

POGLAVLJE 1

Osnovna struktura

Brus Loson

IAKO SU U SREDIŠTU pažnje koju privlači HTML5 novi interfejsi za programiranje aplikacija (API), ima mnogo toga i za one koje zanimaju označavanje (engl. *markup*) i JavaScript. Dodato je trideset novih elemenata s novom semantikom koji se mogu koristiti na tradicionalnim, „statičkim“ stranicama. Postoji i niz novih kontrola za obrasce koje bi mogle da ukinu proveru ispravnosti JavaScript obrazaca.

Prionimo na posao. U ovom poglavlju, postojeću strukturu označavanja elementima `<div>` transformisaćemo u semantički sistem. Novi HTML5 strukturni elementi poput `<nav>`, `<header>`, `<footer>`, `<aside>` i `<article>` označavaju specifične vrste sadržaja. Pogledaćemo kako funkcionišu, te kako HTML5 dokumenti imaju nedvosmislenу strukturu i – dokazivo – bolju semantiku.

<head>

Krenimo od početka : DOCTYPE:

```
<!DOCTYPE html>
```

To je to. Bez glomaznih znakovnih nizova koje čak i najumešnji veb autori moraju da kopiraju i prenose. Bez URL adresa. Bez brojeva verzija. To je sve. Pre je u pitanju bajalica nego naredba: zahtevaju je svi veb čitači (engl. *web browsers*) kojima je deklaracija DOCTYPE neophodna za pokretanje režima Standards (engl. *standards mode*), a ovo je najkraći znakovni niz koji to pouzdano ispunjava. Napisali smo ga velikim slovima da bi bio usaglašen i sa HTML-om i sa XML-om, a predlažemo da i vi tako radite.

Sada treba da definišemo sistem za kodiranje znakova u dokumentu. U suprotnom, privamo prikrivenu ali realnu pretnju (pogledajte dokument <http://blog.php-security.org/archives/38-Wikipedia-UTF-7-XSSCross-Site-Scripting-Hole-Plugged.html>). Definicija kodnog sistema bi trebalo da se nađe u prvih 512 bajtova dokumenta. Za kodiranje znakova preporučujemo UTF-8 (sem ako imate baš dobar razlog da ne koristite taj kodni sistem):

```
<!DOCTYPE html>
```

```
<meta charset=utf-8>
```

Pogledajte pažljivije oznaku <meta>. Čitaoci umešni u pisanju XHTML koda zapišuće tri čudne stvari. Prvo, oznaka <meta> je mnogo kraća od uobičajene – <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">. To je i dalje ispravno, ali je kraći oblik bolji jer je lakše tako pisati a i već svuda funkcioniše.

Primetićete i da atribut u izrazu charset="utf-8" nisam stavio pod navodnike, niti sam zatvorio oznaku <meta charset=utf-8 />.

HTML5 nije XML jezik, pa nema potrebe za tim stvarima; ipak, ako želite, možete i tako da pišete. Sve ovo je ispravan HTML5 kôd:

```
<META CHARSET=UTF-8>
```

```
<META CHARSET=UTF-8 />
```

```
<META CHARSET="UTF-8">
```

```
<META CHARSET="UTF-8" />
```

```
<meta charset=utf-8>
```

```
<meta charset=utf-8 />
```

```
<meTa CHARset="utf-8">
```

```
<meTa CHARset="utf-8" />
```

Zašto toliko slobode u sintaksi? Odgovor je jednostavan: veb čitačima nikada nije bila važna XHTML sintaksa dokle god je kôd slat kao text/html – ona je bila važna samo XHTML validatoru. Zato bi favorizovanje jedne sintakse u jeziku HTML5 bilo potpuno proizvoljno i dovelo bi do toga da stranice koje ne poštuju taj format budu neispravne iako bi savršeno dobro funkcionisale u svakom čitaču. Zbog toga HTML5 ni ne zna koju sintaksu koristite.

Odaberite stil i držite ga se

Možete da koristite bilo koju od prethodno navedenih sintaksi, ali ne treba da ih mešate. To bi bila noćna mora za ažuriranje koda, posebno u slučaju velikih radnih timova.

Savetujemo da odaberete stil koji vam odgovara i da ga se držite. Nije važno koji stil: Remi voli XHTML sintaksu, dok su Brusu draža mala slova, minimizacija atributa (`controls` umesto `controls="controls"`) i atributi pod navodnicima samo kad je neophodno, kao prilikom davanja dve klase elementu – dakle, `<div class=important>`, ali `<div class="important logged-in">`. U knjizi ćete se susretati sa oba stila jer radimo onako kako nam najviše odgovara a vi morate znati da čitate oba.

Kao hrabri novi HTML5 autor imate slobodu izbora – ali držite se onog što izaberete.

Pošto već imamo neverovatnu slobodu u sintaksi (iz perspektive XHTML-a), hajde malo da varamo i da, pošto smo dodali naslov dokumenta, pređemo direktno na sadržaj:

```
<!DOCTYPE html>
<meta charset=utf-8>
<title>Interesting blog</title>
<p>Today I drank coffee for breakfast. 14 hours later,
-I went to bed.</p>
```

Proverimo li valjanost ovog uzbudljivog bloga, videćemo da je ispravan iako nema elemente `<html>`, `<head>` i `<body>` (slika 1.1).

