Colegiul Național “Diaconovici-Titetz” Reșița

**Atestat Informatică**

„Combaterea știrilor false”

**Candidat:**

Munteanu Paul Andrei

**Coordonator:**

Cristian Ogășan

1. **Ce este HTML?**

HTML, sau HyperText Markup Language, este limbajul de marcare folosit pentru a crea documente web. Aceste documente pot include text, imagini, sunet și conținut multimedia și sunt afișate în browser-ul web prin intermediul unei conexiuni la internet.

HTML este un limbaj de marcare ce folosește etichete sau marcaje pentru a specifica modul în care trebuie să fie prezentat conținutul într-un document web. De exemplu, eticheta <p> este utilizată pentru a indica un paragraf de text, în timp ce eticheta <img> este utilizată pentru a insera o imagine într-un document web.

Datorită utilizării etichetelor, HTML permite crearea de documente web cu o structură și organizare clară, care poate fi interpretată și afișată într-un mod consistent de către browser-ul web. De asemenea, HTML permite crearea de link-uri către alte pagini web, prin intermediul etichetei <a>, ceea ce face posibilă navigarea între pagini și crearea de site-uri web complexe.

1. **O scurtă istorie a limbajului**

HTML a fost creat în 1989 de către Tim Berners-Lee, un cercetător de la CERN, în scopul de a putea partaja documente între colegii săi. Limbajul a avut o evoluție rapidă în anii 1990, când a fost adoptat de către întreaga comunitate a internetului. Primul standard al HTML a fost publicat în 1993, sub numele de HTML 2.0.

Cu fiecare versiune nouă a HTML, au fost adăugate noi elemente și caracteristici care au permis crearea de conținut mai complex și mai interactiv pe paginile web. Astfel, HTML a trecut prin mai multe versiuni, cum ar fi HTML 3.2, HTML 4.0 și HTML 4.01, fiecare dintre ele adăugând noi funcționalități și îmbunătățiri față de versiunea precedentă.

HTML5 este ultima versiune a limbajului HTML (Hypertext Markup Language), care este utilizată pentru a crea paginile web. De la lansarea sa în 2014, HTML5 a devenit rapid standardul de facto pentru dezvoltarea de aplicații web, oferind o gamă largă de avantaje față de versiunile anterioare ale HTML.

Una dintre cele mai importante schimbări aduse de HTML5 este adăugarea de noi elemente și atribute, care permit dezvoltatorilor web să structureze conținutul paginilor web într-un mod mai logic și mai semantic. Acest lucru înseamnă că HTML5 permite dezvoltatorilor web să creeze paginile web care sunt mai ușor de înțeles de către utilizatori și de către alte aplicații, cum ar fi motoarele de căutare.

O altă caracteristică importantă a HTML5 este suportul pentru media integrat. Cu HTML5, dezvoltatorii web pot adăuga audio și video direct în pagini web fără a mai fi nevoiți să folosească plugin-uri suplimentare, cum ar fi Adobe Flash. Acest lucru face ca paginile web să fie mai ușor de accesat de către utilizatori, deoarece suportul este implicit.

1. **Legătura cu celelalte limbaje de programare web**

HTML, sau Hypertext Markup Language, este un limbaj de marcare de hipertext utilizat pentru a crea structura paginilor web. Împreună cu alte limbaje de programare, cum ar fi JavaScript și CSS, HTML poate fi utilizat pentru a crea experiențe de utilizator interactive și pentru a controla aspectul și stilul conținutului unei pagini web.

1. **CSS**

CSS (Cascading Style Sheets) este un limbaj de stilizare utilizat pentru a defini aspectul și prezentarea unui document scris în limbajul de marcare de hipertext, cum ar fi HTML. Acest limbaj ne permite să controlăm culorile, fonturile, dimensiunea și alte aspecte ale aspectului unui site web sau aplicație, făcându-le mai ușor de întreținut și de actualizat.

CSS a fost creat în anii 1990 de către World Wide Web Consortium (W3C) pentru a oferi o modalitate standardizată de a stiliza documentele HTML. În prezent, majoritatea site-urilor web și aplicațiilor utilizează CSS pentru a le face mai atractive și mai ușor de utilizat de către utilizatori.

Folosind CSS, puteți crea un aspect consistent pentru întregul site web sau aplicație. De exemplu, dacă doriți să schimbați culoarea de fundal a tuturor paginilor dintr-un site, puteți face acest lucru rapid și ușor folosind CSS, în loc să modificați fiecare pagină în parte. De asemenea, puteți utiliza CSS pentru a crea animații și tranziții ale elementelor.

1. **JavaScript**

JavaScript este un limbaj de programare folosit pentru a adăuga interacțiune și dinamism site-urilor web. Acesta este un limbaj de criptare, ceea ce înseamnă că codul JavaScript este scris într-un fișier separat și apoi încorporat într-un document HTML pentru a fi executat de către browser.

JavaScript este un limbaj de programare interpretat, ceea ce înseamnă că este executat direct de către browser fără a fi nevoie de compilare. Acest lucru permite dezvoltatorilor să scrie și să testeze rapid codul, dar poate avea dezavantajul unei performanțe mai scăzute comparativ cu limbajele compilate.

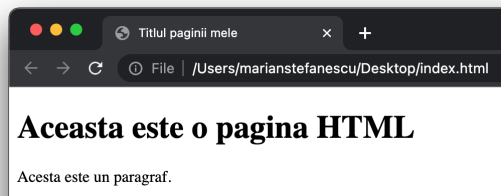
1. **Alegerea editorului de text**

Poți scrie codul HTML intra-un editor de text cum ar fi Note Pad sau TextEdit. Începe prin a deschide editorul de text si a scrie codul HTML in el. Apoi salvează fișierul cu extensia „.html” (de exemplu, „pagina\_mea.html”). După ce ai salvat fișierul, îl poți deschide folosind un browser web (cum ar fi Google Chrome, Firefox sau Safari) si vei putea vedea pagina web pe care ai creat-o. Totuși, ar fi foarte anost să folosești in 2022 un asemenea editor de text. La ora actuală, există alternative mult mai bine. Am scris un articol foarte bun despre cele mai bune editoare de text pentru programare: [check it out](https://code-it.ro/top-10-editoare-de-text-pentru-programare/).

1. **Hello world în HTML**

Începem cu cea mai simplă pagină, un soi de Hello World din HTML. Pagina va conține un titlu de document ce va fi vizibil în bara de titlu a browserului sau tab-ul paginii, un titlu în corpul paginii și un paragraf.

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <title>Titlul paginii mele</title>
5. </head>
6. <body>
7. <h1>Aceasta este o pagina HTML</h1>
8. <p>Acesta este un paragraf.</p>
9. </body>
10. </html>



HTML Hello World

În limbajul HTML există o serie de cuvinte speciale, ce sunt încadrate între <>. Tag-urile pot fi de două tipuri:

* **Taguri pereche**: Un tag HTML pereche începe cu o etichetă de deschidere: numele etichetei inclus în parantezele unghiulare; De exemplu, o etichetă de deschidere a unui paragraf este scrisă ca <p>. Conținutul urmează eticheta de deschidere, care se termină cu o etichetă de sfârșit: numele etichetei care începe cu o bară oblică; de exemplu, o etichetă de sfârșit de paragraf este scrisă ca </p>.
* **Taguri simple**: O etichetă HTML se numește etichetă nepereche sau simplă atunci când eticheta are **doar** o etichetă de deschidere. Uneori sunt denumite și etichete autonome sau etichete singulare.

Codul HTML de mai sus definește o pagină web cu un titlu și un singur paragraf. Elementul <!DOCTYPE html> specifică că aceasta este o pagină HTML5, iar <html> și </html> marchează începutul și sfârșitul paginii HTML. Elementul <head> conține metadate despre pagină, cum ar fi titlul, în timp ce elementul <body> conține elementele care vor fi afișate pe pagina web. În acest caz, pagina conține un titlu de nivel 1 (<h1>) și un paragraf (<p>).

1. **Comentariile in HTML**

Comentariile HTML sunt secțiuni de cod HTML care sunt ignorate de browser atunci când sunt afișate paginile web. Ele sunt utilizate pentru a adăuga observații și instrucțiuni în codul sursă, pentru a ajuta la înțelegerea și întreținerea acestuia. În HTML, comentariile sunt scrise între două simboluri de exclamare și semnul de numărul semnului (<!-- și -->).

1. <!-- Aceasta este un comentariu HTML -->
2. <!-- Aceasta este o secțiune de cod HTML care va fi ignorată de browser -->
3. **Cele mai frecvent utilizate etichete HTML**
4. <p> – folosită pentru a defini un paragraf
5. <br> – folosită pentru a introduce o linie nouă
6. <img> – folosită pentru a include o imagine într-o pagină web
7. <a> – folosită pentru a crea un link către o altă pagină web sau o adresă de email
8. <div> – folosită pentru a grupa elemente HTML într-un container
9. <header> – folosită pentru a defini un container pentru elementele de antet (cum ar fi titlul și meniul de navigare) ale unei pagini web
10. <nav> – folosită pentru a defini o secțiune care conține link-uri de navigare
11. <section> – folosită pentru a defini o secțiune tematică a unei pagini web
12. <footer> – folosită pentru a defini un container pentru elementele de subsol (cum ar fi informațiile de contact și link-urile de politici) ale unei pagini web

Acestea sunt doar câteva dintre etichetele HTML cele mai frecvent utilizate. Există multe alte etichete care pot fi utilizate în HTML pentru a formata și structura conținutul unei pagini web.

1. **Exemplu de pagină web**

O pagină care ar folosi aceste etichete ar putea arată ca în exemplul de mai jos:

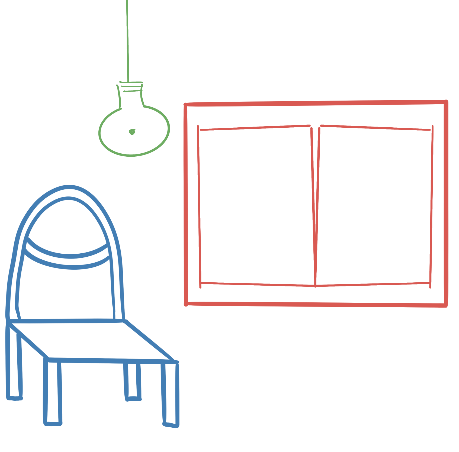
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <title>Titlul Paginii Mele</title>
5. </head>
6. <body>
7. <header>
8. <h1>Titlul Paginii Mele</h1>
9. <nav>
10. <ul>
11. <li><a href="#">Acasă</a></li>
12. <li><a href="#">Despre</a></li>
13. <li><a href="#">Contact</a></li>
14. </ul>
15. </nav>
16. </header>
17. <section>
18. <h2>Despre</h2>
19. <p>Aceasta este o pagină web de exemplu care demonstrează câteva dintre etichetele HTML cele mai frecvent utilizate.</p>
20. </section>
21. <footer>
22. <p>Copyright 2023</p>
23. </footer>
24. </body>
25. </html>

O pagină care utilizează cele mai frecvente etichete HTML

Pentru o înțelegere mai bună, ne putem gândi la **construcția unei case**.

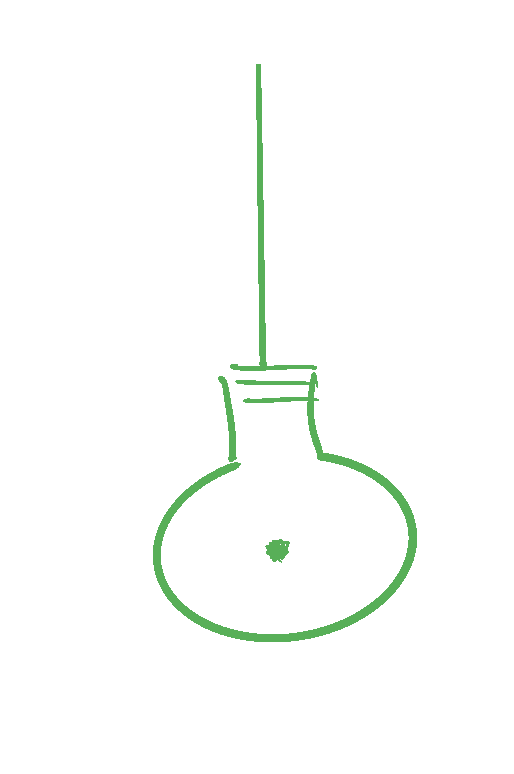
Fundația, pereții și acoperișul fac parte din structura de bază a casei, fără de care nu putem ajunge la un rezultat final. Corespondentul acestei structuri în cazul unei aplicații web este *HTML*.

Așa cum o casă începe cu fundația, construcția unei aplicații web începe cu partea de HTML.



După ce structura de bază este gata, urmează să definim culorile folosite, plasamentul geamurilor, a ușilor, becurilor, mobilei și al altor lucruri ce țin de design.

Pentru un programator FrontEnd, acesta este pasul în care se introduce *CSS*-ul.



La final, trebuie ca geamurile, ușile și becurile puse la pasul anterior să funcționeze corespunzător și să putem interacționa cu ele. Așa cum probabil ați ghicit, în cazul unei aplicații web, aceste operații îi revin *Javascript*-ului.

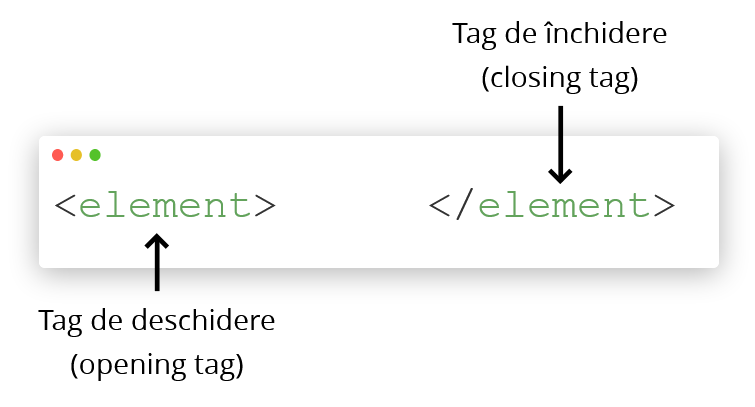
Noi ne vom concentra aici pe primul limbaj dintre cele trei enumerate mai sus, și anume HTML.

**HTML (Hyper Text Markup Language)** este limbajul standard de markup folosit pentru a defini structura de bază a unei pagini web. Nu este un limbaj de programare propriu-zis, ci doar un limbaj descriptiv, bazat pe XML.

Majoritatea persoanelor consideră că HTML este foarte ușor și îl tratează destul de superficial. Chiar dacă HTML este, într-adevăr, mai ușor decât celelalte două limbaje, el trebuie învățat bine și corect. Dacă vei face asta, vei avea o fundație foarte bună peste care vei putea așeza cunoștințele de CSS și JS 💪.

## [#](https://frontend.ro/html/despre-html#elemente)Elemente

Orice fișier HTML este construit din elemente, numite și **tags**. Ele sunt formate din perechi de forma **<element> </element>**. Primul se numește **tag de deschidere**(opening tag), iar al doilea **tag de închidere**(closing tag).

[](https://d3tycb976jpudc.cloudfront.net/public/images/lessons/about-html/opening-closing-tag.png)Tag de deschidere și închidere

Concret, elementul **<section>** are drept conținut alte două elemente (**<h1>** și **<p>**), în timp ce aceste două elemente au în interior doar conținut de tip text.

