

În lecția precedentă, ne-am familiarizat cu elementele de bază ale limbajului HTML. Acum, vom vedea care sunt părțile de bază ale documentelor HTML și vom crea prima noastră pagină. De asemenea, vom învăța și variantele existente ale limbajului și cum și când le putem aplica. Vom afla și ce este [doctype](#) și cum se folosește acesta.

Tag-urile de bază ale paginii HTML

Paginile HTML nu se scriu la întâmplare, ci respectă o structură definită cu precizie. Trei tag-uri sunt obligatorii, deci trebuie neapărat inserate în bază: **<html>**, **<body>** și **<head>**.

Notă:

În HTML5, aceste tag-uri se pot și omite pentru a obține o structură mai liberă a limbajului, **însă nu este recomandabil.**



Imaginea 3.1. Tag-ul HTML

Am întâlnit acest tag în lecția precedentă. Tag-ul HTML de început se află la începutul documentului HTML, în timp ce tag-ul HTML de final se află întotdeauna la sfârșitul documentului. Practic, acest lucru înseamnă că întregul conținut al documentului HTML complet se află în cadrul acestui tag. Tot ceea ce se găsește în cadrul tag-ului (între începutul și sfârșitul tag-ului <HTML>) reprezintă partea paginii HTML. Nu mai putem scrie nimic înainte sau după el, în afară de o excepție, despre care vom vorbi într-una din lecțiile viitoare. De asemenea, acest

element apare doar o singură dată pe pagină.



<head>

Imaginea 3.2. Tag-ul Head

Tag-ul head trebuie scris imediat după deschiderea tag-ului <html> menționat mai devreme. În tag-ul head, se află elemente care nu sunt vizibile în mod direct pe pagină. De exemplu, în partea head inserăm meta tag-urile folosite de motoarele de căutare când indexăm paginile. De asemenea, aici plasăm și linkuri către fișiere CSS externe și către alte fișiere etc.



<body>

Imaginea 3.3. Tag-ul Body

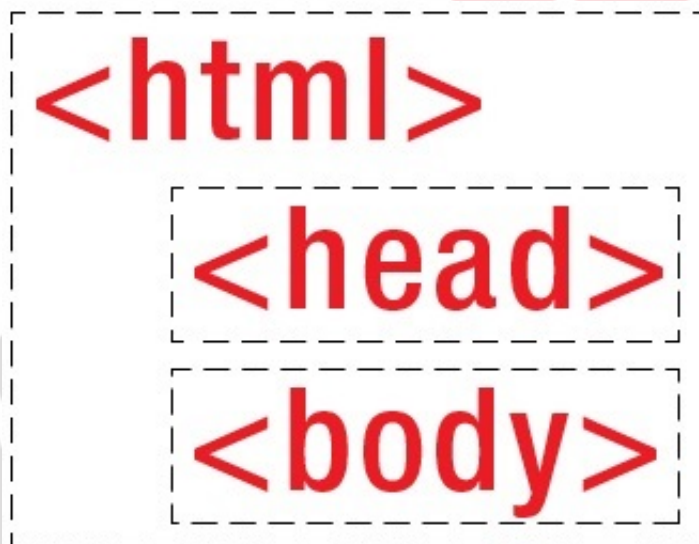
Tag-ul body urmează imediat după tag-ul head și conține tot ceea ce este vizibil direct pe pagină. O greșeală frecventă a începătorilor este că dublează acest tag sau o parte a acestuia, atunci când editează paginile sau când copiază părți ale unei pagini pe altă pagină.

Structura de bază a tuturor paginilor HTML conține aceste tag-uri, iar codul poate arăta astfel:

```
<html>
```

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

respectiv tag-ul HTML conține elementele head și body:



Imaginea 3.4. Trei tag-uri de bază ale paginilor HTML

Rețineți că **toate celelalte elemente, tot codul suplimentar se află fie în elementul head, fie în elementul body. Tag-urile nu trebuie scrise direct în HTML.** De asemenea, rețineți că **elementele html, head și body apar doar o singură dată și întotdeauna în ordinea prezentată mai sus.**

Prima noastră pagină HTML

Deoarece deja am asimilat câteva cunoștințe, haideți să creăm prima

noastră pagină HTML. CSS-ul îl lăsăm pentru mai târziu.

Dacă utilizați Windows, deschideți **Notepad**, iar dacă utilizați Mac, deschideți **TextEdit**.

Deși aplicațiile menționate sunt suficiente pentru acest curs, vă recomandăm să mai instalați și să folosiți Brackets – <http://brackets.io/>. Acest program este disponibil pe toate sistemele de operare populare (Windows, [MacOS](#), Linux). Acest instrument este gratuit, iar pe lângă acest lucru oferă o mulțime de opțiuni care ne facilitează munca. Despre el am vorbit în detaliu în materialul multimedia atașat primei lecții din cadrul acestui curs.