SLIKA 1.1 Šokantno: dokument valjan a nema elemente `head`, `body` i `html`.

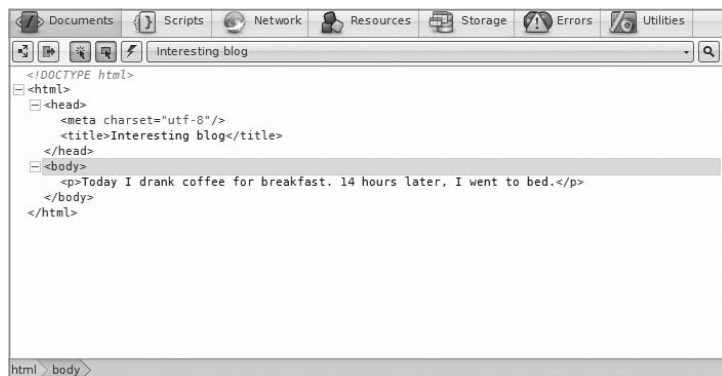
The document is valid HTML5 + ARIA + SVG 1.1 + MathML 2.0
(subject to the utter previewness of this service).

Source

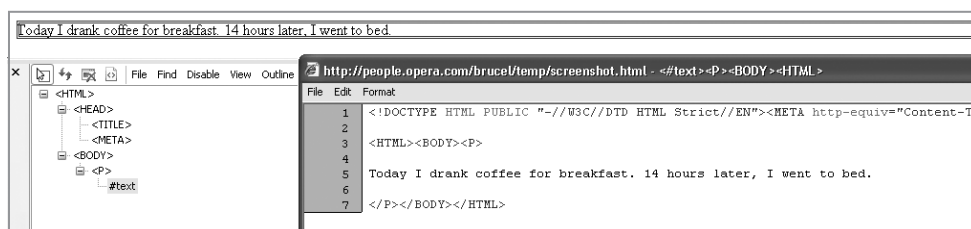
1. `<!DOCTYPE html>` ↵
2. `<meta charset=utf-8>` ↵
3. `<title>Interesting blog</title>` ↵
4. `<p>Today I drank coffee for breakfast. 14 hours later, I went to bed.</p>`

Ovo je možda jedan od onih trenutaka potpune zbunjenosti (*WTF?*) koje sam pomenio u uvodu. Ta tri elementa (XHTML autori, uhvatite se za nešto da ne padnete) potpuno su neobavezna jer ih čitači dopisuju gde treba. Dovoljno je da bacimo pogled ispod haube čitača pomoću razvojne alatke Opera Dragonfly kako bismo dobili potvrdu (slika 1.2).

SLIKA 1.2 Alatka Opera Dragonfly pokazuje da čitač dodaje elemente koji nedostaju.



Slika 1.3 prikazuje to isto pomoću razvojnih alatki Internet Explorera 6.



SLIKA 1.3 Internet Explorer 6, kao svi ostali čitači, dodaje elemente koji nedostaju u DOM. (Međutim, izgleda da stare verzije IE zamenjuju <title> i <meta>.)

HTML5 ne zahteva pomenute elemente pošto ih dodaju čitači veba. Ipak, ako ih izostavite iz svog koda za označavanje, verovatno ćete zbuniti kolege. Osim toga, ako planirate da koristite AppCache (pogledajte poglavlje 7) trebaće vam element <html> u kodu za označavanje. To je ujedno i dobro mesto za zadavanje primarnog jezika dokumenta:

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

Slabovidni korisnici bi mogli da pristupe vašoj veb lokaciji pomoću programa za čitanje ekrana (engl. *screen reader*) koji čita tekst na stranici sintetizovanim glasom. Kada čitač ekrana naiđe na niz znakova „six“, pročitace ga na različite načine zavisno od toga da li je jezik stranice engleski ili, na primer, francuski. Čitači ekrana mogu pokušati da pogode na kom jeziku je sadržaj, ali je mnogo bolje nedvosmisleno ga navesti, kao što sam ja uradio.

Internet Exploreru 8 i starijim verzijama neophodan je element <body> da bi primenili CSS stilove na nove HTML5 elemente, pa ima smisla koristiti i taj element.

Dakle, dodajemo te neobavezne elemente kako bismo napravili verovatno najmanju a ipak održivu HTML5 stranicu:

```
<!DOCTYPE html>
<html lang=en>
<head>
<meta charset=utf-8>
```

```
<title>Interesting blog</title>
</head>
<body>
<p>Today I drank coffee for breakfast. 14 hours later,
-I went to bed.</p>
</body>
</html>
```

Je li provera valjanosti uopšte više važna?

Pošto je sintaksa tako prilagodljiva, možemo da izostavljamo podrazumevane elemente poput `<html>`, `<head>` i `<body>`; i, još važnije, zahvaljujući tome što HTML5 definiše konzistentan DOM za svako eventualno loše označavanje, mogli biste se – kao i mi – zapitati da li je provera valjanosti uopšte bitna.

Mislimo da je i dalje važna kao alatka za proveru kvaliteta. Ali oduvek je i bila samo alatka, pomagalo, a ne cilj sama po sebi.

