

# CURS 1

## Despre Web

### 1.1 Internetul și World Wide Web

Înainte de a scrie prima linie de cod, este esențial să înțelegem infrastructura pe care o folosim zilnic, adesea fără să o vedem. Mulți oameni confundă 'internetul' cu 'Web-ul' — sunt lucruri diferite.

#### Ce este internetul?

Internetul este o rețea globală de calculatoare interconectate care comunică folosind un set comun de protocoale (TCP/IP). Gândește-te la el ca la o rețea imensă de autostrăzi digitale care conectează miliarde de dispozitive: calculatoare, telefoane, servere, camere, IoT.

Internetul există fizic: cabluri subacvatice transoceanice, fibră optică, antene de radio și sateliți de comunicații. Nu este ceva abstract — datele tale călătoresc prin aceste medii la viteze apropiate de viteza luminii.

#### Ce este World Wide Web?

Web-ul (WWW) este un serviciu care rulează PE DEASUPRA internetului — exact cum aplicațiile rulează pe un sistem de operare. Web-ul este un sistem de documente interconectate (pagini HTML) legate prin hiperlink-uri, accesibile prin URL-uri, livrate prin protocolul HTTP.



Analogie: Internetul este autostrada. Web-ul este un tip de vehicul care circulă pe ea — browserele. Email-ul este un alt tip de vehicul. Toți folosesc aceeași autostradă (infrastructura internetului), dar sunt servicii complet diferite.

#### Componentele fundamentale

Concept	Explicație
URL	Adresa unică a unei resurse: <code>https://www.exemplu.ro:443/pagina?id=1#sectiune</code> — componente: protocol, domeniu, port, cale, parametri, fragment.
HTTP/HTTPS	Protocolul de comunicare client↔server. HTTPS adaugă criptare TLS. Toate site-urile moderne folosesc HTTPS obligatoriu.
DNS	Domain Name System — 'cartea de telefon' a internetului. Traduce <code>google.com</code> în adresa IP <code>142.250.185.78</code> . Fără DNS am memora adrese IP.
Adresă IP	Identificator numeric unic pe rețea. IPv4: <code>192.168.1.1</code> . IPv6: <code>2001:0db8::1</code> (pentru mai multe dispozitive).
Browser	Client web: trimite cereri HTTP, primește HTML/CSS/JS, le interpretează și afișează vizual pagina.
Server	Calculator care rulează permanent, ascultă cereri și returnează resurse. Nginx, Apache, Node.js sunt software de server.

## Ce se întâmplă când tastezi o adresă în browser?

Aceasta este una din cele mai frecvente întrebări tehnice în interviuri pentru poziții de dezvoltator web. Urmărește pașii complet:

1. Browser-ul analizează URL-ul: extrage protocolul, domeniul, calea, parametrii
2. Verifică cache-ul DNS local (a mai accesat această adresă recent?)
3. Dacă nu e în cache: interogare DNS recursivă la serverul furnizorului tău de internet (ISP)
4. DNS returnează adresa IP a serverului (ex: 151.101.1.195)
5. Browser-ul inițiază conexiune TCP cu serverul pe portul 443 (HTTPS)
6. Handshake TLS: stabilire canal criptat în 1-2 round-trips
7. Browser trimite cerere HTTP GET la server
8. Serverul returnează răspuns HTTP cu HTML și cod de status (200 OK)
9. Browser parsează HTML, descoperă resurse externe (CSS, JS, imagini), trimite cereri suplimentare
10. Construiește DOM și CSSOM, calculează layout, renderizează pagina — gata!

Toată această secvență durează sub 1 secundă. Fiecare pas afectează viteza percepută - performanța web este o disciplină separată pe care o vom studia în Cursul 13.

## 1.2 HTTP și Modelul Client-Server

HTTP (HyperText Transfer Protocol) este protocolul text care definește exact cum comunică clientul (browser) și serverul. Orice interacțiune web urmează același tipar simplu: **cerere** → **răspuns**.

### Structura unui mesaj HTTP

```
=== CERERE (trimisă de browser) ===
GET /api/speakeri HTTP/1.1
Host: api.devconnect.ro
Authorization: Bearer eyJhbGciOi...
Accept: application/json

=== RĂSPUNS (returnat de server) ===
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Cache-Control: max-age=300

{"speakeri": [...], "total": 12}
```

### Metodele HTTP — operații CRUD

Metodă	CRUD	Utilizare concretă
GET	Read	Recuperează resurse. Nu modifică nimic. Poate fi bookmark-at. Ex: afișare listă speakeri.
POST	Create	Trimite date pentru a crea o resursă nouă. Date în body. Ex: înregistrare participant.
PUT	Update (total)	Înlocuiește complet o resursă. Ex: actualizare profil complet utilizator.
PATCH	Update (parțial)	Modifică doar câmpurile specificate. Ex: schimbare email.
DELETE	Delete	Șterge o resursă. Ex: anulare înregistrare.

## Codurile de stare HTTP — ghid rapid

Cod	Categorie	Exemple importante
<b>2xx</b>	✓ Succes	200 OK · 201 Created (resursă creată) · 204 No Content (succes fără body)
<b>3xx</b>	↪ Redirecționare	301 Moved Permanently · 304 Not Modified (ia din cache)
<b>4xx</b>	✗ Eroare client	400 Bad Request · 401 Unauthorized · 403 Forbidden · 404 Not Found · 422 Unprocessable
<b>5xx</b>	✖ Eroare server	500 Internal Server Error · 502 Bad Gateway · 503 Service Unavailable



Reține: 404 = resursa nu există (verifică URL-ul). 401 = nu ești autentificat (trimite token). 403 = autentificat dar fără permisiune. 500 = problema e pe server (verifică log-urile serverului).