Notă:

Brackets este doar una dintre recomandările pentru folosirea editorului de text pentru acest curs însă voi, desigur, puteți folosi și alte editoare de text, cum este Notepad++, Atom și Sublime. Dacă aveți suficiente cunoștințe anterioare, puteți folosi și editoare mai avansate, cum este Dreamweaver, Eclipse și altele. Important este să alegeți un editor de text care vă place cel mai mult, deoarece codul HTML se poate prelucra în orice editor de text.

După ce ați pornit editorul de text, introduceți următorul cod:

```
<html>
<head>
  <title>My first HTML page</title>
</head>
<body>
  <h1>Hello!</h1>
  <p>I am HTML document.</p>
</body>
</html>
```

Îl puteți copia direct de aici sau îl puteți tasta singuri.

Dacă îl împărțim în elemente, observăm elementele **html**, **head** și **body** pe care le-am menționat mai devreme în lecție. În continuare, în head se află elementul **title** (element obligatoriu), care va fi vizibil într-un tab din browser. În secțiunea body, se află elementele **h1** și **p**, care reprezintă titlul și un paragraf al textului. De aceste tag-uri de text și de restul tag-urilor ne vom ocupa în următoarele lecții.

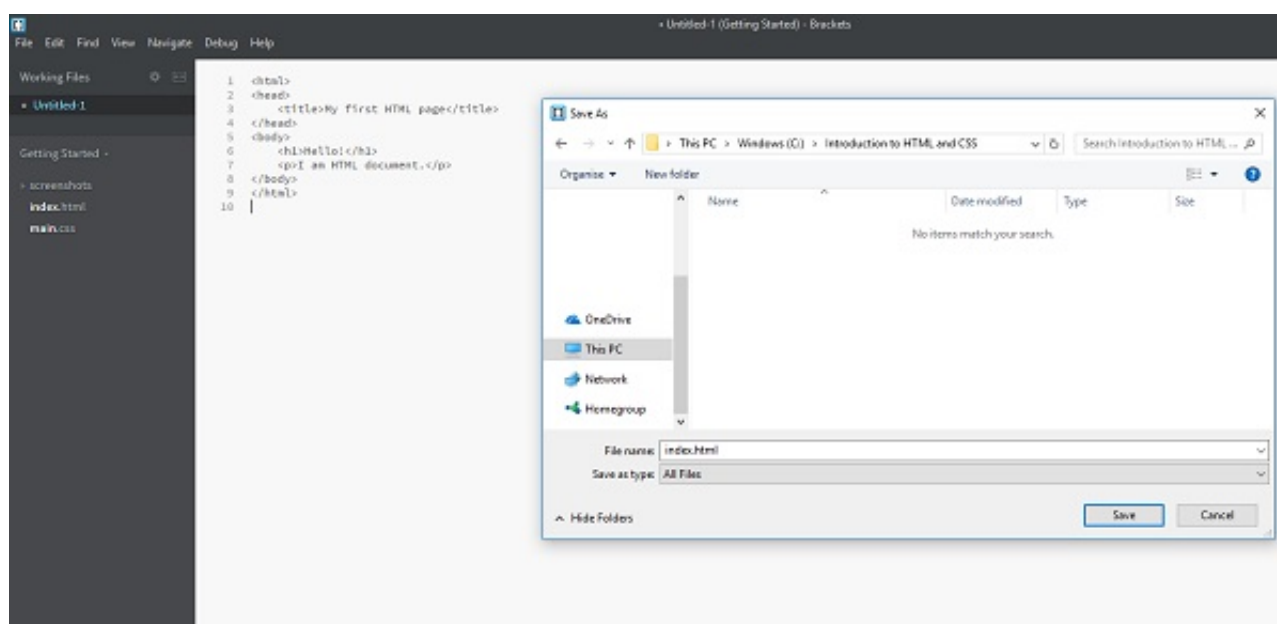
Salvați pagina undeva pe calculator, introduceți numele, dar asigurați-vă că nu o salvați ca document *txt* (care este setarea de bază, *implicită* în toate editoarele de text), ci ca fișier **html**. Nu uitați că toate fișierele html au extensia *.html* sau *.htm*.

Sfat:

De asemenea, există anumite reguli în timpul denumirii fișierelor HTML, folderelor și a celorlalte fișiere din HTML, care ar trebui respectate dacă vreți să evitați erori în timpul lucrului. Regulile sunt următoarele:

- Prima pagină sau pagina inițială, care se afișează atunci când înscriem o adresă web, se numește ca „index.html”, celelalte pagini se pot denumi în funcție de conținutul lor, de exemplu, „despre-noi.html”, „biografie.html” și altele.
- În denumiri nu sunt permise majuscule.
- Nu sunt permise nici diacriticele (ș, ă, î, ț...), sau alte semne speciale (/ , & etc.).
- Nu sunt permise numere.
- Nu sunt permise spații goale (se completează cu semnul minus sau cu liniuța de jos).

