a crea un design armonios și plăcut pentru paginile web.

2.7. Layout-uri și Design Responsiv

Layout-uri cu mai multe coloane

CSS ne permite să construim layout-uri cu mai multe coloane utilizând proprietăți precum `float`, `display: inline-block`, sau mai recent, `flexbox` și `grid`. Acestea ne oferă posibilitatea de a aranja elementele în coloane sau rânduri pentru a crea design-uri complexe.

Exemplu de utilizare a `float` pentru a crea un layout cu două coloane:

```
<style>
.container {
  width: 100%;
}
```

```
.coluna {
  width: 50%;
  float: left;
}
</style>

<div class="container">
  <div class="coluna">Coloana 1</div>
  <div class="coluna">Coloana 2</div>
</div>
```

Acest exemplu va crea două coloane de dimensiuni egale în cadrul containerului.

Grile Flexibile și Grid

Grilele flexibile și grid sunt metode mai avansate de creare a layout-urilor, care oferă un control mai precis asupra poziționării elementelor. Acestea permit distribuirea elementelor în coloane și rânduri și facilitează crearea unor layout-uri complexe.

Exemplu de utilizare a grilei flexibile:

```
<style>
.grid {
  display: flex;
  flex-wrap: wrap;
}
.item {
  flex: 1;
  min-width: 200px;
  margin: 10px;
}
</style>

<div class="grid">
  <div class="item">Element 1</div>
  <div class="item">Element 2</div>
  <div class="item">Element 3</div>
</div>
```

Acest exemplu va crea o grilă flexibilă cu trei elemente distribuite în rânduri, cu lățimea minimă de 200px.

Design Responsiv

Designul responsiv este o abordare esențială în dezvoltarea web modernă. Scopul designului responsiv este de a face ca paginile web să arate bine și să fie ușor de utilizat pe diferite dispozitive și dimensiuni de ecran, cum ar fi calculatoarele desktop, tabletele și telefoanele mobile.

Pentru a crea design-uri responsiv, putem utiliza `media queries`, care ne permit să aplicăm stiluri CSS diferite în funcție de dimensiunea ecranului. Astfel, putem modifica dimensiunile, poziționarea și aspectul elementelor pentru a se potrivi cu diverse dispozitive.

Exemplu de utilizare a `media queries` pentru un design responsiv:

```
.container {
  width: 100%;
}

.coloana {
  width: 100%;
}

/* Stiluri pentru ecrane cu lățimea mai mare de 768px */
@media screen and (min-width: 768px) {
  .coloana {
    width: 50%;
  }
}
```

În acest exemplu, se aplică stiluri diferite pentru elementele `.coloana` în funcție de lățimea ecranului. Pentru ecrane mai mici de 768px, `.coloana` va avea lățimea de 100%, iar pentru ecrane mai mari, va avea lățimea de 50%.

Utilizarea layout-urilor cu mai multe coloane și grile flexibile ne oferă posibilitatea de a crea design-uri complexe și structurate. De asemenea, designul responsiv ne permite să creăm pagini web adaptabile la diverse dispozitive, îmbunătățind astfel experiența utilizatorilor.

2.8. Exemple Practice

Exemplu 1: Stilizarea Textului

```
<style>
.titlu {
  color: #007bff;
  font-size: 32px;
}

.descriere {
  font-style: italic;
  color: #666;
}
</style>

<p class="titlu">Bun venit pe website-ul nostru!</p>
```

```
<p class="descriere">Aici veți găsi cele mai recente știri și noutăți din  
domeniul tehnologiei.</p>
```

Exemplu 2: Stilizarea Imaginilor

```
<style>  
.rotire {  
  transform: rotate(15deg);  
}  
  
.umbra {  
  box-shadow: 3px 3px 5px rgba(0, 0, 0, 0.5);  
}  
</style>  
  
  

```

Exemplu 3: Stilizarea Link-urilor

```
<style>  
.buton {  
  display: inline-block;  
  padding: 10px 20px;  
  background-color: #007bff;  
  color: white;  
  text-decoration: none;  
}  
  
.link {  
  color: #007bff;  
  text-decoration: underline;  
}  
  
.link:hover {  
  color: #ff6600;  
}  
</style>  
  
<a href="#" class="buton">Apasă aici</a>  
<a href="#" class="link">Citește mai mult</a>
```

Exemplu 4: Layout cu Mai Multe Coloane

```
<style>  
.container {  
  width: 100%;  
}  
  
.coloana {  
  width: 50%;
```

```

    float: left;
}
</style>

<div class="container">
    <div class="coloana">Coloana 1</div>
    <div class="coloana">Coloana 2</div>
</div>

```

Exemplu 5: Grilă Flexibilă

```

<style>
.grid {
    display: flex;
    flex-wrap: wrap;
}

.item {
    flex: 1;
    min-width: 200px;
    margin: 10px;
}
</style>

<div class="grid">
    <div class="item">Element 1</div>
    <div class="item">Element 2</div>
    <div class="item">Element 3</div>
</div>

```

Aceste exemple ilustrează modul în care putem aplica stiluri CSS pentru a crea un design atractiv și profesionist pentru paginile web. Combinând diferite concepte și tehnici, putem crea layout-uri complexe și adaptabile, oferind astfel o experiență plăcută utilizatorilor indiferent de dispozitivul pe care îl folosesc.

2.9. Aplicații

1. Creați o pagină web simplă cu un titlu și un paragraf, aplicând diferite stiluri pentru text și fundal.
2. Stilizați o listă neordonată și una ordonată, modificând stilul marcatorelor și al numerelor.
3. Realizați un meniu de navigare cu link-uri interne către ancore pe pagină.
4. Construiți o pagină web cu un tabel care să prezinte informații într-o formă structurată.

5. Adăugați o imagine și aplicați stiluri pentru a o face să fie centrată și cu un efect de umbră.
6. Creați un formular de contact cu câmpuri pentru nume, email și un mesaj, aplicând stiluri la acestea.
7. Stilizați un buton și adăugați un efect de tranziție la hover.
8. Realizați un layout cu două coloane, una fixă și una care să se adapteze la dimensiunea ecranului.
9. Creați un design responsiv pentru meniul de navigare, astfel încât să se afișeze optim pe dispozitive mobile.
10. Implementați o grilă flexibilă pentru a aranja mai multe elemente pe aceeași linie.
11. Stilizați o listă de produse cu imagini, nume și prețuri, aplicând efecte la hover.
12. Realizați un formular de înregistrare cu stiluri personalizate pentru câmpurile de completat.
13. Construiți un meniu dropdown personalizat cu opțiuni pentru utilizatori.
14. Adăugați animații la elementele paginii web pentru a crea o experiență mai interactivă.
15. Stilizați tabelele pentru a le face mai ușor de citit și de înțeles.
16. Creați un carusel cu imagini care să se miște automat și cu săgeți de navigare.
17. Realizați un design responsiv pentru o pagină cu articole de știri, astfel încât acestea să fie aranjate optim pe ecranele mici.
18. Adăugați o hartă interactivă cu locații pe pagina web și stilizați elementele pentru a se potrivi design-ului.
19. Stilizați un element HTML folosind toate cele patru componente principale ale Modelului de Tip Box.
20. Integrați o bibliotecă externă de CSS (de exemplu, Bootstrap sau Materialize) într-o pagină web și utilizați componentele acestuia pentru a crea un design modern și atrăgător.

Bibliografie

1. MDN Web Docs. "Cascading Style Sheets (CSS)". (<https://developer.mozilla.org/ro/docs/Web/CSS>)
2. W3Schools. "CSS Tutorial". (<https://www.w3schools.com/css/>)

3. Flanagan, D. (2011). "CSS: The Definitive Guide". O'Reilly Media.
4. Meyer, E. (2014). "CSS Pocket Reference". O'Reilly Media.
5. Duckett, J. (2018). "HTML and CSS: Design and Build Websites". Wiley.
6. CSS Working Group. "Cascading Style Sheets (CSS) Specifications".
(<https://www.w3.org/Style/CSS/>)

Anexe - Rezolvarea celor 20 de Probleme pentru Acasă

1. Problema: Stilizarea Textului

```
<!DOCTYPE html>
<html>
<head>
  <title>Stilizarea Textului</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <p class="titlu">Bun venit pe website-ul nostru!</p>
  <p class="descriere">Aici veți găsi cele mai recente știri și noutăți din
domeniul tehnologiei.</p>
</body>
</html>
```

```
/* styles.css */
.titlu {
  color: #007bff;
  font-size: 32px;
}

.descriere {
  font-style: italic;
  color: #666;
}
```

2. Problema: Stilizarea Imaginilor

```
<!DOCTYPE html>
<html>
<head>
  <title>Stilizarea Imaginilor</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  
  
```

```
</body>
</html>
```

```
/* styles.css */
.rotire {
  transform: rotate(15deg);
}

.umbra {
  box-shadow: 3px 3px 5px rgba(0, 0, 0, 0.5);
}
```

3. Problema: Stilizarea Link-urilor

```
<!DOCTYPE html>
<html>
<head>
  <title>Stilizarea Link-urilor</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <a href="#" class="buton">Apasă aici</a>
  <a href="#" class="link">Citește mai mult</a>
</body>
</html>
```

```
/* styles.css */
.buton {
  display: inline-block;
  padding: 10px 20px;
  background-color: #007bff;
  color: white;
  text-decoration: none;
}

.link {
  color: #007bff;
  text-decoration: underline;
}

.link:hover {
  color: #ff6600;
}
```

4. Problema: Layout cu Două Coloane

```
<!DOCTYPE html>
<html>
<head>
```

```

<title>Layout cu Două Coloane</title>
<link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="container">
    <div class="coluna">Coloana 1</div>
    <div class="coluna">Coloana 2</div>
  </div>
</body>
</html>

```

```

/* styles.css */
.container {
  width: 100%;
}

.coluna {
  width: 50%;
  float: left;
}

```

5. Problema: Grilă Flexibilă

```

<!DOCTYPE html>
<html>
<head>
  <title>Grilă Flexibilă</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="grid">
    <div class="item">Element 1</div>
    <div class="item">Element 2</div>
    <div class="item">Element 3</div>
  </div>
</body>
</html>

```

```

/* styles.css */
.grid {
  display: flex;
  flex-wrap: wrap;
}

.item {
  flex: 1;
  min-width: 200px;
  margin: 10px;
}

```

```
}
```

6. Problema: Design Responsiv

```
<!DOCTYPE html>
<html>
<head>
  <title>Design Responsiv</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="container">
    <div class="coluna">Conținut coloană 1</div>
    <div class="coluna">Conținut coloană 2</div>
  </div>
</body>
</html>
```

```
/* styles.css */
.container {
  width: 100%;
}

.coluna {
  width: 100%;
}
/* Stiluri pentru ecrane cu lățimea mai mare de 768px */
@media screen and (min-width: 768px) {
  .coluna {
    width: 50%;
  }
}
```

