🌐 Dezvoltare Web pentru Liceeni

Bine ai venit pe platforma ta de învățare! Parcurge pas cu pas lumea fascinantă a **HTML**, **CSS** și **JavaScript** — de la primele taguri până la proiecte reale. Fiecare lecție include exemple practice, exerciții și mini-quizuri.

Începe Cursul | Vezi Proiectul Final



MODULUL 1

Internet & World Wide Web

Ce este Internetul?

Internetul este o rețea globală de calculatoare interconectate care comunică prin protocoale standardizate. WWW (World Wide Web) este un sistem de documente conectate prin hyperlink-uri, accesibil prin browser.

- Protocol HTTP/HTTPS
- DNS – sistemul numelor de domenii
- Browser vs. Server

🧠 Mini-Quiz

Întrebare: Care este diferența dintre Internet și WWW?

Răspuns: Internetul este infrastructura fizică (cabluri, servere), iar WWW este serviciul de pagini web care rulează pe internet.

💡 **Știai că?** Prima pagină web a fost creată de Tim Berners-Lee în 1991 la CERN, Geneva.



Client trimite

Server procesează

Server răspunde

Această arhitectură client-server stă la baza oricărei pagini web pe care o vizitezi zilnic.

Made with GAMMA

MODULUL 1

Exerciții Practice & Test de Verificare

Exerciții Practice Hands-on

Aplică noțiunile învățate cu aceste sarcini simple:

1. **Inspectorul de Elemente:** Deschide orice pagină web, apasă click-dreapta și selectează "Inspect" (sau "Inspect Element"). Identifică un titlu (ex: H1, H2) și schimbă temporar textul acestuia. Ce observi?
2. **Află Adresa IP:** Folosește un instrument online de tip "DNS Lookup" (ex: whatsmydns.net) pentru a găsi adresa IP a `google.com`. Compar-o cu adresa IP a unui alt site.
3. **Vizualizează Sursa Paginii:** Pe o pagină web, apasă click-dreapta și alege "View Page Source". Caută tag-urile `<html>`, `<head>` și `<body>`. Ce crezi că semnifică fiecare secțiune?

Test de Verificare

Răspunde la întrebări pentru a-ți consolida cunoștințele.

1. Care este rolul principal al unui browser web în arhitectura client-server?
2. Ce înseamnă acronimul HTTP și de ce este important?
3. Oferă un exemplu de domeniu de nivel superior (TLD). Ce indică acesta de obicei?

💡 **Reține:** Internetul este rețeaua fizică, iar WWW este doar o parte din serviciile oferite prin această rețea!

Made with GAMMA

MODULUL 2–4

HTML — Scheletul Paginii Web

HTML (HyperText Markup Language) este limbajul de bază al oricărei pagini web. Folosește **taguri** pentru a structura conținutul: titluri, paragrafe, imagini și linkuri.

1

Structura de bază

Orice document HTML începe cu `<!DOCTYPE html>`, urmat de `<html>`, `<head>` și `<body>`.

2

Titluri & Paragrafe

Folosește `<h1>`–`<h6>` pentru titluri și `<p>` pentru paragrafe. Ierarhia titlurilor contează pentru SEO!

3

Linkuri & Imagini

Tag-ul `<a href="...">` creează linkuri, iar `<img src="..." alt="...">` inserează imagini cu descriere.

4

Exercițiu practic

Creează o pagină HTML cu titlul tău, o fotografie și un link către site-ul tău preferat. Salvează fișierul ca `index.html`.

```
<!DOCTYPE html>
<html lang="ro">
 <head>
 <title>Prima mea pagină</title>
 </head>
 <body>
 <h1>Bună ziua, lume!</h1>
 <p>Aceasta este prima mea pagină web.</p>
 <a href="https://google.com">Google</a>
 </body>
</html>
```

Made with GAMMA

MODULUL 2–4

Exerciții Practice HTML & Test de Verificare

Exerciții Practice de Codare HTML

Aplică noțiunile învățate prin aceste exerciții interactive de codare HTML:

1. **Pagină de bază:** Creează un fișier HTML cu un titlu la alegere și un paragraf ce conține textul "Salut, HTML!".
2. **Imagini & Linkuri:** Adaugă o imagine (poți folosi o adresă URL online) și un link către site-ul tău preferat.
3. **Formatare text:** Folosește tag-urile `<b>` și `<i>` pentru a evidenția text într-un paragraf nou.
4. **Liste:** Creează o listă neordonată (`<ul>`) cu trei elemente, reprezentând hobby-urile tale.

Test de Verificare a Cunoștințelor

Verifică-ți înțelegerea conceptelor fundamentale HTML răspunzând la următoarele întrebări:

1. Care este rolul principal al tag-ului `<head>` într-un document HTML?
2. Cum se definește un paragraf în HTML și ce tag este folosit?
3. Ce atribut este esențial pentru tag-ul `<img>` pentru a afișa o imagine și ce informații conține?
4. Explică diferența dintre un link absolut și un link relativ, oferind un exemplu pentru fiecare.

💡 **Sfaturi utile:** Salvează-ți frecvent munca și testează codul HTML direct în browser pentru a vedea rezultatele.

Made with GAMMA

MODULUL 5–7

Liste, Tabele, Formulare & HTML5 Semantic

📋 Liste și Tabele

Listele `<ul>` și `<ol>` organizează informații, iar tabelele `<table>` structurează date în rânduri și coloane.

```
<ul>
  <li>HTML</li>
  <li>CSS</li>
  <li>JavaScript</li>
</ul>
```

📝 Formulare HTML

Formularele colectează date de la utilizatori. Elementele `<input>`, `<select>` și `<textarea>` sunt cele mai folosite.

🧱 Elemente Semantice HTML5

HTML5 introduce taguri cu sens clar: `<header>`, `<nav>`, `<main>`, `<article>`, `<section>`, `<footer>`.

<header>

Antetul paginii

<nav>

Meniu navigare

<main>

Conținut principal

<footer>

Subsol pagină

🎯 **Provocare:** Recreează structura paginii unui site real (ex: Wikipedia) folosind numai taguri semantice HTML5!

Made with GAMMA

MODULUL 5–7

Exerciții Practice & Test de Verificare

Exerciții Practice de Codare HTML

Aplică noțiunile învățate prin aceste exerciții interactive pentru a consolida conceptele de liste, tabele, formulare și semantică HTML5:

1. **Liste imbricate:** Creează o listă neordonată de "Mâncăruri Preferate", unde fiecare element (ex: "Italiană") conține o listă ordonată de 3-4 feluri de mâncare specifice.
2. **Tabel de produse:** Construiește un tabel simplu cu `<thead>`, `<tbody>` și `<tfoot>`, listând cel puțin 3 produse, cantități și prețuri. Adaugă un rând cu totalul în `<tfoot>`.
3. **Formular de înregistrare:** Proiectează un formular basic pentru înregistrarea utilizatorilor cu câmpuri pentru Nume, Prenume (text), Email (email), Parolă (password) și un buton de trimitere.
4. **Structura blogului:** Creează o pagină care conține un `<header>` (cu titlu H1), o zonă de `<nav>` (cu linkuri fake), un `<main>` care include un `<article>` (cu titlu H2 și paragraf) și un `<footer>` (cu copyright).

Test de Verificare a Cunoștințelor

Verifică-ți înțelegerea elementelor avansate HTML răspunzând la următoarele întrebări:

1. Explică diferența structurală și de utilizare dintre `<ul>` și `<ol>`.
2. Care sunt cele trei taguri esențiale pentru structura unui tabel HTML (cu excepția `<table>`)?
3. Ce este atributul `type` în elementul `<input>` și dă două exemple de valori ale sale.
4. De ce este considerată utilizarea tagurilor semantice HTML5 o bună practică în dezvoltarea web modernă?