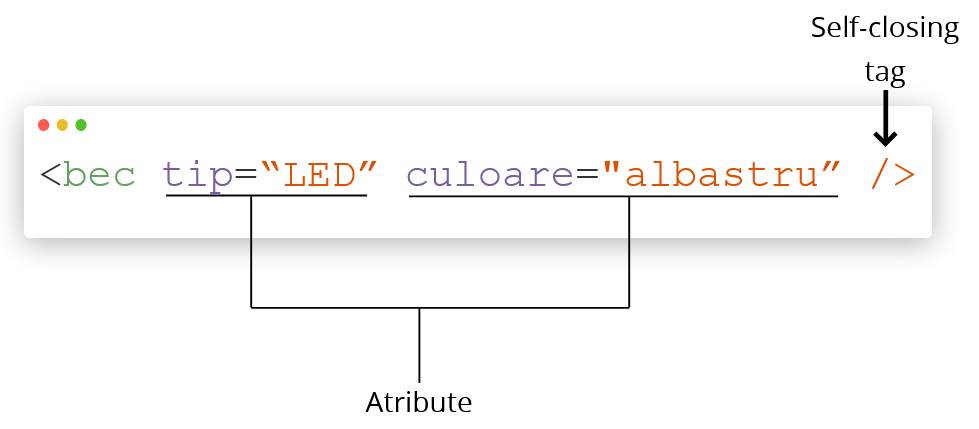
Există și elemente care nu au nevoie de conținut (de exemplu, elementul **input**sau**img**). Acestea se numesc **self-closing tags**, iar modul de scriere este puțin diferit, și anume **<element />**. În exemplul de mai jos putem vedea modul de folosire al unor taguri self-closing:

Probabil te întrebi ce înseamnă **type="text"** si **placeholder="Nume"**. Ele se numesc **atribute**. Hai să vedem care este scopul acestora.

## [#](https://frontend.ro/html/despre-html#atribute)Atribute

Fiecare element poate avea o serie de atribute prin care noi putem fie să **controlam comportamentul acestuia**, fie să îl **identificăm ulterior** din CSS sau JS. Atributele sunt perechi de tip **cheie=“valoare”** care sunt scrise în interiorul unui opening tag.

Revenind la analogia de la început, dacă ne gândim la un bec, atribute posibile ar putea fi tipul becului (LED sau incandescent), culoarea becului, intensitatea acestuia, mărimea, etc. În HTML, un element bec care este de **tip LED**, de **culoare albastră**, ar putea fi scris după cum urmează:

[](https://d3tycb976jpudc.cloudfront.net/public/images/lessons/about-html/attributes.png)Atribute pentru un self-closing tag

## [#](https://frontend.ro/html/despre-html#relatii)Relații de rudenie

Bineînțeles că o aplicație web nu conține doar unul sau două elemente. În majoritatea cazurilor, avem nevoie de zeci sau sute de elemente doar pentru o singură pagină. Aceste elemente sunt adăugate pe baza unor **relații de rudenie**.

Exista trei tipuri de elemente, în funcție de aceste relații:

* elemente de tip părinte - **parent elements**
* elemente de tip copil - **child elements**
* elemente de tip frate / sora - **sibling elements**

În exemplul de mai sus, titlul (**h1**) și paragraful (**p**) sunt siblings, pentru că se află pe același nivel.

Mai jos, se poate observa relația de tip **părinte-copil**, unde elementul **<div>** este părinte pentru elementul **<p>**.

<**div**>

<**p**>Acesta este un paragraf</**p**>

</**div**>

## [#](https://frontend.ro/html/despre-html#resurse)Resurse

Ceea ce am prezentat mai sus este doar vârful iceberg-ului. În lecțiile următoare vei învăța care este structura de bază a unei aplicații web, cum să îți validezi HTML-ul scris, precum și informații mai detaliate despre cele mai importante elemente. Până atunci, îți vom lăsa mai jos o listă de resurse cu elemente și atribute de bază"

### Resurse suplimentare

1. [Lista completă de elemente HTML de pe MDN](https://developer.mozilla.org/en-US/docs/Web/HTML/Element)
2. [Lista completă de atribute posibile de pe MDN](https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes)

Când facem referire însă la un site nu trebuie sa ne gândim doar la ceea ce vedem atunci când îl accesam, ci si la ceea ce se afla in spatele acelei pagini ce ne este afișata. Iar când vorbim despre ceea ce se afla in spatele unei pagini web trebuie sa ne ducem cu gândul la HTML – [Hyper Text Markup Language](https://en.wikipedia.org/wiki/HTML).

Mai mult ca sigur, ai mai auzit pana acum despre acest HTML chiar si daca IT-ul nu este chiar domeniul tău de interes. Putini sunt însă cei care știu ce înseamnă cu adevărat HTML si care este rolul lui in toata aceasta poveste. Ei bine, lucrurile nu sunt chiar atât de simple!

Pe scurt însă, am putea spune ca HTML este un limbaj de editare, prin intermediul lui fiind afișate toate acele informații pe care noi le vedem atunci când accesam o pagina web.

Citește si: [Ce este Bicorn, cum este generata si ce avantaje oferă](https://ceeste.ro/ce-este-bitcoin/)

Practic, prin acest limbaj de editare, cel care construiește pagina web poate forma textul, poate introduce fotografiile, fișierele video sau audio de pe site si așa mai departe. Introducerea acestor texte, fotografii, link-uri, video-uri se face prin intermediul unor comenzi specifice scrise in limbajul HTML.

Sigur, se poate spune ca pentru construirea unui site web exista la ora actuala si alte soluții, precum Word Press spre exemplu. Chiar daca un site se poate construi foarte ușor având la baza un template oferit de Word Press, intr-adevar, limbajul HTML ne poate salva in foarte multe situatii.

Spre exemplu, in cazul in care se intampla ca ceva sa nu functioneze așa cum trebuie pe site, solutia cea mai simpla este sa apelam la HTML si sa facem modificarile necesare in codul sursa al paginii.

## Cum functioneaza acest limbaj



Cod website

In cazul in care este facut corect, codul sursa al paginii, pe care il scriem in format HTML, transmite browser-ului felul in care trebuie sa arate site-ul. Limbajul HTML utilizeaza un set de tag-uri de marcare cu scopul de a descrie paginile web. De cele mai multe ori, aceste tag-uri de marcare sunt denumite tag-uri HTML. Practic, aceste tag-uri HTML sunt niste cuvinte cheie inconjurate de paranteze unghiulare.

In general, acestea sunt folosite in pereche, precum <html> si </html>. In acest caz, primul tag din pereche este considerat a fi tag-ul de inceput sau de deschidere, in timp ce cel de-al doilea tag este considerat a fi tag-ul final sau de incheiere.

Folosind corect aceste tag-uri, cel care construieste site-ul transmite browser-ului felul in care vrea sa arate pagina ce urmeaza sa fie afisata vizitatorului. Prin aceste tag-uri, browser-ul stie ce trebuie sa faca si cum sa afiseze o anumita informatie. Spre exemplu, se poate comanda ingrosarea textului, inserarea unui tabel, introducerea unei fotografii, introducerea unui fisier video si asa mai departe.

         Exemple de tag-uri HTML:

* h1 – Titlul;
* p – Paragraf nou;
* img – Introducerea unei fotografii;
* table – Introducerea unui tabel;
* b – Ingrosarea unui text;
* u – Sublinierea unui text;
* del – Taierea unui text.

Chiar daca toate aceste tag-uri nu sunt vazute de vizitatorul unui site, acestea exista in codul sursa al paginii. Fara aceste tag-uri HTML, toata informatia prezenta pe site nu ar putea fi vazuta in conditii optime. Asadar, putem intelege foarte usor de ce acest limbaj HTML este atat de important atunci cand vorbim despre construirea unei pagini web.

## Dintre miile de editoare de HTML…

Ei bine, ia un editor de texte si apuca-te de scris pentru ca vom invata impreuna **HTML**. Iar cand vorbesc de editoare de text ma refer la cel mai simplu: **Notepad**. Daca insa vrei sa ai si putin ajutor, iti recomand [Notepad++](http://notepad-plus-plus.org/), un editor gratuit, care te ajuta la editarea paginilor HTML prin identarea si colorarea elementelor de limbaj specifice HTML.

**Structura de baza a unei pagini html**, adica elementele fara de care o pagina html nu ar mai fi o pagina html, consta in urmatorul cod:

Dupa cum vezi, codul incepe cu declararea paginii ca fiind una de tip “HTML”, browserul intelegand astfel ce limbaj sa foloseasca atunci cand interpreteaza documentul pentru a-l afisa.

De asemenea, ca orice fiinta, aceasta pagina are un cap (“<head></head>”) si un corp (“<body></body>”). In cadrul capului, este definit titlul paginii. Acest titlu al paginii se regaseste in title bar-ul browserului.

Desi nu este obligatorie, introducerea acestei secvente de cod este importanta.

## Tag-urile

De asemenea, poti observa ca in codul HTML apar multe secvente de cod cuprinse intre “<” si “>”. Acestea sunt numite in mod curent **tag**-uri. **Tag**-ul este folosit pentru a specifica regiuni ale documentului HTML, pe care le va interpreta ulterior browser-ul. **Tag**-ul este o secventa de cod cuprinsa intre caracterele “<” si “>”.

## Elementele

De cele mai multe ori informatiile dintr-o pagina html sunt cuprinse intre doua tag-uri: unul de deschidere (**<tag>**) si unul de inchidere (**</tag>**). Tagul de deschidere si de inchidere formeaza impreuna un element. Spre exemplu, in codul scris mai sus, “<h1>Primul meu titlu</h1>” este un element numit **heading 1**.

## Atributele

**Atributele** sunt folosite pentru a modifica valoarea unui element in HTML. De obicei un element are mai multe atribute.

Despre atribute vom invata mai multe cand le vom intalni. Pentru moment, tot ce trebuie sa stii este ca **un tag este o comanda pe care browser-ul o interpreteaza**, ca **un element este un tag complet**, iar **un atribut modifica un element in HTML**.

# Ce este CSS si pentru ce se poate utiliza

[[https://secure.gravatar.com/avatar/afb26b307ed4fc88c920243b73458220?s=28&d=wavatar&r=g](https://www.thc.ro/blog/author/thcservers/) THC.ro](https://www.thc.ro/blog/author/thcservers/), [1 an in urma](https://www.thc.ro/blog/ce-este-css-si-pentru-ce-se-poate-utiliza/) [0](https://www.thc.ro/blog/ce-este-css-si-pentru-ce-se-poate-utiliza/#comments)  10 min  1976

CSS este unul dintre cele mai importante limbaje de programare. Acesta se utilizeaza pentru a defini stilurile pentru paginile tale web, inclusiv designul, aspectul si variatiile de afisare pentru diferite dispozitive si dimensiuni de ecran. Astfel, cu ajutorul lui se stabilesc fonturi, alineate, culori si multe alte configuratii pe care site-ul tau trebuie sa le aiba. De ce este atat de important? Deoarece, ajuta utilizatorul sa aiba o experienta placuta in timpul navigarii si simplifica modul in care informatiile sunt prezentate.

Cuprins:

1. Ce este CSS ?
2. CSS in HTML – ce inseamna stilizarea paginilor HTML
3. Limbajul de programare CSS – caracteristici
4. Elemente si atribute in CSS

## Ce este CSS ?

Despre CSS se poate spune ca este un limbaj de programare, care se axeaza pe stilul utilizat in prezentarea unui document scris intr-un limbaj de marcare, cum ar fi HTML.  Denumirea CSS provine din expresia Cascading Style Sheets si este o tehnologie de baza a World Wide Web, alaturi de HTML si JavaScript.

El este conceput pentru a permite stilizarea continutului unei pagini web si a prezentarii acesteia, inclusiv a aspectului, fontului si culorilor. Aceasta procedura imbunatateste in mod real accesibilitatea continutului, ofera mai multa flexibilitate si control in ceea ce priveste specificatiile caracteristicilor prezentarii, permite ca mai multe pagini web sa partajeze formatarea si reduce repetarea si complexitatea continutului structurat.

De asemenea, formateaza pagina in asa fel incat sa fie fezabila pe diferite tipuri de ecrane, prin voce, dar si pe dispozitivele tactile Braille. Mai mult, are reguli pentru formatarea alternativa in cazul in care continutul este accesat de pe un dispozitiv mobil si modifica aspectul paginii in asa fel incat sa poate fi vizibil din orice perspectiva.

**CSS este unul dintre cele mai utilizate limbaje de stilizare de pe web, din mai multe motive:**

* **ajuta la economisirea timpului**– codul se poate scrie o singura data si apoi se poate reutiliza in mai multe pagini HTML. De asemenea, puteti defini un stil pentru fiecare element HTML si il poti aplica la oricate pagini web ai nevoie.
* **paginile se incarca mai rapid** – cu cat adaugi mai mult cod unei pagini, cu atat aceasta se va incarca mai greu. CSS iti permite sa utilizezi cat mai putin cod printr-o anumita regula, ce poate fi aplicata la toate etichetele dintr-un document HTML. Astfel, daca ai una sau mai multe pagini de produse care ar trebui sa aiba aceeasi formatare, CSS te va ajuta ca aceasta procedura sa fie realizata doar prin adaugarea unui cod.
* **ofera o experienta mai buna a utilizatorului** – pe langa faptul ca faciliteaza vizibilitatea unei pagini web din perspectiva utilizatorului, permite si formatarea foarte usoara a acesteia. Atunci cand ai butoane si text bine organizat, amplasate in locuri logice pentru cititor, experienta lui se imbunatateste foarte mult.
* **modificari usoare de formatare** – oricand ai nevoie sa modifici anumite formate ale paginilor tale web, CSS iti faciliteaza acest lucru, fara a fi nevoie sa remediezi fiecare pagina in parte. Trebuie sa editezi doar foile de stil corespunzatoare si codul va fi aplicat tuturor paginilor care le utilizeaza.
* **este compatibil pe toate dispozitivele** – nimic nu este mai neplacut decat sa nu poti accesa o pagina anume atunci cand utilizezi un ecran mai mic. Toate paginile ar trebui sa fie vizibile, smart, si foarte usor de navigat, indiferent de dispozitivul care le acceseaza.
* **standarde web globale** – HTML are aproximativ aceleasi atribute ca si CSS, insa este considerat un limbaj invechit pentru stilizare. De aceea este indicat ca in formatarea site-ului tau sa folosesti CSS, pentru ca pe viitor paginile sa fie compatibile cu toate browserele si sa nu intampini probleme in acest sens.
* **este superior HTML-ului** – daca iti doresti sa beneficiezi de o gama cat mai larga de atribute, CSS este limbajul de stilizare pe care trebuie sa il folosesti. Te va ajuta sa oferi un aspect mult mai bine structurat si mai usor de parcurs pentru utilizatori decat HTML. In plus, fata de HTML si alte limbaje accepta utilizarea CSS, inclusiv XHTML, XML simplu, SVG și XUL.

Astfel, CSS este un limbaj care transforma un site web intr-o platforma foarte atragatoare, dar simpla in acelasi timp si usor de vizualizat. In timp ce HTML determina in mare masura continutul texului, CSS determina structura vizuala, estetica si aspectul. Un lucru foarte important de care trebuie sa tii cont inainte de a utiliza CSS este cunoasterea limbajului HTML. Acesta este baza de la care trebuie sa pornesti.

## CSS in HTML – ce inseamna stilizarea paginilor HTML

HTML si CSS sunt limbaje de programare diferite, care utilizeaza coduri diferite, dar reusesc sa se completeze perfect unul cu celalalt. Totusi, o formatare in CSS nu va functiona pentru HTML si invers. De aceea este important ca cele doua sa fie studiate separat, pentru a intelege cat mai bine cum functioneaza.

CSS este un limbaj pentru specificarea modului in care documentele sunt prezentate utilizatorilor – modul în care sunt stilizare, asezate etc. Un document este de obicei un fisier text structurat care foloseste un limbaj de markup – HTML, dar poti intalni si alte limbaje, cum ar fi SVG sau XML.