În imaginea de mai jos puteți vedea dialogul de salvare din cadrul instrumentului Brackets, în care este selectată salvarea fișierului ca document *.html*.



Imaginea 3.5. Salvarea paginii noastre, Brackets (Windows)

Notă:

În imagine puteți vedea că numele fișierului este definit ca index. Aceasta nu este întâmplător sau o alegere liberă a programatorului, deoarece prima pagină a fiecărui site trebuie să aibă întotdeauna denumirea index. De exemplu, când ne conectăm la un site. De exemplu, dacă accesăm www.link-academy.com/, serverul ne direcționează în mod automat la pagina acestui site, mai exact, în cazul acestui exemplu vom fi direcționați la pagina: www.link-academy.com/index.html chiar dacă noi nu vedem acest lucru.

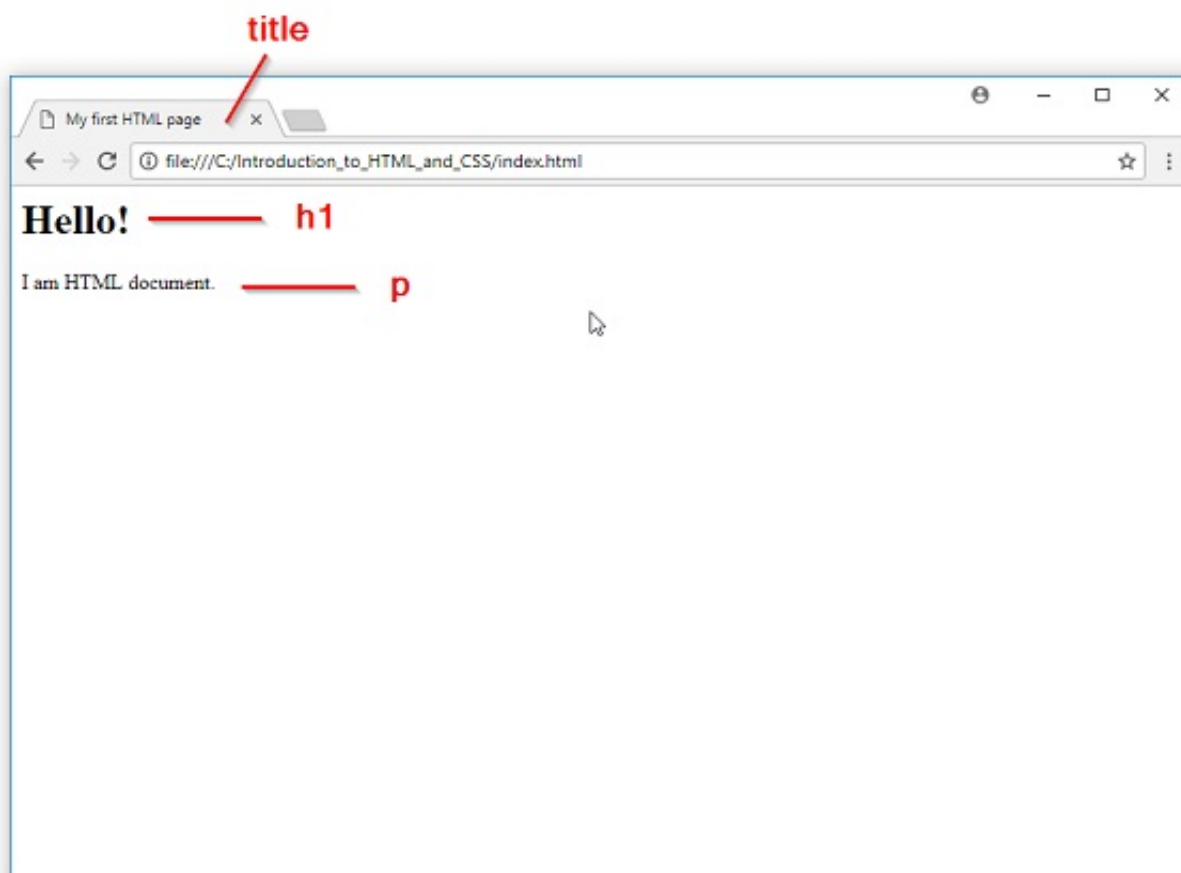
Vă întrebați de ce există acest sistem? De exemplu, pentru ca pagina noastră să se numească homepage, în momentul în care utilizatorul introduce adresa manual, acesta ar trebui să tasteze întreaga cale și să știe denumirea paginii principale. Exemplu, www.link-academy.com/homepage.html. De aceea, pentru server, fișierul index este pagina implicită când se accesează un site. Desigur, aceasta se poate schimba în setările serverului, setând ca o

altă pagină să fie implicită (ceea ce se și întâmplă în ziua de astăzi cu majoritatea site-urilor moderne).