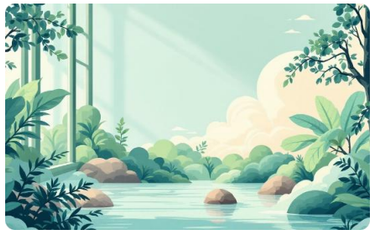
💡 **Sfaturi utile:** Nu te teme să experimentezi! Testează fiecare exercițiu în browser și inspectează codul cu instrumentele de dezvoltator.

Made with GAMMA

MODULUL 8–10

CSS — Stilul și Designul Paginii

CSS (Cascading Style Sheets) transformă structura HTML într-o pagină vizual atractivă. Controlezi culori, fonturi, spațiere și layout-ul întregii pagini.



Selectori & Proprietăți

Selectori de **element**, **clasă** (.clasa) și **ID** (#id) aplică stiluri țintit. Proprietăți esențiale: `color`, `font-size`, `background`.



Box Model

Fiecare element HTML este o cutie cu: **content**, **padding**, **border** și **margin**. Înțelegerea box model-ului este esențială pentru layout corect.



Flexbox & Grid

Flexbox aliniază elemente pe o axă, **CSS Grid** creează layout-uri complexe în 2D. Ambele sunt indispensabile pentru design responsive modern.

```
/* Exemplu CSS */
body {
 font-family: 'Arial', sans-serif;
 background-color: #f0f4f8;
 color: #333;
}
h1 {
 color: #437066;
 text-align: center;
}
```

Made with GAMMA

MODULUL 8–10

Exerciții Practice CSS & Test de Verificare

Exerciții Practice de Codare CSS

Aplică stiluri paginilor tale HTML pentru a le transforma vizual, folosind conceptele fundamentale de CSS.

1. **Stilizare Selectoare:** Creează un fișier CSS extern. Stilizează un paragraf cu clasa `.highlight` pentru a avea text albastru și fundal gri deschis.
2. **Box Model:** Definește un `<div>` în HTML. În CSS, aplică-i o lățime de `200px`, un `padding` de `20px`, o `border` de `2px solid black` și o `margin` de `10px`.
3. **Flexbox Simplu:** Creează o secțiune cu trei `<div>`-uri copil. Folosește Flexbox pentru a le alinia orizontal și a distribui spațiul uniform între ele.

Test de Verificare a Cunoștințelor

Verifică-ți înțelegerea conceptelor cheie din CSS răspunzând la aceste întrebări:

1. Care sunt cele trei modalități principale de a include CSS într-un document HTML și care sunt avantajele fiecăreia?
2. Explică pe scurt rolul proprietăților `margin`, `border` și `padding` în modelul de cutie (Box Model) CSS.
3. Ce este specificitatea (specificity) în CSS și cum ajută la rezolvarea conflictelor de stilizare?
4. Ce problemă rezolvă Flexbox în designul web și cum se diferențiază de CSS Grid?

💡 **Explorare suplimentară:** Încearcă să folosești instrumentele de dezvoltator ale browserului pentru a inspecta și modifica stilurile în timp real.

Made with GAMMA

MODULUL 11–15

JavaScript — Logica și Interactivitatea

De ce JavaScript?

JavaScript aduce viață paginilor web: butoane interactive, validare formulare, animații și comunicare cu serverul. Este singurul limbaj de programare care rulează nativ în browser.

- Variabile: `let`, `const`, `var`
- Tipuri: string, number, boolean, array, object
- Funcții: declarate și arrow functions

Structuri de Control

```
// Condiții
if (nota >= 5) {
  console.log("Promovat!");
} else {
  console.log("Corigent!");
}

// Bucla for
for (let i = 1; i <= 10; i++) {
  console.log("Linia " + i);
}

// Funcție
function salut(nume) {
  return "Salut, " + nume + "!";
}
```

🎯 **Mini-Quiz:** Ce va afișa `console.log(salut("Maria"))`?
Răspuns: *"Salut, Maria!"*

01

Variabile & Tipuri de Date

Declară variabile cu `let` pentru valori modificabile și `const` pentru constante.