Cilj je semantičko označavanje: obezbeđivanje da elementi koje ste odabrali što približnije definišu značenje sadržaja, a da ne opisuju način predstavljanja. Moguće je da stranica bude potpuno valjana ali da sadrži samo oznake `display:table`, `div` i `span`, što nema nikakvu semantičku vrednost ni za koga. I obrnuto, izvanredno strukturirana, semantički bogata stranica može zbog jednog jedinog nekodiranog ampersenda da bude pogrešna, ali je i dalje semantička.

Kada vodimo razvojne timove, insistiramo da je potvrda valjanosti neophodna pre bilo kakve revizije koda, a kamoli pre objavljivanja koda. To je odličan način da se osigura da kôd zaista funkcioniše kako želite – čitači možda mogu da naprave konzistentan DOM od lošeg koda za označavanje, ali to ne mora biti željeni DOM.

Uz to, HTML5 raščlanjivača (engl. *parsers*) još uvek nema svuda, pa biste morali da osigurate ispravnost stranica kako biste obezbedili očekivano ponašanje CSS i JavaScript koda.

Preporučujemo <http://validator.w3.org/> ili <http://html5.validator.nu>. Očekujemo poboljšanja validatora poput opcija nametanja izbora pri kodiranju – na primer, odabracete da budete upozoreni zbog nekorišćenja XHTML sintakse, iako specifikacija to ne zahteva. Prilično dobra alatka za to je <http://lint.brihten.com>, mada nismo sigurni da li su njene procedure provere ažurne.

Korišćenje novih strukturnih elemenata

Godine 2004. Jan Hikson, urednik specifikacije za HTML5, pregledao je milijardu veb stranica pomoću Google Indexa, želeći da vidi šta čini „pravi“ veb. Jedna od analiza koje je potom objavio (<http://code.google.com/webstats/2005-12/classes.html>) predstavljala je listu najpopularnijih imena klasa u tim HTML dokumentima.

U skorije vreme, 2009. godine, Operin pretraživač MAMA ponovo je pregledao atribute klasa na 2.148.723 slučajno odabrane URL adrese, kao i id atribute elemenata (što nije bilo obuhvaćeno pretragom pomoću Google Indexa) na 1.806.424 URL adrese (tabela 1.1 i tabela 1.2).

TABELA 1.1 Imena klasa

POPULARNOST	VREDNOST	UČESTALOST
1	footer	179.528
2	menu	146.673
3	style1	138.308
4	msonormal	123.374
5	text	122.911
6	content	113.951
7	title	91.957
8	style2	89.851
9	header	89.274
10	copyright	86.979
11	button	81.503
12	main	69.620
13	style3	69.349
14	small	68.995
15	nav	68.634
16	clear	68.571
17	search	59.802
18	style4	56.032
19	logo	48.831
20	body	48.052

TABELA 1.2 ID imena

POPULARNOST	VREDNOST	UČESTALOST
1	footer	288.061
2	content	228.661
3	header	223.726
4	logo	121.352
5	container	119.877
6	main	106.327
7	table1	101.677
8	menu	96.161
9	layer1	93.920
10	autonumber1	77.350
11	search	74.887
12	nav	72.057
13	wrapper	66.730
14	top	66.615
15	table2	57.934
16	layer2	56.823
17	sidebar	52.416
18	image1	48.922
19	banner	44.592
20	navigation	43.664

Kao što vidite, kada uklonimo klase koje očigledno služe za predstavljanje, dobijamo prilično dobru sliku o strukturama koje autori pokušavaju da koriste na svojim stranicama.

Kao što HTML 4 odražava rani veb naučnika i inženjera (pa imamo elemente poput `<kbd>`, `<samp>` i `<var>`), HTML5 je odraz veba kakav je bio tokom razvoja: sadrži 30 novih elemenata od kojih su mnogi inspirisani klasama i id atributima iz ovih tabela jer je to ono što su koristili projektanti veb strana.

Pa hajde da ih upotrebimo dok nas drži ovo pragmatično raspoloženje a ne filozofsko. Evo primera početne strane (engl. *home page*) bloga, označene onako kako je uobičajeno za HTML 4 – pomoću semantički neutralnog elementa `<div>`:

```
<div id="header">
  <h1>My interesting life</h1>
</div>
<div id="sidebar">
  <h2>Menu</h2>
  <ul>
    <li><a href="last-week.html">Last week</a></li>
    <li><a href="archive.html">Archives</a></li>
  </ul>
```

```

</div>
<div class="post">
  <h2>Yesterday</h2>
  <p>Today I drank coffee for breakfast. 14 hours later,
  ~ I went to bed.</p>
</div>
<div class="post">
  <h2>Tuesday</h2>
  <p>Ran out of coffee, so had orange juice for breakfast.
  ~ It was from concentrate.</p>
</div>
<div id="footer">
  <p><small> This is copyright by Bruce Sharp. Contact me to
  ~ negotiate the movie rights.</small></p>
</div>

```

Jednostavan CSS poslužiće za stilizaciju:

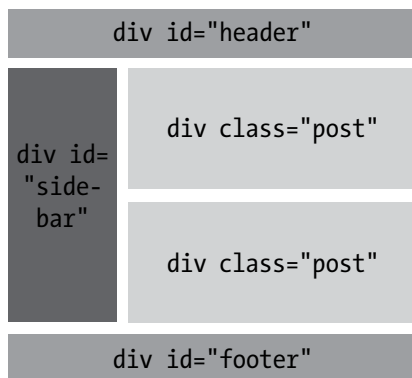
```

#sidebar {float:left; width:20%;}
.post {float:right; width:79%;}
#footer {clear:both;}

```

Šematski prikaz stranice dat je na slici 1.4.