## Cele trei straturi ale paginii web

Strat	Tehnologie	Responsabilitate
<b>Structură</b>	<b>HTML</b>	Scheletul — ce elemente există și în ce ordine. Ca oasele corpului.
<b>Prezentare</b>	<b>CSS</b>	Aspectul vizual — culori, fonturi, layout. Ca hainele și machiajul.
<b>Comportament</b>	<b>JavaScript</b>	Interactivitatea — animații, date dinamice, comunicare server. Ca mușchii.



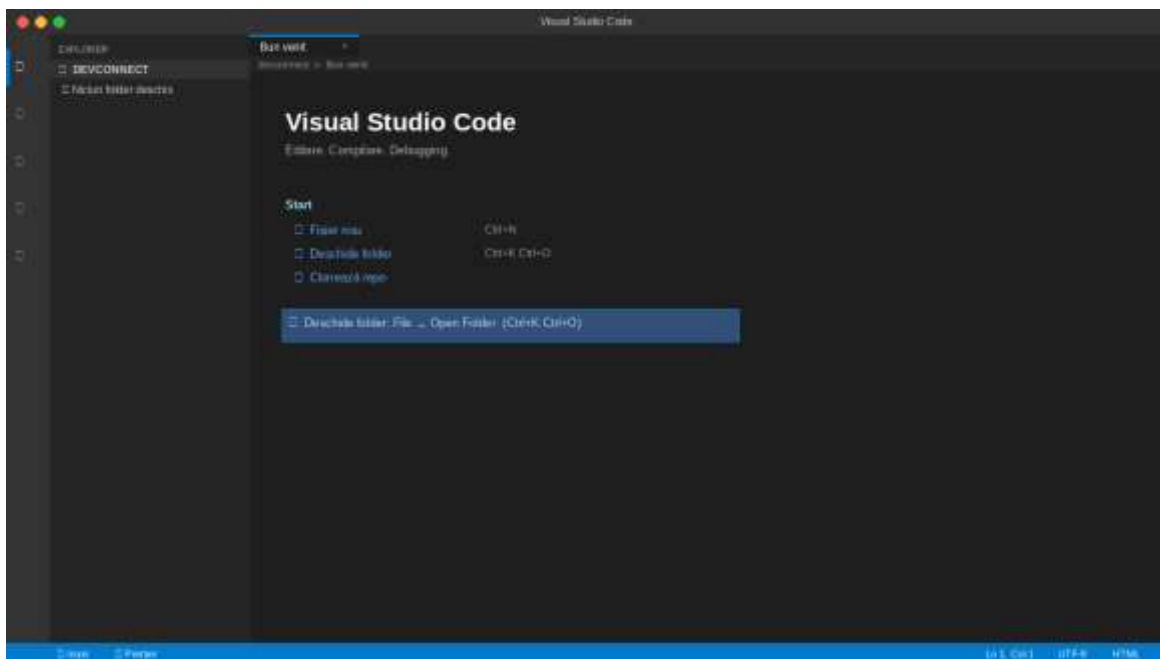
Greșeala clasică a începătorilor: amestecarea celor trei straturi — stiluri CSS scrise direct ca attribute inline HTML, sau logică JavaScript înglobată în HTML. Ține mereu cele trei straturi separate în fișiere distincte.

## 1.3 Configurarea VS Code - Pas cu Pas

Visual Studio Code (VS Code) este editorul de cod pe care îl vom folosi pe tot parcursul cursului. Este gratuit, open-source și cel mai popular editor web în industrie (Stack Overflow Survey 2024: peste 73% din dezvoltatori îl folosesc). Urmărește pașii de mai jos exact — fiecare screenshot îți arată ce trebuie să vezi pe ecranul tău.

### Pasul 1 - Instalare VS Code

11. Deschide browserul și accesează <https://code.visualstudio.com>
12. Descarcă versiunea pentru sistemul tău de operare (Windows / macOS / Linux)
13. Rulează installer-ul cu setările implicite (nu modifica nimic)
14. La prima deschidere vei vedea ecranul de bun venit:



**Ecranul de bun venit VS Code - prima deschidere. Observă opțiunile din stânga pentru a deschide sau crea un folder de proiect.**

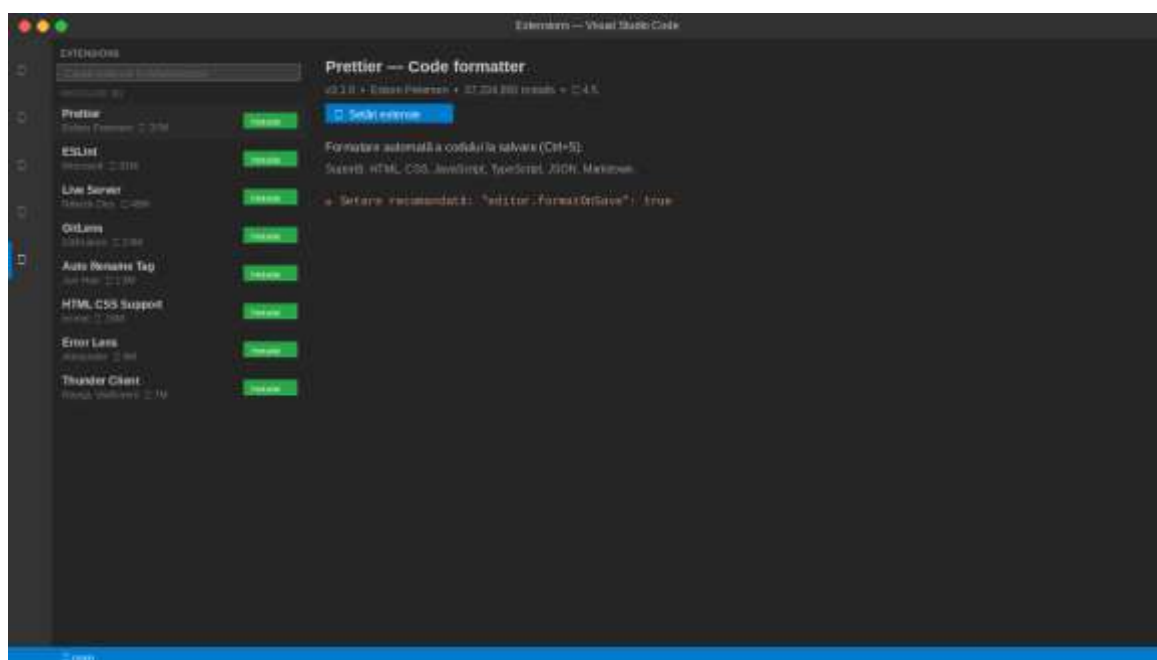


Ce vezi în acest ecran: bara de activitate (stânga, cu iconițe pentru Explorer, Search, Git, Debug, Extensions), zona de bun venit (centru) și bara de stare (jos, albastră). Acestea sunt elementele permanente ale interfeței VS Code.