02

Structuri Condiționale

Folosește `if/else` și `switch` pentru a lua decizii în cod.

03

Bucle

Repetă acțiuni cu `for` și `while`. Atenție la bucle infinite!

04

Funcții

Grupează cod reutilizabil în funcții. Arrow functions: `const f = () => {}`

Made with GAMMA

MODULUL 11–15

Exerciții Practice JavaScript & Test de Verificare

Exerciții Practice de Codare JavaScript

Aplică-ți cunoștințele de JavaScript prin aceste exerciții interactive pentru a înțelege mai bine variabilele, funcțiile și structurile de control de bază:

1. **Declarații de Variabile:** Declară o variabilă `let nume = "Ion"`, `let varsta = 30` și o constantă `const PI = 3.14`. Afișează valorile în consolă.
2. **Funcție Simplă:** Scrie o funcție care primește două numere și returnează suma lor. Apelează funcția cu `5` și `10` și afișează rezultatul în consolă.
3. **Condiție `if/else`**: Implementează o condiție care verifică dacă `varsta` este mai mare sau egală cu `18`. Afișează "Este adult" sau "Este minor", în funcție de rezultat.

Test de Verificare a Cunoștințelor

Verifică-ți înțelegerea conceptelor cheie de JavaScript răspunzând la aceste întrebări:

1. Care este principala diferență dintre `let` și `const` în JavaScript?
2. De ce sunt funcțiile esențiale în programare și cum ajută la organizarea codului?
3. Explicați utilitatea unei structuri condiționale `if/else` și dați un exemplu simplu de aplicare.
4. Ce înseamnă "event handling" în JavaScript și cum poate face o pagină web mai interactivă?

💡 **Sfat:** Utilizează consola browserului (F12 în majoritatea browserelor) pentru a testa rapid fragmentele de cod JavaScript!

Made with GAMMA

MODULUL 16–17

DOM & Evenimente JavaScript

DOM (Document Object Model) este reprezentarea arborescente a paginii HTML în memorie. JavaScript poate **selecta**, **modifica** și **șterge** orice element din pagină în timp real.

Manipularea DOM

```
// Selectare element
const titlu = document.getElementById("titlu");

// Modificare conținut
titlu.textContent = "Titlu nou!";

// Modificare stil
titlu.style.color = "#437066";

// Adăugare element
const p = document.createElement("p");
p.textContent = "Paragraf nou";
document.body.appendChild(p);
```

Evenimente

```
// Click pe buton
const btn = document.querySelector("#btn");
btn.addEventListener("click", function() {
  alert("Ai apăsat butonul!");
});

// Eveniment la hover
btn.addEventListener("mouseover", () => {
  btn.style.background = "#437066";
});
```

🎯 **Provocare:** Creează o pagină cu un contor care crește la fiecare click pe un buton, folosind DOM și evenimente!

Made with GAMMA

MODULUL 16–17

Exerciții Practice DOM & Evenimente JavaScript

Acest modul vă provoacă să interacționați direct cu elementele unei pagini web, manipulându-le dinamic și răspunzând la acțiunile utilizatorilor.

Exerciții Practice de Codare DOM & Evenimente

Puneți în practică ce ați învățat despre DOM și evenimente pentru a crea interfețe web dinamice:

1. **Selector & Modificare:** Selectați un element `` după clasa sa, schimbați-i textul și adăugați-i o clasă CSS care să-i modifice culoarea și font-size.
2. **Creare Dinamică:** Creați un element `
3. ` nou, adăugați-i textul "Element Adăugat Dinamic" și inserați-l într-o listă `

` existentă.