SLIKA 1.4 Struktura našeg bloga kada se koristi HTML 4.



Iako u ovom kodu nema grešaka (i funkcionisaće bez problema u novom HTML5 svetu), najveći deo strukture je potpuno nepoznat veb čitaču jer je jedini pravi HTML element koji možemo da upotrebimo za ove važne orijentire (engl. *landmarks*) na stranici semantički neutralan `<div>` (definisan u HTML 4 specifikaciji kao „generički mehanizam za dodavanje strukture dokumentima“).

Dakle, ako se lepo prikazuje, šta nije u redu s ovim? Zašto bismo dodavali još elemenata kako bismo obogatili semantiku?

Recimo da imamo pametan veb čitač s tasterskom prečicom za direktan prelazak na navigaciju stranice. Kako da zna na šta treba da pređe? Pojedini autori pišu `<div class="menu">`, drugi koriste `class="nav"`, `class="navigation"`, `class="links"` ili bilo koji ekvivalent na jeziku koji nije engleski. Navedene tabele iz programa Opera MAMA sugerišu da bi menu, nav,

sidebar i navigation mogli biti sinonimi, ali nema garancija da je tako: recimo, na veb lokaciji (engl. *web site*) nekog restorana, izraz `<div class="menu">` mogao bi da se koristi ne za navigaciju već za listanje menija s jelima.

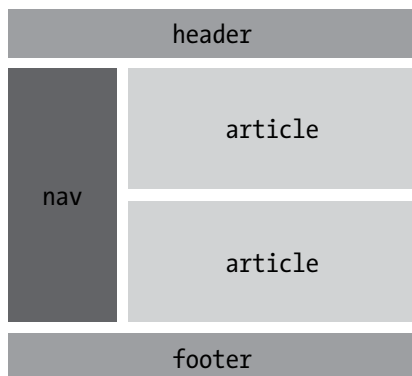
HTML5 sadrži nove elemente koji nedvosmisleno označavaju orijentire na stranici. Hajde da ponovo napišemo istu stranicu koristeći neke od tih elemenata:

```
<header>
  <h1>My interesting life</h1>
</header>
<nav>
  <h2>Menu</h2>
  <ul>
    <li><a href="last-week.html">Last week</a></li>
    <li><a href="archive.html">Archives</a></li>
  </ul>
</nav>
<article>
  <h2>Yesterday</h2>
  <p>Today I drank coffee for breakfast. 14 hours later,
  ↳ I went to bed.</p>
</article>
<article>
  <h2>Tuesday</h2>
  <p>Ran out of coffee, so had orange juice for breakfast.
  ↳ It was from concentrate.</p>
</article>

<footer>
  <p><small>This is copyright by Bruce Sharp. Contact me to
  ↳ negotiate the movie rights.</small></p>
</footer>
```

Šematski prikaz HTML5 verzije dat je na slici 1.5.

SLIKA 1.5 Struktura našeg bloga kada se koristi HTML5.



Pre nego što detaljnije razmotrimo kada koristiti ove nove elemente i šta oni znače, stilizujmo osnovne strukture stranice.

Zašto nema elementa <content>?

Lako je videti kako bi naša hipotetička tasterska prečica „jump to nav“ funkcionisala, ali se češće zahteva prelazak na oblast s glavnim sadržajem. Pojedini dizajneri koji vode računa o pristupačnosti strana osobama s posebnim potrebama dodaju vezu „skip links“ pri vrhu stranice kako bi omogućili korisnicima čitača ekrana da zaobiđu stavke navigacije. Zar ne bi bilo sjajno kada bi veb čitači omogućavali da se jednim pritiskom na taster odmah pređe na glavni sadržaj?

Međutim, HTML5 nema element <content> na koji bi se prešlo. Kako onda čitač da zna gde počinje glavni sadržaj stranice?

Zapravo, to se jednostavno postiže pomoću nečeg što zovem algoritam Scubi Du. Znamo da je osoba iza maske duha opaki čuvar pomenutog zabavnog parka, jer je on jedina osoba u epizodi koja nije Fred, Dafne, Velma, Šegi ili Skubi. Slično tome, prvi segment sadržaja koji nije u elementu <header>, <nav>, <aside> ili <footer> predstavlja početak glavnog sadržaja, bez obzira na to da li je u elementu <article> ili <div>, ili da li je direktni potomak elementa <body>.

To bi bilo zgodno korisnicima čitača ekrana, a proizvođači mobilnih telefona bi, recimo, mogli postići da se prozor veb čitača direktno pozicionira na centralni sadržaj.

Ako biste ipak želeli stilizaciju pomoću elementa <content>, upotrebite paket WAI-ARIA i primenite izraz `role=main` na bilo koji element koji omotava glavni sadržaj; to omogućava i stilizovanje pomoću CSS selektora atributa (IE6 ih ne podržava) – na primer, `div[role=main] {float:right;}` (više o paketu WAI-ARIA u poglavlju 2).