### Pasul 2 - Instalare Extensii Recomandate

Extensiile transformă VS Code dintr-un editor simplu într-un mediu de dezvoltare profesional. Instalează-le pe toate acum — le vei folosi în fiecare laborator.

**Cum se instalează:** clic pe iconița puzzle 🧩 din bara stângă (sau Ctrl+Shift+X) → tastează numele extensiei → clic Install.

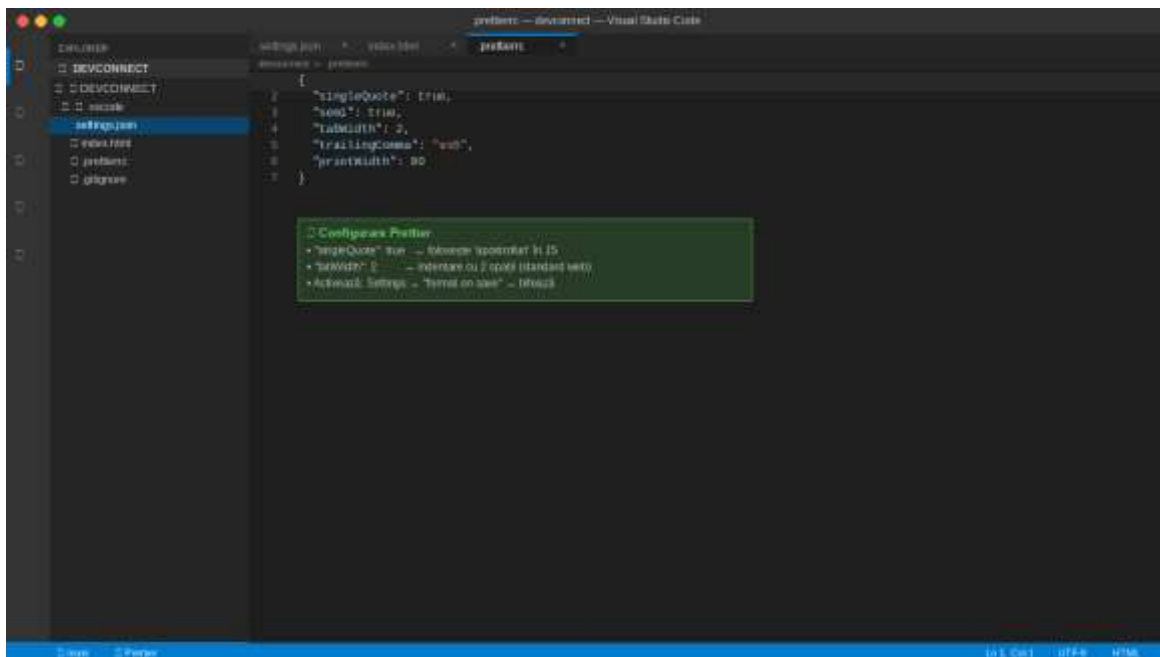


**Panoul Extensions din VS Code. În stânga: lista extensiilor instalate. În dreapta: detalii despre extensia selectată (Prettier). Observă butonul 'Instalat' verde lângă fiecare extensie.**

Extensie	De ce este importantă
<b>Prettier - Code formatter</b>	Formatează automat codul la Ctrl+S. Elimină erorile de indentare și stil — codul tău va arăta profesional întotdeauna.
<b>ESLint</b>	Detectează erori JavaScript înainte să rulezi codul. Ca un corector gramatical pentru cod.
<b>Live Server</b>	Deschide pagina ta HTML în browser cu reîncărcare automată la salvare. Esențial pentru Lab-uri.
<b>Auto Rename Tag</b>	Când redenumești <div> în <section>, redenumește automat și </div> în </section>.
<b>GitLens</b>	Arată în editor cine a scris fiecare linie de cod și când — util la debugging.
<b>HTML CSS Support</b>	Autocompletare pentru clase CSS în fișiere HTML.
<b>Error Lens</b>	Afișează erorile direct pe linia cu probleme, nu doar în bara de jos.
<b>Thunder Client</b>	Client API REST integrat — vom testa API-urile Express direct din VS Code.

### Pasul 3 - Configurarea Prettier (formatare automată)

Prettier va formata automat codul la fiecare salvare. Configurează-l în fișierul .prettierrc în folderul proiectului:



**Fișierul .prettierrc deschis în VS Code. Observă colorarea sintaxă JSON (cheile în albastru, valorile în portocaliu/verde). Panoul din stânga arată structura proiectului DevConnect.**

```
// Creează fișierul .prettierrc în root-ul proiectului:
{
  "singleQuote": true,
  "semi": true,
  "tabWidth": 2,
  "trailingComma": "es5",
  "printWidth": 80
}
```

Activare formatare la salvare: apasă **Ctrl+,** (Settings) → caută 'format on save' → bifează opțiunea. De acum, **Ctrl+S** formatează și salvează automat.