7. Problema: Meniu Dropdown Personalizat

```
<!DOCTYPE html>
<html>
<head>
  <title>Meniu Dropdown Personalizat</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="menu">
    <button class="dropbtn">Meniu</button>
    <div class="dropdown-content">
      <a href="#">Acasă</a>
      <a href="#">Despre Noi</a>
      <a href="#">Servicii</a>
      <a href="#">Contact</a>
    </div>
  </div>
```

```
</div>
</body>
</html>
```

```
/* styles.css */
.menu {
  position: relative;
  display: inline-block;
}

.dropbtn {
  background-color: #007bff;
  color: white;
  padding: 10px 20px;
  border: none;
  cursor: pointer;
}

.dropdown-content {
  display: none;
  position: absolute;
  background-color: #f9f9f9;
  min-width: 160px;
  box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
}

.dropdown-content a {
  display: block;
  padding: 12px 16px;
  text-decoration: none;
  color: #333;
}

.dropdown-content a:hover {
  background-color: #ddd;
}

.menu:hover .dropdown-content {
  display: block;
}
```

8. Problema: Stilizarea Tabelului

```
<!DOCTYPE html>
<html>
<head>
  <title>Stilizarea Tabelului</title>
  <link rel="stylesheet" href="styles.css">
</head>
```

```

<body>
  <table class="tabel">
    <tr>
      <th>Produs</th>
      <th>Preț</th>
    </tr>
    <tr>
      <td>Laptop</td>
      <td>2500 lei</td>
    </tr>
    <tr>
      <td>Telefon</td>
      <td>1500 lei</td>
    </tr>
  </table>
</body>
</html>

```

```

/* styles.css */
.tabel {
  border-collapse: collapse;
  width: 100%;
}

.tabel th, .tabel td {
  border: 1px solid #ddd;
  padding: 8px;
  text-align: left;
}

.tabel th {
  background-color: #f2f2f2;
}

```

9. Problema: Carusel cu Imagini

```

<!DOCTYPE html>
<html>
<head>
  <title>Carusel cu Imagini</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="slideshow-container">
    <div class="slide fade">
      
    </div>
    <div class="slide fade">
      
    </div>
  </div>

```

```

    </div>
    <div class="slide fade">
        
    </div>
    <a class="prev" onclick="plusSlides(-1)">#10094;</a>
    <a class="next" onclick="plusSlides(1)">#10095;</a>
  </div>
</body>
</html>

```

```

/* styles.css */
.slide {
    display: none;
}

.prev, .next {
    cursor: pointer;
    position: absolute;
    top: 50%;
    width: auto;
    padding: 16px;
    margin-top: -22px;
    color: white;
    font-weight: bold;
    font-size: 18px;
    transition: 0.6s ease;
    border-radius: 0 3px 3px 0;
}

.next {
    right: 0;
    border-radius: 3px 0 0 3px;
}

.fade {
    animation: fade 2s infinite;
}

@keyframes fade {
    from {opacity: .4}
    to {opacity: 1}
}

```

10. Problema: Design Responsiv pentru Articole de Știri

```

<!DOCTYPE html>
<html>
<head>
  <title>Design Responsiv pentru Articole de Știri</title>

```

```

    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <div class="container">
        <div class="articol">
            
            <h2>Titlul Stire 1</h2>
            <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed euismod elit sit amet neque blandit, ac rutrum mauris vestibulum.</p>
        </div>
        <div class="articol">
            
            <h2>Titlul Stire 2</h2>
            <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed euismod elit sit amet neque blandit, ac rutrum mauris vestibulum.</p>
        </div>
        <div class="articol">
            
            <h2>Titlul Stire 3</h2>
            <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed euismod elit sit amet neque blandit, ac rutrum mauris vestibulum.</p>
        </div>
    </div>
</body>
</html>

```

```

/* styles.css */
.container {
    display: flex;
    flex-wrap: wrap;
}

.articol {
    flex: 1;
    min-width: 300px;
    margin: 10px;
}

img {
    width: 100%;
}

```

11. Problema: Design pentru Formularul de Contact

```

<!DOCTYPE html>
<html>
<head>
    <title>Design pentru Formularul de Contact</title>
    <link rel="stylesheet" href="styles.css">

```

```

</head>
<body>
  <form class="formular">
    <label for="nume">Nume:</label>
    <input type="text" id="nume" name="nume" required>

    <label for="email">Email:</label>
    <input type="email" id="email" name="email" required>

    <label for="mesaj">Mesaj:</label>
    <textarea id="mesaj" name="mesaj" required></textarea>

    <input type="submit" value="Trimite">
  </form>
</body>
</html>

```

```

/* styles.css */
.formular {
  width: 300px;
  margin: 0 auto;
}

label {
  display: block;
  margin-bottom: 5px;
}

input, textarea {
  width: 100%;
  padding: 10px;
  margin-bottom: 15px;
  border: 1px solid #ddd;
  border-radius: 5px;
}

input[type="submit"] {
  background-color: #007bff;
  color: white;
  cursor: pointer;
}

```

12. Problema: Meniu Fixat în Partea de Sus

```

<!DOCTYPE html>
<html>
<head>
  <title>Meniu Fixat în Partea de Sus</title>
  <link rel="stylesheet" href="styles.css">

```

```

</head>
<body>
  <div class="menu">
    <a href="#">Acasă</a>
    <a href="#">Despre Noi</a>
    <a href="#">Servicii</a>
    <a href="#">Contact</a>
  </div>
</body>
</html>

```

```

/* styles.css */
.menu {
  position: fixed;
  top: 0;
  left: 0;
  width: 100%;
  background-color: #007bff;
  padding: 10px;
}

.menu a {
  color: white;
  margin-right: 20px;
  text-decoration: none;
}

.menu a:hover {
  color: #ff6600;
}

```

13. Problema: Efecte de Tranziție

```

<!DOCTYPE html>
<html>
<head>
  <title>Efecte de Tranziție</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="box"></div>
</body>
</html>

```

```

/* styles.css */
.box {
  width: 100px;
  height: 100px;
}

```

```

background-color: #007bff;
transition: width 1s, height 1s, background-color 1s;
}

.box:hover {
width: 150px;
height: 150px;
background-color: #ff6600;
}

```

14. Problema: Meniu Navigare Orizantal

```

<!DOCTYPE html>
<html>
<head>
  <title>Meniu Navigare Orizantal</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <ul class="menu">
    <li><a href="#">Acasă</a></li>
    <li><a href="#">Despre Noi</a></li>
    <li><a href="#">Servicii</a></li>
    <li><a href="#">Contact</a></li>
  </ul>
</body>
</html>

```

```

/* styles.css */
.menu {
  list-style-type: none;
  margin: 0;
  padding: 0;
  display: flex;
}

.menu li {
  margin-right: 20px;
}

.menu a {
  color: #007bff;
  text-decoration: none;
}

.menu a:hover {
  color: #ff6600;
}

```

15. Problema: Stilizarea Butonului Radio

```
<!DOCTYPE html>
<html>
<head>
  <title>Stilizarea Butonului Radio</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <label class="container">Opțiune 1
    <input type="radio" name="radio">
    <span class="checkmark"></span>
  </label>
  <label class="container">Opțiune 2
    <input type="radio" name="radio">
    <span class="checkmark"></span>
  </label>
</body>
</html>
```

```
/* styles.css */
.container {
  position: relative;
  padding-left: 30px;
  cursor: pointer;
  user-select: none;
}

.container input {
  position: absolute;
  opacity: 0;
  cursor: pointer;
}

.checkmark {
  position: absolute;
  top: 0;
  left: 0;
  height: 20px;
  width: 20px;
  border: 1px solid #ccc;
}

.container:hover .checkmark {
  border: 1px solid #007bff;
}

.container input:checked ~ .checkmark {
  background-color: #007bff;
}
```

```

}

.checkmark:after {
  content: "";
  position: absolute;
  display: none;
}

.container input:checked ~ .checkmark:after {
  display: block;
}

.container .checkmark:after {
  left: 7px;
  top: 3px;
  width: 5px;
  height: 10px;
  border: solid white;
  border-width: 0 3px 3px 0;
  transform: rotate(45deg);
  -webkit-transform: rotate(45deg);
  -ms-transform: rotate(45deg);
}

```

16. Problema: Design pentru Butonul de Înregistrare

```

<!DOCTYPE html>
<html>
<head>
  <title>Design pentru Butonul de Înregistrare</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <button class="buton-primar">Înregistrare</button>
</body>
</html>

```

```

/* styles.css */
.buton-primar {
  background-color: #007bff;
  color: white;
  padding: 10px 20px;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}

.buton-primar:hover {
  background-color: #ff6600;
}

```

```
}
```

17. Problema: Design pentru Lista cu Iconițe

```
<!DOCTYPE html>
<html>
<head>
  <title>Design pentru Lista cu Iconițe</title>
  <link rel="stylesheet" href="styles.css">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.0.0-beta3/css/all.min.css">
</head>
<body>
  <ul class="lista-iconite">
    <li><i class="fas fa-check"></i> Element 1</li>
    <li><i class="fas fa-check"></i> Element 2</li>
    <li><i class="fas fa-check"></i> Element 3</li>
  </ul>
</body>
</html>
```

```
/* styles.css */
.lista-iconite {
  list-style-type: none;
  padding: 0;
}

.lista-iconite li {
  padding-left: 30px;
  position: relative;
}

.lista-iconite li i {
  position: absolute;
  left: 0;
}
```

18. Problema: Design pentru Efect de Umbră la Imagini

```
<!DOCTYPE html>
<html>
<head>
  <title>Design pentru Efect de Umbră la Imagini</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  
</body>
</html>
```

```
/* styles.css */
.umbra {
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
}
```

19. Problema: Efect de Tranziție la Hover

```
<!DOCTYPE html>
<html>
<head>
    <title>Efect de Tranziție la Hover</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <div class="box-tranzitie"></div>
</body>
</html>
```

```
/* styles.css */
.box-tranzitie {
    width: 100px;
    height: 100px;
    background-color: #007bff;
    transition: background-color 0.5s ease;
}

.box-tranzitie:hover {
    background-color: #ff6600;
}
```

20. Problema: Meniu cu Efect de Deschidere și Închidere

```
<!DOCTYPE html>
<html>
<head>
    <title>Meniu cu Efect de Deschidere și Închidere</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <div class="menu-toggle">
        <button class="buton-toggle" onclick="toggleMenu()">Meniu</button>
        <ul class="menu-lista" id="menu-lista">
            <li><a href="#">Acasă</a></li>
            <li><a href="#">Despre Noi</a></li>
            <li><a href="#">Servicii</a></li>
            <li><a href="#">Contact</a></li>
        </ul>
    </div>
```

```
</body>
</html>
```

```
/* styles.css */
.meniu-toggle {
  position: relative;
}

.buton-toggle {
  background-color: #007bff;
  color: white;
  padding: 10px 20px;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}

.meniu-lista {
  display: none;
  position: absolute;
  top: 40px;
  right: 0;
  background-color: #f9f9f9;
  list-style-type: none;
  padding: 0;
  border: 1px solid #ddd;
  border-radius: 5px;
}

.meniu-lista li {
  padding: 8px 16px;
}

.buton-toggle.active + .menu-lista {
  display: block;
}
```

LUCRAREA 3

3.1. Obiectivele lucrării

În această lucrare, vom introduce conceptele de bază legate de tabele și formulare în HTML și vom discuta importanța lor în dezvoltarea paginilor web.

