

## HTML1 - 5

### Scurt istoric

Până în prezent, HTML a trecut prin cinci standarde majore, inclusiv cel mai recent 5.0. Următoarele secțiuni oferă o scurtă trecere în revistă a diferitelor versiuni și tehnologii.

HTML 1.0 – Nu a fost specificat în mod oficial de W3C, deoarece W3C a venit prea târziu. HTML 1.0 a fost folosită specificația originală Mosaic 1.0 și suporta puține elemente. Ceea ce nu se putea face pe o pagină este mai interesant decât ceea ce se putea face. Nu se putea seta culoarea de fundal sau imaginea de fundal a paginii. Nu existau tabele sau ferestre (frames). Nu se putea seta fontul. Toate imaginile de pe o singură linie (inline) trebuiau să fie GIF. Toate paginile arătau în mare parte la fel : fundal gri și fontul Times Roman. Link-urile erau afișate cu albastru până când erau vizitate, atunci deveneau roșii. HTML 1.0 a fost implementat doar în Mosaic și Lynx (un browser doar de text care rulează sub UNIX).

HTML 2.0 – Pași mari înainte s-au realizat între HTML 1.0 și HTML 2.0. HTML 1.1 de fapt, a existat, creat de Netscape pentru a sprijini ceea ce putea face primul său browser. Deoarece numai Netscape și Mosaic au fost disponibile la momentul respectiv (ambele scrise sub conducerea

lui Marc Andreessen), producătorii de browsere au avut obiceiul de a adăuga noi caracteristici și de a crea nume pentru elementele HTML pentru a utiliza aceste caracteristici. Între HTML 1.0 și HTML 2.0, a apărut W3C, sub conducerea lui Tim Berners-Lee, fondatorul Web. HTML 2.0 a fost o mare îmbunătățire față de HTML 1.0. Culoarele de fundal și imaginile puteau fi setate. Formularele au devenit disponibile cu un set

limitat de câmpuri, pentru prima dată vizitatorii de pe o pagină Web puteau să trimită informațiile. Tabelele au devenit posibile.

HTML 3.2 – De ce nu 3.0 ? W3C nu a putut obține o listă de specificații în timp util pentru acord, de către membri. HTML 3.2 a fost cu mult mai bogat decât HTML 2.0. Acesta a inclus suport pentru CSS. Chiar dacă CSS a fost suportat în HTML 3.2, producătorii de browsere nu au sprijinit CSS destul de bine pentru ca designeri să poată profita de această calitate. HTML 3.2 a extins numărul de attribute care a permis designerilor să personalizeze aspectul unei pagini (exact opusul HTML 4). HTML 3.2 nu a inclus suport pentru ferestre (frames) , dar producătorii de browsere le-au implementat oricum.

HTML 4.0 – Ce a adus nou HTML 4.0 ? Nu atât de multe elemente noi , dar totuși există câteva, cum ar fi regândirea direcției pe care o ia HTML. Până în prezent HTML încuraja intercalarea elementelor de prezentare în pagina. HTML 4.0 dezaprobă în mod clar orice utilizare a HTML care se referă la forțarea browser-ului pentru a formata un element într-un anumit fel. Tot ce ține de format a fost mutat în fișierul CSS . Cu informațiile de format răspândite peste tot prin pagini , HTML 3.2 a atins un punct unde întreținerea era costisitoare și dificilă. Această mișcare de a separa prezentarea într-un fișier CSS , o dată pentru totdeauna, ar trebui să faciliteze creșterea continuă și rapidă a Web-ului.

HTML 5.0 – A fost publicat în Octombrie 2014 de către W3C pentru a îmbunătăți limbajul cu suport pentru cele mai recente multimedia, în timp ce îl menține ușor de citit de către oameni și bine înțeles de către calculatoare și dispozitive , cum ar fi browsere web , parsere, etc. HTML 5.0 include modele detaliate de procesare pentru a încuraja mai mult implementările

interoperabile. HTML 5.0 este de asemenea un candidat pentru aplicațiile mobile, deoarece include caracteristici proiectate pentru dispozitive cum ar fi tabletele sau telefoanele inteligente. Multe caracteristici sintactice sunt incluse. Pentru a include și manipula nativ multimedia și conținut grafic au fost introduse elementele : <video>, <audio> și <canvas>. Pentru a îmbogăți conținutul semantic al documentelor, noi elemente de structură a paginii au fost adăugate, cum ar fi: <main> , <section> , <article>, <header>, <footer>, <aside>, <nav> și <figure>. Noi attribute au fost introduse, unele

elemente și attribute au fost eliminate, iar altele, cum ar fi <a>, <cite> și <menu> au fost modificate, redefinite sau standardizate.