Notă:

Salvați fișierul ca html fie prin alegerea tipului de fișier (Notepad++), fie prin tastarea manuală a extensiei .html la sfârșitul numelui (Notepad++ și TextWrangler).

Acum, puteți localiza fișierul pe hard disk și îl puteți porni. Se va deschide browser-ul web setat ca implicit, care va afișa pagina noastră. Vom obține ceva similar următoarei imagini:



Imaginea 3.6. Afișarea paginii noastre în browser-ul Chrome

Puteți afișa codul paginii și din browser. Dați clic dreapta pe pagină și din meniul derulant selectați View page [source](#), respectiv *View source* (în funcție de browser). Se va afișa același cod pe care tocmai l-ați introdus în editor. Din această fereastră de dialog a browser-ului, codul nu poate fi schimbat, dar este util atunci când doriți să vizualizați HTML-ul unei alte pagini pe care nu ați creat-o dvs.

Să privim exemplele unor site-uri populare precum Wikipedia (wikipedia.org) și Amazon (amazon.com). Dacă deschidem sursa site-ului Wikipedia, vom observa că și această pagină are tag-urile de bază `<html>`, `<body>` și `<head>`, chiar dacă are și foarte multe rânduri ale altui cod.

Principalele tag-uri menționate sunt marcate în imaginile de mai jos cu roșu:


```
1 <!DOCTYPE html>
2 <html lang="mul" dir="ltr">
3 <head>
4 <!-- Sysops: Please do not edit the main template directly; update /temp and synch
5 <meta charset="utf-8">
6 <title>Wikipedia</title>
7 <meta name="title" content="Wikipedia">
8 <meta name="description" content="Wikipedia, the free encyclopedia that anyone can
9 <meta name="author" content="Wikimedia Foundation">
10 <meta name="copyright" content="Creative Commons Attribution-Share Alike 3.0 and G
11 <meta name="publisher" content="Wikimedia Foundation">
12 <meta name="language" content="Many">
13 <meta name="robots" content="index, follow">
14 <!--[[if lt IE 7]]<meta http-equiv="imageToolbar" content="no"></endif-->
149 .central-textlogo img {
150   width: 100px;
151   height: auto;
152 }
153 }
154 </style>
155 </head>
156 <body id="www-wikipedia-org">
157
158 <div class="central-textlogo">
159 
161 </div>
162 <!-- container div for the central logo and the links to the largest language edit
163 <div class="central-featured">
164
165 </div>
166
167 <!-- Site info -->
168 <div class="wm-site-info">
169 <a href="//wikimediafoundation.org/" title="A Wikimedia Project"></a>
171 </div>
172
173 <script src="//bits.wikimedia.org/meta.wikimedia.org/load.php?debug=false&lang
174 </body>
175 </html>
```

Imaginea 3.7. Exemplu de cod al site-ului wikipedia.org

```

21
22 <html>
23 <head>
24 <script type="text/javascript">var ue_t0=ue_t0||+new Date();</script>
25
26 <script type='text/javascript'>document.write('<!--')</script>
27 <noscript>
28 <meta http-equiv='set-cookie' content='jsState=dsab; path=/; domain=.amazon.co
29 <meta http-equiv='refresh' content='0;URL=/gp/navigation/js-enabled-transition
%3Fie%3DUTF8%26jsr%3D1%26rId%3D0ZS90SXN1HBEPQYGONJT' />
30 </noscript>
31
32 amznJQ.addLogical('csm-base', { "http://z-ecx.images-amazon.com/images/G/01/or
33 amznJQ.available('csm-base', function() {});
34 }
35
36 </script>
37 </head>
38 <body>
39 <a name="top"></a>
40 <div style="position:absolute; left:0px; top:-500px; width:1px;height:1px; ove
41 <a href="/access">A different version of this web site containing similar cont
42 www.amazon.com/access</a>
43 </div>
44 {window.addEventListener('load',a.onload,false);else if(window.attachEvent){windo
45 (a.ue_pr==3||a.ue_pr==4)}{a.ue._uep()}}}(ue_csm);
46 </script>
47
48 <a href="/gp/uedata/187-8484943-0799202?tepes=1&id=0ZS90SXN1HBEPQYGONJT">v</a>
49 <noscript>
55 </body>
56 </html>
57

```

Imaginea 3.8. Exemplu de cod pentru site-ul amazon.com

Variantele limbajului HTML

HTML s-a dezvoltat de-a lungul anilor, așa că astăzi avem mai multe versiuni actuale. Deși poate nu se deosebesc la prima vedere, cu siguranță există diferențe și acestea influențează drastic modul de scriere a codului, respectiv modul de creare a documentelor noastre, însă și modul în care vor fi reprezentate documentele respective.