Stilizovanje HTML5 koda pomoću CSS-a

Stilizovanje ovih novih elemenata prilično je jednostavno u svim veb čitačima izuzev jednog: CSS možete primeniti na proizvoljan element jer, po specifikaciji, CSS je „jezik opisa stilova koji omogućava autorima i korisnicima da pridruže stil... strukturiranim elementima (na primer, HTML dokumentima i XML aplikacijama)“, a XML aplikacije mogu imati elemente koje poželite.

Dakle, kada se primeni CSS, element <nav> može da pluta, elementi <header> i <footer> mogu da dobiju granice a dodavanje margina i dopune (engl. *padding*) elementu <article> gotovo je jednostavno kao pomoću elementa <div>.

Iako nove HTML5 elemente možete koristiti odmah, stariji veb čitači ne moraju uvek da ih razumeju. Ne rade ništa posebno s njima i odnose se prema njima kao prema nepoznatim elementima koje ste izmislili.

Možda će vas iznenaditi to što CSS podrazumeva da su ti elementi tipa `display:inline`; znači, ako zadate samo visine i širine strukturnih elemenata – kao što radimo s elementom <div> – stranica neće pravilno funkcionisati u starim veb čitačima dok čitaču eksplicitno ne napišemo `display:block`. Čitači sadrže rudimentaran, ugrađen opis stila koji redefiniše

podrazumevano stilizovanje za one elemente za koje smatramo da su izvorno na nivou bloka (jedan takav opis stila može se naći na adresi <http://www.w3.org/TR/CSS2/sample.html>). Stariji čitači nemaju pravila koja nove HTML elemente poput `<header>`, `<nav>`, `<footer>`, `<article>` definišu pomoću parametra `display:block`, pa to moramo da uradimo sami, u našem CSS opisu stila. Modernim čitačima će naš red biti višak, ali neškodljiv, a za starije čitače – za koje svi znamo da umeju da se motaju po vebu dugo pošto ih njihovi proizvođači šutnu – biće korisna pomoć.

Dakle, da bismo stilizovali naš HTML5 kôd tako da odgovara našem HTML 4 dizajnu, dovoljni su nam naredni stilovi:

```
header, nav, footer, article {display:block;}
nav {float:left; width:20%;}
article {float:right; width:79%;}
footer {clear:both;}
```

I predivna HTML5 stranica je rođena – osim u jednom veb čitaču.

Stilizovanje HTML5 koda u Internet Exploreru 6, 7, 8

U starim (ali nažalost i dalje korišćenim) verzijama Internet Explorera, CSS se na odgovarajući način primenjivao na HTML 4 elemente koje IE podržava, ali se novi HTML5 elementi – nepoznati veb čitaču – ne stilizuju. To može da izgleda... pa, neprijatno.

Stari IE ćete naterati da primeni CSS na HTML5 tako što ćete ga bocnuti štapićem nanoštrenim JavaScriptom. Zašto? To je tajna koja se ne sme otkriti i – kad bismo vam rekli – morali bismo da vas ubijemo. (Zapravo, ni sami ne znamo.) Dodajte naredni segment JavaScript koda u zaglavlje stranice

```
<script>
document.createElement('header');
document.createElement('nav');
document.createElement('article');
document.createElement('footer');
</script>
```

i IE će magično primeniti stilove na te elemente, pod uslovom da u kodu za označavanje postoji element `<body>`. Dovoljno je da jednom napravite svaki element, koliko god puta se pojavljivao na stranici.

Podsećamo da HTML5 ne zahteva element `body`, ali je on neophodan specifičnoj mešavini koju čine Internet Explorer 8 (i ranije verzije), CSS, HTML5 i JavaScript. IE9 radi kao drugi veb čitači i nije mu potreban JavaScript.

Iako ovaj segment koda na JavaScriptu nije neophodan za druge čitače, neće im naškoditi. Međutim, možda biste hteli da optimizujete brzinu učitavanja stranice tako što ćete preuzimati i izvršavati ovaj skript u Internet Exploreru samo uz uslovne komentare (pogledajte tekst <http://dev.opera.com/articles/view/supporting-ie-with-conditional-comments/>).

NAPOMENA Za element `<script>` ne morate više da navodite vrstu skripta – podrazumeva se da je reč o JavaScriptu. To važi i za zastarele čitače, pa ovo možete odmah početi da primenjujete.

Skript za podršku

Drugo rešenje je da koristite Remijev kratak skript <http://remysharp.com/2009/01/07/html5-enabling-script/> koji u jednom potezu obavlja ovo za sve nove elemente, a obuhvata i IE Print Protector Džona Nila (<http://www.iecss.com/print-protector>) koji osigurava da HTML5 elementi budu pravilno stilizovani i prilikom štampanja dokumenata u Internet Exploreru.

Korisnik sa onemogućenim JavaScriptom po sopstvenom izboru ili zbog korporativnih bezbednosnih pravila, moći će da pristupi vašem sadržaju ali će videti delimično stilizovanu ili nestilizovanu stranicu. To će vas možda odbiti od opisanog rešenja. (Takvo siromašno veb iskustvo ima korisnik s prastarim Internet Explorerom i bez JavaScripta, pa teško da će vaša veb lokacija biti najgora s kojim se susreće.) Sajmon Piters je pokazao sledeće: ako znate kako DOM izgleda, možete delom da stilizujete HTML5 bez JavaScripta, ali rezultat nije skalabilan niti se da održavati; pogledajte tekst „Styling HTML5 markup in IE without script“ na adresi <http://blog.whatwg.org/styling-ie-noscript>.