Prezentarea unui document unui utilizator inseamna transformarea acestuia într-un formular utilizabil de publicul site-ului tau. Browserele sunt concepute pentru a prezenta documente vizual, de exemplu, pe un ecran de computer, tableta, telefon, etc. CSS poate fi folosit pentru stilul textului documentului de baza si se foloseste in schimbarea culorii si dimensiunii titlurilor si a linkurilor. De asemenea, poate fi utilizat chiar si pentru efecte precum animatia.

Pentru a intelege cat mai bine cum functioneaza CSS cu HTML, poti face un exercitiu de imaginatie. De exemplu, HTML este motorul unei masini, iar CSS caroseria acesteia. Masina va functiona fara acestea din urma, dar cu siguranta nu va avea un aspect prea placut. Acelasi lucru se intampla si in cazul site-urilor web. Ele poate fi accesate, insa daca ofera o experienta neplacuta utilizatorului, acesta poate fi considerat un client pierdut. Astfel, CSS face ca partea din fata a unui site sa creeze o placere, atat in ceea ce priveste vizualul, cat si in partea redactata pe care cineva ar trebui sa o citeasca. Fara CSS, site-ul tau va avea un aspect mai putin placut si va fi mult mai greu de navigat.

Practic, utilizarea CSS se imparte in doua categorii principale: prima consta in stabilirea unor fonduri, culori, dimensiuni si stiluri de text, iar ce-a de-a doua in definirea elementelor in cadrul unei pagini. Exista foarte multe programe permit construirea unui site standard, cu elemente predefinite, iar aceasta abordare este potrivita pentru paginile care sunt la inceput de drum si inca se lucreaza la estetica lor. Totusi, ele nu sunt potrivite pentru site-urile web care au un trafic mare de utilizatori si isi doresc sa se dezvolte, prezentand produse multiple si content din ce in ce mai mult. Aici intervine CSS. Cu ajutorul lui iti poti proiecta dupa bunul plac aspectul site-ului tau.

De asemenea, un lucru important de care trebuie sa tii cont pentru a te asigura ca site-ul tau functioneaza la parametrii maximi este [gazduirea web](https://www.thc.ro/). Orice pagina care isi doreste sa aiba un flux constant de vizitatori si content de care acestia ar trebui sa fie constienti se bazeaza pe servicii de hosting de inalta calitate. Mai mult decat atat, folosirea acestora iti va sustine site-ul si se va asigura ca ai partea de o viteza de incarcare superioara, precum si [securitate](https://www.thc.ro/certificate-ssl), care este extrem de esentiala.

## Limbajul de programare CSS – caracteristici

CSS este un limbaj de programare foarte usor de folosit, destinat simplificarii procesului de prezentare a paginilor web. Astfel, daca iti doresti ca utilizatorii tai sa aiba parte de o experienta inedita este indicat sa apelezi la un inginer software care sa lucreze in domeniul dezvoltarii web. Limbajul de codare CSS este interschimbabil. Scopul acestuia de baza este de a aduce o practica mai standardizata pentru dezvoltarea web. Folosind un cod CSS sau vei putea urmari rapid efortul de dezvoltare web, deoarece iti permite sa utilizezi elemente web predefinite.

Stilurile pentru o pagina se pot defini in **Head**-ul documentului HTML, intr-un fisier css extern sau in partea de **Body**.

**Toate stilurile se defintesc prin taguri (coduri) si se pot folosi pentru:**

* culoarea unei pagini (background)
* selectarea unei imagini de fundal a paginii web
* repetarea unei imagini de fundal pe orizontala
* pozitionarea imaginii de fundal
* setarea latimii celor patru frontiere ale paginii (stanga sus, jos/ dreapta sus, jos)
* setarea stilului celor patru frontiere ale paginii (stanga sus, jos/ dreapta sus, jos)
* setarea culorilor
* capitonarea unui element in pagina
* stabilirea unei culori a textului in cadrul diferitelor elemente
* alinierea si decorarea textului
* spatierea dintre caractere
* aliniera unei imagini in interiorul unui text
* setarea fondului, dimensiunii, stilului si culorii unui text
* crearea cutiilor tip link
* adaugarea tabelelor in pagina
* adaugarea conturului unui text sau a unei imagini
* introducerea [adreselor URL](https://www.thc.ro/blog/adresele-url-modul-ingenios-prin-care-navigarea-web-este-accesibila-oamenilor/)
* crearea galeriilor de imagini
* transformarea 2D a diferitelor elemente (rotire, micsorare, marire, mutare)
* adaugarea unei animatii intr-un element
* configurarea imaginilor
* adaugarea butoanelor
* paginare
* adaugarea de culoare in cadrul paginii

Astfel, folosirea limbajului CSS este esentiala. Cu el iti poti customiza pagina web dupa bunul tau plac. Pe langa acesta, hostingul si baza de date sunt celelalte elemente de care site-ul tau trebuie sa beneficieze pentru a te asigura ca functioneaza asa cum trebuie.

## Elemente si atribute in HTML si CSS

Cei mai importanti termeni ai limbajului de programare HTML sunt elementele, tagurile şi atributele:

* **elementele** – definesc continutul si structura obiectelor care exista intr-o pagina. Cele mai cunoscute elemente sunt antetele (de la h1 pana la h6), paragrafele si care se scriu **<p> </p>**. Simbolurile de la inceputul numelui elementului („mai mic” si „mai mare” sunt numite **taguri**sau **etichete**, care se gasesc mereu in pereche si nu pot exista separat. Linia orizontala de la final ii ofera programului informatia ca respectivul text s-a incheiat. Astfel, tagul de incheiere este „/”. Ceea ce se afla intre cele doua taguri („<” si „>”) reprezinta continutul acelui element. Exemplu:

**<p> …text… </p>**

* **atributele**– de cele mai multe ori acestea se folosesc pentru a oferi mai multe informatii despre un element. De exemplu: **id**(identifica un element unic) **class**(clasifica un element) **src**(mentioneaza sursa unui continut) si **href**(inseamna referinta unui hyperlink catre o resursa. Atributele se plaseaza in cadrul unui tag, chiar la inceputul acestuia, dupa numele elementului. Ele includ o valoare si un nume. De exemplu:

**nume, urmat de egal si de valoarea atributului, intre ghilimele**

**Cei mai importanti termeni ai limbajului de programare CSS sunt selectorii, valorile şi proprietarile:**

* **selectorii**– acest tool identifica carui element din codul HTML i se asociaza stilul (**style**), precum marime, culoare, pozitie, etc. Acestia pot include o anumita selectie multipla, cum ar fi un singur rand dintr-un paragraf, doar anumite paragrafe sau toate paragrafele. Selectorii sunt urmati de acolade. In interiorul acestora se mentioneaza aspectul pe care vrei ca acel element sa il aiba. Se pot folosi proprietati precum **background**,**height sau width**, **color**, etc.  De exemplu:

**p {**

**color: …;**

**font size: …;**

**}**

* **valorile**– dupa ce au fost selectate elementele si s-a stabilit aspectul pe care il doresti ca acestea sa il aiba, urmeaza sa se determine proprietatea lor, prin folosirea unei valori. De exemplu, in ceea ce priveste culoare, valoare poate fi „galben”, iar in ceea ce priveste font size-ul valoarea poate sa fie „12”. Ele se amplaseaza intre cele doua puncte si punctul si virgula de la final. Luand exemplul de mai sus, codul se va transforma in :

**p {**

**color: yellow;**

**font-size: 12px;**

**}**

Prin aceste coduri, textul care apare in pagina ta web se poate modifica dupa bunul plac. CSS si HTML nu sunt limbaje de programare foarte complicate, insa trebuie sa exersezi constant pentru a le putea folosi la un nivel avansat.

Asadar, daca iti doresti ca pagina ta web sa aiba un aspect vizual ingrijit si placut si sa poata fi accesata cat mai eficient, trebuie sa apelezi la un limbaj de programare precum CSS. De asemenea, nu uita de faptul ca orice site cu un flux mare de utilizatori are nevoie de o gazduire web performanta.

Pentru folosirea (css-ului), acestui limbaj de stilizare, neaparat trebuie sa cunoasteti limbajul html. Daca limbajul html nu va este familiar atunci va sfatuiesc sa urmati aceste [tutoriale html](https://it.webdesign-galaxy.ro/gazduire/tutoriale-html/).  
**Cum arata un stil CSS?**

p {  
font-family:Arial;  
font-size:14px;  
color:#003300;  
}

**Acest stil definit mai sus, rezulta intr-un stil care arata asa:**  
Acestui text a fost aplicat stilul de mai sus.

## Unde definim stilurile CSS ?

Stilurile pentru o pagina pot fi definite **in partea de Head** a documentului html, pot fi definite intr-un **fisier css extern**, pe care putem chema tot din partea de head a paginii, sau **putem aplica un stil diferit in partea Body** a fisierului html, la fiecare tag html in parte.

## Stil css definit in HEAD

**Exemplu**

<head>  
<title>Stil definit in head</title>  
<style type=”text/css”>  
p {  
font-family:Arial;  
font-size:14px;  
color:#003300;  
}  
</style>  
</head>  
<body>

Paragraf la care se va aplica stilul definit mai sus!

</body>

Dupa cum vedeti stilurile se definesc in partea **head** a documentului cu ajutorul tagurilor **<style type=”text/css”>** si **</style>**. In acest exemplu am definit cu css un stil pentru toate paragrafele din pagina respectiva, adaugand un stil la tagul **<p>**.  
Haideti sa vedem cum am putea suprascrie in pagina un stil definit deja.

## Stil css in partea BODY ( css inline )

In partea **body** a fisierului html, putem adauga un stil diferit la fiecare tag html, ba chiar putem si suprascrie un stil definit in partea **head**.  
**Exemplu**

<body>  
<p style=”font-family:Arial; color:#478600;font-size:14px;”>  
Paragraf care a fost stilizat in interiorul tagului p! </p>

</body>

**Reulta in:**

Paragraf care a fost stilizat in interiorul tagului p!

In acest exemplu am folosit **Css inline** pentru a adauga un stil. Cu aceasta metoda putem adauga un stil cu css la orice tag html, si putem suprascrie stilul definit in partea head. Pentru a adauga un stil diferit fata de restul documentului, am folosit **atributul style** pentru a redefini, a adauga stilul respectiv. Parametri ca si in cazul stilurilor definite in head, sunt separate prin (;) punct si virgula.

**Forma: tag-html style=”….” >**

## Definirea stilurilor cu un fisier extern

Pentru a tine toate stilurile intr-un fisier extern, tot ce trebuie sa facem, este sa definim stilurile intr-un fisier cu extensia **css**. exemplu: **stiluri.css**pe care putem invoca, chema din partea de **head** a documentului.  
**Exemplu:**

<head>  
<link rel=”stylesheet” href=”stiluri.css”/>  
</head>

## Esenta Css-ului In ce consta puterea – Tutorial Css

Cu ajutorul stilurilor CSS putem sa stilizam situri intregi dintr-un singur fisier. Asta rezulta in mai put in cod, care va face paginile sa se incarce mai repede. Daca in viitor vrem sa facem o modificare pe site, nu va trebui sa editam fiecare pagina, articol in parte, ci vom modifica un singur fisier, fisierul css extern.  
Css-ul ne da libertatea sa suprascriem un stil deja definit, pentru ca lucreaza cu o anumita ierarhie, adica:  
**1:**– Prima data va fi luat in considerare **stilul definit in fisierul extern**, dupa care urmeaza **2: stilul definit in partea head**, iar ultima data **3:** cu **css-ul inline**, adica stilul definit intr-un tag html cu ajutorul atributului **style** putem suprascrie orice stil definit anterior.

**CSS3 - Borduri**

Acum CSS3 oferă posibilitatea de a crea borduri cu colțurile rotunjite fară a folosi elemente grafice de fundal așa cum se folosea anterior acestui upgrade.

Proprietatea CSS3 border-radius definește prin valorile exprimate in pixeli cat de rotunjite vor fi colțurile unui element HTML sau unei imagini. Fiecare colț poate avea o alta valoare exprimată in pixeli diferită de un alt colț al aceluiași element. Prin urmare putem folosi pana la 4 valori diferite atribuite unui element HTML sau imagine.

Exemplu:

border-radius: 5px ;

- definește valoarea de 5px radius pentru toate cele 4 colțuri ale elementului.

border-radius: 5px 7px 12px 4px;

- aceste valori multiple definesc cat de mult vor fi rotunjite colțurile elementului HTML, iar pentru fiecare colt este specificata valorarea. Colțul stanga-sus are valoarea border-radius de 5px, colțul dreapta-sus are valoarea border-radius de 7px, colțul dreapta-jos al elementului HTML are valoarea de 12px iar colțul din stanga-jos are valoarea de 4px.

CSS3 - Borduri Rotunjite - Optimizat

Varianta ne-comprimată sau ne-optimizată:

border-radius-left: 5px;

border-radius-right: 7px;

border-radius-top: 12px;

border-radius-bottom: 4px;

Varianta mimificată, compresată/optimizată:

border-radius: 5px 7px 12px 4px;

Ambele variante sunt corecte și acceptate de clientul browser.

CSS3 - Borduri Rotunjite - Compatibilitate Browser

Pentru compatibilitatea cu diferite browsere se folosesc prefixe: *-webkit- , -moz- , -o-*

Compatibilitate: Internet Explorer (IE) - 0.9 , Chrome folosește prefixul -webkit- pentru 4,0 , Firefox folosește prefixul -moz- pentru versiunea 3.0, Safari folosește prefixul -webkit- pentru versiunea 3.1, Opera 10.5 prefix -o-

**Exemplu CSS3 border-radius:**

div {

border: 2px solid #333333;

padding: 10px 40px;

background: #dddddd;

width: 300px;

border-radius:25px;

}

Elementul HTML div este definit de urmatoarele proprietăți CSS: dimensiunea in lungime este redată de valoarea in pixeli a proprietății width, folosește o bordură de 2 pixeli, o bordură solidă de culoare gri-inchis definită de caloarea HEX #333333. Culoarea de fundal este gri deschis definită de HEX #dddddd. Bordura rotunjită este de 25 pixeli pentru toate cele 4 colțuri.

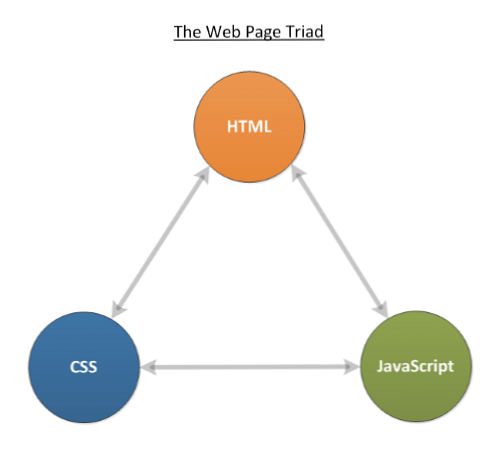
# Ce este JavaScript și cum funcționează?

JavaScript este un limbaj de programare pentru web. Este susținută de majoritatea browserelor web, inclusiv Chrome, Firefox, Safari, Internet Explorer, Edge, Opera etc. Cele mai multe browsere mobile pentru telefoane inteligente suportă și JavaScript.

Este folosit în primul rând pentru a îmbunătăți paginile web pentru a oferi o experiență mai ușor de utilizat. Acestea includ actualizarea dinamică a paginilor web, îmbunătățiri ale interfeței cu utilizatorul, cum ar fi meniuri și casete de dialog, animații, grafică 2D și 3D, hărți interactive, playere video și multe altele. Acest mod de utilizare a JavaScript în browser-ul web este, de asemenea, cunoscut sub numele de **client javascript**.