Fiecare versiune nouă a limbajului HTML aducea anumite îmbunătățiri și elemente noi, în timp ce unele elemente vechi și anumite reguli erau eliminate. În general, limbajul HTML a crescut și a fost îmbunătățit odată cu apariția versiunilor noi, ceea ce implică și actuala versiune HTML5. Totuși, în paralel, din versiunea menționată este eliminată practic întreaga stilizare HTML. Astăzi, putem spune că avem 3 versiuni

actuale (cu unele variante ale acestora)

HTML 4

Această variantă a fost publicată în anul 1997, ceea ce în domeniul designului web reprezintă o perioadă de timp destul de lungă. HTML4 suportă o anumită stilizare direct prin HTML. În acest scop, există tag-uri precum sunt <center>, și altele. Totuși, astăzi, pentru fiecare stilizare se utilizează CSS. Practic, HTML4 nici nu se mai folosește, însă se poate recunoaște pe unele site-uri mai vechi. Recomandarea noastră este să nu creați pagini noi în HTML4.

xHTML 1

Publicat în anul 2000, xHTML1 reprezintă o combinație a limbajului HTML4 și XML, de unde a și primit numele de xHTML. Pe baza limbajului HTML4 sunt adăugate unele reguli foarte importante, dar și mult mai stricte din XML, în care intră și următoarele:

- fiecare element (tag) trebuie să aibă un tag de deschidere și unul de închidere, excepție fiind elementele cu autoînchidere (precum);
- numele atributelor în tag-uri trebuie scrise cu minuscule;
- toate atributele din tag-uri trebuie să aibă valori inserate între ghilimele;
- sunt interzise unele tag-uri din HTML4.

xHTML a permis ca documentele scrise în acest standard să poată funcționa ușor cu instrumentele prevăzute pentru XML. De asemenea, xHTML a asigurat și implementarea elementelor în alte formate care se bazează pe XML, cum sunt [SVG](#), MathML și altele.

Există trei tipuri de xHTML:

- **Strict XHTML 1.0** – Noile reguli XML introduse trebuie respectate în totalitate.

- **Transitional XHTML 1.0** – În acest tip, este permis un mod de scriere mai liber al documentelor. Este permisă utilizarea tag-urilor care nu se mai folosesc, precum sunt `<center>` și ``, de exemplu. Din denumirea acestui tip (transitional - tranzițional, de trecere), putem concluziona că acesta este conceput ca un tip tranzițional, temporar, însă în continuare se folosește împreună cu varianta strict.
- **XHTML1.0 Frameset** – Această variantă nu se mai folosește deloc, însă ea presupunea definirea regiunilor, a frame-urilor pe pagină, în care apoi erau încărcate pagini separate.

xHTML (strict și transitional) era, în urmă cu un an, un standard care se folosea foarte des. Totuși, de când a apărut versiunea stabilă de HTML5, acesta a devenit lider, pentru ca în octombrie 2014 să fie finalizată versiunea care obține totodată și statusul Recommendation, așadar putem spune că xHTML nu se va mai folosi.

HTML5

După cum am menționat deja, HTML5 este standardul recomandat momentan pentru crearea documentelor HTML. Tot ceea ce vom crea în cadrul acestui curs vom crea în această versiune.

Trebuie să menționăm și că HTML5 este deja implementat în toate browser-ele moderne de pe calculatoare (versiunile actuale), dar și pe dispozitivele mobile și pe alte dispozitive. De exemplu, Android, iOS și Windows Phone, cele mai populare sisteme de operare, deja suportă HTML5.

Totuși, trebuie să avem grijă ca paginile noastre să fie accesibile și vizibile în mod corect și în browser-e mai vechi, chiar dacă acest lucru necesită scrierea elementelor suplimentare. Se poate întâmpla ca unele elemente avansate să nu fie deloc vizibile în vechile browser-e (deoarece nu suportă HTML5), însă nu putem permite ca pagina să nu se afișeze deloc sau să nu fie funcțională. Această problemă apare la calculatoarele pe care sunt instalate browser-e mai vechi. De exemplu,

Microsoft Internet Explorer 8, lansat în anul 2009 și Internet Explorer 7, din anul 2006, nu suportă specificațiile HTML5, dar sunt browser-e care încă se mai folosesc (nu într-o măsură mare, dar sunt totuși prezente pe piață).

Am văzut ce variante de limbaj există. Pentru determinarea variantei pe care o folosim, avem la dispoziție **Doctype**-ul de la începutul paginii.

Doctype

Fiecare document trebuie să posede o declarație (în care se găsesc datele despre document și caracteristicile limbajului) și un element root, sub care se găsește structura documentului.