**HTML (Hypertext Markup Language) este un limbaj folosit în publicarea paginilor statice web dezvoltat de către W3C (World Wide Web Consortium), o comunitate internațională de dezvoltatori condusă de către Tim Berners-Lee și Jeffrey Jaffe<sup>1</sup>. Misiunea W3C este să dezvolte la potențial maxim standardul World Wide Web prin protocoale și asigurarea creșterii pe termen lung a standardului Web<sup>2</sup>.**

**Rolul HTML este de a asigura prezentarea uniformă a informațiilor**

**În acest moment se fac eforturi serioase pentru lansarea versiunii HTML5 împreună cu WHATWG (Web Hypertext Application Technology Working Group),-**

HTML5 este doar o recomandare W3C și nu o variantă finală a acestui limbaj<sup>4</sup>, modificări importante putând fi aduse în viitorul apropiat.

**WHATWG este o comunitate care asigură dezvoltarea standardului Web focalizată pe îmbunătățirea standardelor HTML și APIs (Application Programming Interface)** necesare aplicațiilor web. Organizația WHATWG a fost fondată de către dezvoltatori de la Apple, Fundația Mozilla și Opera Software în 2004 în urmă unui congres W3C îngrijorați de lipsa de interes pentru HTML și nerespectarea nevoilor creatorilor independenți de cod HTML. Această a fost de altfel și misiunea acestui grup<sup>5</sup>.

**HTML este scris sub formă de elemente numite etichete (tags) care vin de cele mai multe ori în perechi (de deschidere / de închidere) dar nu neapărat.** În interiorul acestor elemente se introduc informațiile text sau chiar alte elemente HTML. HTML este un cod care stă la baza paginilor web și poate fi editat într-un program de tipul text editor fiind salvate ulterior cu extensia html sau htm.

**HTML DOM (Document Object Model)** descrie obiecte și metode specifice documentelor HTML oferind funcționalități de manipulare a documentelor structurate ierarhic, a

elementelor și atributelor HTML. El expune un număr de metode și proprietăți convenabile care sunt în concordanță cu modelele existente.

Structura unui astfel de fișier trebuie să fie de formă descrisă mai jos unde:

- etichetele `<html>` delimitează conținutul HTML al paginii
- etichetele `<head>` marchează descrieri de tip `<meta>`, link-uri către alte fișiere (scripturi, stiluri CSS etc) sau stiluri pentru formatarea textului, `title`
- etichetele `<body>` vor conține etichetele HTML împreună cu informațiile text pe care le conțin și care vor fi afișate în fereastră browser-ului

De precizat că prima linie a acestui fișier trebuie să conțină eticheta `<DOCTYPE>` care descrie versiunea HTML folosită la creare codului din fișier. Această etichetă poate să aibe una din variantele de mai jos<sup>6</sup>:

➤ HTML 4.01 Strict:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML  
4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

➤ HTML 4.01 Transitional:

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML  
4.01 Transitional//EN"

"http://www.w3.org/TR/html4/loose.dtd">

➤ HTML 4.01 Frameset:

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML  
4.01 Frameset//EN"

"http://www.w3.org/TR/html4/frameset.dtd">

➤ HTML 5:

<!DOCTYPE html>

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<!-- taguri din head -->
```