## Triada paginii web

Când luați în considerare componentele care formează o pagină web, JavaScript formează cea de-a treia componentă a triadei, HTML și CSS fiind celelalte două. HTML descrie pagina, inclusiv textul, graficele etc. CSS este folosit pentru a controla și personaliza aspectul paginii web, inclusiv culorile, fonturile etc. JavaScript-ul este folosit pentru a adăuga o componentă dinamică la pagina web și pentru a face cea mai mare parte elemente de pe pagina programabilă.



## Istoricul JavaScript

Dezvoltarea JavaScript a început în 1995 la Netscape Communications, producătorii browserului Netscape. Ei și-au dat seama că adăugarea a “limba lipiciului” pentru a îmbunătăți experiența utilizatorilor web ar crește absorbția utilizatorilor. Așa că au adus-o pe Brendan Eich pentru a încorpora limba de programare a programului. Cu toate acestea, deoarece Java a fost, la acea vreme, noua limbă nouă a webului, ei au decis să facă limba mai aproape de sintaxa Java. Rezultatul a fost JavaScript, cu caracteristici ale Schemei, orientarea obiectului SmallTalk și sintaxa Java. Prima versiune a acestei limbi a fost numită de fapt Mocha în mai 1995, redenumită la LiveScript în septembrie 1995 și redenumită din nou la JavaScript în decembrie 1995.

În 1996, JavaScript a fost trimis la ECMA International pentru finalizarea ca specificație standard. În iunie 1997, prima specificație oficială pentru limbă a fost lansată ca ECMA-262. Cea mai recentă versiune a limbii este ECMAScript 2017, care a fost lansată în iunie 2017.

## Ce pot face cu JavaScript?

JavaScript este un interpret de limbaj de programare integrat încorporat în browserul dvs. web. Puteți face orice în JavaScript care permite o limbă obișnuită, cum ar fi Java. Acestea includ:

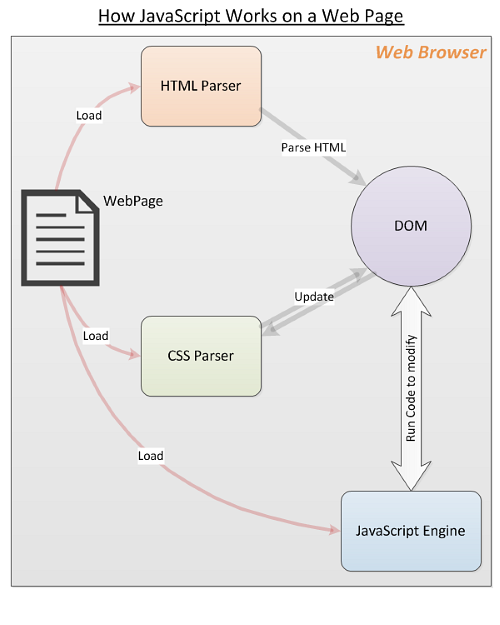
* Declarați variabilele
* Stocați și preluați valorile
* Definirea și invocarea funcțiilor
* Definiți propriile clase
* Încărcați și utilizați module externe
* Scrieți instrucțiuni de gestionare a evenimentelor care răspund la evenimentele utilizatorilor și alte evenimente
* Și mult mai mult

Browserul web încarcă o pagină web, analizează codul HTML și creează ceea ce este cunoscut sub numele de Document Object Model (DOM) din conținut. DOM-ul prezintă o vizualizare live a paginii web la codul dvs. JavaScript. Codul dvs. poate apoi să actualizeze DOM-ul și să îl prezinte instantaneu utilizatorului. Browserul vă permite, de asemenea, să vă înregistrați codul pentru a fi notificat la evenimentele interfeței cu utilizatorul, cum ar fi mișcarea mouse-ului, apăsarea butonului etc. Utilizând toate aceste facilități, puteți crea aplicații puțin interesante (și nu atât de puțin).

***Avertizare:*** Deoarece JavaScript este un limbaj atât de puternic JavaScript și dezvoltare web: Utilizarea modelului de obiect de documente JavaScript și dezvoltarea web: Utilizarea modelului de obiect de document Acest articol vă va prezenta scheletul de documente cu care lucrează JavaScript. Având o cunoaștere de lucru a acestui model de obiect abstract, puteți scrie JavaScript care funcționează pe orice pagină web. Citiți mai multe, este, de asemenea, posibil să scrieți programe malware, viruși, atacatori ai browser-ului și alte dispozitive pentru a le aplica utilizatorilor. Acestea variază de la furtul cookie-urilor de browser, parolelor, cardurilor de credit la descărcarea virușilor pe calculatorul dvs..

## Cum functioneaza JavaScript?

Când browserul web încarcă o pagină web, parserul HTML începe să analizeze codul HTML și să creeze DOM. Ori de câte ori parserul întâlnește o directivă CSS sau JavaScript (încărcat în interior sau extern), el este predat parserului CSS sau motorului JavaScript, după cum este necesar. Motorul JavaScript încarcă fișiere JavaScript externe și cod inline, dar nu rulează imediat codul. Se așteaptă ca parsingul HTML și CSS să se finalizeze. Odată ce acest lucru este realizat, JavaScript este executat în ordinea în care au fost găsite pe pagina web: variabilele și funcțiile sunt definite, invocările funcțiilor sunt executate, procesatorii de evenimente sunt declanșați etc. Aceste activități duc la actualizarea DOM prin JavaScript și este redat instantaneu de browser.



## Încărcarea JavaScript într-o pagină web

Cea mai obișnuită metodă de a încărca JavaScript într-o pagină web este de a utiliza **scenariu** Etichetă HTML. În funcție de cerințele dvs., puteți utiliza una dintre următoarele metode.

* Încărcați un fișier javascript extern într-o pagină web după cum urmează:

* Puteți specifica adresa URL completă dacă javascriptul este dintr-un domeniu diferit de pagina web, după cum urmează:

* JavaScript poate fi încorporat direct în codul HTML. Următoarele cauzează paginii web să afișeze o casetă de alertă atunci când este încărcată.

În afară de aceste metode există modalități de încărcare dinamic a codului JavaScript la cerere. De fapt, există cadre întregi dedicate încărcării și executării modulelor JavaScript cu dependențe adecvate rezolvate la momentul executării. Discuțiile despre aceste tehnici trebuie amânate la un articol avansat.

## Unele Fragmente de cod JavaScript JavaScript

Iată câteva exemple simple de cod JavaScript pentru a ilustra cât de ușor este de utilizat pe pagina dvs. web.

* Următoarele selectează toate **îndrăzneţ** elemente din document și stabilește culoarea primei culori în roșu.

var elems = document.getElementsByTagName ('b'); elems [0] .style.color = "roșu";

* Doriți să modificați imaginea într-un **img** etichetă? Următorii asociați un gestionar de evenimente pentru clic evenimentul unui buton.

Schimbați imaginea

* Actualizați conținutul text al unui paragraf (**p**) element? Seteaza **innerHTML** proprietatea elementului după cum se arată:

Salut Lume

La începutul și mijlocul anilor ’90, internetul a devenit accesibil unui public mai larg datorită computerelor personale și a browserelor. În acel moment, în războiul browserelor existau doi actori cheie care își disputau dominația pe internet: Netscape și Microsoft. Navigator, browserul emblematic al Netscape, și Internet Explorer al Microsoft se aflau față în față.

JavaScript a fost creat inițial pentru a fi un limbaj de scripting doar pentru browserul Netscape, dar după lansarea sa, tot mai multe browsere au adăugat suport pentru JavaScript.

Web-ul avea nevoie de o modalitate de a deveni mai dinamic, cu animații, interacțiuni și mici automatizări. Iar JavaScript a fost soluția pentru a face acest lucru. De atunci, limbajul a ajuns să fie cel mai utilizat limbaj de programare. Și există o mulțime de beneficii dacă decizi să-l stăpânești.

Să vedem lista noastră de beneficii! Iată de ce **decizia de a învăța JavaScript este cea corectă**, pentru un viitor asigurat în programare.

## Nu poți ocoli JavaScript pe web

JavaScript este cel mai universal limbaj de programare de pe internet. În prezent, este utilizat de 97,6% din toate site-urile web, conform [datelor statistice](https://w3techs.com/technologies/details/cp-javascript). Așadar, nu este o exagerare când spunem că nu-l poți ignora.

Rezultatele căutării Google care apar pe măsură ce tastezi sunt obținute cu ajutorul JavaScript. Microsoft a construit browserul web Edge cu ajutorul acestuia. Iar Facebook a inventat propriul său mod de a furniza JavaScript pentru a accelera site-ul său (fiecare secțiune a paginii Facebook este o colecție de aplicații JavaScript independente).

World Wide Web este construit cu acest limbaj de programare, așa că, în mod firesc, nu există nicio modalitate de a-l ocoli. Și cu siguranță nu ar trebui.

## 2. Este o abilitate foarte solicitată pe piața muncii

Dacă consulți [sondajul mondial din 2021 realizat de](https://www.statista.com/statistics/793628/worldwide-developer-survey-most-used-languages/)Statista, vei vedea că JavaScript este oficial cel mai utilizat limbaj de programare din lume. Și este un limbaj omniprezent, apărând în cel mai mare procent de anunțuri de angajare în total. Așadar, deși nu este neapărat cel mai iubit limbaj (toată lumea are preferințele sale), dar este cu siguranță cel mai utilizat. Și, de asemenea, câștigă tot mai multă popularitate și relevanță în multe domenii.

Conform [raportului DevSkiller pentru 2020 al](https://www.infoworld.com/article/3441178/what-is-javascript-the-full-stack-programming-language.html)site-ului de testare a competențelor dezvoltatorilor, JavaScript este cea mai solicitată competență IT în acest moment. Așadar, învățarea acesteia îți poate oferi avantaje uimitoare pe piața muncii și te poate ajuta să-ți asiguri competențele tehnice pentru viitor.

## Poți aduce caracteristici uimitoare pe site-urile web

Cândva, internetul era doar o simplă afișare de texte, fără niciun comportament interactiv. Astăzi, JavaScript aduce interactivitate, capacitate de reacție și, adesea, personalitate site-urilor. Toate browserele acceptă JavaScript, ceea ce permite afișarea frumoasă a videoclipurilor, animațiilor și a altor materiale media. Componenta principală a web-ului sunt HTML și CSS, care marchează conținutul site-urilor, dar este nevoie de JavaScript pentru a le da viață.

Deschide uimitorul [roman grafic online de la SBS.](https://www.sbs.com.au/theboat/)Sau conectează-te online la diferite posturi de radio din lume [cu Radio Garden.](http://radio.garden/visit/zagreb/wRtG4LE8) Nimic din acest tip de conținut web nu s-ar putea realiza fără JavaScript.

Așadar, dacă ți se cere să creezi site-uri web frumoase, receptive, cu o UX (experiență a utilizatorului) uimitoare, pune [JavaScript pe lista de „de învățat”](https://codecool.com/en/blog/future-proof-your-career-with-the-51-most-wanted-programming-languages/)rapid.

## Este mai mult decât un simplu limbaj de programare front-end

Este un [limbaj de scripting](https://en.wikipedia.org/wiki/Scripting_language)(deci scripturile sale trebuie să ruleze într-un browser sau într-un interpretor) și nu este limitat doar la dezvoltarea front-end sau back-end. Atunci când este combinat cu interpretori precum Node.js, [versatilitatea sa pare într-adevăr nesfârșită](https://medium.com/@TheAnmolNigam/the-versatility-of-javascript-223f93d546c0). În afară de elemente interactive pentru site-uri web, poți utiliza JavaScript pentru:

* A crea aplicații web și mobile – Vezi câteva dintre aplicațiile de top ale unor [companii de renume mondial,](https://wiredelta.com/10-most-popular-nodejs-apps-of-2021/)precum Netflix, Linkedin, Paypal sau Uber – [toate au fost create cu ajutorul interpretoruluiNode.js,](https://wiredelta.com/10-most-popular-nodejs-apps-of-2021/)care este un mediu de execuție pentru codul JavaScript. Dezvoltatorii pot folosi interpretorii și diverse cadre pentru a dezvolta și construi aplicații web și mobile, nu doar pagini web.
* A construi servere web și aplicații de tip server-side – Dincolo de site-urile web și de aplicații, dezvoltatorii pot folosi JavaScript pentru a construi servere web simple și pentru a dezvolta infrastructura back-end folosind Node.js.

Avem, mai jos, o listă detaliată ce oferă răspuns la întrebarea – *Ce poți face cu JavaScript?*

## Este ușor de învățat.

JavaScript este un limbaj de scripting multi-paradigmă, astfel încât acceptă și stiluri de programare orientate pe obiecte, imperative și funcționale. Am spune că este, cu siguranță, unul dintre limbaje de programare de învățat pentru începători. De asemenea, dacă stăpânești JavaScript, vei putea face o asemenea varietate de lucruri cu el încât nu vei regreta efortul depus.

Este foarte bine să îl înveți printre primele tale limbaje, deoarece poți realiza multe cu el într-un timp scurt. S-ar putea să poți învăța Python un pic mai repede datorită limbajului său simplist. Dar nu vei putea face atât dezvoltare front-end, cât și back-end cu Python – poți face lucruri mai spectaculoase cu JavaScript mult mai devreme.

Nu ai nevoie de prea multe cunoștințe tehnice pentru a face codul JavaScript să funcționeze. Îl poți rula pur și simplu cu un browser. Cu toate acestea, dacă vrei să mergi mai departe decât codul, vor exista lucruri mai complexe pe care trebuie să le înțelegi. Iar un plan de studiu bine structurat, plus niște îndrumări îți vor fi de mare folos.

## De ce este limbajul JavaScript atât de popular?

**Programarea JavaScript**vine cu o mulțime de avantaje apreciate de profesioniștii care îl folosesc zi de zi. Iată câteva caracteristici care stau la baza popularității JavaScript:

### Accesibilitate

JavaScript există de mult timp, iar acest aspect vine cu avantaje precum cărți, tutoriale, instrumente de asistență și alte beneficii ale comunității. În plus, natura sa dinamică le permite programatorilor să înceapă dezvoltarea proiectelor cât mai repede posibil.

### Tastare dinamică

JavaScript este un limbaj dinamic în aproape toate aspectele sale: variabilele sunt dinamice (atât ca tip, cât și ca existență), chiar și codul este dinamic. Se pot crea variabile noi în timpul execuției și tot atunci se determină tipul de variabile, iar noi funcții pot fi adăugate oricând.

### Tehnologii moderne

Framework-uri moderne precum React, Angular și Vue.js s-au stabilizat și sunt optimizate pentru o performanță mai bună. Aceste tehnologii sunt prietenoase cu programatorii care le accesează atunci când dezvoltă site-uri web sau aplicații.

### Omni-platform

Limbajul JavaScript este apreciat și capacitatea sa de a executa funcțiile programate pe orice tip de platformă. Fie că vorbim de dispozitive precum telefoane mobile, tablete și laptopuri, sau servere, JavaScript un limbaj universal valabil.

### Fără compilare

Limbajul JavaScript poate fi executat imediat ce este scris, fără a fi nevoie de compilare. Acest lucru permite o dezvoltare mult mai rapidă a prototipurilor, un avantaj apreciat de programatori.

### Interoperabilitate

Dacă ai dezvoltat un întreg website într-un alt limbaj de programare, de obicei nu este nicio problemă să introduci pur și simplu conținut suplimentar în JavaScript.

## La ce poți folosi programarea JavaScript?

Deși este cel mai adesea asociat cu browserele web,**limbajul Javascript are multe alte aplicații în lumea tehnologiei.**

Iată **ce poți face cu JavaScript**în lista ta de abilități:

### 1. Web design

JavaScript tinde să fie utilizat ori de câte ori este implicat un browser web. Dar este de reținut faptul că este rareori folosit izolat, se găsește mai des alături de HTML și CSS. Cele trei limbaje de programare se completează bine și, prin urmare, pentru cele mai bune rezultate, un programator trebuie să le cunoască pe toate.