Când este vorba de HTML, declarația, de obicei, implică versiunea limbajului și adresa fișierului în care se găsește definiția pentru acest limbaj. Declarația se numește **Document Type Declaration** și se găsește la începutul fiecărei paginii HTML în „tag”-ul DOCTYPE. Motivul pentru care am introdus cuvântul tag între ghilimele este că acest DOCTYPE, de fapt, nu este un tag HTML adevărat, ci este o instrucțiune dată browser-ului (din cauza semnului ! după deschiderea tag-ului). Instrucțiunea poate fi dată și serverului web, dar o recunoaștem (ca și instrucțiunea browser-ului), în așa fel încât la începutul său conține caractere speciale (de exemplu, <? , <% ...).

În ceea ce privește instrucțiunea DOCTYPE (declarația), de obicei, aceasta arată astfel:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

Acest exemplu definește pagina ca XHTML 1.0 transitional menționat mai devreme sau ca:

<!DOCTYPE html>

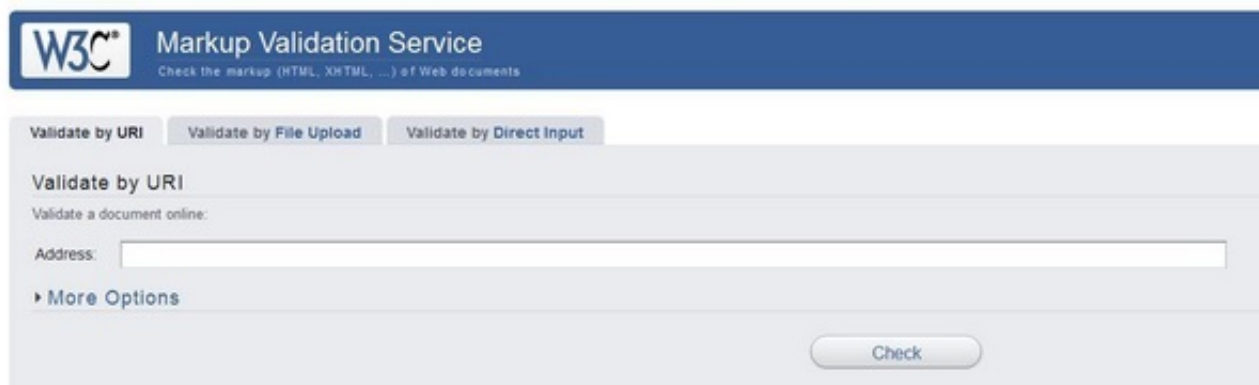
Ce definește pagina ca HTML5?

La ce se folosește definiția tipului de document?

DOCTYPE se poate imagina ca gramatica documentului HTML. La început, declarăm ce DTD, adică ce set de reguli vom folosi. Dacă respectăm regulile respective și dacă pagina noastră este creată conform acestora și nu există erori în cod, atunci putem spune că documentul nostru este **valid**. Regulile provin din versiunile limbajului HTML menționate mai devreme.

Cu alte cuvinte, la începutul documentului, noi definim singuri dacă vrem să creăm HTML5, XHTML1 transitional sau al treilea tip de pagină. Când determinăm aceasta, în continuare respectăm regulile stabilite pentru versiunea aleasă.

Documentul este valid dacă în el nu avem erori. Erorile pot include și utilizarea unei anumite reguli dintr-o altă versiune care nu este suportată în versiunea pe care o utilizăm. De exemplu, în XHTML am putut să folosim atributul [cellspacing](#) în tag-ul tabelului, în timp ce acest atribut concret este eliminat din HTML5. Dacă scriem atributul în XHTML, codul va fi valid, în timp ce, dacă îl scriem în HTML5, codul nu va fi valid (indiferent dacă browserul l-a recunoscut sau nu). Tabelele și detaliile legate de acestea vor fi abordate mai târziu în acest curs.



The image shows the W3C Markup Validation Service interface. At the top, there is a blue header with the W3C logo and the text "Markup Validation Service" and "Check the markup (HTML, XHTML, ...) of Web documents". Below the header, there are three tabs: "Validate by URI", "Validate by File Upload", and "Validate by Direct Input". The "Validate by URI" tab is selected. Under this tab, there is a text input field labeled "Address:" and a "Check" button. There is also a link for "More Options".

Imaginea 3. 9. W3C Validator

Validitatea documentului se poate verifica pe site-ul W3C, creat tocmai cu acest scop (<http://validator.w3.org/>). Când la adresa respectivă inserăm linkul URL până la site sau încărcăm o pagină sau o parte din pagină, validatorul va verifica erorile în cod (precum tag-urile care nu sunt închise, attributele care lipsesc și altele), dar va verifica și dacă codul este în concordanță cu DTD-ul pe care l-am setat. De exemplu, dacă folosim varianta XHTML strict și introducem tag-ul *font* care este interzis în această variantă, validatorul va semnală existența unei erori. Totuși, dacă folosim XHTML transitional, atunci nu va fi semnalată nicio eroare.