## Comenzi esențiale VS Code

Shortcut	Acțiune
<b>Ctrl+Shift+P</b>	Command Palette — accesează ORICE comandă VS Code. Cel mai important shortcut.
<b>Ctrl+P</b>	Deschide rapid un fișier după nume (în proiect mare cu zeci de fișiere).
<b>Ctrl+`</b>	Deschide/închide terminalul integrat — terminal fără a ieși din VS Code.
<b>Ctrl+S</b>	Salvează fișierul curent (și formatează dacă Prettier e configurat).
<b>Ctrl+/</b>	Comentează/decomentează linia sau selecția curentă.
<b>Alt+Z</b>	Toggle word wrap — afișează liniile lungi pe mai multe rânduri vizuale.
<b>Ctrl+D</b>	Selectează următoarea apariție a cuvântului selectat — editare multiplă.
<b>F12</b>	Du-te la definiția funcției/variabilei.
<b>Shift+Alt+F</b>	Formatează tot fișierul (alternativă la format on save).

## 1.4 Git și Primul Proiect DevConnect

Git este sistemul de control al versiunilor standard în industrie. Permite urmărirea tuturor modificărilor din cod, colaborarea în echipă și revenirea la versiuni anterioare. Gândește-te la Git ca la un sistem de salvări automate cu mesaje descriptive — fiecare 'commit' este o fotografie a proiectului la un moment dat.

### Instalare și configurare Git

- Windows: descarcă de la <https://git-scm.com> — instalează cu setările implicite (include Git Bash)
- macOS: deschide Terminal și tastează `git --version` — dacă nu e instalat, oferă să-l instaleze automat
- Linux (Ubuntu): `sudo apt install git`

### Configurare inițială (o singură dată pe calculatorul tău):

```
# Tastează aceste comenzi în terminalul VS Code (Ctrl+`):
git config --global user.name "Prenume Nume"
git config --global user.email "email@exemplu.com"
git config --global init.defaultBranch main
git config --global core.editor "code --wait"

# Verifică configurarea:
git config --list
```

### Crearea Proiectului DevConnect în VS Code

Acum vom crea proiectul pe care îl vom construi pe tot parcursul cursului. Urmărește exact pașii:

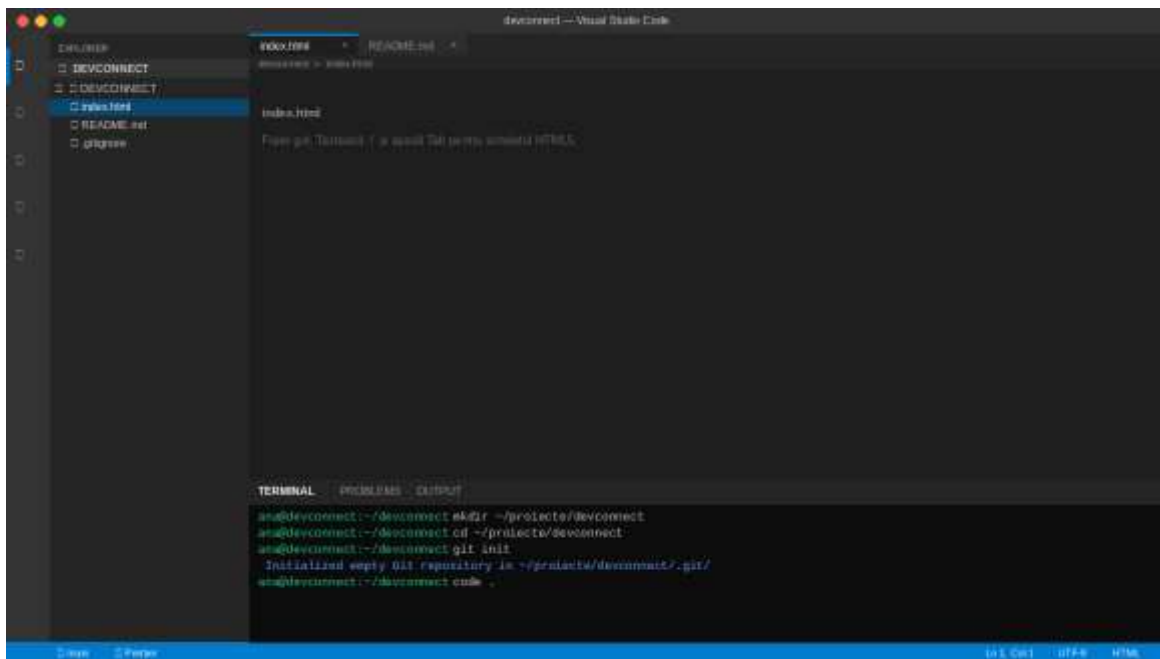
15. Deschide VS Code
16. Apasă `Ctrl+`` pentru a deschide terminalul integrat
17. Execută comenzile de mai jos:

```
# Creează folderul proiectului
mkdir ~/proiecte/devconnect
cd ~/proiecte/devconnect

# Inițializează Git în acest folder
git init

# Deschide VS Code în folderul curent
code .
```

VS Code se redeschide în folderul devconnect. Vei vedea ceva similar cu:



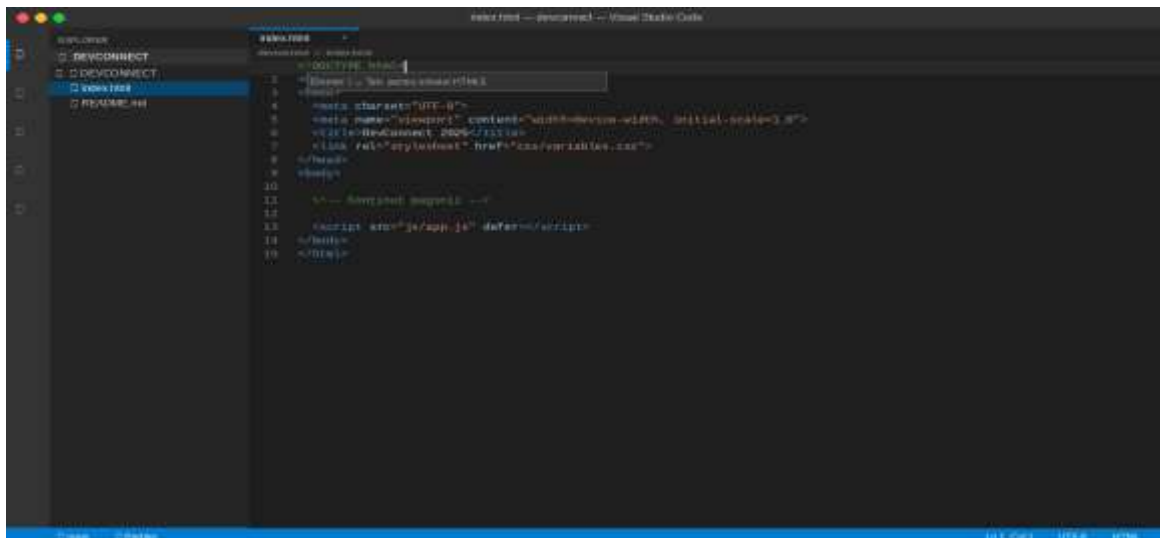
**VS Code după deschiderea folderului devconnect. Stânga: Explorer cu structura proiectului (inițial gol). Jos: terminalul integrat cu comenzile Git executate și confirmarea inițializării repository-ului.**