**La cursul nostru de Full-Stack Developer** poți să înveți JavaScript, HTML, CSS și alte limbaje și tehnologii care fac parte din formarea unui bun programator.

### 2. Aplicații mobile

Cu ajutor din partea framework-urilor moderne, dezvoltatorii JavaScript pot crea produse atât pentru iOS, cât și pentru Android, folosind același set de abilități de bază. Crearea de jocuri Android este, de obicei, un loc excelent pentru dezvoltatorii care se familiarizează cu programarea.

### 3. Jocuri

Desigur, poți folosi JavaScript pentru a crea jocuri! Dacă ai o idee grozavă pentru un nou joc, Javascript îți va permite să îl creezi fie în 2D, fie în 3D. Motoare de jocuri precum Pixi.js și Physics JS, de exemplu, pot fi folosite pentru a construi un joc video frumos.

### 4. Realitate virtuală

VR este un domeniu relativ nou și interesant, cu limite care abia încep să fie explorate, iar programarea JavaScript deschide calea. Realitatea virtuală este folosită nu doar în jocuri video, profesioniști din alte domenii, de la chirurgi la designeri de interior apelează la VR în jobul lor.

### 5. Inteligența artificială

În prezent, majoritatea aplicațiilor de inteligență artificială folosesc R sau Python, dar JavaScript are un viitor mare ca limbaj de inteligență artificială. Sunt în curs de apariție mai multe cadre JavaScript AI care fac posibilă executarea de sarcini AI în browser. Biblioteca [Tensorflow](https://www.tensorflow.org/)Javascript este, de asemenea, disponibilă pentru sarcini de învățare automată.

### 6. Hărți interactive

JavaScript este perfect și atunci când vine vorba de crearea de hărți interactive. Acestea pot fi mărite și suprapuse cu informații despre locații existente, dar limbajul poate fi folosit chiar și pentru a crea hărți ale locațiilor care nu există în lumea reală, doar în cea virtuală.

### 7. Sisteme de rezervări

Programarea Javascript este folosită și în cazul sistemelor de rezervări care se aplică în cazul restaurantelor, cinematografelor, teatrelor, parcurilor tematice și aproape tuturor atracțiilor.

### 8. Animații

JavaScript este folosit și atunci când este nevoie de elemente vizuale, sau animații care să ofere utilizatorului o experiență inedită. Cele realizate cu Javascript sunt interactive si pot fi ajustate pentru a se potrivi dispozitivului de afișare.

## Ce opțiuni ai pentru a învăța JavaScript?

Începătorilor le scapă deseori un aspect important atunci când se decid să învețe JavaScript: acest limbaj de programare va aduce o mulțime de întrebări, iar existența unei structuri și a unei îndrumări suplimentare în călătoria de învățare va fi de neprețuit.

Când vrei să treci dincolo de simpla scriere a codului, va trebui să te familiarizezi cu soluțiile server-side, cu gestionarea bazelor de date, cu dezvoltarea front-end și cu UX (experiența utilizatorului). JavaScript va impune o „viziune de dezvoltare full-stack” pentru oricine îl abordează în profunzime. Și toate acestea pot fi greu de înțeles fără îndrumarea unui specialist cu experiență și fără o viziune imersivă și atotcuprinzătoare asupra dezvoltării.

### Pe cont propriu: cărți și tutoriale JavaScript

Dacă este prima ta întâlnire cu programarea, studiul individual este o bună opțiune pentru a te familiariza cu noțiunile de bază. Și o simplă căutare pe Google îți va genera suficiente resurse pentru a începe, provocarea fiind să le alegi pe cele potrivite pentru tine. Când analizezi **tutorialele JavaScript**, poți să ai în vedere criterii precum aprecierile raportate la numărul total de vizionari, dar și comentariile lăsate de alte persoane care au accesat materialul.

Pentru cei ce preferă **învățarea tradițională, o carte JavaScript** și câteva ore dedicate zilnic pot să dea rezultate în faza de început și să ajungi să scrii linii simple de cod. Iată **câteva cărți JavaScript**numai bune de parcurs:

* **A Smarter Way to Learn JavaScript, autor Mark Myers**

Această carte este scrisă pentru începătorii în JavaScript, dar este foarte utilă si pentru dezvoltatorii experimentați care doresc să-și perfecționeze abilitățile. Cuprinde elementele fundamentale ale limbajului și câteva concepte avansate, cum ar fi constructori și prototipuri. În plus, conține o serie de exerciții gratuite, online, interactive, la fiecare capitol.

* **Head First JavaScript Programming: A Brain-Friendly Guide, autor Elisabeth Robson**

Cartea este concepută în baza celor maii recente cercetări în știința cognitivă și teoria învățării pentru a oferi o experiență de învățare multi-senzorială. Cuprinde totul despre elementele fundamentale ale JavaScript, la subiecte avansate prin jocuri interactive și puzzle-uri. Pentru oricine preferă învățarea vizuală decât formatele grele de text, această carte este răspunsul.

* **JavaScript: The Definitive Guide, autor David Flanagan**

Poți opta pentru cartea JavaScript folosită de programatori de aproape 25 de ani. Cea mai recentă ediție este actualizată pentru a acoperi versiunea 2020 de Javascript împreună cu câteva capitole nou adăugate pe o serie de subiecte. Începe cu elementele de bază ale limbajului și acoperă până la subiecte avansate precum platformele web și Node.js.

**Deși studiul pe cont propriu** funcționează foarte bine pentru persoanele autodidacte, chiar și acestea pot ajunge în momente de blocaj și frustrare, pe măsură ce avansează către concepte mai complexe.

**Ce este jQuery?**

**Este cea mai utilizata si stabila librarie JavaScript** cu ajutorul careia putem adauga **efecte si comportamente**elementelor ce alcatuiesc**DOM-ul** (Document Object Model) paginii web si putem crea cereri Ajax.

**Avantaje**

– Sloganul jQuery este „**Write less, do more**” deci **scrierea** este cu mult **comprimata** in comparatie cu utilizarea limbajului de programare JavaScript pur;  
– Avand in vedere succesul de care se bucura, succes verificat prin gradul difuziunii sale, libraria jQuery este una dintre cele mai active in ceea ce priveste producerea de cod nou si de plugin-uri, bucurandu-se de o comunitate de dezvoltatori in continua crestere;  
– Multi colosi din industria IT utilizeaza jQuery cum sunt: IBM, Netflix, Nokia, Wikipedia, Google si Microsoft;  
– Libraria jQuery are dimensiuni reduse de circa 96 KB pentru intreaga librarie in varianta comprimata.

**Download jQuery**

Pentru utilizarea jQuery avem doua optiuni: putem sa descarcam release-ul cel mai recent pe calculator si sa il includem fizic in proiectul nostru sau putem folosi libraria prin linkarea la un **server CDN** (Content Ditribution Network).

**Descarcarea librariei pe calculator**

Pentru a folosi fisierul descarcat fizic pe calculatorul nostru vom merge pe [site-ul jQuery](https://jquery.com/). Facand un click pe butonul de DOWNLOAD vom deschide pagina unde avem mai multe optiuni pentru descarcarea librariei.

In acest moment release-ul cel mai recent pentru versiunea 3x este 3.5.1 pe care il putem descarca in varianta comprimata (production) sau necomprimata (development), vom vedea mai jos care sunt diferentele dintre cele doua. Se mai poate folosi si versiunea slim, dar care exlude modulele ajax si efectele de animatie.

**Includerea prin CDN**

Aceasta optiune ne da posibilitatea de a utiliza libraria prin reteaua **CDN – Content Delivery Network** pusa la dispozitie atat de jQuery, cat si de Google sau Microsoft.

**Google** gazduieste pe reteaua sa numeroase librarii open source, printre care mai multe librarii JavaScript deci si jQuery. **Avantajul utilizarii librariei jQuery prin CDN-ul Google** este ca astfel vom fi avantajati de viteza si de performanta instrastructurii Google care ne da posibilitatea de a utiliza versiunea cea mai noua. Un alt avantaj este faptul ca multi vizitatori au descarcat deja libraria de la CDN-ul Google prin vizualizarea altor site-uri web si deci viteza de incarcare a site-ului nostru va fi astfel favorizata.

Pentru versiunile cele mai recente link-urile la CDN sunt:

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.4/jquery.min.js"></script>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>

Pentru a vedea care este link-ul la versiunea cea mai recenta vom merge pe [site-ul Google Developers.](https://developers.google.com/speed/libraries/devguide#jquery)

Bineinteles ca putem linka si catre CDN-ul jQuery:

https://code.jquery.com/jquery-3.5.1.min.js

https://code.jquery.com/jquery-1.11.0.min.js

**Varianta comprima sau cea necomprimata?**

**Pentru producerea codului este preferabila utilizarea versiuni comprimate (min)** in care au fost eliminate spatiile si randurile si in care numele variabilelor a fost prescurtat. Utilizand varianta comprimata vom obtine un fisier mai usor decat originalul pentru a reduce astfel lungimea de banda si **pentru a spori viteza de incarcare a paginii**.

Daca insa folosim pentru primele dati jQuery si ne intereseaza sa il studiem pentru a putea intelege mai bine sintaxa atunci avem nevoie de versiunea necomprimata in care codul este inteligibil.

**Unde plasam libraria in documentul nostru HTML?**

Link-urile la libraria jQuery in documentul nostru HTML vor fi plasate ca buna regula **inainte de inchiderea tag-ului body** al paginii, fie ca vorbim despre libraria jQuery descarcata pe calculator sau de un link la unul dintre CDN-uri.

Link-ul in cazul salvarii librariei pe calculator va fi asemanator cu acesta, unde fisierul este salvat in cadrul site-ului nostru in folder-ul cu numele JS iar versiunea descarcata este cea comprimata:

<script type=”text/javascript” src=”JS/jquery-1.11.1.min.js”></script>

Si link-ul catre unul dintre variile CDN-uri disponibile va fi plasat de regula tot inainte de inchiderea tag-ului body al documentului.

<body>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>

</body>

</html>

Daca insa vrem sa pozitionam aceste link-uri intr-o alta parte a documentului o putem face in interiorul tag-ului head ca in exemplul de mai jos, numai ca in acest caz libraria va fi incarcata deodata cu celelalte elemente ale paginii, si nu dupa ca in cazul in care pozitionarea se face la sfarsitul paginii:

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>

</head>

## Definiție - Ce înseamnă jQuery?

jQuery este o bibliotecă JavaScript rapidă, concisă și rapidă, care poate fi utilizată pentru a simplifica gestionarea evenimentelor, traversarea documentelor HTML, interacțiunile Ajax și animația pentru dezvoltarea rapidă a site-ului. jQuery simplifică scripturile HTML din partea clientului, simplificând astfel dezvoltarea aplicațiilor Web 2.0.

jQuery este o bibliotecă gratuită, open-source și cu licență dublă, sub licența publică generală GNU. Este considerată una dintre bibliotecile JavaScript (JS) preferate disponibile astăzi. Începând cu 2012, acesta este utilizat de mai mult de jumătate dintre site-urile de top ale Web-ului.

## Techopedia explică jQuery

jQuery a fost introdus în ianuarie 2006 de John Resig la BarCamp NYC.

jQuery oferă, de asemenea, funcționalitate care permite dezvoltatorilor să creeze pluginuri, pe lângă biblioteca JavaScript. Aceasta permite dezvoltarea abstractizărilor pentru animație și interacțiune la nivel scăzut, efecte sofisticate și widget-uri tematice, la nivel înalt. Mecanismul modular al bibliotecii jQuery facilitează dezvoltarea de aplicații web și pagini Web puternice, extrem de eficiente.

Biblioteca jQuery oferă mai multe strategii și funcții prietenoase pentru dezvoltarea unei aplicații bogate. Deoarece funcțiile jQuery sunt simple, este foarte popular printre dezvoltatori. jQuery poate fi utilizat în toate aplicațiile bazate pe Web, în ​​ciuda tehnologiei. Poate fi utilizat cu ASP, PHP, JSP, CGI, Servlets și majoritatea limbajelor de programare Web.

Sau, in loc de a avea fisierul cu jQuery pe serverul dv., o metoda alternativa de a include libraria jQuery este prin Google:

<script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1.6.1/jquery.min.js"></script>

Acest tutorial foloseste versiunea lansat in mai 2011 (*jQuery 1.6.1*).  
  
Pentru a putea interactiona cu elementele HTML in pagina, instructiunile script-ului trebuie executate dupa incarcarea paginii, astfel tot codul jQuery se scrie in interiorul unei functii speciale "*document ready*", aceasta executa codul din ea dupa incarcarea paginii.  
  Sintaxa:

**<script src="jquery\_file.js"></script>**

**$(document).ready(function() {**

**// tot codul jQuery se scrie aici**

**});**

- Aproape toate comenzile si functiile jQuery se adauga in acest "*document-ready*". In interiorul functiei "*ready()*" puteti adauga orice cod JavaScript, precum variabile noi, obiecte, instructiuni if(), for(), etc.  
  
In urmatorul exemplu creem o simpla fereastra alert de test, ca sa ne asiguram ca am inclus bine libraria jQuery:

<!doctype html>

<html>

<head>

<meta charset="utf-8" />

<title>Simplu jQuery</title>

**<script type="text/javascript" src="jquery\_library.js"></script>**

<script type="text/javascript"><!--

**$(document).ready(function() {**

**alert('Mesaj de test');**

**});**

--></script>

</head>

<body>

Continut HTML

</body>

</html>

Se poate inlocui **$()** cu **jQuery()** ; acesta e util ca sa se evite conflicte cu alte aplicatii JavaScript care folosesc "*$()*". Exemplu:

<script type="text/javascript"><!--

**jQuery(document).ready(function() {**

**alert('Mesaj de test');**

**});**

--></script>

### Selectare elemente HTML in jQuery

Elementele (X)HTML se selecteaza cu jQuery prin adaugarea lor in **$("")** (sau ***jQuery("")*** , se pot folosi ghilimele duble sau simple).  
De exemplu, daca vreti sa selectati toate elementele <h3>, se foloseste:   **$('h3')** (sau  ***jQuery('h3')*** ).  
Daca vreti sa selectati tag-ul cu *id="unid"* , se scrie   **$('#unid')**.  
  
Urmatorul exemplu afiseaza o fereastra Alert cu continutul unui DIV cu *id="div1"*:

<!doctype html>

<html>

<head>

<meta charset="utf-8" />

<title>Simplu select</title>

<script type="text/javascript"

src="jquery\_library.js"></script>

<script type="text/javascript"><!--

$(document).ready(function() {

var ctn = $('#div1').html(); // preia continutul din tag-ul cu id="div1"

alert(ctn);

});

--></script>

</head>

<body>

<div id="div1">Curs jQuery</div>

<div id="div2">marplo.net/javascript/curs-jquery-tutoriale-js</div>

</body>

</html>

Cand adaugati un selector jQuery in $(), returneaza un obiect jQuery ce contine un set cu elementele selectate, potrivite cu acel selector.  
Functia **html()** este o metoda jQuery care returneaza continutul HTML din elementul la care e aplicata.  
Selectorii jQuery sunt foarte asemanatori cu selectorii CSS, iata cateva example:  
  
  **$('\*')**   - selecteaza toate elementele.  
  **$('div')**   - selecteaza toate tag-urile <div>.  
  **$('#un\_id')**   - selecteaza un tag HTML cu id="un\_id".  
  **$('.a\_class')**   - selecteaza toate tag-urile HTML cu class="a\_class".  
  **$('p#un\_id')**   - selecteaza tagul <p> cu id="un\_id".  
  **$('li.a\_class')**   - toate tag-urile <li> cu class="a\_class".  
  **$('li a')**   - toate tag-urile <a> din elementele <li>.  
  **$('div a.a\_class')**   - tag-urile <a> cu class="a\_class", care sunt adaugate in DIV-uri.  
  **$('div.a\_class p span')**   - toate tag-urile <span> din <p>-uri care sunt adaugate in <div>-uri cu class="a\_class".  
  
jQuery suporta folosirea tuturor selectorilor CSS, inclusiv cei din CSS3. Iata cateva example:  
  **$('p>a')**   - selecteaza toate tag-urile <a> care sunt incluse direct in paragrafe (<p>).  
  **$('a:first')**   - selecteaza primul tag <a>.  
  **$('h3:last')**   - selecteaza ultimul <h3> din pagina.  
  **$('input[type=text]')**   - selecteaza elementele input care au tipul (type) specificat la text.  
  **$('p:odd')**   - selecteaza toate paragrafele cu numar de ordine impar.  
  **$('li:first-child')**   - selecteaza primul <li> din fiecare lista cu tag-uri <li>.  
  
jQuery are si cativa selectori proprii:  
  **$(':button')**   - selecteaza elementele de tip buton (input sau button).  
  **$(':radio')**   - selecteaza butoanele tip radio.  
  **$(':checkbox')**   - selecteaza checkbox.  
  **$(':checked')**   - selecteaza elementele checkbox sau radio care sunt selectate.  
  **$(':header')**   - selecteaza elementele de tip Header (h1, h2, h3, etc.).  
  **$(':contains("String")')**   - selecteaza elementele care contin textul specificat la "String".  
  