Validatorul nu este atotputernic și nu ne va corecta erorile, dar ne informează și ne arată unde am greșit, pentru a putea să corectăm ceea ce nu este scris corect.

Notă:

Vă recomandăm folosirea unui validator în timpul realizării site-urilor, deoarece vă poate ajuta foarte mult în timpul verificării validității documentului, chiar și atunci când există probleme cu fișierul vostru HTML. În special este util atunci când există o mică eroare, cum ar fi un tag incorect închis, care se observă mai greu.

Este important să menționăm că există trei moduri în care validatorul oferă un feedback cu privire la fișierul HTML. Cele trei moduri sunt:

- cu roșu este marcată eroarea,
- cu galben sunt marcate sfaturile referitoare la modul în care puteți standardiza fișierul vostru HTML,
- cu verde este marcat că nu există probleme cu documentul și că el este realizat conform standardului.

În cadrul materialului video atașat lecției puteți vedea în detaliu modul în care se folosește acest serviciu.

Variantele Doctype

Analog variantelor HTML, respectiv variantelor limbajului XHTML disponibile, există mai multe variante de declarații Doctype.

HTML4

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

Transitional XHTML 1.0

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0  
Transitional//EN""http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

Strict XHTML 1.0

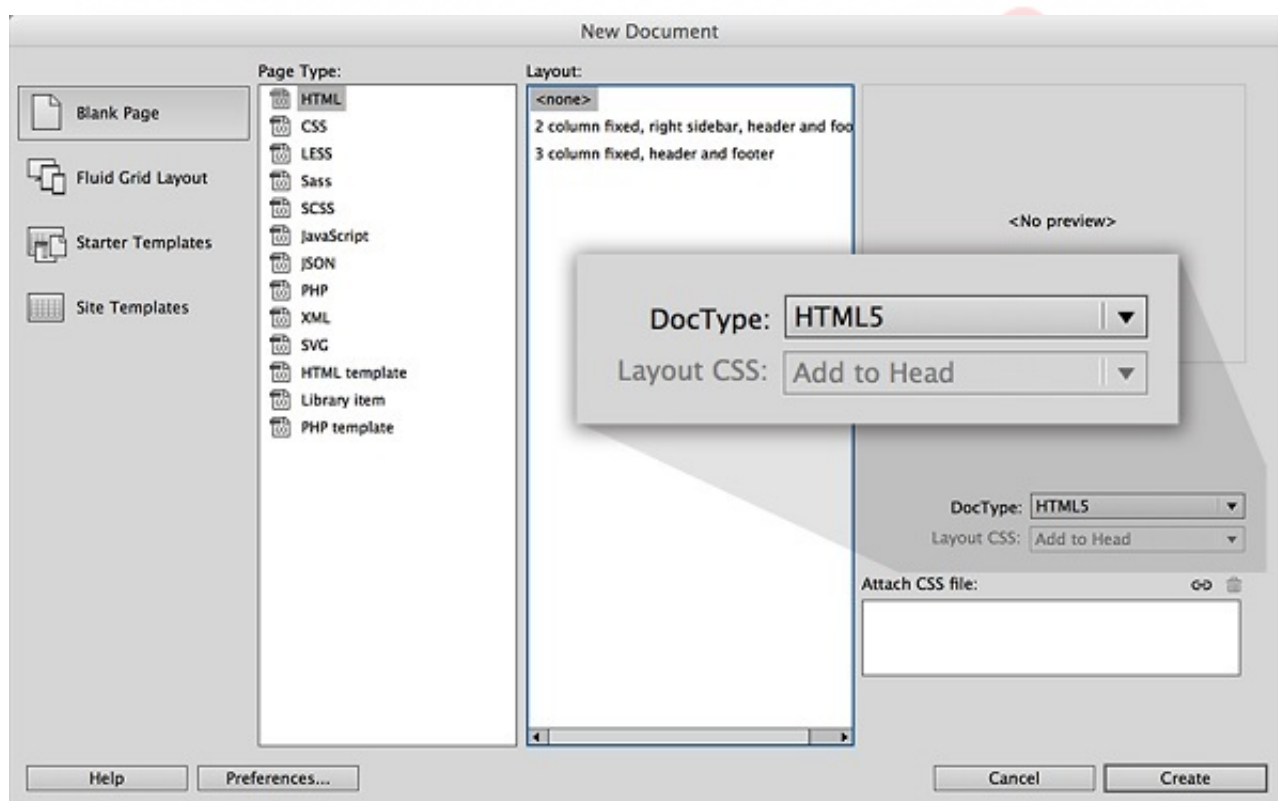
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0  
Strict//EN""http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

HTML5

La această variantă, Doctype este foarte mult simplificat, așadar scriem doar:

```
<!DOCTYPE html>
```

Acestea nu trebuie învățate pe de rost, ci le copiați doar la începutul noului document. Dacă folosiți un instrument mai avansat, precum, de exemplu, Adobe Dreamweaver, în timp ce creați documentul nou puteți să alegeți tipul de document, iar Dreamweaver se va ocupa de doctype.