3.2. Introducere

Tabelele și formularele sunt două elemente importante ale limbajului HTML care permit structurarea și interacțiunea cu datele pe o pagină web. Tabelele sunt utilizate pentru a organiza datele în rânduri și coloane, oferind o modalitate eficientă de prezentare a informațiilor într-un format structurat și ușor de înțeles pentru utilizatori. Acestea pot fi folosite pentru a crea liste, rapoarte sau grile de date.

Pe de altă parte, formularele sunt utilizate pentru a colecta informații de la utilizatori. Ele conțin diverse elemente precum casete de text, câmpuri de selectare, butoane de radio sau de bifa, care permit utilizatorilor să introducă și să transmită date către server. Formularele sunt esențiale pentru interacțiunea dintre utilizatori și aplicații web, cum ar fi înregistrarea utilizatorilor, căutările, trimiterea de mesaje sau completarea de sondaje online.

Importanța tabelelor și formularelor în dezvoltarea web

Tabelele și formularele joacă un rol crucial în dezvoltarea paginilor web, deoarece permit organizarea și prezentarea datelor într-un mod structurat, ușor de navigat și înțeles de către utilizatori. Tabelele facilitează structurarea informațiilor într-un format tabular, permitând utilizatorilor să vadă datele într-o manieră ordonată și logică.

De asemenea, formularele oferă interactivitate și posibilitatea de a colecta date de la utilizatori. Prin intermediul formularelor, site-urile web devin mai dinamice și capabile să interacționeze cu utilizatorii, oferindu-le o experiență personalizată și facilitând procesele de transmitere a informațiilor.

În concluzie, cunoașterea și utilizarea corespunzătoare a tabelelor și formularelor în HTML sunt esențiale pentru dezvoltatorii web, deoarece acestea permit crearea de pagini web atractive, funcționale și interactivă, îmbunătățind astfel experiența utilizatorilor și eficiența site-

urilor web. În capitolele următoare, vom explora în detaliu modul de utilizare a tabelelor și formularelor în HTML, vom învăța despre elementele și atributele corespunzătoare și vom realiza exemple practice pentru a ilustra utilizarea lor în cadrul paginilor web.

3.3. Tabele în HTML

Tabelele sunt utilizate pentru a organiza datele într-un format tabular, cu rânduri și coloane. În HTML, utilizăm următoarele elemente pentru a crea tabele:

`<table>`

Elementul `<table>` este containerul principal pentru întreaga tabelă. Acesta definește începutul și sfârșitul tabelului.

`<tr>`

Elementul `<tr>` reprezintă un rând în cadrul tabelului. Acesta conține una sau mai multe celule de tabel (`<td>` sau `<th>`).

`<td>`

Elementul `<td>` reprezintă o celulă de tabel și conține datele propriu-zise ale tabelului. Acesta este utilizat pentru a insera informații într-un rând.

`<th>`

Elementul `<th>` reprezintă o celulă de antet (header cell) și este utilizat pentru a defini capul de tabel. De obicei, conține text mai mare și bold pentru a distinge antetele de restul datelor.

Stilizarea tabelelor cu CSS

După ce am creat o tabelă în HTML, putem stiliza aspectul acesteia cu ajutorul CSS. CSS ne permite să ajustăm culorile, dimensiunile, fonturile și marginile pentru a face tabelele mai atractive și a se potrivi cu design-ul general al site-ului.

Putem folosi selecționarea elementelor `<table>`, `<tr>`, `<td>` și `<th>` sau clase și ID-uri pentru a aplica stiluri diferite pentru fiecare parte a tabelului.

De exemplu, pentru a schimba culoarea de fundal a tabelului, putem folosi următoarea regulă CSS:

```
table {
  background-color: #f2f2f2;
}
```

Pentru a stiliza antetul tabelului cu o culoare de fundal diferită și text alb, putem folosi:

```
th {  
  background-color: #333;  
  color: #fff;  
}
```

Astfel, putem personaliza aspectul tabelelor pentru a se potrivi cu design-ul site-ului și pentru a face datele mai ușor de citit și înțeles pentru utilizatori.

3.4. Formulare în HTML

Elementele de bază ale formularelor în HTML

Formularele reprezintă o componentă esențială a paginilor web, permitând interacțiunea utilizatorilor cu site-ul prin colectarea și trimiterea datelor către server. În HTML, utilizăm următoarele elemente pentru a construi formulare:

`<form>`

Elementul `<form>` este containerul principal al formularului și definește începutul și sfârșitul acestuia. Acesta include diverse elemente precum câmpuri de text, butoane, casete de selectare și altele, care permit utilizatorilor să introducă date.

`<input>`

Elementul `<input>` este utilizat pentru a crea diferite tipuri de câmpuri de intrare, cum ar fi casete de text, butoane radio, casete de bifa și multe altele. Atributele precum `type`, `name` și `value` sunt folosite pentru a specifica tipul de intrare, numele câmpului și valoarea inițială.

`<textarea>`

Elementul `<textarea>` permite utilizatorilor să introducă texte lungi sau să ofere comentarii extinse. Acesta are un tag de deschidere și închidere și poate fi redimensionat pentru a permite introducerea de texte mai lungi.

`<select>` și `<option>`

Elementul `<select>` este utilizat pentru a crea liste de selecție, iar elementul `<option>` definește opțiunile din cadrul listei. Acesta permite utilizatorilor să selecteze o singură opțiune dintr-o listă predefinită.

Validarea datelor în formulare

Validarea datelor introduse de utilizatori în formulare este un aspect important pentru asigurarea corectitudinii și securității informațiilor colectate. HTML5 introduce o serie de

attribute speciale care permit validarea direct în cadrul formularului, cum ar fi `required`, `pattern`, `min`, `max` și altele.

De exemplu, putem utiliza atributul `required` pentru a forța utilizatorii să completeze un câmp înainte de a trimite formularul:

```
<input type="text" name="username" required>
```

De asemenea, putem folosi atributul `pattern` pentru a specifica un format specific pentru un câmp:

```
<input type="text" name="phone" pattern="[0-9]{10}" title="Introduceți un număr de telefon valid (10 cifre)">
```

În plus, pentru validări mai complexe sau personalizate, putem utiliza JavaScript pentru a verifica datele introduse de utilizatori și pentru a afișa mesaje de eroare corespunzătoare.

Transmiterea datelor din formulare

După ce utilizatorii completează formularele și apasă butonul de trimitere (Submit), datele trebuie să fie prelucrate și transmise către server pentru a fi procesate. Pentru acest scop, utilizăm atributul `action` al elementului ``, care specifică URL-ul sau adresa către care vor fi trimise datele.

```
<form action="/procesare_formular.php" method="post">
  <!-- Elemente de intrare a datelor -->
  <!-- Butonul de trimitere -->
</form>
```

În exemplul de mai sus, datele vor fi trimise către fișierul `procesare\_formular.php` de pe server folosind metoda HTTP `POST`.

Elemente speciale pentru formulare

În afara elementelor de bază precum `` și ``, există și alte elemente speciale care pot fi utilizate în formulare pentru a colecta diverse tipuri de informații.</p></div><div data-bbox="173 699 613 718" data-label="Text"><p>`<input type="radio">` și `<input type="checkbox">`</p></div><div data-bbox="113 726 884 794" data-label="Text"><p>Aceste elemente permit utilizatorilor să selecteze opțiuni dintr-o listă de variante, fie pentru a alege una dintre ele (`radio`) sau pentru a bifa una sau mai multe variante (`checkbox`).</p></div><div data-bbox="173 801 589 819" data-label="Text"><p>`<input type="submit">` și `<input type="reset">`</p></div><div data-bbox="113 827 884 894" data-label="Text"><p>Elementul `<input>` cu atributul `type="submit"` permite utilizatorilor să trimită formularul, în timp ce `<input type="reset">` permite resetarea valorilor câmpurilor la valorile inițiale.</p></div><div data-bbox="478 922 518 937" data-label="Page-Footer"><p>- 4 -</p></div>

```
`<input type="file">`
```

Acest element permite utilizatorilor să încarce fișiere către server.

Implementarea formularului cu stiluri CSS

Formularele pot fi îmbunătățite estetic prin adăugarea stilurilor CSS. Putem schimba aspectul câmpurilor de intrare, culorile, dimensiunile și altele pentru a se potrivi cu design-ul general al site-ului.

De exemplu, putem stiliza un câmp de intrare cu o culoare de fundal personalizată și o margine mai mare:

```
input[type="text"] {  
  background-color: #f2f2f2;  
  border: 2px solid #ccc;  
  padding: 10px;  
}
```

Validarea datelor cu JavaScript

În timp ce atributele HTML5 precum `required`, `pattern`, `min`, `max` pot oferi o validare de bază în cadrul formularului, putem folosi și JavaScript pentru validarea mai avansată a datelor introduse de utilizatori. Aceasta ne oferă flexibilitate și control mai mare asupra validării.

Putem crea funcții JavaScript personalizate pentru a verifica datele din câmpurile de intrare și pentru a afișa mesaje de eroare corespunzătoare. De exemplu, să presupunem că avem un câmp de intrare pentru introducerea vârstei utilizatorului și dorim ca acesta să fie un număr întreg pozitiv:

```
<label for="age">Vârsta:</label>  
<input type="number" id="age" name="age">  
  
<p id="age-error" style="color: red; display: none;">Vă rugăm să introduceți o  
vârstă validă.</p>  
  
<script>  
  const ageInput = document.getElementById("age");  
  const ageError = document.getElementById("age-error");  
  
  ageInput.addEventListener("input", function() {  
    const age = parseInt(ageInput.value);  
    if (isNaN(age) || age < 0) {  
      ageError.style.display = "block";  
    } else {  
      ageError.style.display = "none";  
    }  
  });  
</script>
```

```
}  
});  
</script>
```

În exemplul de mai sus, am creat o funcție care verifică dacă valoarea introdusă în câmpul de vârstă este un număr întreg pozitiv. Dacă nu este, se afișează un mesaj de eroare.

Transmiterea datelor către server

După ce utilizatorii completează formularele și datele sunt validate cu succes, acestea pot fi trimise către server pentru prelucrare. În general, acest lucru se face prin intermediul metodelor HTTP `POST` sau `GET`, în funcție de cerințele aplicației web.

La nivel de HTML, specificăm metoda de trimitere a datelor folosind atributul `method` în elementul ``:

```
<form action="/procesare_formular.php" method="post">  
  <!-- Elemente de intrare a datelor -->  
  <!-- Butonul de trimitere -->  
</form>
```

În exemplul de mai sus, datele vor fi trimise prin metoda `POST` către fișierul `procesare\_formular.php` de pe server.