Observă în Explorer (stânga): folderul DEVCONNECT apare cu fișierele create. Bara de stare (jos, albastră) arată '┌ main' — ești pe branch-ul principal Git. Terminalul arată confirmate comenzile executate.

### Crearea fișierului index.html

18. În VS Code Explorer (stânga), clic pe iconița 'New File' de lângă DEVCONNECT
19. Tastează index.html și apasă Enter
20. Fișierul se deschide în editor — este gol
21. Tastează exact ! și apasă Tab — Emmet generează automat scheletul HTML5 complet:



**index.html deschis în VS Code după expandarea abrevierii Emmet '!'. Observă: colorarea sintaxă HTML (etichete în albastru, atribute în deschis, valori în portocaliu), numerele de linie în stânga, fișierul activ în tab-ul de sus.**

```

<!-- Acesta este codul generat automat de Emmet cu ! + Tab: -->
<!DOCTYPE html>
<html lang="ro">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>DevConnect 2025</title>
</head>
<body>

</body>
</html>

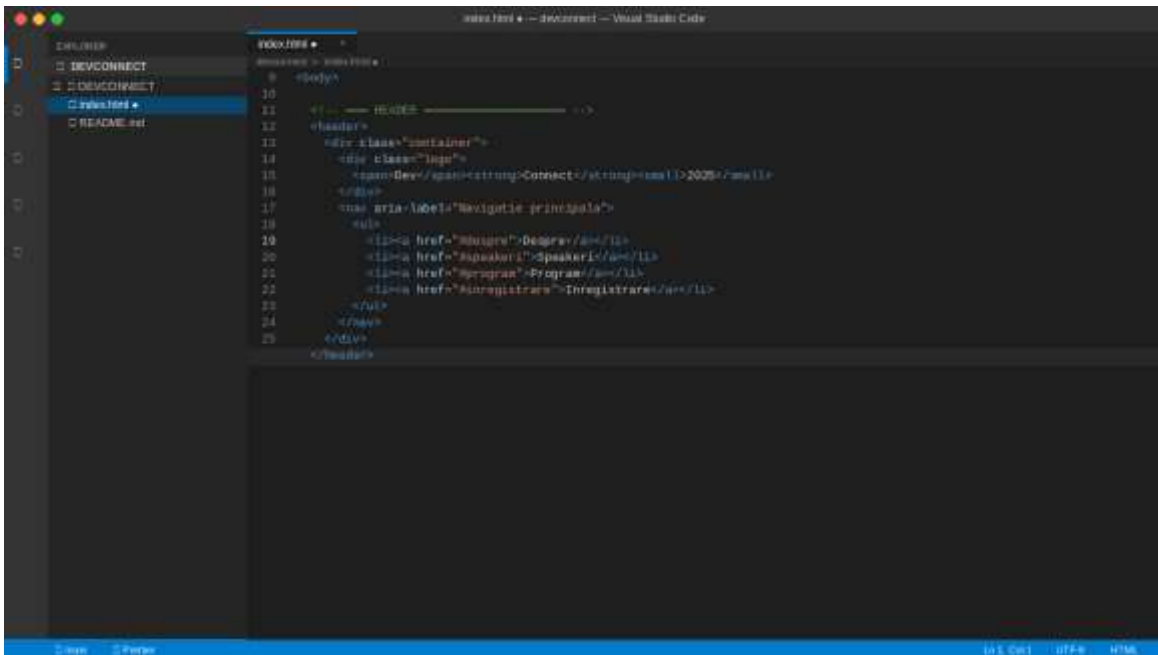
```



Nu copia-lipi codul — tastează-l manual sau folosește Emmet. Tastarea activă consolidează memorarea sintaxei de 3-4 ori mai eficient decât copia-lipirea. Erorile de tastare sunt normale și instructive.

## Adăugarea structurii HTML — Header și Navigație

Adaugă în `<body>` elementul `<header>` cu navigația. Observă autocompletarea VS Code pe măsură ce tastezi:



**VS Code cu codul HTML al header-ului DevConnect. Observă: indentarea automată (Prettier), colorarea etichetelor HTML (albastru) vs atribute (deschis-albastru) vs valori (portocaliu), indicatoarele de modificare (●) în tab — fișierul are modificări nesalvate.**

```

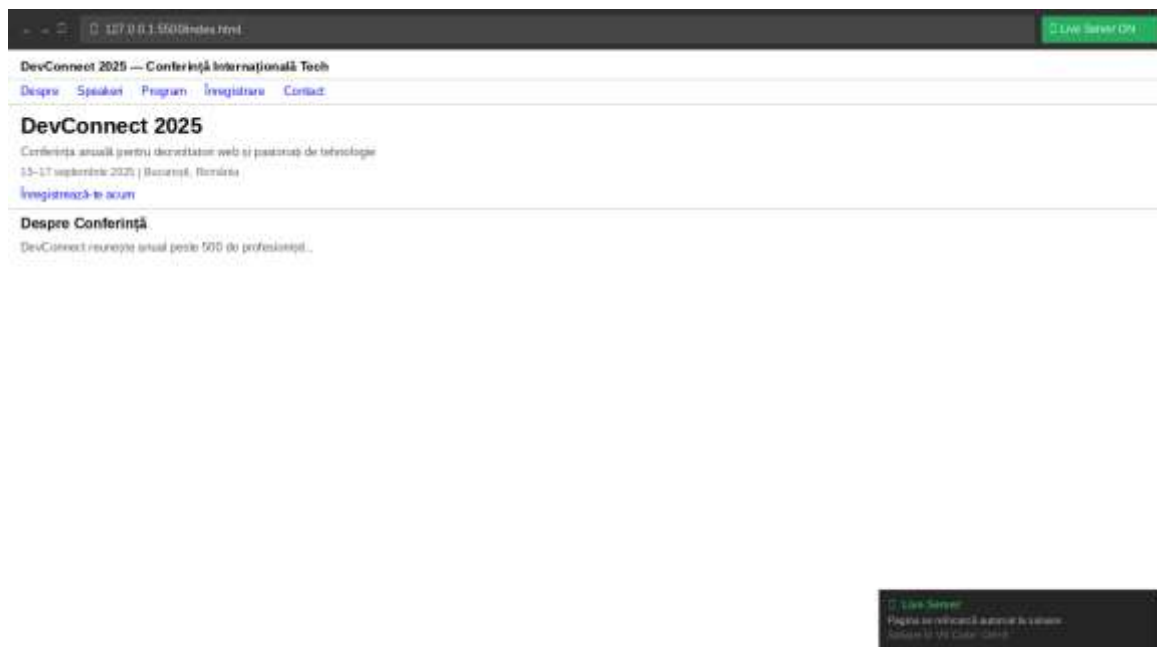
<!-- Adaugă în <body>: -->
<header>
  <div class="container">
    <div class="logo">
      <span>Dev</span><strong>Connect</strong><small>2025</small>
    </div>
    <nav aria-label="Navigatie principala">
      <ul>
        <li><a href="#despre">Despre</a></li>
        <li><a href="#speakeri">Speakeri</a></li>
        <li><a href="#program">Program</a></li>
        <li><a href="#inregistrare">Inregistrare</a></li>

```

```
</ul>  
</nav>  
</div>  
</header>
```

## Testare cu Live Server

Clic dreapta pe index.html în Explorer → 'Open with Live Server'. Browserul se deschide automat:



***Pagina DevConnect deschisă în browser prin Live Server (127.0.0.1:5500). Pagina apare fără stilizare CSS — text simplu, liste cu bullet, linkuri albastre. Aceasta este forma corectă a unui document HTML5 valid dar nestilizat — baza pe care adăugăm CSS în Laboratorul 3.***



Testează Live Server: modifică orice text în VS Code (ex: 'DevConnect 2026' în titlu), apasă Ctrl+S. Browserul se reîncarcă instant fără să apeși F5. Aceasta este puterea Live Server — feedback imediat la orice modificare.

## 1.5 Fluxul de Lucru Git

Git urmărește modificările din proiect. Gândește-te la el ca la un sistem de fotografii (commit-uri) ale codului - poți reveni oricând la orice fotografie anterioară.

### Comenzile Git esențiale

Comandă	Ce face
<code>git init</code>	Inițializează un repository Git în folderul curent (creează <code>.git/</code> ascuns).
<code>git status</code>	Arată fișierele modificate, adăugate sau șterse față de ultimul commit.
<code>git add .</code>	Adaugă TOATE fișierele modificate în staging area (zona de pregătire).
<code>git commit -m "mesaj"</code>	Salvează un snapshot (fotografie) cu mesajul specificat.
<code>git log --oneline</code>	Afișează istoricul commit-urilor compact (un commit per linie).
<code>git push origin main</code>	Trimite commit-urile locale pe GitHub.
<code>git pull</code>	Descarcă modificările de pe GitHub în local.
<code>git clone url</code>	Clonează un repository existent de pe GitHub.

### Fluxul complet: primul commit și push pe GitHub

Urmărește secvența de comenzi în terminalul VS Code:

```

an@deconnect:~/deconnect$ git status
On branch main
Changes not staged for commit:
  modified:   index.html

an@deconnect:~/deconnect$ git add
an@deconnect:~/deconnect$ git commit -m "feat: adauga structura HTML pagina principala"
[main 3a7f2bc] feat: adauga structura HTML pagina principala
1 file changed, 45 insertions(+), 1 deletion(-)

an@deconnect:~/deconnect$ git push origin main

```

**Terminalul VS Code (jos) cu fluxul Git complet: `git status` → `git add` → `git commit` → `git push`. Observă răspunsurile Git: confirmarea commit-ului cu hash-ul (3a7f2bc), numărul de fișiere modificate și statisticile push.**

```

# 1. Verifică starea curentă
git status
# Arată: index.html modificat (nesalvat în Git)

# 2. Adaugă toate fișierele în staging
git add .

```

```
# 3. Creează commit-ul
git commit -m "feat: adauga structura HTML pagina principala DevConnect"


# 4. Publică pe GitHub
# (Creează mai întâi repo-ul pe github.com)
git remote add origin https://github.com/username/devconnect.git
git branch -M main
git push -u origin main

# 5. Verifică istoricul
git log --oneline
```



Mesajele de commit contează enorm! Evită: 'fix', 'update', 'asdfgh'. Folosește formatul: 'tip: descriere'. Tipuri: feat (funcționalitate nouă), fix (rezolvare bug), docs (documentație), style (CSS/formatare), refactor (restructurare).