    - **Contor cu Evenimente:** Implementați un contor simplu care crește sau scade cu o unitate la apăsarea a două butoane distincte ("Crește" și "Scade"). Afișați valoarea contorului într-un ``.
    - **Toggle Clasă:** Creați un buton care, la fiecare click, adaugă sau elimină o clasă CSS unui `

`, schimbându-i stilul (ex: fundal, margine).

Test de Verificare a Cunoștințelor

Evaluați-vă înțelegerea conceptelor cheie legate de manipularea DOM și gestionarea evenimentelor:

1. Enumerați trei metode diferite de a selecta elemente din DOM folosind JavaScript și când ați folosi fiecare metodă.
2. Ce este `addEventListener` și cum se utilizează pentru a atașa o funcție la un eveniment specific al unui element?
3. Explicați conceptul de "obiect eveniment" (event object) și dați un exemplu de proprietate utilă a acestuia.
4. Care este diferența principală între `textContent` și `innerHTML` atunci când manipulați conținutul unui element?

> 💡 **Sfaturi:** Folosiți `console.log()` pentru a verifica selecțiile DOM și valorile contorului. Nu uitați de `event.preventDefault()`!

Made with GAMMA

MODULUL 18–20

JSON, AJAX & Arhitectura Client-Server

JSON

JSON (JavaScript Object Notation) este formatul universal de schimb de date. Ușor de citit de oameni și mașini. Folosit în API-uri moderne.

```
{"nume":"Ana","varsta":17}
```

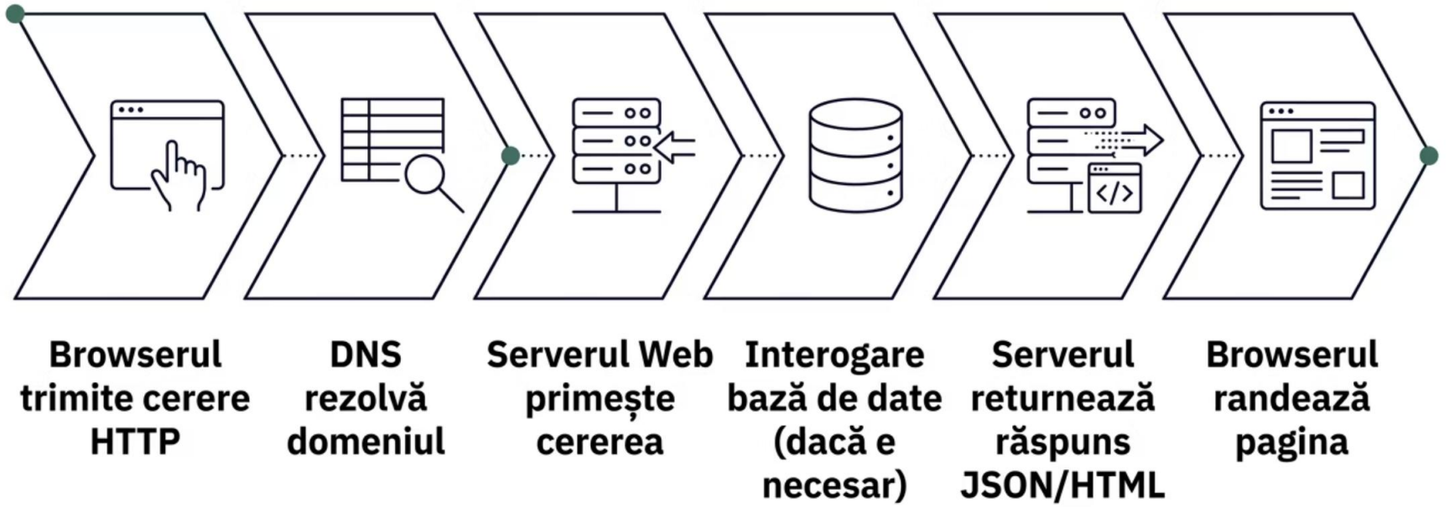
AJAX

AJAX permite actualizarea parțială a paginii fără reîncărcare. `fetch()` este metoda modernă pentru cereri HTTP asincrone.

```
fetch('/api/date')
  .then(r => r.json())
  .then(d => console.log(d));
```

Client-Server

Browserul (client) trimite cereri HTTP către server. Serverul procesează și returnează răspunsuri HTML, JSON sau fișiere. Baza oricărei aplicații web.