Implementarea formularului cu stiluri CSS

Aspectul formularului poate fi îmbunătățit cu ajutorul stilurilor CSS. Putem schimba fonturile, culorile, dimensiunile, stilul butoanelor și altele pentru a crea o experiență vizuală mai plăcută pentru utilizatori.

```
/* Exemplu de stilizare a butonului de trimitere */  
input[type="submit"] {  
  background-color: #4CAF50;  
  color: white;  
  padding: 12px 20px;  
  border: none;  
  border-radius: 4px;  
  cursor: pointer;  
}  
  
/* Exemplu de stilizare a câmpului de intrare pentru e-mail */  
input[type="email"] {  
  padding: 10px;  
  border: 1px solid #ccc;  
}
```

Exemple practice de formulare

Pentru a înțelege mai bine cum se implementează formularele în HTML și cum le stilizăm cu CSS, vom studia câteva exemple practice.

Exemplu 1: Formular de înregistrare utilizator

Vom crea un formular simplu pentru înregistrarea utilizatorilor, solicitând informații precum numele, adresa de email și parola.

```
<form action="/inregistrare" method="post">
  <label for="nume">Nume:</label>
  <input type="text" id="nume" name="nume" required>

  <label for="email">Email:</label>
  <input type="email" id="email" name="email" required>

  <label for="parola">Parolă:</label>
  <input type="password" id="parola" name="parola" required>

  <input type="submit" value="Înregistrare">
</form>
```

Exemplu 2: Formular de contact

Vom crea un formular pentru contact, în care utilizatorii pot trimite întrebări și comentarii.

```
<form action="/contact" method="post">
  <label for="nume">Nume:</label>
  <input type="text" id="nume" name="nume" required>

  <label for="email">Email:</label>
  <input type="email" id="email" name="email" required>

  <label for="mesaj">Mesaj:</label>
  <textarea id="mesaj" name="mesaj" rows="4" required></textarea>

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

Stilizarea formularului cu CSS

Folosind CSS, putem personaliza aspectul formularului pentru a se potrivi cu design-ul general al site-ului. Putem schimba fonturile, culorile, dimensiunile câmpurilor și stilul butoanelor pentru a crea o experiență vizuală coerentă.

```
/* Stilizarea câmpurilor de intrare */
input[type="text"],
input[type="email"],
```

```

input[type="password"],
textarea {
    width: 100%;
    padding: 12px;
    margin: 8px 0;
    border: 1px solid #ccc;
    border-radius: 4px;
}

/* Stilizarea butoanelor */
input[type="submit"] {
    background-color: #4CAF50;
    color: white;
    padding: 12px 20px;
    border: none;
    border-radius: 4px;
    cursor: pointer;
}

input[type="submit"]:hover {
    background-color: #45a049;
}

```

Validarea formularului cu JavaScript

Validarea formularului este un aspect esențial pentru asigurarea că datele introduse de utilizatori sunt corecte și pot fi prelucrate corespunzător. În cazul în care utilizatorii nu completează corect sau omit anumite câmpuri, putem folosi JavaScript pentru a le avertiza și a le solicita să completeze informațiile lipsă.

Vom crea o funcție JavaScript pentru validarea formularului de înregistrare, în care vom verifica dacă câmpurile pentru nume, email și parolă au fost completate corect:

```

<script>
    function validaFormular(){
        var nume = document.getElementById("nume").value;
        var email = document.getElementById("email").value;
        var parola = document.getElementById("parola").value;

        if (nume == "" || email == "" || parola == "") {
            alert("Vă rugăm să completați toate câmpurile.");
            return false;
        }
        return true;
    }
</script>

```

În exemplul de mai sus, am creat funcția `validaFormular()` care verifică dacă câmpurile pentru nume, email și parolă au fost completate. Dacă vreunul dintre aceste câmpuri este gol, utilizatorul va primi o alertă cu mesajul "Vă rugăm să completați toate câmpurile." și trimiterea formularului va fi oprită.

Pentru a aplica funcția de validare, o putem adăuga ca atribut `onsubmit` în elementul ``:

```
<form action="/inregistrare" method="post" onsubmit="return validaFormular()">
  <!-- Elemente de intrare a datelor -->
  <!-- Butonul de trimitere -->
</form>
```

Trimiterea formularului către server

După ce datele au fost validate cu succes, formularul poate fi trimis către server pentru prelucrare. În cazul în care utilizăm un limbaj de programare server-side cum ar fi PHP, datele pot fi preluate și procesate pentru a fi salvate într-o bază de date sau pentru a fi trimise pe email.

Implementarea formularului cu stiluri CSS

Cu ajutorul CSS, putem adăuga stiluri pentru a îmbunătăți aspectul formularului și pentru a-l face mai atractiv. Putem schimba culorile, fonturile, dimensiunile și alte proprietăți pentru a se potrivi cu design-ul general al site-ului.

```
/* Stilizarea câmpurilor de intrare */
input[type="text"],
input[type="email"],
input[type="password"],
textarea {
  width: 100%;
  padding: 12px;
  margin-bottom: 10px;
  border: 1px solid #ccc;
  border-radius: 4px;
}

/* Stilizarea butonului de trimitere */
input[type="submit"] {
  background-color: #4CAF50;
  color: white;
  padding: 12px 20px;
  border: none;
  border-radius: 4px;
  cursor: pointer;
```

```

}

input[type="submit"]:hover {
    background-color: #45a049;
}

```

Exemple practice de formulare

Exemplu 3: Formular de căutare

Vom crea un formular simplu pentru căutarea unui termen într-un motor de căutare.

```

<form action="/cautare" method="get">
    <input type="text" name="cuvant_cautat" placeholder="Introduceți termenul
căutat">
    <input type="submit" value="Caută">
</form>

```

Exemplu 4: Formular de abonare la newsletter

Vom crea un formular pentru abonarea la newsletter, unde utilizatorii pot introduce adresa de email pentru a primi actualizări periodice.

```

<form action="/abonare" method="post">
    <label for="email">Adresă email:</label>
    <input type="email" id="email" name="email" required>
    <input type="submit" value="Abonează-te">
</form>

```

Casete de selectare (Checkboxes) și Casete radio (Radio buttons)

Casetele de selectare (checkboxes) și casetele radio (radio buttons) sunt elemente importante ale formulelor care permit utilizatorilor să facă alegeri multiple sau să selecteze o opțiune dintr-o listă.

Vom crea un exemplu de casetă de selectare pentru a permite utilizatorilor să selecteze opțiuni multiple, cum ar fi limbajele de programare preferate:

```

<form action="/preferinte" method="post">
    <label for="limbaj1">HTML</label>
    <input type="checkbox" id="limbaj1" name="limbaj" value="HTML">

    <label for="limbaj2">CSS</label>
    <input type="checkbox" id="limbaj2" name="limbaj" value="CSS">

    <label for="limbaj3">JavaScript</label>
    <input type="checkbox" id="limbaj3" name="limbaj" value="JavaScript">

    <input type="submit" value="Trimite">
</form>

```

Vom crea un exemplu de casetă radio pentru a permite utilizatorilor să selecteze o singură opțiune, cum ar fi genul:

```
<form action="/gen" method="post">
  <label for="gen1">Masculin</label>
  <input type="radio" id="gen1" name="gen" value="masculin">

  <label for="gen2">Feminin</label>
  <input type="radio" id="gen2" name="gen" value="feminin">

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

Liste derulante (Dropdowns)

Listele derulante (dropdowns) sunt utilizate pentru a permite utilizatorilor să selecteze o opțiune dintr-o listă, care se deschide într-o casetă derulantă.

```
<form action="/oras" method="post">
  <label for="oras">Selectați orașul:</label>
  <select id="oras" name="oras">
    <option value="bucuresti">București</option>
    <option value="cluj">Cluj</option>
    <option value="iasi">Iași</option>
    <option value="timisoara">Timișoara</option>
  </select>

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

Gestionarea datelor introduse de utilizatori

După ce utilizatorii completează formularul și trimit datele către server, acestea pot fi prelucrate cu ajutorul unui limbaj de programare server-side cum ar fi PHP. Datele pot fi salvate într-o bază de date, trimise prin email sau folosite în alte scopuri specifice aplicației.

3.5. Studiu de Caz: Sistem de Rezervări Online

Vom analiza cum sunt utilizate tabelele și formularele într-un sistem de rezervări online pentru a colecta și afișa informații despre rezervări.

Sistemul de rezervări online permite utilizatorilor să facă rezervări pentru diverse servicii sau evenimente, cum ar fi rezervări pentru hoteluri, restaurante, bilete la spectacole sau participarea la evenimente. Acesta implică utilizarea tabelor pentru organizarea și afișarea disponibilității și a opțiunilor de rezervare, precum și formularele pentru colectarea datelor de contact și preferințele utilizatorilor.

Exemplu de Tabel pentru Disponibilitate

```
<table>
  <tr>
    <th>Data</th>
    <th>Disponibilitate</th>
    <th>Preț</th>
  </tr>
  <tr>
    <td>2023-08-15</td>
    <td>Libere</td>
    <td>100 lei</td>
  </tr>
  <tr>
    <td>2023-08-16</td>
    <td>Ocupate</td>
    <td>150 lei</td>
  </tr>
  <tr>
    <td>2023-08-17</td>
    <td>Libere</td>
    <td>100 lei</td>
  </tr>
</table>
```

Exemplu de Formular de Rezervare

```
<form action="/rezervare" method="post">
  <label for="nume">Nume:</label>
  <input type="text" id="nume" name="nume" required>

  <label for="email">Email:</label>
  <input type="email" id="email" name="email" required>

  <label for="data">Data rezervării:</label>
  <input type="date" id="data" name="data" required>

  <label for="mesaj">Mesaj (opțional):</label>
  <textarea id="mesaj" name="mesaj" rows="4"></textarea>