```
</head>
```

```
<body>
```

```
</body>
```

```
</html>
```

**HTML5 este o continuare a standardului HTML care îmbunătățește codul HTML oferind suport pentru cele mai noi aplicații multimedia și în același timp menține modul simplu de înțelegere a codului de către browser-ele de pe calculator și mobile.** Acest standard încearcă să elimine caracteristicile cu specificații diferite, caracteristicile diferite de software, browsere și erorile de sintaxă în documentele web deja existente. Cu alte cuvinte, se încearcă să se obțină un singur limbaj de marcare care poate fi scris în oricare din limbajele HTML. HTML5 extinde, îmbunătățește și raționalizează disponibilitățile documentelor web aducând în plus marcaje și resurse APIs pentru creare de aplicații web complexe.

HTML5 aduce caracteristici sintactice noi, elemente ca `<video>`, `<audio>`, `<header>`, `<canvas>` și altele, dar și renunță la unele elemente cum ar fi `<object>` care este prea generic. În acest fel se simplifică includerea și manipularea conținutului multimedia sau grafic fără a avea nevoie de plugin-uri externe codului HTML.

APIs și HTML DOM sunt părți esențiale în codul HTML5. Exemple de interfațe DOM pentru HTML5 extinse cu noi APIs:

- **Canvas** – pentru modul desen 2D/3D
- **Timed media playback** - Redarea media cu cronometru

- **Offline storage database** – pentru aplicații web offline
- **Editarea de documente**
- **Funcționalitatea Drag-and-drop**
- **Mesageria Cross-Document**
- **Managementul istoric al browserului**
- **MIME type=""** (Multipurpose Internet Mail Extensions) și protocolul de manipulare a înregistrărilor

*Schimbările apărute în HTML5 și-au făcut deja simțite prezența în randare pentru majoritatea browser-elor. Cu mici excepții HTML5 poate fi folosit la o anumită capacitate în toate browser-ele moderne.*

Nu toate elementele HTML5 sunt recunoscute încă de browser-e. În funcție de cerințele de dezvoltare a aplicațiilor web HTML5, trebuie verificate mereu starea de implementare a standardului.

Evoluția compatibilităților HTML5 în browsere pentru desktop, tablete și mobile sunt

<http://html5test.com/results/desktop.html>.

structura a unei paginii HTML5

```
<!DOCTYPE html>
```



```
<html>

<head>

  <meta charset="utf-8">

  <title>...</title>

</head>

<body>

  <header>...</header>

  <nav>...</nav>

  <article>

    <section>    ...    </section>

  </article>

  <aside>    ...    </aside>

  <footer>    ...    </footer>

</body>  </html>
```

## Semantica HTML5

HTML5 are o **sintaxa derivată din HTML** care este compatibilă cu **HTML4 și XHTML**. Desigur că nu are exact

aceeași sintaxa, însă regulile stricte din XHTML ca etichetele să fie scrise cu litere mici, atributele să aibe valori sau existența etichetelor de închidere nu mai sunt necesare.

**HTML5 oferă o flexibilitate mai mare și se vrea un limbaj mult simplificat. Scopul noii sintaxe este de a genera un cod cât mai scurt prin scoaterea părților de cod care nu pot influența rezultatul paginii HTML.**

**Modificările în sintaxa HTML5 apar încă de la prima declarație din pagină, cea a tipului de document la care nu mai este necesară trimiterea la o referință DTD (Document Type Definition). Astfel, declararea tipului de document se face mult mai simplu și fără a trebui să fie scrisă cu caractere mari. Mai jos este o comparație a declarării tipului de document comparativ HTML4 / HTML5.**

```
<!-- HTML 4.01 Strict -->
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML  
4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

```
<!-- HTML5 -->
```

```
<!DOCTYPE html>
```

În etichetă <head>, declararea datelor <meta> s-au schimbat și ele radical. Declararea setului de caractere este și ea modificată. Mai jos este o comparație a declarării setului de caractere, comparativ HTML4 / HTML5.

```
<!-- HTML4 – Char Set -->

<meta http-equiv="Content-Type"
content="text/html; charset=UTF-8">

<!-- HTML5 – Char Set -->

<meta charset="UTF-8">
```

Elementul <script> din <head> este el modificat, HTML5 renunțând la informațiile extra. Mai jos este o comparație a declarării elementului <script>, comparativ HTML4 / HTML5.

```
<!-- HTML4 – Script -->

<script type="text/JavaScript" src="scriptfile.js"></script>

<!-- HTML5 – Script -->

<script src="scriptfile.js"></script>
```

Declararea fișierelor externe de stil (elementul <link>) este și el modificat analog elementului <script> cu mențiunea

că nu există eticheta de închidere și adăugarea atributului “rel” cu valoarea ”stylesheet”. Mai jos este o comparație a declarării elementului <link>, comparativ HTML4 / HTML5.

```
<!-- HTML4 – Link -->
```

```
<link rel="stylesheet" type="text/css" href="stylefile.css">
```

```
<!-- HTML5 – Link -->
```

```
<link rel="stylesheet" href="stylefile.css">
```

Elementele HTML5 sunt folosite între eticheta de deschide și eticheta de închidere a elementului respectiv. Aceste etichete pot fi scrise cu caractere mici sau mari, însă pot fi scrise și ca o combinație de litere mici și mari. Există însă convenția că etichetele să fie scrise cu litere mici. Cu toate că cea mai mare parte a elementelor sunt conținute între eticheta de deschidere și cea de închidere, există totuși și excepții: etichetă br, hr, link, meta și altele.