### Interfața Git din VS Code (fără terminal)

VS Code are o interfață grafică Git integrată — iconița  din bara laterală stângă. Alternativă vizuală la comenzile de terminal:

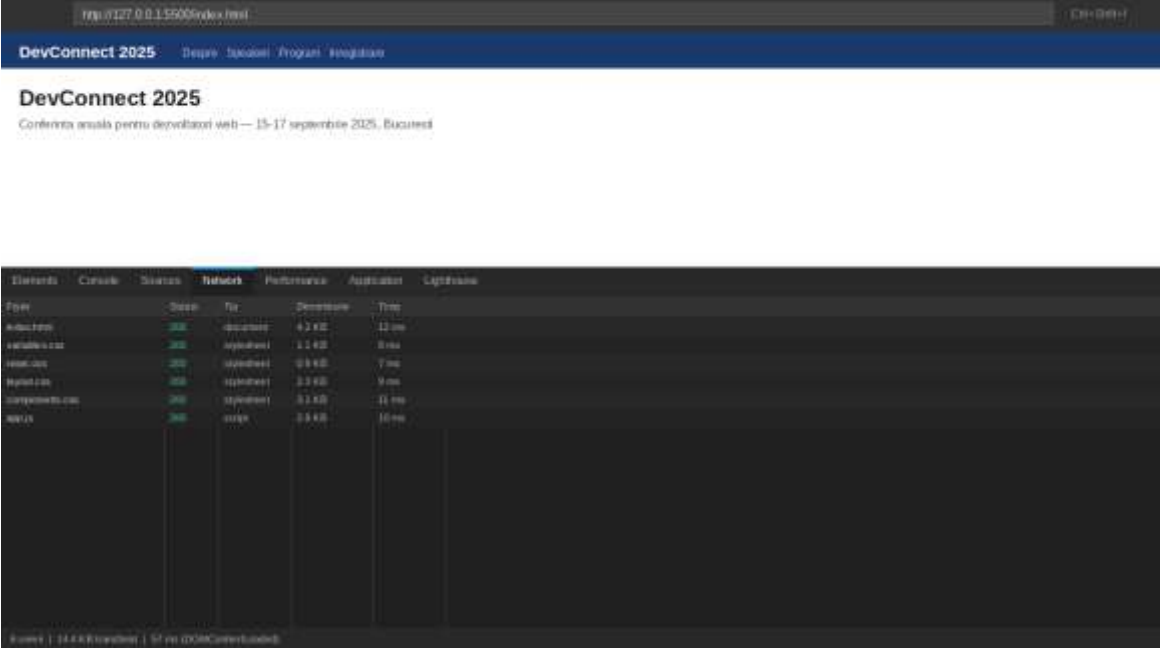
- Bara laterală → iconița Source Control (Ctrl+Shift+G)
- Fișierele modificate apar în lista 'Changes'
- Clic + pentru a face stage, tastează mesajul în câmpul de sus și apasă Ctrl+Enter pentru commit
- Butonul 'Sync Changes' face push și pull simultan



Recomandare: învață mai întâi comenzile terminal (consolidează înțelegerea), apoi folosește interfața grafică pentru eficiență. Ambele metode sunt corecte.

## 1.6 Browser DevTools - Instrumentul zilnic al dezvoltatorului

DevTools (Instrumentele pentru dezvoltatori) sunt integrate în orice browser modern. Sunt cel mai important instrument de debugging și investigare a paginilor web. Deschide-le cu F12 sau Ctrl+Shift+I.



The screenshot shows the Network tab in Chrome DevTools. The page title is 'DevConnect 2025'. The Network panel displays a list of requests:

File	Status	Type	Size	Time
index.html	200	document	4.2 KB	12 ms
css/index.css	200	stylesheet	1.2 KB	5 ms
js/index.js	200	script	0.8 KB	7 ms
js/app.js	200	script	2.2 KB	9 ms
components.css	200	stylesheet	3.2 KB	11 ms
utils	200	script	0.8 KB	10 ms

**Browser DevTools — panoul Network. Observă lista de cereri HTTP pe care pagina le face: index.html (200 OK, 4.2 KB, 12ms), fișierele CSS, JavaScript. Fiecare linie este o cerere HTTP separată cu toate detaliile ei.**

### Panourile DevTools esențiale

Panou	Ce face și cum se folosește
<b>Elements</b>	Inspectează și editează live DOM-ul și CSS-ul. Clic dreapta pe orice element din pagină → Inspect. Modificările sunt temporare (se pierd la reîncărcare) — ideal pentru experimentat CSS.
<b>Console</b>	Rulează JavaScript direct, vezi log-uri (console.log), erori, avertismente. Cel mai folosit panou în debugging.
<b>Network</b>	Monitorizează TOATE cererile HTTP: URL, metodă, status, dimensiune, timp. Esențial pentru debugging API și performanță.
<b>Sources</b>	Inspectează fișierele sursă JS/CSS. Setează breakpoints și execută codul pas cu pas.
<b>Application</b>	Inspectează localStorage, sessionStorage, cookies. Util pentru debugging stocare date.
<b>Lighthouse</b>	Audit automat: performanță, accesibilitate, SEO. Generează scor și recomandări concrete.

## Exerciții Practice DevTools

Execută acești pași cu DevTools deschis pe pagina DevConnect (Live Server):

22. Elements: clic dreapta pe titlul H1 → Inspect. Modifică textul direct în panou. Observă că pagina se actualizează instant. Reîncarcă — modificarea dispare.
23. Console: tastează `document.title` și apasă Enter. Vei vedea titlul paginii afișat. Tastează `document.querySelector('h1').style.color = 'red'` — titlul devine roșu instant.
24. Network: reîncarcă pagina cu panoul Network deschis. Observă fiecare fișier solicitat, ordinea, dimensiunile și timpii.
25. Responsive: apasă `Ctrl+Shift+M` pentru modul device emulation. Selectează 'iPhone SE' (375px). Observă cum arată pagina pe mobil.