  <input type="submit" value="Rezervă Acum">
</form>
```

3.6. Aplicații

- I. Creați un tabel simplu pentru a afișa informații despre produsele dintr-un magazin online. Tabelul trebuie să conțină coloane pentru numele produsului, preț și stoc disponibil.

2. Construiți un formular de înregistrare pentru un eveniment. Formularul trebuie să conțină câmpuri pentru nume, adresă de email, număr de telefon și opțiunea de a selecta tipul de bilet dorit (normal, VIP, etc.).
3. Realizați un tabel cu programul de lucru al unei companii, afișând orele de început și de sfârșit ale fiecărei ture de lucru.
4. Creați un formular de contact pentru un site web, cu câmpuri pentru nume, adresă de email, subiectul mesajului și conținutul mesajului.
5. Construiți un tabel pentru a afișa informații despre studenții unei școli, inclusiv nume, vârstă, clasă și medie.
6. Realizați un formular de înscriere la un newsletter, cu un câmp pentru adresa de email și un buton de subscriere.
7. Creați un tabel pentru a afișa detalii despre cărțile disponibile într-o bibliotecă, precum numele cărții, autorul, anul publicării și stocul disponibil.
8. Construiți un formular pentru a colecta recenzii de la clienți, cu câmpuri pentru nume, adresa de email, evaluarea și comentariul.
9. Realizați un tabel pentru a afișa rezultatele unei competiții sportive, cu coloane pentru numele sportivilor și punctajul obținut.
10. Creați un formular de înregistrare pentru un curs online, cu câmpuri pentru nume, adresă de email, număr de telefon și opțiunea de a selecta varianta de plată (în rate sau integral).
11. Construiți un tabel cu detalii despre angajații unei companii, inclusiv numele, funcția, data angajării și salariul.
12. Realizați un formular pentru a colecta datele de contact ale participanților la un eveniment, cu câmpuri pentru nume, adresă de email și număr de telefon.
13. Creați un tabel pentru a afișa programul de proiecții la un cinematograf, cu coloane pentru numele filmului, ora de început și durata.
14. Construiți un formular pentru a colecta cereri de oferte de la potențiali clienți, cu câmpuri pentru nume, adresă de email, tipul de serviciu solicitat și un câmp pentru descrierea cererii.
15. Realizați un tabel cu informații despre echipa unui club sportiv, inclusiv numele jucătorilor, poziția și numărul de tricou.
16. Creați un formular de înscriere la un workshop, cu câmpuri pentru nume, adresă de email, ocupație și opțiunea de a alege unul din mai multe ateliere disponibile.

17. Construiți un tabel pentru a afișa rezultatele unui sondaj de opinie, cu coloane pentru întrebări și opțiunile de răspuns.
18. Realizați un formular pentru a colecta datele de contact ale clienților pentru un magazin online, cu câmpuri pentru nume, adresă de livrare, adresă de facturare și număr de telefon.
19. Creați un tabel pentru a afișa programul de concerte la un festival, cu coloane pentru numele formațiilor, data și locația.
20. Construiți un formular pentru a colecta feedback de la vizitatorii unui site web, cu câmpuri pentru nume, adresă de email, categoria feedback-ului (pozitiv, negativ sau sugestie) și un câmp pentru comentarii.

Bibliografie

1. MDN Web Docs - Tabele în HTML:
<https://developer.mozilla.org/ro/docs/Web/HTML/Element/table>
2. MDN Web Docs - Formulare în HTML:
<https://developer.mozilla.org/ro/docs/Web/HTML/Element/form>
3. W3Schools - HTML Tables: https://www.w3schools.com/html/html_tables.asp
4. W3Schools - HTML Forms: https://www.w3schools.com/html/html_forms.asp
5. HTML.com - How to Create Tables in HTML: <https://html.com/tables/>
6. HTML.com - How to Create Forms in HTML: <https://html.com/forms/>
7. Shay Howe - Learn to Code HTML & CSS: <https://learn.shayhowe.com/html-css/>
8. Jon Duckett - HTML & CSS: Design and Build Websites
9. Jennifer Niederst Robbins - Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics
10. David Sawyer McFarland - HTML, CSS & JavaScript: Web Publishing in One Hour a Day
11. Eric Meyer - CSS: The Definitive Guide: Visual Presentation for the Web
12. Smashing Magazine - The CSS Grid Layout:
<https://www.smashingmagazine.com/2018/05/guide-css-layout/>
13. Bootstrap - The most popular HTML, CSS, and JS library in the world:
<https://getbootstrap.com/>
14. Materialize CSS - A modern responsive front-end framework based on Material Design:
<https://materializecss.com/>
15. CSS-Tricks - Guides, Almanac, and Reference for CSS: <https://css-tricks.com/>

Anexe - Rezolvarea celor 20 de Probleme pentru Acasă

1. Creați un tabel simplu pentru a afișa informații despre produsele dintr-un magazin online.

Tabelul trebuie să conțină coloane pentru numele produsului, preț și stoc disponibil.

Pentru a rezolva această cerință, vom crea un tabel simplu pentru a afișa informații despre produsele dintr-un magazin online. Tabelul va avea trei coloane: una pentru numele produsului, alta pentru preț și a treia pentru stocul disponibil.

```
<!DOCTYPE html>
<html>
<head>
  <title>Magazin Online - Produse</title>
</head>
<body>
  <h1>Magazin Online - Produse</h1>
  <table border="1">
    <tr>
      <th>Nume Produs</th>
      <th>Preț</th>
      <th>Stoc Disponibil</th>
    </tr>
    <tr>
      <td>Telefon Smart</td>
      <td>1500 Lei</td>
      <td>10 bucăți</td>
    </tr>
    <tr>
      <td>Laptop Ultrabook</td>
      <td>3500 Lei</td>
      <td>5 bucăți</td>
    </tr>
    <tr>
      <td>Televizor LED</td>
      <td>2500 Lei</td>
      <td>8 bucăți</td>
    </tr>
  </table>
</body>
</html>
```

În acest exemplu, am creat un tabel cu trei coloane: "Nume Produs", "Preț" și "Stoc Disponibil". Fiecare rând din tabel reprezintă un produs diferit, iar în fiecare celulă am introdus informațiile corespunzătoare despre produs. Pentru a face tabelul mai ușor de citit, am adăugat o bordură în jurul celulelor utilizând atributul `border="1"`.

Acesta este un exemplu simplu, dar puteți extinde tabelul și să adăugați mai multe produse și informații, în funcție de nevoile specifice ale magazinului online.

2. *Construiți un formular de înregistrare pentru un eveniment. Formularul trebuie să conțină câmpuri pentru nume, adresă de email, număr de telefon și opțiunea de a selecta tipul de bilet dorit (normal, VIP, etc.).*

Pentru a rezolva această cerință, vom construi un formular de înregistrare pentru un eveniment. Formularul va conține câmpuri pentru nume, adresă de email, număr de telefon și opțiunea de a selecta tipul de bilet dorit (normal, VIP, etc.).

```
<!DOCTYPE html>
<html>
<head>
  <title>Formular de Înregistrare - Eveniment</title>
</head>
<body>
  <h1>Formular de Înregistrare - Eveniment</h1>
  <form action="/submit" method="post">
    <label for="nume">Nume:</label>
    <input type="text" id="nume" name="nume" required><br>

    <label for="email">Email:</label>
    <input type="email" id="email" name="email" required><br>

    <label for="telefon">Număr de telefon:</label>
    <input type="tel" id="telefon" name="telefon" pattern="[0-9]{10}"
required><br>

    <label for="bilet">Tip Bilet:</label>
    <select id="bilet" name="bilet">
      <option value="normal">Bilet Normal</option>
      <option value="VIP">Bilet VIP</option>
      <option value="premium">Bilet Premium</option>
    </select><br>

    <input type="submit" value="Înregistrează-te">
  </form>
</body>
</html>
```

În acest exemplu, am creat un formular de înregistrare pentru un eveniment. Formularul conține câmpurile necesare pentru a colecta informații de la utilizator, cum ar fi numele, adresa de email și numărul de telefon. Am folosit etichetele `<label>` pentru a descrie

fiecare câmp și am atribuit fiecărui câmp un ID unic (`id`) și un nume (`name`) pentru a le identifica în formular.

Pentru a asigura completarea corectă a formularului, am adăugat atributul `required` la câmpurile de nume, email și număr de telefon. Astfel, utilizatorul trebuie să completeze aceste câmpuri înainte de a trimite formularul.

Pentru a permite utilizatorului să aleagă tipul de bilet dorit, am creat un câmp de tip select cu opțiunile "Bilet Normal", "Bilet VIP" și "Bilet Premium".

În final, utilizatorul poate completa toate câmpurile și apoi apăsa butonul "Înregistrează-te" pentru a trimite formularul către server, unde datele pot fi procesate și înregistrate.

3. Realizați un tabel cu programul de lucru al unei companii, afișând orele de început și de sfârșit ale fiecărei ture de lucru.

Pentru a rezolva această cerință, vom crea un tabel care va afișa programul de lucru al unei companii, includând orele de început și de sfârșit ale fiecărei ture de lucru.

```
<!DOCTYPE html>
<html>
<head>
  <title>Program de Lucru - Companie</title>
</head>
<body>
  <h1>Program de Lucru - Companie</h1>
  <table border="1">
    <tr>
      <th>Zi</th>
      <th>Ora de Început</th>
      <th>Ora de Sfârșit</th>
    </tr>
    <tr>
      <td>Luni</td>
      <td>08:00</td>
      <td>16:00</td>
    </tr>
    <tr>
      <td>Marți</td>
      <td>09:00</td>
      <td>17:00</td>
    </tr>
    <tr>
      <td>Miercuri</td>
      <td>08:30</td>
      <td>16:30</td>
    </tr>
```

```

        <tr>
            <td>Joi</td>
            <td>08:00</td>
            <td>16:00</td>
        </tr>
        <tr>
            <td>Vineri</td>
            <td>09:00</td>
            <td>17:00</td>
        </tr>
    </table>
</body>
</html>

```

În acest exemplu, am creat un tabel cu trei coloane: "Zi", "Ora de Început" și "Ora de Sfârșit". Fiecare rând din tabel reprezintă o zi a săptămânii, iar în fiecare celulă am introdus orele corespunzătoare pentru începutul și sfârșitul turei de lucru a companiei în ziua respectivă.

Aceasta este o modalitate simplă de a prezenta programul de lucru al unei companii într-un tabel. Puteți extinde tabelul pentru a include mai multe zile și să le adaptați în funcție de nevoile specifice ale companiei.

4. Creați un formular de contact pentru un site web, cu câmpuri pentru nume, adresă de email, subiectul mesajului și conținutul mesajului.

Pentru a rezolva această cerință, vom crea un formular de contact pentru un site web, utilizând și stilizând CSS. Formularul va conține câmpuri pentru nume, adresă de email, subiectul mesajului și conținutul mesajului.

```

<!DOCTYPE html>
<html>
<head>
    <title>Formular de Contact</title>
    <style>
        body {
            font-family: Arial, sans-serif;
        }

        form {
            max-width: 400px;
            margin: 0 auto;
        }

        label {
            display: block;

```

```

        margin-bottom: 5px;
    }

    input[type="text"],
    input[type="email"],
    textarea {
        width: 100%;
        padding: 8px;
        margin-bottom: 10px;
        border: 1px solid #ccc;
        border-radius: 5px;
    }

    textarea {
        height: 100px;
    }

    input[type="submit"] {
        background-color: #4CAF50;
        color: white;
        padding: 10px 20px;
        border: none;
        border-radius: 5px;
        cursor: pointer;
    }

    input[type="submit"]:hover {
        background-color: #45a049;
    }
</style>
</head>
<body>
    <h1>Formular de Contact</h1>
    <form action="/submit" method="post">
        <label for="nume">Nume:</label>
        <input type="text" id="nume" name="nume" required>

        <label for="email">Adresă de Email:</label>
        <input type="email" id="email" name="email" required>

        <label for="subiect">Subiect:</label>
        <input type="text" id="subiect" name="subiect" required>

        <label for="mesaj">Mesaj:</label>
        <textarea id="mesaj" name="mesaj" required></textarea>

        <input type="submit" value="Trimite Mesajul">
    </form>
</body>

```

```
</html>
```

În acest exemplu, am creat un formular de contact și l-am stilizat folosind CSS. Am setat fontul și am centrat formularul orizontal pe pagină. Câmpurile pentru nume, adresă de email, subiect și mesaj au fost stilizate cu un aspect simplu și elegant, folosind o margine și o umplere pentru a le evidenția. Câmpul de mesaj a fost setat să aibă o înălțime de 100 de pixeli pentru a permite utilizatorilor să introducă mesaje mai lungi.

Butonul de trimitere a mesajului a fost stilizat cu culori pentru a se diferenția de restul formularului, iar la hover (când utilizatorul trece cursorul peste el), culoarea se schimbă pentru a indica că butonul este interactiv.

Acesta este doar un exemplu de cum puteți stiliza un formular de contact cu CSS. Puteți ajusta stilul și design-ul în funcție de preferințele și aspectul general al site-ului web.

5. Construiți un tabel pentru a afișa informații despre studenții unei școli, inclusiv nume, vârstă, clasă și medie.

Pentru a rezolva această cerință, vom construi un tabel pentru a afișa informații despre studenții unei școli, inclusiv nume, vârstă, clasă și medie. Vom utiliza și stilizăm CSS pentru a face tabelul să arate mai atractiv.