Ostali problemi sa zastarelim veb čitačima

Ima i drugih problema sa zastarelim čitačima kada je reč o stilizovanju HTML5 elemenata. Starije verzije programa Firefox (pre verzije 3) i Camino (pre verzije 2) imale su grešku čije je rešenje opisano na adresi <http://html5doctor.com/how-to-get-html5-working-in-ie-and-firefox-2/>.

Ne smatramo da treba praviti opširan spisak ovakvih ponašanja; reč je o privremenim problemima za koje očekujemo da će brzo nestati sa objavljivanjem novih verzija i prelaskom korisnika na njih.

Kada koristiti nove HTML5 strukturne elemente

Upotrebili smo nove elemente za označavanje naše stranice i stilizovali ih; iako već njihova imena sugerišu ulogu koju imaju, vreme je da ih detaljnije proučimo.

<header>

U prethodnom primeru, kao na većini veb lokacija, zaglavlje će biti prvi element na stranici. Sadrži ime lokacije, logotipe, veze za povratak na početnu stranu itd. Evo šta kaže specifikacija:

„Element header predstavlja grupu uvodnih ili navigacionih alatki... Napomena: glavna namena elementa header je da obuhvati naslov odeljka (element h1–h6 ili element hgroup), ali to nije obavezno. Element header može da posluži i za omotavanje tabele sa sadržajem odeljka, obrasca za pretraživanje ili svakog relevantnog logotipa.“

Razmotrimo to. Prva važna stvar je da element **<header>** nije obavezan; u prethodnom primeru je suvišan jer okružuje samo element **<h1>**. Njegova vrednost je u grupisanju „uvodnih ili navigacionih“ elemenata, pa evo realnijeg primera:

```

<header>
  <a href="/"><img src=logo.png alt="home"></a>
  <h1>My interesting blog</h1>
</header>

```

Mnoge veb lokacije imaju naslov i podnaslov (engl. *tagline, subtitle*). Da bi se podnaslov izuzeo iz algoritma za strukturiranje, engl. *outlining algorithm* (tako da glavni naslov i podnaslov sačinjavaju jednu logičku celinu, što je objašnjeno u poglavlju 2), glavni naslov i podnaslov se mogu grupisati u novi element `<hgroup>`:

```

<header>
  <a href="/"><img src=logo.png alt="home"></a>
  <hgroup>
    <h1>My interesting blog</h1>
    <h2>Tedium, dullness and monotony</h2>
  </hgroup>
</header>

```

I zaglavlje može da sadrži navigaciju. To može biti veoma korisno kad je reč o globalnoj navigaciji (tj. navigacionom sistemu cele lokacije), a posebno za lokacije zasnovane na šablonima (engl. *template-driven sites*) gde se može desiti da ceo element `<header>` potiče iz datoteke šablona (engl. *template file*). Na primer, horizontalna globalna navigacija na adresi www.thaicookery.co.uk može se napisati na sledeći način (rezultat je prikazan na slici 1.6.):

```

<header>
  <hgroup>
    <h1>Thai Cookery School</h1>
    <h2>Learn authentic Thai cookery in your own home.</h2>
  </hgroup>
  <nav>
    <ul>
      <li>Home</li>
      <li><a href="courses.html">Cookery Courses</a></li>
      <li><a href="contact.html">Contact</a></li>
    </ul>
  </nav>
</header>

```

SLIKA 1.6 Zaglavlje za www.thaicookery.co.uk.



Naravno, nije obavezno da element `<nav>` bude u elementu `<header>`. Primer s tajlandskim restoranom mogao se lako napisati tako da glavni element `<nav>` bude izvan elementa `<header>`:

```
<header>
  <hgroup>
    <h1>Thai Cookery School</h1>
    <h2>Learn authentic Thai cookery in your own home.</h2>
  </hgroup>
</header>
<nav>
  <ul>
    <li>Home</li>
    <li><a href="courses.html">Cookery Courses</a></li>
    <li><a href="contact.html">Contact</a></li>
  </ul>
</nav>
```

Rešenje u velikoj meri zavisi od toga smatrate li da globalna navigacija (engl. *sitewise navigation*) pripada zaglavlju cele lokacije, kao i od pragmatičnih aspekata lakšeg stilizovanja. Uzmimo kao primer moju ličnu veb lokaciju, s veoma dugačkom globalnom navigacijom smeštenom levo od oblasti s glavnim sadržajem; takva navigacija može biti mnogo duža od samog teksta na strani. Ako bi se element `<nav>` stavio u element `<header>`, bilo bi veoma teško namestiti glavni sadržaj na pravo mesto i postaviti podnožje, tako da je u ovom slučaju globalna navigacija izvan elementa `<header>` i potomak je elementa `<body>`, kao u primeru na slici 1.7.



SLIKA 1.7 Tipična stranica s globalnom navigacijom izvan glavnog zaglavlja.

Trenutno pravimo samo glavni element `<header>` za stranicu, ali može ih biti i više; na to ćemo se osvrnuti u poglavlju 2.