Elementele în HTML5 ca și în HTML4 pot conține attribute pentru setarea mai multor proprietăți. Unele attribute sunt definite global iar altele sunt definite doar pentru un anumit element. Toate attributele sunt perechi de date care definesc numele atributului și valoarea pe care o conține. Mai jos este un

exemplu în care unui element <div> i se atașează un atribut “class” cu valoarea ”header”.

```
<div class="header">...</div>
```

Față de HTML4, noul HTML5 a adăugat elemente noi dar a și scos (“deprecate”). Exemple de elemente la care s-a renunțat sunt:

- <applet> - specifică un applet (.class) ---<object>
- <acronym> - specifică un acronim
- <basefont> - specifică un font de bază
- <center> - centrează un text
- <dir> - specifică o listă de directoare
- <big> - specifică un text mare
- <tt> - specifică un text teletype
- <font> - specifică mărimea și culoarea unui text
- <frame> - specifică o fereastră
- <frameset> - specifică un set de ferestre
- <noframes> - specifică o secțiune fără ferestre
- <u> - specifică un text subliniat
- <strike><s> - specifică un text tăiat
- <plaintext> - transformă documentul în text simplu

Totodată HTML5 a venit și cu elemente noi:

- <article> - specifică o bucată de conținut independent de document
- <aside> - specifică o bucată de conținut care este asemănătoare cu restul conținutului
- <audio> - specifică un fișier audio
- <canvas> - randează dinamic imagini grafice, grafice sau jocuri
- <command> - specifică o comandă pe care un utilizator o poate invoca
- <datalist> - specifică o listă creată dinamic
- <details> - specifică informații adiționale pe care un utilizator le poate obține
- <embed src="img/a.swf"> - specifică conținut extern
- <footer> - specifică o secțiune din document unde se trec de obicei date despre autor
- <header> - specifică o secțiune unde se trece de obicei un menu
- <nav> - specifică o secțiune destinată navigației
- <section> - specifică un document generic
- <time> - specifică o dată sau o ora
- <video> - specifică un fișier video

Cele mai folosite **atribute**, de altfel suportate de cele mai multe elemente HTML5 sunt date mai jos:

- **accesskey** – specifică o combinație de taste pentru accesarea elementului respectiv
- **align** – aliniază elementul orizontal (stânga, centru, dreapta)
- **background** – plasează o imagine sub element
- **bgcolor** – setează o culoare de fundal sub element (valoare numerică, RGB sau hexazecimal)
- **contenteditable** – specifică dacă utilizatorul poate edita sau nu conținutul elementului
- **data-XXXX** – crează attribute personalizate de către autor
- **draggable** - specifică dacă utilizatorul poate muta sau nu un elementul în pagină
- **spellcheck** – specifică dacă un element trebuie sau nu verificat (ortografic sau gramatical)
- **tabindex** – specifică ordinea tab-urilor

Pentru crearea structurii documentul HTML5 se pot utiliza următoarele elemente:

- **<section>** - reprezintă un document generic care împreună cu elementele de scriere

<h1>,<h2>,<h3>,<h4>,<h5> si <h6> indică structura textului afișat în browser

- <article> - specifică o parte de conținut independent de document
- < aside> - specifică o bucată de conținut care este asemănătoare cu restul conținutului
- <header> - specifică o secțiune unde se trece de obicei un menu
- < footer> - specifică o secțiune din document unde se trec de obicei date despre autor, informații despre drepturi de autor etc
- < nav> - specifică o secțiune destinată navigației
- < dialog> - specifică marcarea unei conversații
- < figure> - specifică un titlu și o parte a conținutului preluat extern paginii (image, audio, video etc)

#### Lista tagurilor noi in HTML 5.0

Tag-ul	Descrierea
<article>	marcheaza un articol
<aside>	marcheaza un continut aparte fata de continutul paginii, dar care are legatura cu el.