```
<!DOCTYPE html>
<html>
<head>
  <title>Tabel cu Informații despre Studenți</title>
  <style>
    body {
      font-family: Arial, sans-serif;
    }

    table {
      width: 100%;
      border-collapse: collapse;
      margin-bottom: 20px;
    }

    th, td {
      border: 1px solid #ccc;
      padding: 8px;
      text-align: center;
    }

    th {
      background-color: #f2f2f2;
    }
  </style>
</head>
<body>
  <table>
    <tr>
      <th>Nume</th>
      <th>Vârstă</th>
      <th>Clasă</th>
      <th>Medie</th>
    </tr>
    <tr>
      <td>Ion Popescu</td>
      <td>18</td>
      <td>Clasa 10</td>
      <td>8.5</td>
    </tr>
    <tr>
      <td>Maria Ionescu</td>
      <td>17</td>
      <td>Clasa 9</td>
      <td>7.8</td>
    </tr>
    <tr>
      <td>Andrei Nistor</td>
      <td>19</td>
      <td>Clasa 11</td>
      <td>9.2</td>
    </tr>
    <tr>
      <td>Elena Popescu</td>
      <td>16</td>
      <td>Clasa 8</td>
      <td>7.0</td>
    </tr>
  </table>
</body>
</html>
```

```

        tr:nth-child(even) {
            background-color: #f2f2f2;
        }

        tr:hover {
            background-color: #e0e0e0;
        }
    </style>
</head>
<body>
    <h1>Tabel cu Informații despre Studenți</h1>
    <table>
        <tr>
            <th>Nume</th>
            <th>Vârstă</th>
            <th>Clasă</th>
            <th>Medie</th>
        </tr>
        <tr>
            <td>Popescu Ion</td>
            <td>16 ani</td>
            <td>A IX-a</td>
            <td>8.75</td>
        </tr>
        <tr>
            <td>Ionescu Maria</td>
            <td>15 ani</td>
            <td>A X-a</td>
            <td>9.20</td>
        </tr>
        <tr>
            <td>Georgescu Andrei</td>
            <td>17 ani</td>
            <td>A XI-a</td>
            <td>9.50</td>
        </tr>
    </table>
</body>
</html>

```

În acest exemplu, am creat un tabel pentru a afișa informații despre studenții unei școli. Tabelul are patru coloane: "Nume", "Vârstă", "Clasă" și "Medie". Am aplicat un stil simplu la tabel pentru a-l face mai ușor de citit și mai atractiv.

Am utilizat CSS pentru a seta o margine și o umplere pentru celulele tabelului, iar culorile de fundal au fost aplicate alternativ pentru a evidenția mai bine fiecare rând. Când

utilizatorul plasează cursorul peste un rând, culoarea de fundal se schimbă pentru a indica interactivitatea.

Acest exemplu oferă o bază simplă pentru afișarea informațiilor despre studenții unei școli, iar stilul și aspectul tabelului pot fi personalizate în funcție de preferințele și design-ul general al paginii web.

6. Realizați un formular de înscriere la un newsletter, cu un câmp pentru adresa de email și un buton de subscriere.

Pentru a rezolva cerința, vom crea un formular de înscriere la un newsletter cu un câmp pentru adresa de email și un buton de subscriere. Vom utiliza CSS pentru a stiliza formularul și butonul.

```
<!DOCTYPE html>
<html>
<head>
  <title>Formular de Înscriere la Newsletter</title>
  <style>
    body {
      font-family: Arial, sans-serif;
    }

    form {
      max-width: 300px;
      margin: 0 auto;
    }

    input[type="email"] {
      width: 100%;
      padding: 8px;
      margin-bottom: 10px;
      border: 1px solid #ccc;
      border-radius: 5px;
    }

    input[type="submit"] {
      background-color: #4CAF50;
      color: white;
      padding: 10px 20px;
      border: none;
      border-radius: 5px;
      cursor: pointer;
    }

    input[type="submit"]:hover {
      background-color: #45a049;
```

```

    }
  </style>
</head>
<body>
  <h1>Formular de Înscriere la Newsletter</h1>
  <form action="/subscribe" method="post">
    <label for="email">Adresa de Email:</label>
    <input type="email" id="email" name="email" required>

    <input type="submit" value="Abonează-te">
  </form>
</body>
</html>

```

În acest exemplu, am creat un formular de înscriere la un newsletter și l-am stilizat folosind CSS. Am setat fontul și am centrat formularul orizontal pe pagină. Câmpul pentru adresa de email a fost stilizat cu un aspect simplu și elegant, folosind o margine și o umplere pentru a-l evidenția.

Butonul de subscriere a fost stilizat cu culori pentru a se diferenția de restul formularului, iar la hover (când utilizatorul trece cursorul peste el), culoarea se schimbă pentru a indica că butonul este interactiv.

Acesta este un exemplu simplu de cum puteți crea un formular de înscriere la un newsletter și să-l stilizați cu CSS. Puteți personaliza aspectul formularului și butonului pentru a se potrivi cu design-ul general al paginii web.

7. Creați un tabel pentru a afișa detalii despre cărțile disponibile într-o bibliotecă, precum numele cărții, autorul, anul publicării și stocul disponibil.

Pentru a rezolva această cerință, vom crea un tabel pentru a afișa detalii despre cărțile disponibile într-o bibliotecă, inclusiv numele cărții, autorul, anul publicării și stocul disponibil. Vom utiliza și stilizăm CSS pentru a face tabelul să arate mai atractiv.

```

<!DOCTYPE html>
<html>
<head>
  <title>Tabel cu Detalii despre Cărți</title>
  <style>
    body {
      font-family: Arial, sans-serif;
    }

    table {
      width: 100%;

```

```

        border-collapse: collapse;
        margin-bottom: 20px;
    }

    th, td {
        border: 1px solid #ccc;
        padding: 8px;
        text-align: center;
    }

    th {
        background-color: #f2f2f2;
    }

    tr:nth-child(even) {
        background-color: #f2f2f2;
    }

    tr:hover {
        background-color: #e0e0e0;
    }
</style>
</head>
<body>
    <h1>Tabel cu Detalii despre Cărți</h1>
    <table>
        <tr>
            <th>Numele Cărții</th>
            <th>Autor</th>
            <th>Anul Publicării</th>
            <th>Stoc Disponibil</th>
        </tr>
        <tr>
            <td>Harry Potter și Piatra Filozofală</td>
            <td>J.K. Rowling</td>
            <td>1997</td>
            <td>10</td>
        </tr>
        <tr>
            <td>Mândrie și Prejudecată</td>
            <td>Jane Austen</td>
            <td>1813</td>
            <td>5</td>
        </tr>
        <tr>
            <td>Crimă și Pedepsă</td>
            <td>Fyodor Dostoevsky</td>
            <td>1866</td>
            <td>8</td>
        </tr>
    </table>

```

```

    </tr>
  </table>
</body>
</html>

```

În acest exemplu, am creat un tabel pentru a afișa detalii despre cărțile disponibile într-o bibliotecă. Tabelul are patru coloane: "Numele Cărții", "Autor", "Anul Publicării" și "Stoc Disponibil". Am aplicat un stil simplu la tabel pentru a-l face mai ușor de citit și mai atractiv.

Am utilizat CSS pentru a seta o margine și o umplere pentru celulele tabelului, iar culorile de fundal au fost aplicate alternativ pentru a evidenția mai bine fiecare rând. Când utilizatorul plasează cursorul peste un rând, culoarea de fundal se schimbă pentru a indica interactivitatea.

Acest exemplu oferă o bază simplă pentru afișarea informațiilor despre cărțile disponibile într-o bibliotecă, iar stilul și aspectul tabelului pot fi personalizate în funcție de preferințele și design-ul general al paginii web.

8. Construiți un formular pentru a colecta recenzii de la clienți, cu câmpuri pentru nume, adresa de email, evaluarea și comentariul.

Pentru a rezolva cerința, vom crea un formular pentru a colecta recenzii de la clienți, cu câmpuri pentru nume, adresa de email, evaluarea și comentariul. Vom utiliza CSS pentru a stiliza formularul.

```

<!DOCTYPE html>
<html>
<head>
  <title>Formular de Recenzii de la Clienți</title>
  <style>
    body {
      font-family: Arial, sans-serif;
    }

    form {
      max-width: 400px;
      margin: 0 auto;
    }

    label {
      display: block;
      margin-bottom: 8px;
    }

    input[type="text"],

```

```

input[type="email"],
textarea {
    width: 100%;
    padding: 8px;
    margin-bottom: 15px;
    border: 1px solid #ccc;
    border-radius: 5px;
}

select {
    width: 100%;
    padding: 8px;
    margin-bottom: 15px;
    border: 1px solid #ccc;
    border-radius: 5px;
}

input[type="submit"] {
    background-color: #4CAF50;
    color: white;
    padding: 10px 20px;
    border: none;
    border-radius: 5px;
    cursor: pointer;
}

input[type="submit"]:hover {
    background-color: #45a049;
}
</style>
</head>
<body>
    <h1>Formular de Recenzii de la Clienți</h1>
    <form action="/submit-review" method="post">
        <label for="name">Nume:</label>
        <input type="text" id="name" name="name" required>

        <label for="email">Adresa de Email:</label>
        <input type="email" id="email" name="email" required>

        <label for="rating">Evaluare:</label>
        <select id="rating" name="rating" required>
            <option value="5">5 stele</option>
            <option value="4">4 stele</option>
            <option value="3">3 stele</option>
            <option value="2">2 stele</option>
            <option value="1">1 stea</option>
        </select>
    </form>

```

```

<label for="comment">Comentariu:</label>
<textarea id="comment" name="comment" rows="4" required></textarea>

<input type="submit" value="Trimite Recenzia">
</form>
</body>
</html>

```

În acest exemplu, am creat un formular pentru a colecta recenzii de la clienți și l-am stilizat folosind CSS. Am setat fontul și am centrat formularul orizontal pe pagină.

Am utilizat etichete ``<label>`` pentru a asocia fiecare câmp cu o descriere și pentru a le face mai accesibile pentru utilizatori.

Câmpurile pentru nume și adresa de email au fost stilizate cu o margine și o umplere pentru a le evidenția, iar textarea și câmpul select au primit și ele un stil simplu și plăcut.

Butonul de trimitere a recenziei a fost stilizat cu culori pentru a se diferenția de restul formularului, iar la hover (când utilizatorul trece cursorul peste el), culoarea se schimbă pentru a indica că butonul este interactiv.

Acesta este un exemplu simplu de cum puteți crea un formular pentru a colecta recenzii de la clienți și să-l stilizați cu CSS. Puteți personaliza aspectul formularului și butonului pentru a se potrivi cu design-ul general al paginii web.

9. Realizați un tabel pentru a afișa rezultatele unei competiții sportive, cu coloane pentru numele sportivilor și punctajul obținut.

Pentru a rezolva cerința, vom crea un tabel pentru a afișa rezultatele unei competiții sportive, cu coloane pentru numele sportivilor și punctajul obținut. Vom utiliza CSS pentru a stiliza tabelul.