Imaginea 3.10. Alegerea Doctype în timpul creării documentului HTML în instrumentul Dreamweaver

XMLNS

Mai devreme, am menționat că la începutul documentului inserăm tag-ul `<html>`. Acesta urmează imediat după Doctype. Dacă folosim o versiune mai veche de xHTML, atunci pagina „goală” va arăta astfel:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
  <title>New document</title>
</head>

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

Atenție la acest exemplu de pagină xHTML! Știm că este xHTML, fiindcă putem vedea în primul rând în Doctype. Ceea ce observăm este existența atributului **xmlns** în tag-ul HTML. Acest atribut este necesar în variantele xHTML, deoarece definește [namespace](#)-ul necesar în fișierele xml (ceea ce după structura și regulile sale este și xHTML). În general, vom folosi HTML5, așadar, acest detaliu nu ne va fi necesar.

Dacă observăm doar acest rând, vedem că acesta este un tag html clasic pe care l-am folosit deja, dar în plus avem și acest atribut xmlns.

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

Ca material suplimentar al acestei lecții, aveți la dispoziție o arhivă ZIP cu un doctype inclus, dar și cu alte elemente de bază pentru cele mai des folosite variante de HTML. Puteți descărca arhiva de la următorul [link](#).

Meta pentru UTF-8

Un detaliu foarte important, care, deseori, se poate anticipa, este setarea lui [encoding](#), respectiv inserarea lui meta tag, care va determina modul în care browser-ul trebuie să citească pagina noastră și cum să afișeze caracterele/literele specifice unei limbi.

Se recomandă ca encoding-ul să fie setat întotdeauna la [UTF-8](#). În caz contrar, caracterele specifice limbii române/diacriticele (de exemplu, ă,

î, ș, ț, â și altele) s-ar fi pierdut în totalitate, dar în special ar fi putut apărea probleme dacă s-ar fi folosit, de exemplu, alfabetul chirilic. În imaginea 3.11 puteți vedea cum omiterea definiției charset influențează cuvântul care folosește diacriticele ă, î, ș, ț, â.



Imaginea 3.11. Problemă de afișare a caracterului

Utf-8 rezolvă problema limbii și suportă aproape toate limbile active și recunoscute.

De aceea, în partea *head* (pentru HTML5) inserăm întotdeauna:

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

În timp ce pentru xHTML inserăm următoarele:

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

Acesta este meta tag-ul care definește encoding-ul. Nu trebuie să-l memorăm, ci îl copiem întotdeauna în aceeași formă în partea head.

După cum puteți observa, și meta tag-ul pentru encoding este

simplificat în versiunea HTML5.

Notă:

Având în vedere faptul că am menționat tag-urile meta, vom spune pe scurt ceva și despre ele. Tag-urile `<meta>` se introduc în partea head a paginii, rolul lor fiind de a oferi informații suplimentare despre pagina pe care se găsesc. Totuși, aceste informații nu sunt vizibile utilizatorului la prima vedere, deoarece se află în partea head a paginii și, prin urmare, nu sunt afișate în browser. Aceste informații sunt folosite doar de browsere, cu scopul prezentării corecte, precum și de motoarele de căutare (Google și altele), cu scopul indexării paginilor. Tag-ul `<meta>` este cu autoînchidere și de aceea nu are tag-ul inițial și cel final, ci toate informațiile sunt plasate în attribute asupra tag-ului propriu-zis, așa cum am putut vedea și în definiția pentru charset-ul documentului nostru.

Atributul Lang

În cadrul tag-ului HTML scriem încă un atribut, este vorba de atributul lang. Atributul lang îi comunică motorului de căutare în ce limbă sunt scrise conținuturile de pe pagină. Acest atribut nu este obligatoriu, însă pe validator veți primi întotdeauna o avertizare prin care sunteți informați că este adăugat acest atribut. Atributul lang se scrie în felul următor:

```
<html lang="en">
```

În acest caz am stabilit că documentul nostru HTML este scris în limba engleză. Dacă scriem conținuturi în limba română, am fi scris tag-ul în felul următor:

```
<html lang="ro">
```

Pentru cei care vor să afle mai multe detalii, la următorul link pot găsi întreaga listă de limbi cu prescurtările lor în limbajul HTML: <https://www.html.am/reference/iso-language-codes.cfm>

Notă:

Când, în cele din urmă, luăm în considerare tot ceea ce am spus în această lecție, baza paginii noastre HTML5 va arăta **întotdeauna** așa:

```
<!DOCTYPE HTML><html><head><meta charset="UTF-8"><title>HTML5</title></head><body></body></html>
```

Este foarte important să rețineți această bază, deoarece o veți folosi pentru fiecare pagină HTML nouă pe care o veți scrie.