<audio>	marcheaza introducerea de continut audio
<canvas>	marcheaza introducerea de continut grafic
<command>	marcheaza un buton de comanda
<datalist>	marcheaza un meniu drop-down
<details>	marcheaza detaliile unui element
<dialog>	marcheaza un dialog, o conversatie
<embed>	marcheaza continut interactiv extern sau introducerea unui plugin
<figure>	marcheaza un grup de elemente care au legatura unul cu celalalt si care pot fi considerate in pagina, continut de sine statator.
<footer>	marcheaza sectiunea footer a pagini
<header>	marcheaza sectiunea header a pagini
<hgroup>	marcheaza o sectiune a pagini
<keygen>	marcheaza un cod generat automat intr-un formular
<mark>	marcheaza text evidentiati
<meter>	marcheaza valoarea unei unitati de masura cunoscute

<code>&lt;nav&gt;</code>	marcheaza o bara de navigare cu linkuri
<code>&lt;output&gt;</code>	marcheaza diferite tipuri de rezultate ale unui script oarecare.
<code>&lt;progress&gt;</code>	marcheaza o bara de progres fie ea grafica sau numerica
<code>&lt;rp&gt;</code>	defineste continut care va fi afisat in cazul in care browser-ul nu suporta tag-ul ruby
<code>&lt;rt&gt;</code>	defineste o regula sau o explicatie pentru tagul ruby
<code>&lt;ruby&gt;</code>	folosit impreuna cu caracterele asiatice
<code>&lt;section&gt;</code>	marcheaza o sectiune oarecare (header, footer, bara de navigare, capitole sau orice alta sectiune)
<code>&lt;source&gt;</code>	marcheaza sursa fisierului multimedia
<code>&lt;time&gt;</code>	marcheaza ora / data
<code>&lt;video&gt;</code>	marcheaza introducerea unui video

### Atribute noi pentru Tag-ul **input**:

Tag-ul	Descrierea
tel	Valoare de tip numar de telefon
search	Valoare de tip camp de cautare
url	Valoare de tip URL. (link)
email	Valoare de tip adresa de email.
datetime	Valoarea este de tip data sau timp
date	Valoarea este de tip data
month	Valoarea reprezinta una din lunile anului
week	Valoarea reprezinta una din saptamanile zilelor.
time	Valoarea este de tip timp
datetime-local	Valoarea este de tip date/time local
number	Valoarea este un numar
range	Valoarea este un numar dintr-un interval dat
color	Valoarea este o culoare in hexazecimal. Ex: #FF8800
placeholder	Specifica un indiciu care descrie valoarea preconizata a unui camp.

## Elemente de stocare pe web

HTML5 îmbunătățește mult sistemul HTML4 de stocare locală de date. **Modalitatea clasică de stocare** a unor date necesare rulării aplicațiilor web sub formă dorită de dezvoltator este cea prin intermediul așa numitelor ”**cookie**”. Un cookie este o parte mică de informație trimisă de website către browser și salvată local. De fiecare dată când pagina se încarcă browser-ul trimite informația către server care decide dacă o modifică sau nu și o retrimite înapoi către browser. Cookie-urile sunt create pentru a reține informații despre activitatea utilizatorului, informații necesare aplicației de server pentru afișarea în browser a datelor noi cerute conform navigării. Cookie-urile sunt perechi de informație (nume și valoare), iar pagina web poate accesa doar datele salvată de ea. **Dezavantajele acestui proces sunt cantitatea mică de informație care se poate salva (4kb) și reducerea performanței aplicației web prin trimiterea repetată a acestor date către server.**

Dar toate acestea se schimbă cu noul model HTML5 în care cookie-ul este înlocuit cu așa numitul **"web storage"**, o bază de date locală în care informațiile sunt salvate ca text sub formă de perechi cheie / valoare. Dimensiunea acestei baze de date este de maxim **10MB** (mult mai mare decât cea suportată de cookie) și accesul la ea se face la cerere, informația nefiind trimisă către server la fiecare accesare a paginii web. Procesul de preluare și modificare a informației web storage se face printr-un API în cadrul unui script conținut în pagină HTML.

În acest moment toate browser-ele moderne suportă această tehnologie, excepție făcând IE7 și versiunile mai vechi de IE7.

#### **Observatie:**

Web storage poate salva date local (**"local storage"** - fără a avea data de expirare cum se întâmplă la cookie, fiind accesibile chiar dacă browserul se redeschide) sau poate salva date despre sesiune (**"session storage"**). Datele privitoare la sesiune sunt șterse **când browser-ul se închide.**

HTML5 Web Storage asigură două noi obiecte (JavaScript) pentru a putea salva sau prelua datele dorite:

- **window.localStorage** accesează datele salvate local
- **code.sessionStorage** accesează datele privitoare la sesiune