• jQuery poate selecta mai multi selectori intr-o singura declaratie, separati prin virgula, intr-un singur sir.  
De exemplu, daca vrem sa selectam fiecare <h2>, <div> cu *class="clas1"*, si <p> cu *class="cls2"* intr-o singura expresie, folosim urmatoarea instructiune:  
                  ***$('h2, div.clas1, p.cls2')***  
  
- Lista completa cu selectorii folositi in jQuery o gasiti pe site-ul oficial, la pagina [Selectori](http://api.jquery.com/category/selectors/).  
  
• Pentru a obtine numarul de elemente selectate se foloseste proprietatea **length**.  
Sintaxa:

**$('selector').length**

Exemplu:

<script type="text/javascript"><!--

$(document).ready(function() {

var div\_nr = $('div').length; // obtine numarul de DIV-uri din pagina

alert(div\_nr);

});

--></script>

<div id="div1">Curs jQuery</div>

<div id="div2">marplo.net/javascript/curs-jquery-tutoriale-js</div>

### Evenimente in jQuery

Evenimentele sunt actiuni pe care utilizatorul le efectueaza in pagina web, cum ar fi click, miscarea mouse-ului, deschiderea unei ferestre de browser, derularea paginii, apasarea unui buton la tastatura, etc..  
Se poate specifica executarea unor instructiuni cand un eveniment este declansat, folosind urmatoarea sintaxa:

**$('selector').tip\_eveniment(function() {**

**// instructiuni care sa fie executate**

**// cand actiunea specificata la "tip\_eveniment"**

**// este efectuata la elementele reprezentate de "selector"**

**});**

Exemplu:

<script type="text/javascript"><!--

$(document).ready(function() {

// inregistreaza un eveniment de tip click la tag-urile <div>

$('div').click(function() {

var div\_txt = $(this).text(); // preia textul din DIV-ul pe care s-a apasat click

alert(div\_txt);

});

});

--></script>

<div id="div1">Curs jQuery</div>

<div id="div2">marplo.net/javascript/curs-jquery-tutoriale-js</div>

- Instructiunea ***$(this)*** returneaza elementul curent, "*acesta*" (la care s-a declansat evenimentul).  
- Metoda ***text()*** returneaza textul din obiectul la care e aplicata.  
- Codul din interiorul **function()** va fi executat doar cand se apasa click pe un <div>.  
Demo (*apasati click pe urmatoarele linii de text*)

Curs jQuery

marplo.net/javascript/curs-jquery-tutoriale-js

• Iata alte cateva tipuri de evenimente ce pot fi utilizate in jQuery:  
  **blur**   - e declansat cand un element pierde "atentia" (cursorul nu mai e pe el).  
  **focus**   - e declansat cand se intra intr-un element.  
  **hover**   - executa instructiunile cand mouse-ul este in cadrul obiectului.  
  **mouseover**   - declansat cand mouse-ul intra in cadrul obiectului.  
  **mousemove**   - declansat cand mouse-ul se misca in cadrul elementului.  
  **keydown**   - actionat cand utilizatorul apasa un buton pe tastatura.  
  **load**   - actionat cand un element si toate sub-elementele lui au fost incarcate.  
  **resize**   - trimis la obiectul window cand dimensiunea ferestrei browser-ului e modificata.  
  **scroll**   - declansat cand pagina (sau un anume obiect) e derulata.  
  **submit**   - actionat cand utilizatorul incearca sa trimita datele dintr-un formular.  
  **select**   - declansat cand utilizatorul selecteaza un text adaugat intr-o caseta <input type="text">, sau <textarea>.  
  
- O lista completa cu evenimentele folosite in jQuery, gasiti pe site-ul oficial, la pagina [Evenimente jQuery](http://api.jquery.com/category/events/).

## Citire proprietati CSS

Proprietatile CSS ale unui element HTML se pot citi /prelua cu urmatoarea sintaxa:

**$('selector').css('proprietate\_css');**

- *selector* - reprezinta un element HTML.  
- *proprietate\_css* - proprietatea CSS ce va fi citita.  
  
Codul din urmatorul exemplu preia valorile proprietatilor "width" si "background-color" ale primului <div> cu *class="cls"*, apoi afiseaza aceste valori intr-o fereastra Alert, cand butonul cu *id="btn"* e apasat:

<!doctype html>

<html>

<head>

<meta charset="utf-8" />

<title>Citire stil CSS</title>

<style type="text/css"><!--

.cls {

width:250px;

background-color:#bcefcd;

border:2px solid blue;

}

--></style>

<script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1.6.1/jquery.min.js"></script>

<script type="text/javascript"><!--

$(document).ready(function() {

// preia width si background-color ale primului <div> cu class="cls"

var div1\_width = $('div.cls:first').css('width');

var div1\_bgcolor = $('div.cls:first').css('background-color');

// afiseaza un Alert cand tag-ul HTML cu id="btn" e apasat

$('#btn').click(function(){

alert('width = '+ div1\_width+ ' , bgcolor = '+ div1\_bgcolor);

});

});

--></script>

</head>

<body>

<div class="cls">Curs jQuery</div>

<div class="cls">marplo.net/javascript/curs-jquery-tutoriale-js</div>

<button id="btn">Click</button>

</body>

</html>

Demo:

Curs jQuery

marplo.net/javascript/curs-jquery-tutoriale-js

Click

jQuery obtine valoarea poprietatii CSS data de browser, nu cea adaugata in definitia CSS. Astfel, daca aveti un DIV cu inaltimea de 300 pixeli setata in CSS, dar continutul din el determina o inaltime mai mare de 300, jQuery va furniza inaltimea actuala a acelui DIV, nu cea de 300 specificata in CSS.

### Setare proprietati CSS

Pentru a seta o proprietate CSS la un element HTML cu jQuery, se foloseste urmatoarea sintaxa:

**$('selector').css('proprietate\_css', 'valoare');**

Exemplu:

<script type="text/javascript"><!--

$(document).ready(function() {

// seteaza font-size la toate tag-urile <div> cu class="cls", cand tag-ul cu id="btn" e apasat

$('#btn').click(function(){

$('div.cls').css('font-size', '20px');

});

});

--></script>

<div class="cls">Curs jQuery</div>

<div class="cls">marplo.net/javascript/curs-jquery-tutoriale-js</div>

<button id="btn">Click</button>

Codul de sus seteaza proprietatea CSS *font-size:20px;* la toate tag-urile <div> cu class="cls", cand tag-ul cu id="btn" e apasat.  
Demo:

Curs jQuery

marplo.net/javascript/curs-jquery-tutoriale-js

Click

Daca vreti sa setati mai multe proprietati CSS intr-o singura expresie, puteti folosi urmatoarea sintaxa:

**$('selector').css({ 'prop1': 'valoare', 'prop2': 'valoare' });**

- Puteti adauga oricate perechi *'prop':'valoare'*, separate prin virgula intre acolade.  
  
Exemplu:

<script type="text/javascript"><!--

$(document).ready(function() {

// setare mai multe proprietati CSS la <div> cu class="cls", cand e apasat

$('div.cls').click(function(){

$(this).css({'margin': '3px auto', 'background': '#bcfecd', 'color': 'blue'});

});

});

--></script>

<div class="cls">Curs jQuery</div>

<div class="cls">marplo.net/javascript/curs-jquery-tutoriale-js</div>

Codul de sus seteaza: *margin:2px auto; background':#bcfecd; color:blue;* la oricare <div> cu class="cls", cand se apasa click pe el.  
Ca sa vedeti efectul, apasati click pe urmatoarele doua linii de text:

Curs jQuery

marplo.net/javascript/curs-jquery-tutoriale-js

### Adaugare si stergere clase

jQuery permite cu usurinta adaugarea, stergerea si comutarea claselor CSS. Iata cateva sintaxe prin care se realizeaza aceste actiuni:  
  
    **$('selector').addClass('nume\_clasa');**   - adauga class "nume\_clasa" la elementele HTML reprezentate de "selector".  
    **$('selector').addClass('nume\_clasa1 nume\_clasa2');**   - adauga class "nume\_clasa1" si "nume\_clasa2" la elementele HTML reprezentate de "selector".  
    **$('selector').removeClass('nume\_clasa');**   - sterge clasa "nume\_clasa" din elementele HTML reprezentate de "selector".  
    **$('selector').toggleClass('nume\_clasa');**   - comuta class "nume\_clasa" (o adauga daca nu e setata, si o sterge daca e adaugata).  
  
Exemplu:

<!doctype html>

<html>

<head>

<meta charset="utf-8" />

<title>Adaugare, Stergere clase CSS</title>

<style type="text/css"><!--

.cls1 { border:2px solid #01da02; color:blue; }

.cls2 { background:#bcfecd; font-size:18px; }

--></style>

<script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1.6.1/jquery.min.js"></script>

<script type="text/javascript"><!--

$(document).ready(function() {

// adauga class "cls2" la tag-ul cu id="id1", sterge clasa "cls1" din toate <div>

// cand se apasa click pe tag-ul cu id="btn"

$('#btn').click(function(){

$('#id1').addClass('cls2');

$('div').removeClass('cls1');

});

});

--></script>

</head>

<body>

<div class="cls1">Cursuri pt. Web Development</div>

<div id="id1">Web site: marplo.net</div>

<button id="btn">Click</button>

</body>

</html>

Codul jQuery de mai sus adauga class "cls2" la tag-ul cu id="id1", si sterge clasa "cls1" din toate <div> cand se apasa click pe eticheta HTML cu id="btn".  
Demo:

Cursuri pt. Web Development

Web site: marplo.net

Click

### Verificare daca o clasa e adaugata

Cu jQuery se poate de asemenea verifica daca elementul selectat are o anumita clasa CSS, folosind metoda **hasClass()**.  
Sintaxa:

**if ($(unElement').hasClass('nume\_clasa')) {**

**// efectueaza ceva aici**

**}**

- Daca elementul HTML specificat de "*unElement*" are clasa CSS "*nume\_clasa*", browser-ul va executa codul dintre acolade.  
  
Exemplu:

<script type="text/javascript"><!--

$(document).ready(function() {

// cand mouse-ul e pe un DIV cu class="cls", ii adauga o bordura

$('div').mouseover(function() {

if($(this).hasClass('cls')) {

$(this).css('border', '2px solid blue');

}

});

});

--></script>

<div class="cls">Div cu class="cls"</div>

<div>DIV fara class</div>

<div class="cls">Alt div cu class="cls"</div>



## Informatii generale

* Categorie: Web
* Judetul: Caraș Severin
* Surse: [GitHub - Pcarioca/Fake-News-Identifier2 13](https://github.com/Pcarioca/Fake-News-Identifier2)
* Homepage: [https://identifyfakenews.netlify.app/ 30](https://identifyfakenews.netlify.app/)

## Descriere

Ca și advocat al web developmentului, cred ca toată lumea ar trebui să aiba acces la surse de încredere de informații online. Astfel, “Fake News Identifier” vine in ajutor, conferind utilizatorilor libertatea de a vota site-urile care sunt sau nu sunt de încredere. De asemenea, prima pagina oferă un catalog al știrilor care au fost votate cel mai mult ca fiind de încredere, pe categorii si ușor de navigat.

## Tehnologii

Pentru a crea aplicația web, am folosit in principal trio-ul HTML/CSS/JS. Baza de date este consturită prin mongoDB, iar ca și biblioteci front-end am utilizat Bootstrap și Fontawesome. Mai multe detalii despre resursele folosite, cât si modul de preparare al bazei de date găsiti in secțiunea “About me” a site-ului.

## Cerinte sistem

Singurele cerințe sunt instalarea Node.js pe computer, cât și o conexiune la internet. De menționat ca link-ul reprezentând adresa lucrării ofera acces strict la partea de front-end.

Fake news identifier este aplicația web menită la identificarea Știrilor false.

Link repository Github: <https://github.com/Pcarioca/Fake-News-Identifier2>

Link wesite: [https://identifyfakenews.netlify.app](https://identifyfakenews.netlify.app/) (fără a se folosi baza de date)

Cum pregătim hostul local pentru rulare?

Instalați nodeJS si MongoDB atlas.

Navigați in folderul backend folosind o linie de comanda la alegere (recomandat Powershell)

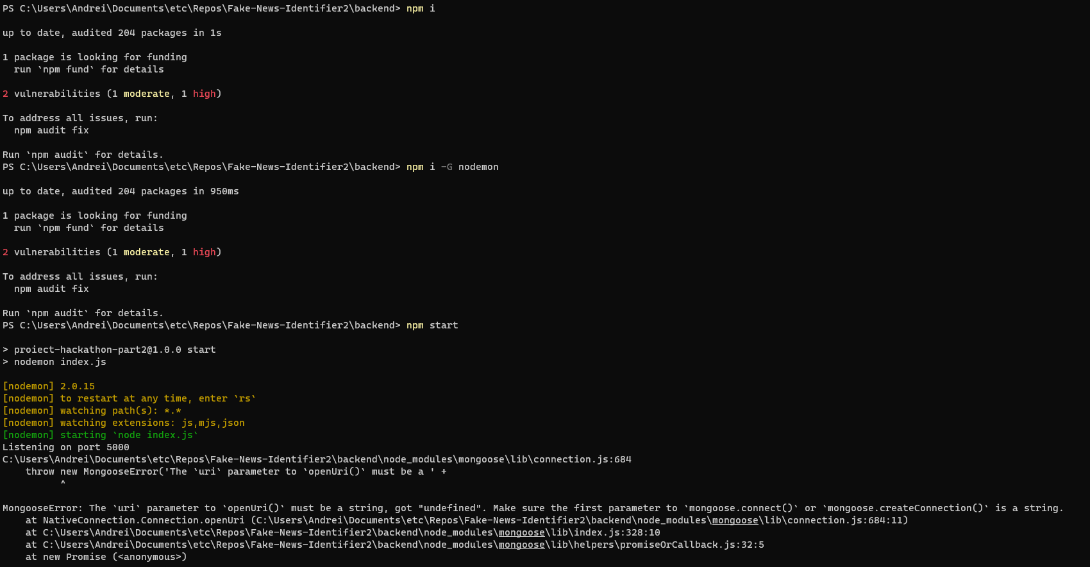
Creați in backend un fișier denumit .env cu următorul text:

DB\_URL= mongodb://localhost:27017/fake

Aici, rulați comenzile in ordine:

1. Npm i
2. Npm i -G nodemon
3. npm start.

Ignorați orice mesaje de avertizare.