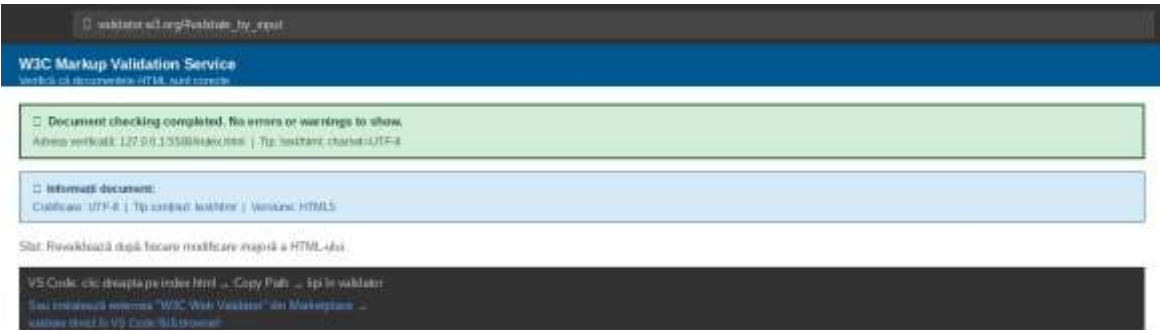


Sfat profesionist: lasă DevTools deschis mereu în timpul dezvoltării. Panoul Console îți arată erorile JavaScript imediat. Panoul Network îți arată când o cerere API eșuează. Aceste două panouri economisesc ore de debugging.

## 1.7 Validarea HTML și finalizarea cursului

Un fișier HTML valid este fundamentul oricărei pagini web profesionale. Validarea verifică că respectăm standardele W3C și că browserele vor interpreta corect codul nostru.

### Validarea HTML cu W3C Validator



The screenshot shows the W3C Markup Validation Service interface. At the top, it says "W3C Markup Validation Service" and "verifică că documentele HTML sunt corecte". Below that, a green box displays the message: "Document checking completed. No errors or warnings to show." It also shows the address "https://127.0.0.1:55000/index.html" and the encoding "charset=UTF-8". A blue box below provides document information: "Codificarea: UTF-8 | Tipul documentului: HTML5". At the bottom, there is a footer with instructions on how to use the service, including a note about VS Code.

**Rezultatul W3C Validator pentru pagina DevConnect: zero erori, zero avertismente. Mesajul verde confirmă că documentul HTML5 este complet valid. Informațiile suplimentare arată codificarea UTF-8 detectată corect.**

Pași pentru validare:

26. Deschide [https://validator.w3.org/#validate\\_by\\_input](https://validator.w3.org/#validate_by_input) în browser
27. În VS Code, selectează tot codul (Ctrl+A) și copiază (Ctrl+C)
28. Lipește în câmpul de text al validatorului (Ctrl+V)
29. Clic 'Check'
30. Obiectivul: 'No errors or warnings to show' (mesaj verde)
31. Dacă există erori: rezolvă-le una câte una, revalidează



Nu trimite niciodată Lab-ul fără să fi trecut validatorul W3C cu zero erori. Erorile de HTML indică probleme structurale care pot cauza afișare incorectă în browsere diferite.

## Rezumat: Ce ai configurat în Curs 1

Unealtă/Concept	Status și utilizare în curs
VS Code + extensii	<input checked="" type="checkbox"/> Instalat și configurat — editor principal pentru tot cursul
Node.js + npm	<input checked="" type="checkbox"/> Instalat — necesar din Cursul 7 (React) și Cursul 9 (Express)
Git	<input checked="" type="checkbox"/> Configurat cu user.name și user.email — control versiuni pentru fiecare lab
GitHub	<input checked="" type="checkbox"/> Repository devconnect creat și publicat — vei publica fiecare laborator
Live Server	<input checked="" type="checkbox"/> Activ — feedback instantaneu la modificări HTML/CSS
Browser DevTools	<input checked="" type="checkbox"/> Familiar cu panourile cheie — debugging zilnic
W3C Validator	<input checked="" type="checkbox"/> Bookmark-at — validare obligatorie înainte de orice submission

## Referințe și Resurse Suplimentare

### Resurse online recomandate

- ★ [MDN Web Docs — Cum funcționează Web-ul](#)
- ★ [VS Code — Getting Started \(video-uri oficiale\)](#)
- ★ [GitHub Skills — Introduction to GitHub \(curs interactiv gratuit\)](#)
- [Pro Git — cartea oficială Git \(gratuită online, inclusiv în română\)](#)
- [How HTTPS Works — explicație vizuală interactivă](#)
- [How DNS Works — explicație vizuală \(benzi desenate\)](#)
- [The Odin Project — Foundations \(curriculum complet gratuit\)](#)

### Cărți recomandate

- Duckett, J. (2011). HTML and CSS: Design and Build Websites. Wiley. — cea mai accesibilă introducere vizuală
- Robbins, J. N. (2018). Learning Web Design, 5th ed. O'Reilly. ISBN: 978-1491960202
- Chacon, S. & Straub, B. (2014). Pro Git, 2nd ed. Apress. — disponibilă gratuit la [git-scm.com/book](https://git-scm.com/book)



La finalul Cursului 1 ai: mediul de dezvoltare complet configurat, primul proiect DevConnect pe GitHub, și înțelegerea conceptelor fundamentale ale Web-ului. Ești pregătit pentru Cursul 2 — HTML semantic complet.

### Notă privind elaborarea materialelor de curs

Vreau să fiu transparent cu voi: structura și conținutul acestor note de curs au fost generate cu ajutorul unui instrument de inteligență artificială (Claude, de la Anthropic), pe baza cerințelor și direcțiilor pe care le-am formulat eu ca titular de curs.

De ce vă spun asta? Pentru că:

- Nu pot garanta că fiecare noțiune tehnică are 100% acuratete sau este actualizată
- Vă încurajez să verificați activ sursele bibliografice indicate
- Utilizarea responsabilă a AI în educație înseamnă transparență, nu ascundere

Considerați aceste materiale un ghid structurat de studiu, nu un manual definitiv. Dacă identificați o eroare sau o neclaritate, veniți cu ea la curs - este o oportunitate de învățare pentru toți.