```

<!DOCTYPE html>
<html>
<head>
  <title>Tabel cu Rezultate Sportive</title>
  <style>
    body {
      font-family: Arial, sans-serif;
    }

    table {
      width: 100%;
      border-collapse: collapse;
      margin-bottom: 20px;
    }

```

```

    th, td {
        border: 1px solid #ccc;
        padding: 8px;
        text-align: center;
    }

    th {
        background-color: #f2f2f2;
    }

    tr:nth-child(even) {
        background-color: #f2f2f2;
    }

    tr:hover {
        background-color: #e0e0e0;
    }
</style>
</head>
<body>
    <h1>Tabel cu Rezultate Sportive</h1>
    <table>
        <tr>
            <th>Numele Sportivilor</th>
            <th>Punctaj Obținut</th>
        </tr>
        <tr>
            <td>John Doe</td>
            <td>85</td>
        </tr>
        <tr>
            <td>Jane Smith</td>
            <td>92</td>
        </tr>
        <tr>
            <td>Mike Johnson</td>
            <td>78</td>
        </tr>
        <tr>
            <td>Laura Brown</td>
            <td>95</td>
        </tr>
    </table>
</body>
</html>

```

În acest exemplu, am creat un tabel pentru a afișa rezultatele unei competiții sportive. Tabelul are două coloane: "Numele Sportivilor" și "Punctaj Obținut".

Am utilizat CSS pentru a seta o margine și o umplere pentru celulele tabelului, iar culorile de fundal au fost aplicate alternativ pentru a evidenția mai bine fiecare rând. Când utilizatorul plasează cursorul peste un rând, culoarea de fundal se schimbă pentru a indica interactivitatea.

Acest exemplu oferă o bază simplă pentru afișarea rezultatelor unei competiții sportive, iar stilul și aspectul tabelului pot fi personalizate în funcție de preferințele și design-ul general al paginii web.

10. Creați un formular de înregistrare pentru un curs online, cu câmpuri pentru nume, adresă de email, număr de telefon și opțiunea de a selecta varianta de plată (în rate sau integral).

Pentru a rezolva cerința, vom crea un formular de înregistrare pentru un curs online, cu câmpuri pentru nume, adresă de email, număr de telefon și opțiunea de a selecta varianta de plată (în rate sau integral). Vom utiliza CSS pentru a stiliza formularul.

```
<!DOCTYPE html>
<html>
<head>
    <title>Formular de Înregistrare la Curs Online</title>
    <style>
        body {
            font-family: Arial, sans-serif;
        }

        form {
            max-width: 400px;
            margin: 0 auto;
        }

        label {
            display: block;
            margin-bottom: 8px;
        }

        input[type="text"],
        input[type="email"],
        input[type="tel"],
        select {
            width: 100%;
            padding: 8px;
            margin-bottom: 15px;
            border: 1px solid #ccc;
            border-radius: 5px;
        }
```

```

        input[type="submit"] {
            background-color: #4CAF50;
            color: white;
            padding: 10px 20px;
            border: none;
            border-radius: 5px;
            cursor: pointer;
        }

        input[type="submit"]:hover {
            background-color: #45a049;
        }
    </style>
</head>
<body>
    <h1>Formular de Înregistrare la Curs Online</h1>
    <form action="/submit-registration" method="post">
        <label for="name">Nume:</label>
        <input type="text" id="name" name="name" required>

        <label for="email">Adresă de Email:</label>
        <input type="email" id="email" name="email" required>

        <label for="phone">Număr de Telefon:</label>
        <input type="tel" id="phone" name="phone" required>

        <label for="payment">Varianta de Plată:</label>
        <select id="payment" name="payment" required>
            <option value="integral">Integral</option>
            <option value="rate">În Rate</option>
        </select>

        <input type="submit" value="Înscrie-te la Curs">
    </form>
</body>
</html>

```

În acest exemplu, am creat un formular de înregistrare pentru un curs online și l-am stilizat folosind CSS. Am setat fontul și am centrat formularul orizontal pe pagină.

Am utilizat etichete ``<label>`` pentru a asocia fiecare câmp cu o descriere și pentru a le face mai accesibile pentru utilizatori.

Câmpurile pentru nume, adresă de email, număr de telefon și câmpul select pentru varianta de plată au fost stilizate cu o margine și o umplere pentru a le evidenția.

Butonul de înscriere a fost stilizat cu culori pentru a se diferenția de restul formularului, iar la hover (când utilizatorul trece cursorul peste el), culoarea se schimbă pentru a indica că butonul este interactiv.

Acesta este un exemplu simplu de cum puteți crea un formular de înregistrare pentru un curs online și să-l stilizați cu CSS. Puteți personaliza aspectul formularului și butonului pentru a se potrivi cu design-ul general al paginii web.

11. Construiți un tabel cu detalii despre angajații unei companii, inclusiv numele, funcția, data angajării și salariul.

Pentru a rezolva cerința, vom construi un tabel cu detalii despre angajații unei companii, inclusiv numele, funcția, data angajării și salariul. Vom utiliza CSS pentru a stiliza tabelul.

```
<!DOCTYPE html>
<html>
<head>
  <title>Tabel cu Detalii despre Angajații Companiei</title>
  <style>
    body {
      font-family: Arial, sans-serif;
    }

    table {
      width: 100%;
      border-collapse: collapse;
      margin-bottom: 20px;
    }

    th, td {
      border: 1px solid #ccc;
      padding: 8px;
      text-align: center;
    }

    th {
      background-color: #f2f2f2;
    }

    tr:nth-child(even) {
      background-color: #f2f2f2;
    }

    tr:hover {
      background-color: #e0e0e0;
    }
  </style>
```

```

</head>
<body>
  <h1>Tabel cu Detalii despre Angajații Companiei</h1>
  <table>
    <tr>
      <th>Nume</th>
      <th>Funcție</th>
      <th>Data Angajării</th>
      <th>Salariu</th>
    </tr>
    <tr>
      <td>John Doe</td>
      <td>Manager</td>
      <td>2023-07-15</td>
      <td>5000 lei</td>
    </tr>
    <tr>
      <td>Jane Smith</td>
      <td>Contabil</td>
      <td>2023-06-01</td>
      <td>4500 lei</td>
    </tr>
    <tr>
      <td>Mike Johnson</td>
      <td>Programator</td>
      <td>2023-04-10</td>
      <td>6000 lei</td>
    </tr>
    <tr>
      <td>Laura Brown</td>
      <td>Asistent</td>
      <td>2023-03-20</td>
      <td>3500 lei</td>
    </tr>
  </table>
</body>
</html>

```

În acest exemplu, am construit un tabel pentru a afișa detalii despre angajații unei companii. Tabelul are patru coloane: "Nume", "Funcție", "Data Angajării" și "Salariu".

Am utilizat CSS pentru a seta o margine și o umplere pentru celulele tabelului, iar culorile de fundal au fost aplicate alternativ pentru a evidenția mai bine fiecare rând. Când utilizatorul plasează cursorul peste un rând, culoarea de fundal se schimbă pentru a indica interactivitatea.

Acest exemplu oferă o bază simplă pentru afișarea detaliilor despre angajații unei companii, iar stilul și aspectul tabelului pot fi personalizate în funcție de preferințele și design-ul general al paginii web.

12. Realizați un formular pentru a colecta datele de contact ale participanților la un eveniment, cu câmpuri pentru nume, adresă de email și număr de telefon.

```
<!DOCTYPE html>
<html>
<head>
  <title>Formular de Înregistrare la Eveniment</title>
  <style>
    body {
      font-family: Arial, sans-serif;
    }

    form {
      max-width: 400px;
      margin: 0 auto;
    }

    label {
      display: block;
      margin-bottom: 8px;
    }

    input[type="text"],
    input[type="email"],
    input[type="tel"] {
      width: 100%;
      padding: 8px;
      margin-bottom: 15px;
      border: 1px solid #ccc;
      border-radius: 5px;
    }

    input[type="submit"] {
      background-color: #4CAF50;
      color: white;
      padding: 10px 20px;
      border: none;
      border-radius: 5px;
      cursor: pointer;
    }

    input[type="submit"]:hover {
      background-color: #45a049;
    }
  </style>
</head>
<body>
  <div>
    <h1>Formular de Înregistrare la Eveniment</h1>
    <div>
      <div>
        <input type="text" value="Nume" />
        <input type="text" value="Prenume" />
      </div>
      <div>
        <input type="email" value="Email" />
        <input type="tel" value="Telefon" />
      </div>
      <input type="submit" value="Înregistrare" />
    </div>
  </div>
</body>
</html>
```

```

    }
  </style>
</head>
<body>
  <h1>Formular de Înregistrare la Eveniment</h1>
  <form action="/submit-contact" method="post">
    <label for="name">Nume:</label>
    <input type="text" id="name" name="name" required>

    <label for="email">Adresă de Email:</label>
    <input type="email" id="email" name="email" required>

    <label for="phone">Număr de Telefon:</label>
    <input type="tel" id="phone" name="phone" required>

    <input type="submit" value="Înscrie-te la Eveniment">
  </form>
</body>
</html>

```

13. Creați un tabel pentru a afișa programul de proiecții la un cinematograf, cu coloane pentru numele filmului, ora de început și durata.

```

<!DOCTYPE html>
<html>
<head>
  <title>Tabel Program Proiecții Cinematograf</title>
  <style>
    body {
      font-family: Arial, sans-serif;
    }

    table {
      width: 100%;
      border-collapse: collapse;
      margin-bottom: 20px;
    }

    th, td {
      border: 1px solid #ccc;
      padding: 8px;
      text-align: center;
    }

    th {
      background-color: #f2f2f2;
    }
  </style>

```

```

        tr:nth-child(even) {
            background-color: #f2f2f2;
        }

        tr:hover {
            background-color: #e0e0e0;
        }
    </style>
</head>
<body>
    <h1>Tabel Program Proiecții Cinematograf</h1>
    <table>
        <tr>
            <th>Nume Film</th>
            <th>Ora de Început</th>
            <th>Durata</th>
        </tr>
        <tr>
            <td>Avatar</td>
            <td>18:00</td>
            <td>2h 42min</td>
        </tr>
        <tr>
            <td>Inception</td>
            <td>20:30</td>
            <td>2h 28min</td>
        </tr>
        <tr>
            <td>The Dark Knight</td>
            <td>22:45</td>
            <td>2h 32min</td>
        </tr>
    </table>
</body>
</html>

```

14. Construiți un formular pentru a colecta cereri de oferte de la potențiali clienți, cu câmpuri pentru nume, adresă de email, tipul de serviciu solicitat și un câmp pentru descrierea cererii.