<nav>

Element **<nav>** je namenjen označavanju navigacije. Pod navigacijom podrazumevamo veze (engl. *links*) oko stranice (na primer, tabelu sa sadržajem, engl. *table of contents*, na vrhu članka, s vezama ka odredišnim tačkama na toj strani) ili unutar veb lokacije. Ipak, nije svaki skup veza element **<nav>**; to, na primer, nisu lista sponzorisanih veza niti stranica s rezultatima pretraživanja, jer je tu reč o glavnom sadržaju stranice.

<nav> ili ne, pitanje je sad?

Ranije sam patio od navitisa – poriva da sve veze ka drugim delovima veb lokacije obuhvatim elementom **<nav>**.

Izlečio sam se razmatrajući ko će imati koristi od upotrebe elementa **<nav>**. Prethodno smo govorili o prečici koja bi omogućila korisniku tehnologije za lakši pristup da direktno pređe na navigacioni meni. Ako imamo desetine elemenata **<nav>**, korisniku će biti teško da nađe najvažnije. Sada zagovaram označavanje samo najvažnijih navigacionih blokova, poput onih za celu lokaciju (odnosno ceo odeljak) ili za tabele sadržaja za dugačke stranice.

Dobro orijentaciono pravilo bi bilo sledeće: element **<nav>** koristite ako možete da zamislite da je iznad veza koje omotavate naslov „Navigacija“. Ukoliko su te veze dovoljno važne da zaslužuju naslov (bez obzira na to da li sadržaj ili dizajn zahtevaju takav naslov), dovoljno su važne i da budu označene elementom **<nav>**.

Kako stoji u specifikaciji: „Nije potrebno da sve grupe veza na stranici budu u elementu **nav** – taj element je prevashodno namenjen odeljcima koji se sastoje od glavnih navigacionih blokova.“

S druge strane, specifikacija sugerise da se veze koje se odnose na „zakonsku regulativu“ (autorska prava, kontakt, slobodan pristup informacijama, pravila privatnosti itd.) i koje se često smeštaju u podnožje stranice, ne obuhvataju elementom **<nav>**: „Uobičajeno je da se u podnožju nađe kratka lista veza ka različitim stranama veb lokacije, kao što je strana sa uslovima korišćenja, početna strana i strana o autorskim pravima. Za takve veze je dovoljan element **footer**; može se upotrebiti i element **nav** ali je to obično nepotrebno.“

Zanemarite specifikaciju i koristite element **<nav>** u tim slučajevima. Na mnogim lokacijama postoje i veze ka informacijama o pristupačnosti (engl. *accessibility*), gde je objašnjeno kako, na primer, zatražiti sadržaj u drugim formatima. Takve informacije su najpotrebnije ljudima kojima bi najviše odgovarali korisnički agenti (engl. *user agents*) sposobni da ih prebace direktno na elemente označene sa **<nav>**.

Kao i u slučaju elemenata **<header>** i **<footer>** (i svih ostalih novih elemenata), niste ograničeni na samo jedan element **<nav>** po strani. Možete u zaglavlju da imate **<nav>** za globalnu navigaciju, zatim **<nav>** za tabelu sa sadržajem tekućeg članka a ispod toga **<nav>** za veze ka drugim srodnim člancima na vašoj veb lokaciji.

Sadržaj elementa **<nav>** verovatno će biti lista veza označena kao neuređena lista, engl. *unordered list*, (što je postalo tradicija od značajnog teksta Marka Njuhausa „CSS Design: Taming Lists“ (<http://www.alistapart.com/articles/taminglists/>) ili – u slučaju menija posećenih mesta (engl. *breadcrumb trails*) – kao uređenu listu (engl. *ordered list*). Element **<nav>** je omotač (engl. *wrapper*), i ne menja elemente **** ili **** već ih omotava. Na taj način, zastareli čitači veba koji ne razumeju element videće samo listu i stavke liste i ponašaće se kako treba.

Iako ima smisla koristiti listu (što otvara više mogućnosti za upotrebu CSS-a), to nije obavezno. I ovo je sasvim ispravno:

```
<nav>
  <p><a href="/">Home</a></p>
  <p><a href="/about">About</a></p>
</nav>
```

Možete da dodate i naslove za navigaciju:

```
<nav>
  <h2>Pages</h2>
  <ul>
    <li><a href="/about">About me</a></li>
    <li><a href="/news">News</a></li>
  </ul>

  <h2>Categories</h2>
  <ul>
    <li><a href="/happy">Happy Pirates</a></li>
    <li><a href="/angry">Angry Pirates</a></li>
  </ul>
</nav>
```

Grupisanje elemenata <nav> i drugih elemenata u bočni stubac

Mnoge veb lokacije imaju bočni stubac (engl. *sidebar*) s više navigacionih blokova i sadržaja drugog tipa. Uzmimo kao primer moju ličnu lokaciju, www.brucelawson.co.uk (slika 1.8).

SLIKA 1.8 Bočni stubac mog bloga koji je (nekad) obuhvatao navigaciju, kolofon i slike zgodnih tipova.



Bočni stubac levo od glavnog sadržaja ima jednu navigacionu oblast s podlistama za stranice, kategorije, arhive i najnovije komentare. U prvom izdanju ove knjige preporučio sam da se takav sadržaj označi kao niz uzastopnih elemenata `<nav>`; sada sam se predomislio i omotao podliste jednim elementom `<nav>`. (Ako imate dva ili više važnih navigacionih blokova koji ne slede jedan za drugim, svakako ih označite zasebnim elementima `<nav>`.)