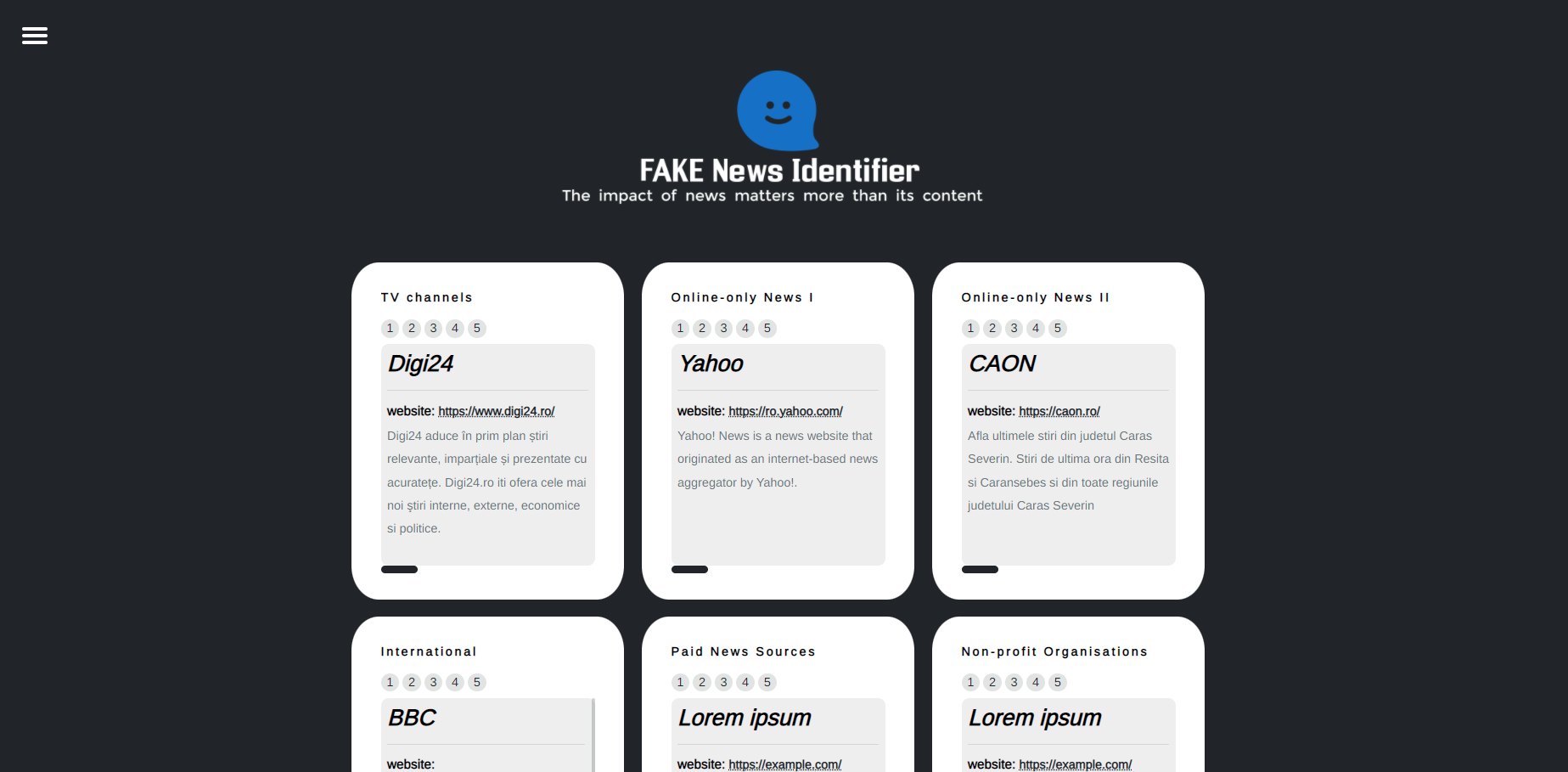


Cum funcționează Fake News Identifier?

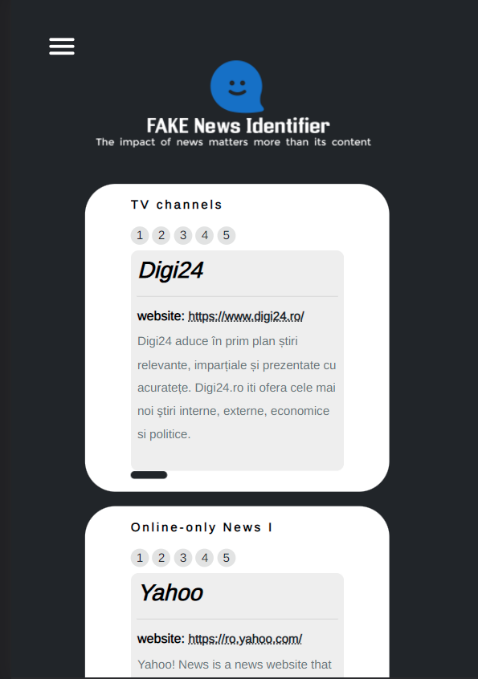
Acesta are multiple funcții.

1. **Lista de surse de încredere**

Prima pagina conține carduri formate din diferite tipuri de știri, unde se găsesc sursele propriu-zise si o scurta descriere:

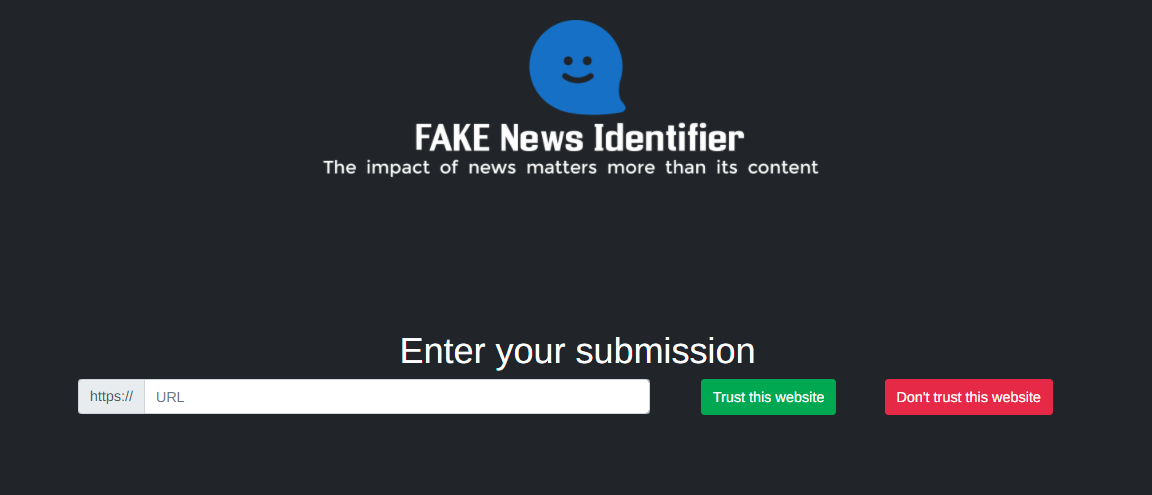
****

De asemenea, precum toate paginile, site-ul este total responsive:

****

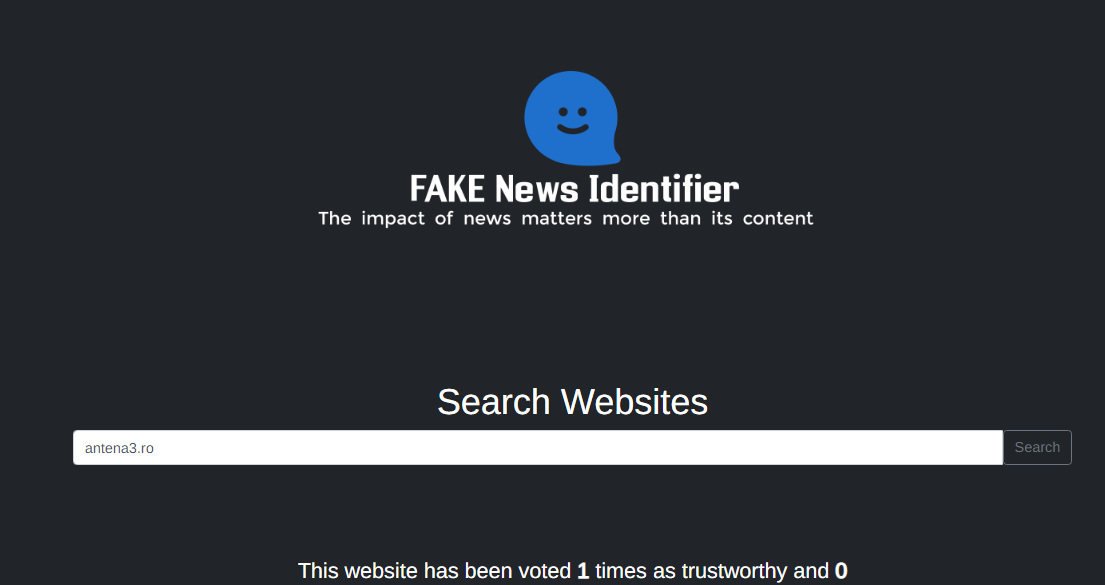
1. **Raportare site-uri ca si surse sigure/nesigure de știri.**

Navigând la tab-ul “Make a submission”, putem găsi câmpul de input.

****

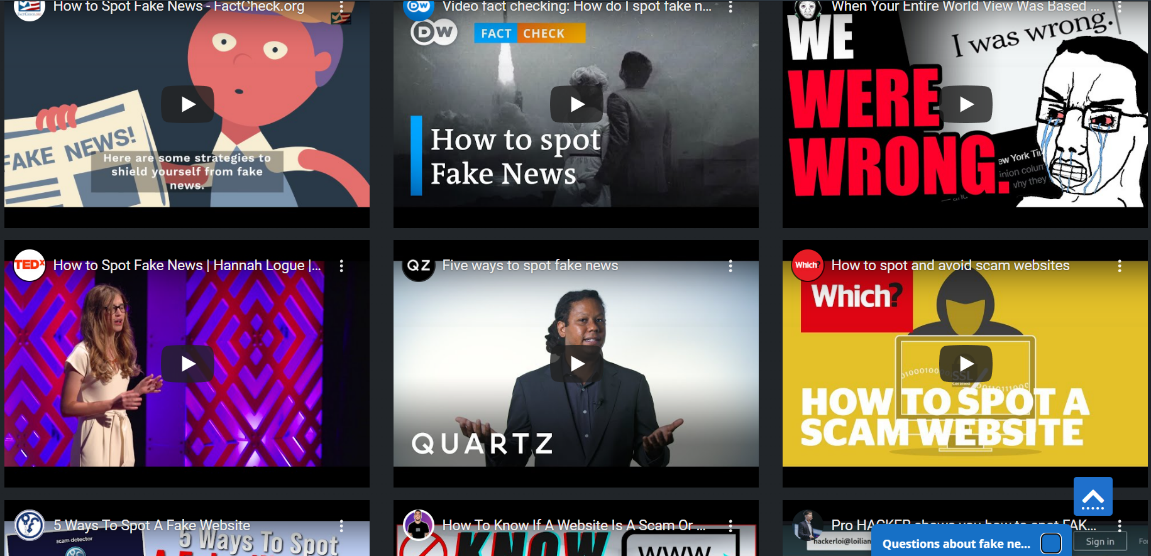
Aplicația ne oferă feedback pentru submission-uri, si nu ne permite sa votam de mai multe ori același website:

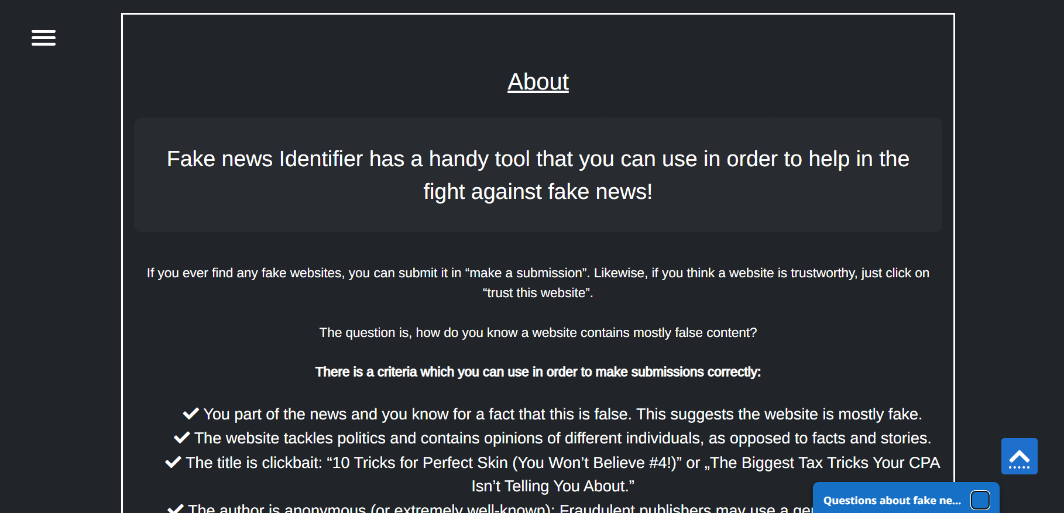
**3.Căutare website-uri**

****Aplicația ne permite sa aflam de cate ori a fost votat website-ul ca si trustworthy sau untrustworthy.

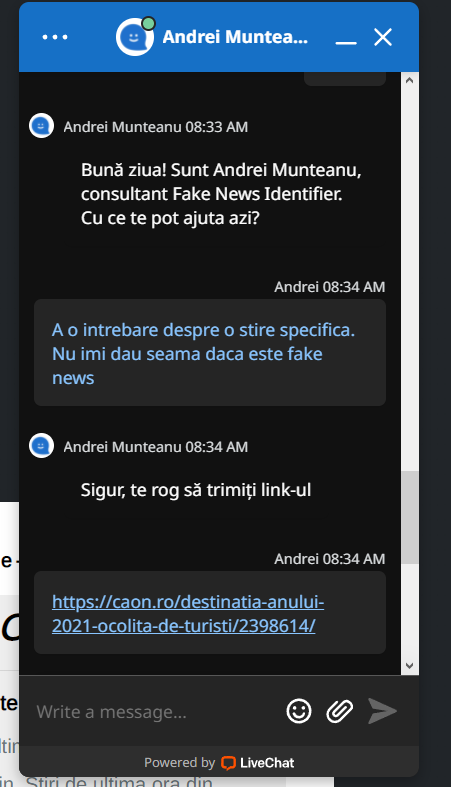
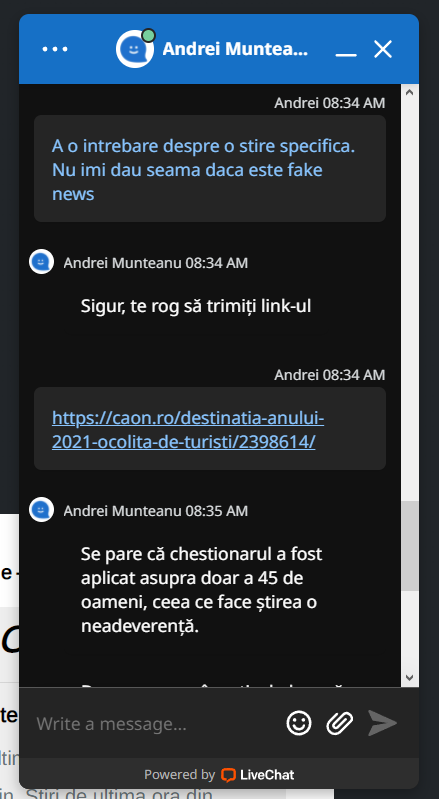
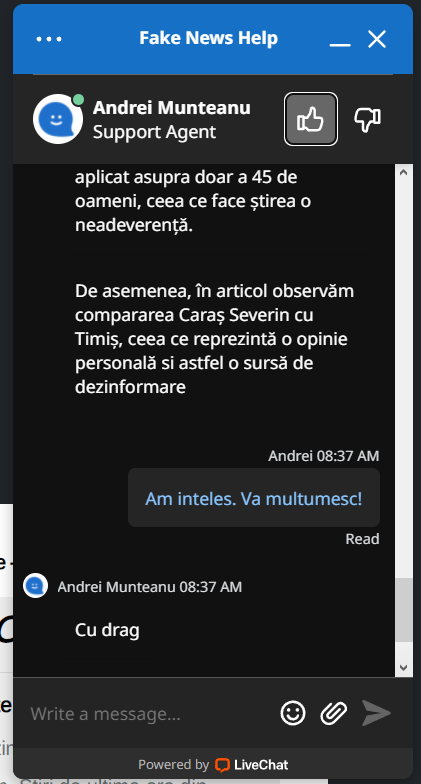
**4.Informatii despre Fake News**

Pe pagina de info, avem detalii despre cum putem sa ne protejam de fake news, si inherent cum notam corect site-urile.