```

<!DOCTYPE html>
<html>
<head>
    <title>Formular Cerere Ofertă</title>
    <style>
        body {
            font-family: Arial, sans-serif;
        }
    </style>

```

```

    form {
        max-width: 400px;
        margin: 0 auto;
    }

    label {
        display: block;
        margin-bottom: 8px;
    }

    input[type="text"],
    input[type="email"] {
        width: 100%;
        padding: 8px;
        margin-bottom: 15px;
        border: 1px solid #ccc;
        border-radius: 5px;
    }

    textarea {
        width: 100%;
        height: 100px;
        padding: 8px;
        margin-bottom: 15px;
        border: 1px solid #ccc;
        border-radius: 5px;
    }

    input[type="submit"] {
        background-color: #4CAF50;
        color: white;
        padding: 10px 20px;
        border: none;
        border-radius: 5px;
        cursor: pointer;
    }

    input[type="submit"]:hover {
        background-color: #45a049;
    }
</style>
</head>
<body>
    <h1>Formular Cerere Ofertă</h1>
    <form action="/submit-quote-request" method="post">
        <label for="name">Nume:</label>
        <input type="text" id="name" name="name" required>

        <label for="email">Adresă de Email:</label>

```

```

<input type="email" id="email" name="email" required>

<label for="service">Tipul de Serviciu Solicitat:</label>
<input type="text" id="service" name="service" required>

<label for="description">Descriere Cerere:</label>
<textarea id="description" name="description" required></textarea>

<input type="submit" value="Trimite Cererea">
</form>
</body>
</html>

```

15. Realizați un tabel cu informații despre echipa unui club sportiv, inclusiv numele jucătorilor, poziția și numărul de tricou.

```

<!DOCTYPE html>
<html>
<head>
  <title>Tabel Informații Echipă Club Sportiv</title>
  <style>
    body {
      font-family: Arial, sans-serif;
    }

    table {
      width: 100%;
      border-collapse: collapse;
      margin-bottom: 20px;
    }

    th, td {
      border: 1px solid #ccc;
      padding: 8px;
      text-align: center;
    }

    th {
      background-color: #f2f2f2;
    }

    tr:nth-child(even) {
      background-color: #f2f2f2;
    }

    tr:hover {
      background-color: #e0e0e0;
    }
  </style>

```

```

    </style>
</head>
<body>
    <h1>Tabel Informații Echipă Club Sportiv</h1>
    <table>
        <tr>
            <th>Nume Jucător</th>
            <th>Poziție</th>
            <th>Număr Tricou</th>
        </tr>
        <tr>
            <td>John Doe</td>
            <td>Fundaș</td>
            <td>5</td>
        </tr>
        <tr>
            <td>Jane Smith</td>
            <td>Mijlocăș</td>
            <td>10</td>
        </tr>
        <tr>
            <td>Michael Johnson</td>
            <td>Atacant</td>
            <td>7</td>
        </tr>
    </table>
</body>
</html>

```

16. Creați un formular de înscriere la un workshop, cu câmpuri pentru nume, adresă de email, ocupație și opțiunea de a alege unul din mai multe ateliere disponibile.

```

<!DOCTYPE html>
<html>
<head>
    <title>Formular Înscriere Workshop</title>
    <style>
        body {
            font-family: Arial, sans-serif;
        }

        form {
            max-width: 400px;
            margin: 0 auto;
        }

        label {
            display: block;

```

```

        margin-bottom: 8px;
    }

    input[type="text"],
    input[type="email"] {
        width: 100%;
        padding: 8px;
        margin-bottom: 15px;
        border: 1px solid #ccc;
        border-radius: 5px;
    }

    select {
        width: 100%;
        padding: 8px;
        margin-bottom: 15px;
        border: 1px solid #ccc;
        border-radius: 5px;
    }

    input[type="submit"] {
        background-color: #4CAF50;
        color: white;
        padding: 10px 20px;
        border: none;
        border-radius: 5px;
        cursor: pointer;
    }

    input[type="submit"]:hover {
        background-color: #45a049;
    }
</style>
</head>
<body>
    <h1>Formular Înscriere Workshop</h1>
    <form action="/register-for-workshop" method="post">
        <label for="name">Nume:</label>
        <input type="text" id="name" name="name" required>

        <label for="email">Adresă de Email:</label>
        <input type="email" id="email" name="email" required>

        <label for="occupation">Ocupație:</label>
        <input type="text" id="occupation" name="occupation" required>

        <label for="workshop">Alegeți un Atelier:</label>
        <select id="workshop" name="workshop" required>
            <option value="">-- Alegeți --</option>

```

```

        <option value="design-grafic">Design Grafic</option>
        <option value="dezvoltare-web">Dezvoltare Web</option>
        <option value="fotografie">Fotografie</option>
    </select>

    <input type="submit" value="Înscrieți-vă">
</form>
</body>
</html>

```

17. Construiți un tabel pentru a afișa rezultatele unui sondaj de opinie, cu coloane pentru întrebări și opțiunile de răspuns.

```

<!DOCTYPE html>
<html>
<head>
    <title>Tabel Rezultate Sondaj de Opinie</title>
    <style>
        body {
            font-family: Arial, sans-serif;
        }

        table {
            width: 100%;
            border-collapse: collapse;
            margin-bottom: 20px;
        }

        th, td {
            border: 1px solid #ccc;
            padding: 8px;
            text-align: center;
        }

        th {
            background-color: #f2f2f2;
        }

        tr:nth-child(even) {
            background-color: #f2f2f2;
        }

        tr:hover {
            background-color: #e0e0e0;
        }
    </style>
</head>
<body>

```

```

<h1>Tabel Rezultate Sondaj de Opinie</h1>
<table>
  <tr>
    <th>Întrebare</th>
    <th>Răspuns 1</th>
    <th>Răspuns 2</th>
    <th>Răspuns 3</th>
  </tr>
  <tr>
    <td>Ce vă place cel mai mult la produsul nostru?</td>
    <td>30%</td>
    <td>45%</td>
    <td>25%</td>
  </tr>
  <tr>
    <td>Cum ați aflat despre produsul nostru?</td>
    <td>20%</td>
    <td>15%</td>
    <td>65%</td>
  </tr>
</table>
</body>
</html>

```

18. Realizați un formular pentru a colecta datele de contact ale clienților pentru un magazin online, cu câmpuri pentru nume, adresă de livrare, adresă de facturare și număr de telefon.

```

<!DOCTYPE html>
<html>
<head>
  <title>Formular Date de Contact</title>
  <style>
    body {
      font-family: Arial, sans-serif;
    }

    form {
      max-width: 400px;
      margin: 0 auto;
    }

    label {
      display: block;
      margin-bottom: 8px;
    }

    input[type="text"],
    input[type="email"] {

```

```

        width: 100%;
        padding: 8px;
        margin-bottom: 15px;
        border: 1px solid #ccc;
        border-radius: 5px;
    }

    input[type="submit"] {
        background-color: #4CAF50;
        color: white;
        padding: 10px 20px;
        border: none;
        border-radius: 5px;
        cursor: pointer;
    }

    input[type="submit"]:hover {
        background-color: #45a049;
    }
</style>
</head>
<body>
    <h1>Formular Date de Contact</h1>
    <form action="/submit-contact-details" method="post">
        <label for="name">Nume:</label>
        <input type="text" id="name" name="name" required>

        <label for="shipping-address">Adresă de Livrare:</label>
        <input type="text" id="shipping-address" name="shipping-address"
required>

        <label for="billing-address">Adresă de Facturare:</label>
        <input type="text" id="billing-address" name="billing-address"
required>

        <label for="phone">Număr de Telefon:</label>
        <input type="text" id="phone" name="phone" required>

        <input type="submit" value="Trimite">
    </form>
</body>
</html>

```

19. Creați un tabel pentru a afișa programul de concerte la un festival, cu coloane pentru numele formațiilor, data și locația.

```

<!DOCTYPE html>
<html>

```

```

<head>
  <title>Tabel Program Concerte Festival</title>
  <style>
    body {
      font-family: Arial, sans-serif;
    }

    table {
      width: 100%;
      border-collapse: collapse;
      margin-bottom: 20px;
    }

    th, td {
      border: 1px solid #ccc;
      padding: 8px;
      text-align: center;
    }

    th {
      background-color: #f2f2f2;
    }

    tr:nth-child(even) {
      background-color: #f2f2f2;
    }

    tr:hover {
      background-color: #e0e0e0;
    }
  </style>
</head>
<body>
  <h1>Tabel Program Concerte Festival</h1>

  <table>
    <tr>
      <th>Formație</th>
      <th>Data</th>
      <th>Locația</th>
    </tr>
    <tr>
      <td>Formația X</td>
      <td>10 august</td>
      <td>Scena Principală</td>
    </tr>
    <tr>
      <td>Formația Y</td>

```

```

        <td>11 august</td>
        <td>Scena B</td>
    </tr>
    <tr>
        <td>Formația Z</td>
        <td>12 august</td>
        <td>Scena C</td>
    </tr>
</table>
</body>
</html>

```

20. Construiți un formular pentru a colecta feedback de la vizitatorii unui site web, cu câmpuri pentru nume, adresă de email, categoria feedback-ului (pozitiv, negativ sau sugestie) și un câmp pentru comentarii.

```

<!DOCTYPE html>
<html>
<head>
    <title>Formular Feedback Vizitatori</title>
    <style>
        body {
            font-family: Arial, sans-serif;
        }

        form {
            max-width: 400px;
            margin: 0 auto;
        }

        label {
            display: block;
            margin-bottom: 8px;
        }

        input[type="text"],
        input[type="email"] {
            width: 100%;
            padding: 8px;
            margin-bottom: 15px;
            border: 1px solid #ccc;
            border-radius: 5px;
        }

        textarea {
            width: 100%;
            height: 100px;
        }
    </style>

```

```

        padding: 8px;
        margin-bottom: 15px;
        border: 1px solid #ccc;
        border-radius: 5px;
    }

    input[type="radio"] {
        margin-right: 5px;
    }

    input[type="submit"] {
        background-color: #4CAF50;
        color: white;
        padding: 10px 20px;
        border: none;
        border-radius: 5px;
        cursor: pointer;
    }

    input[type="submit"]:hover {
        background-color: #45a049;
    }
</style>
</head>
<body>
    <h1>Formular Feedback Vizitatori</h1>
    <form action="/submit-feedback" method="post">
        <label for="name">Nume:</label>
        <input type="text" id="name" name="name" required>

        <label for="email">Adresă de Email:</label>
        <input type="email" id="email" name="email" required>

        <label>Feedback:</label>
        <input type="radio" id="positive" name="feedback" value="pozitiv"
required>
        <label for="positive">Pozitiv</label>

        <input type="radio" id="negative" name="feedback" value="negativ"
required>
        <label for="negative">Negativ</label>

        <input type="radio" id="suggestion" name="feedback" value="sugestie"
required>
        <label for="suggestion">Sugestie</label>

        <label for="comments">Comentarii:</label>
        <textarea id="comments" name="comments" required></textarea>

```

```
        <input type="submit" value="Trimite Feedback">
    </form>
</body>
</html>
```