Made with GAMMA

MODULUL 18–20

Exerciții Practice JSON, AJAX & Arhitectura Client–Server

Acest modul vă provoacă să lucrați cu schimbul de date și comunicarea asincronă, elemente fundamentale pentru dezvoltarea aplicațiilor web moderne.

Exerciții Practice de Codare

1. **Manipulare JSON:** Creați un obiect JavaScript cu proprietăți precum `nume` și `email`. Convertiți-l într-un șir JSON, apoi parsați șirul JSON înapoi într-un obiect JavaScript. Afișați ambele în consolă.
2. **Cerere AJAX cu Fetch:** Folosiți `fetch()` pentru a face o cerere GET către `https://jsonplaceholder.typicode.com/posts/1`. Parcelați răspunsul JSON și afișați titlul și corpul postării în consolă.
3. **Gestionarea Erorilor:** Modificați exercițiul anterior pentru a include un bloc `.catch()` pentru a prinde și afișa orice erori de rețea sau de parsare a răspunsului.

Test de Verificare a Cunoștințelor

1. Ce avantaje oferă JSON față de alte formate de schimb de date, cum ar fi XML, în contextul dezvoltării web moderne?
2. Explicați conceptul de "asincron" în contextul AJAX și de ce este crucial pentru o experiență utilizator fluidă.
3. Descrieți pașii principali ai unei cereri HTTP de la client la server și ce tipuri de informații sunt schimbate.
4. Care este diferența fundamentală dintre o arhitectură client-server și o aplicație desktop independentă?

💡 **Sfat:** Pentru a depana cererile AJAX, utilizați fila "Network" din uneltele de dezvoltare ale browserului (F12)!

Made with GAMMA

PROIECT FINAL

🚀 Construiește-ți Propriul Site Web!

Aplică tot ce ai învățat într-un proiect complet: un **portofoliu personal** sau un **site de prezentare** cu multiple pagini, stilizare CSS avansată și interactivitate JavaScript.

1 — Săptămâna 1

Structura HTML completă: header, nav, secțiuni, footer semantic

2 — Săptămâna 2

Stilizare CSS: tema de culori, fonturi, Flexbox layout, responsive design

3 — Săptămâna 3

JavaScript interactiv: meniu mobil, formular validat, efecte animate

4 — Săptămâna 4

Publicare pe GitHub Pages și prezentarea proiectului final!

🌟 **Inspirație:** Vizitează **github.com**, **codepen.io** și **frontendmentor.io** pentru exemple reale și provocări de nivel real!

Made with GAMMA

RESURSE & SFATURI

💡 Sfaturi pentru Începători & Resurse Utile

🎯 Top 5 Sfaturi

1. Scrie cod în fiecare zi, chiar și 15 minute
2. Nu memora — înțelege conceptele și caută documentația
3. Greșelile sunt normale — citește erorile cu atenție
4. Construiește proiecte reale, nu doar exerciții
5. Alătură-te comunităților online de programatori

📚 Resurse Recomandate

→ MDN Web Docs

Documentația oficială pentru HTML, CSS și JavaScript — `developer.mozilla.org`

→ freeCodeCamp

Cursuri gratuite interactive cu certificate recunoscute — `freecodecamp.org`

→ W3Schools

Referință rapidă și editor online pentru testare imediată — `w3schools.com`

20+
Lecții complete
De la zero la proiect real

3
Tehnologii învățate
HTML, CSS și JavaScript

100%
Gratuit
Toate resursele sunt accesibile tuturor

1
Proiect final
Site complet publicat online

Începe de la Lecția 1

Descarcă Ghidul PDF

Made with GAMMA