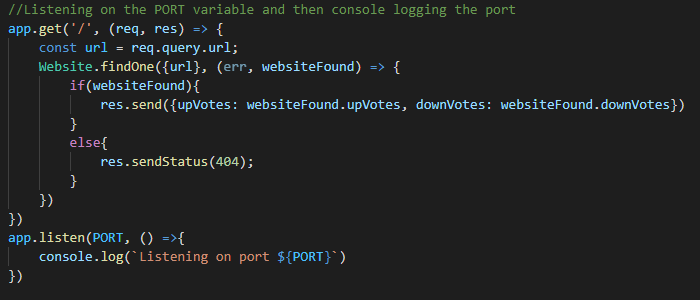


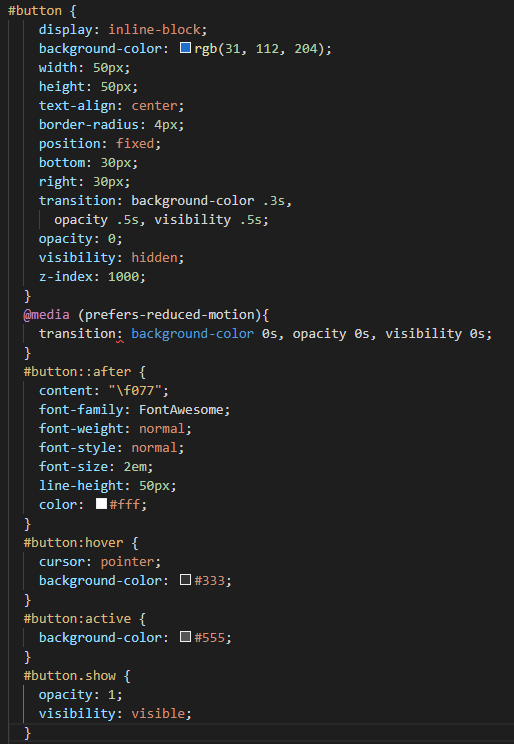
**5. Live chat cu un consilier**

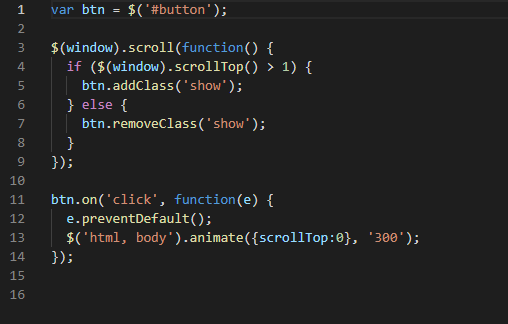
Nesigur despre o anumita stire care se cauzeaza distres? Intra in conversatie cu un consultant apasand pe bula de chat!

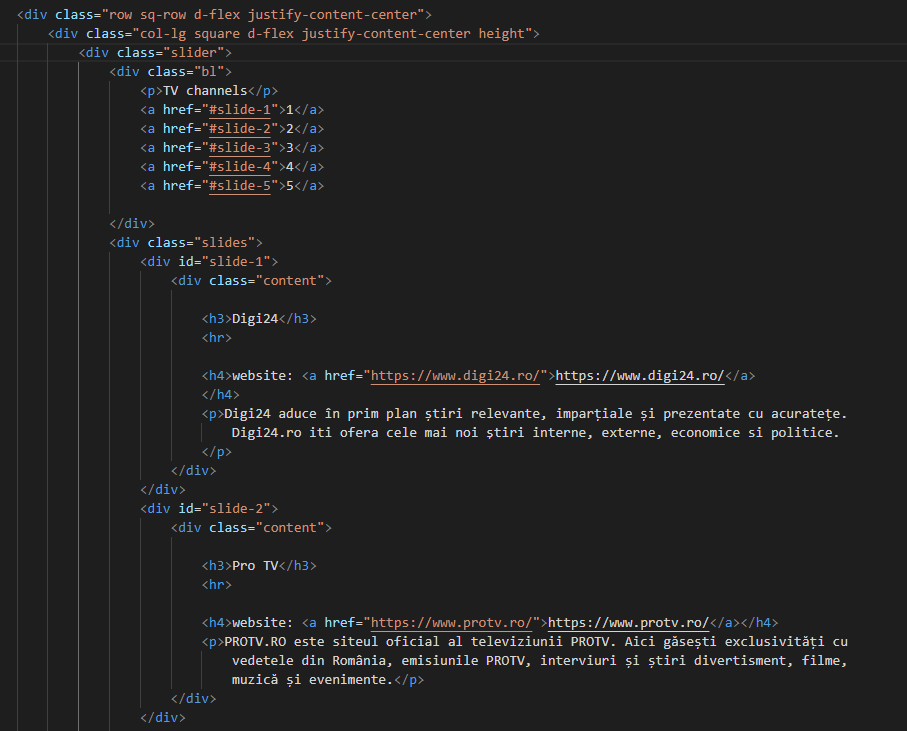
# 

# Pentru a crea aplicația, am utilizat faimosul triplet HTML/CSS/JS, de asemenea Mongo DB pentru baza de date. Iată câteva exemple:









Bibliografie: -Logo: [Namecheap Logo maker](https://www.namecheap.com/logo-maker)  
-Navbar: [Eric Terwan via codepen.io](https://codepen.io/erikterwan/pen/EVzeRP)  
-Footer:[Material Design for bootstrap](https://mdbootstrap.com/docs/standard/navigation/footer/)  
-Resources Carousel: [CSS Tricks Website](https://css-tricks.com/css-only-carousel/)  
-Icons: [Fontawesome.com](https://fontawesome.com/)  
-Go to top button: [Matthew Cain via codepen.io](https://codepen.io/matthewcain/pen/ZepbeR)

<https://www.freecodecamp.org/news/learn-mongodb-a4ce205e7739/>

https://www.w3schools.com/

Vă mulțumesc.

Munteanu Paul Andrei

Colegiul Național “Diaconovici-Tietz”

Resita

## Bibliografie

* [Jeffrey Zeldman](https://ro.wikipedia.org/w/index.php?title=Jeffrey_Zeldman&action=edit&redlink=1) (2009): *Designing With Web Standards*, New Riders, [ISBN 978-0321616951](https://ro.wikipedia.org/wiki/Special:Referin%C8%9Be_%C3%AEn_c%C4%83r%C8%9Bi/9780321616951) (paperback) ([book's companion site](http://www.zeldman.com/dwws/))
* [Dan Cederholm](https://ro.wikipedia.org/w/index.php?title=Dan_Cederholm&action=edit&redlink=1) (2009): *Web Standards Solutions, The Markup and Style Handbook*, Friends of Ed, [ISBN 978-1430219200](https://ro.wikipedia.org/wiki/Special:Referin%C8%9Be_%C3%AEn_c%C4%83r%C8%9Bi/9781430219200) (paperback) ([Author's site](http://www.simplebits.com/publications/solutions/) [Arhivat](https://web.archive.org/web/20060409154734/http:/www.simplebits.com/publications/solutions/) în 9 aprilie 2006, la [Wayback Machine](https://ro.wikipedia.org/wiki/Wayback_Machine).)
* [Cascading Style Sheets](http://people.opera.com/howcome/2006/phd/) *Cascading Style Sheets*, PhD thesis, by Håkon Wium Lie – provides an authoritative historical reference of CSS
* [*Cascading Style Sheets: Separating Content from Presentation*](http://friendsofed.com/book.html?isbn=978-1590592311)[[*nefuncțională*](https://ro.wikipedia.org/wiki/Wikipedia:Leg%C4%83turi_externe)], (co-authored by Owen Briggs, Steven Champeon, Eric Costello, and Matt Patterson), Friends of Ed (2004), [ISBN 978-1590592311](https://ro.wikipedia.org/wiki/Special:Referin%C8%9Be_%C3%AEn_c%C4%83r%C8%9Bi/9781590592311)
* [Keith Schengili-Roberts](https://ro.wikipedia.org/w/index.php?title=Keith_Schengili-Roberts&action=edit&redlink=1) (2003): *Core CSS, 2nd Edition*, Prentice Hall, [ISBN 0-13-009278-9](https://ro.wikipedia.org/wiki/Special:Referin%C8%9Be_%C3%AEn_c%C4%83r%C8%9Bi/0130092789)
* [On the Analysis of Cascading Style Sheets](http://dl.acm.org/citation.cfm?doid=2187836.2187946), [Pierre Geneves](https://ro.wikipedia.org/w/index.php?title=Pierre_Geneves&action=edit&redlink=1), Nabil Layaida, and Vincent Quint, Proceedings of the 21st International Conference on World Wide Web (WWW'12), p. 809–818, 2012.
* [\*\*\* - CSS: Tehnici esentiale - elefant.ro](https://www.elefant.ro/css-tehnici-esentiale_bc85dc2c-f186-4334-9d5e-d3273a11b9a4)
* [Amazon.it: A Smarter Way to Learn JavaScript: The new approach that uses technology to cut your effort in half - Myers, Mark - Libri](https://www.amazon.it/dp/1497408180?tag=hackr069-21&geniuslink=true)
* [JavaScript: The Definitive Guide: Activate Your Web Pages : David Flanagan: Amazon.it: Libri](https://www.amazon.it/dp/0596805527?tag=hackr069-21&geniuslink=true)
* [JavaScript & jQuery: The Missing Manual (Missing Manuals): McFarland, David Sawyer: 9781491947074: Amazon.com: Books](https://www.amazon.com/gp/product/1491947071/ref=as_li_qf_sp_asin_il_tl?ie=UTF8&tag=whatpixel-20&camp=1789&creative=9325&linkCode=as2&creativeASIN=1491947071&linkId=e09da3bd0fa9d741cfbcaae3d01c325c)
* [Amazon.com: Learning jQuery Fourth Edition eBook : Chaffer, Jonathan, Swedberg, Karl: Kindle Store](https://www.amazon.com/gp/product/B00DMYO2KA/ref=as_li_qf_sp_asin_il_tl?ie=UTF8&tag=whatpixel-20&camp=1789&creative=9325&linkCode=as2&creativeASIN=B00DMYO2KA&linkId=5d671bc5a9f5356f3e21027adad4a034)

Cuprins

[#Elemente 7](#_Toc126230610)

[#Atribute 8](#_Toc126230611)

[#Relații de rudenie 9](#_Toc126230612)

[#Resurse 10](#_Toc126230613)

[Resurse suplimentare 10](#_Toc126230614)

[Cum functioneaza acest limbaj 11](#_Toc126230615)

[Dintre miile de editoare de HTML… 12](#_Toc126230616)

[Tag-urile 13](#_Toc126230617)

[Elementele 13](#_Toc126230618)

[Atributele 13](#_Toc126230619)

[Ce este CSS si pentru ce se poate utiliza 13](#_Toc126230620)

[Ce este CSS ? 13](#_Toc126230621)

[CSS in HTML – ce inseamna stilizarea paginilor HTML 14](#_Toc126230622)

[Limbajul de programare CSS – caracteristici 15](#_Toc126230623)

[Elemente si atribute in HTML si CSS 16](#_Toc126230624)

[48.1 Unde definim stilurile CSS ? 17](#_Toc126230625)

[48.2 Stil css definit in HEAD 17](#_Toc126230626)

[48.3 Stil css in partea BODY ( css inline ) 18](#_Toc126230627)

[48.4 Definirea stilurilor cu un fisier extern 18](#_Toc126230628)

[48.5 Esenta Css-ului In ce consta puterea – Tutorial Css 19](#_Toc126230629)

[Ce este JavaScript și cum funcționează? 20](#_Toc126230630)

[Triada paginii web 20](#_Toc126230631)

[Istoricul JavaScript 21](#_Toc126230632)

[Ce pot face cu JavaScript? 22](#_Toc126230633)

[Cum functioneaza JavaScript? 22](#_Toc126230634)

[Încărcarea JavaScript într-o pagină web 23](#_Toc126230635)

[Unele Fragmente de cod JavaScript JavaScript 24](#_Toc126230636)

[Nu poți ocoli JavaScript pe web 25](#_Toc126230637)

[2. Este o abilitate foarte solicitată pe piața muncii 25](#_Toc126230638)

[Poți aduce caracteristici uimitoare pe site-urile web 25](#_Toc126230639)

[Este mai mult decât un simplu limbaj de programare front-end 26](#_Toc126230640)

[Este ușor de învățat. 26](#_Toc126230641)

[De ce este limbajul JavaScript atât de popular? 26](#_Toc126230642)

[Accesibilitate 26](#_Toc126230643)

[Tastare dinamică 26](#_Toc126230644)

[Tehnologii moderne 26](#_Toc126230645)

[Omni-platform 27](#_Toc126230646)

[Fără compilare 27](#_Toc126230647)

[Interoperabilitate 27](#_Toc126230648)

[48.6 La ce poți folosi programarea JavaScript? 27](#_Toc126230649)

[1. Web design 27](#_Toc126230650)

[2. Aplicații mobile 27](#_Toc126230651)

[3. Jocuri 27](#_Toc126230652)

[4. Realitate virtuală 27](#_Toc126230653)

[5. Inteligența artificială 27](#_Toc126230654)

[6. Hărți interactive 28](#_Toc126230655)

[7. Sisteme de rezervări 28](#_Toc126230656)

[8. Animații 28](#_Toc126230657)

[Ce opțiuni ai pentru a învăța JavaScript? 28](#_Toc126230658)

[Pe cont propriu: cărți și tutoriale JavaScript 28](#_Toc126230659)

[Definiție - Ce înseamnă jQuery? 31](#_Toc126230660)

[Techopedia explică jQuery 31](#_Toc126230661)

[Selectare elemente HTML in jQuery 33](#_Toc126230662)

[Evenimente in jQuery 35](#_Toc126230663)

[Citire proprietati CSS 36](#_Toc126230664)

[Setare proprietati CSS 37](#_Toc126230665)

[Adaugare si stergere clase 38](#_Toc126230666)

[Verificare daca o clasa e adaugata 39](#_Toc126230667)

[Informatii generale 40](#_Toc126230668)

[Descriere 40](#_Toc126230669)

[Tehnologii 40](#_Toc126230670)

[Cerinte sistem 41](#_Toc126230671)

[49 44](#_Toc126230672)

[50 46](#_Toc126230673)

[51 Pentru a crea aplicația, am utilizat faimosul triplet HTML/CSS/JS, de asemenea Mongo DB pentru baza de date. Iată câteva exemple: 46](#_Toc126230674)

[Bibliografie 47](#_Toc126230675)