NAPOMENA Pre nego što odbacite ovu knjigu zgađeni mojom nedoslednošću, naglašavam da retko postoji samo jedan pravi način za označavanje sadržaja. HTML je opšti jezik i ne sadrži milion elemenata koji bi pokrili sve mogućnosti (samo se takvim ponekad čini)!

Sva glavna navigacija za moju veb lokaciju sadržana je u elementu `<aside>` koji se „može koristiti za tipografske efekte poput citata ili bočnih stubaca, za oglašavanje/reklamiranje, za grupe elemenata `nav`, i za druge sadržaje koji se smatraju razdvojenim od glavnog sadržaja stranice“ (<http://dev.w3.org/html5/spec/semantics.html#the-aside-element>).

```
<aside>
  <nav>
    <h2>Pages</h2>
    <ul> .. </ul>
    <h2>Categories</h2>
    <ul> .. </ul>

    <h2>Recent comments</h2>
    <ul> ... </ul>
  </nav>

  <section>
    <h2>blah blah</h2>
    <a href="#">Web hosting by LovelyHost</a>
    
    <p>Powered by <a href="#">WordPress</a></p>
    <p><a href="#">Entries (RSS)</a> and <a href="#">
      ↳ Comments (RSS)</a></p>
  </section>
</aside>
```

Primetićete da odeljak „blah blah“ nije označen sa `<nav>`, jer mi se veza ka mom davacu veb usluga, moja slika i dve RSS veze ne čine kao „odeljak koji se sastoji od glavnih navigacionih blokova“ – kako specifikacija definiše `<nav>`. Umesto toga, odeljak je omotan elementom `<section>` pa naslovi bočnog stupca ostaju na istom nivou u algoritmu za strukturiranje (više informacija o tome u poglavlju 2).

<footer>

<footer> je u specifikaciji definisan kao element koji predstavlja „podnožje za sadržaj sekcionisanja najbližeg pretka ili elementa korena sekcionisanja.“ („Sadržaj sekcionisanja“, engl. *sectioning content*, obuhvata elemente `article`, `aside`, `nav` i `section`, dok su „elementi korena sekcionisanja“, engl. *sectioning root elements*, `blockquote`, `body`, `details`, `fieldset`, `figure` itd.)

Skrećemo pažnju na sledeće: kao u slučaju elementa `header`, na stranici može biti više elemenata `footer` – tome ćemo se vratiti u poglavlju 2. Za sada imamo samo jedan element `footer` na stranici koji je dete elementa `body`. U specifikaciji piše: „Kada je element `body` sadržaj sekcionisanja najbližeg pretka ili element korena sekcionisanja, onda se odnosi na celu stranicu.“

Dalje piše: „Podnožje obično sadrži informacije o svom odeljku (sekciji) kao što su ime autora, veze ka srodnim dokumentima, podaci o autorskim pravima i slično.“

Naš element `footer` sadrži podatke o autorskim pravima koje smo omotali i elementom `<small>`. Definicija elementa `<small>` je izmenjena u jeziku HTML5; prethodno je to bio prezentacioni element, ali sada ima semantiku: predstavlja dopunske komentare ili tekst napisan sitnim slovima i „obično služi za označavanje pravnih ograda, upozorenja, zakonskih ograničenja ili autorskih prava. Sitna slova se ponekad koriste i za navođenje izvora ili za zadovoljavanje zahteva licence.“

Podnožje vaše veb lokacije verovatno sadrži više od napomene o autorskim pravima. Možda su tu i veze ka pravilima o zaštiti privatnosti, informacijama o dostupnosti (zašto to krijete?) i druge slične veze. Predlažem da ih omotate elementom `<nav>` iako specifikacija kaže drugačije (pogledajte prethodni odeljak o elementu `<nav>`).

U specifikaciji se navodi da „pojedine veb lokacije imaju nešto što se ponekad naziva 'debelim podnožijma' (engl. *fat footers*). Reč je o podnožijima koja sadrže mnogo materijala, uključujući slike, veze ka drugim člancima, ka stranicama za slanje povratnih informacija, specijalne ponude... moglo bi se reći, kompletna 'glavna strana' u podnožju.“ Specifikacija vas savetuje da omotate informacije elementom `<nav>` unutar elementa `<footer>`.

Ako ste u iskušenju da koristite „debelo podnožje“, razmislite da li takvim vezama uopšte treba element `<nav>` – navitis se teško leći. Zapitajte se i da li je takvim vezama mesto u elementu `<footer>`; možda bi bilo bolje smestiti ih u bratski element elementa `<footer>`, element `<aside>` cele stranice?

<article>

Glavni sadržaj početne strane ovog bloga sadrži nekoliko priloga. Svaki smo omotali elementom `<article>`. Specifikacija za element `<article>` kaže: „Samodovoljna kompozicija u dokumentu, na stranici, u aplikaciji ili na veb lokaciji, što u principu znači da se nezavisno distribuira ili koristi. To može biti prilog na forumu, članak u časopisu ili u novinama, prilog na blogu, korisnički komentar, interaktivne mini-aplikacije ili bilo kakva druga stavka sa sadržajem.“

Prilog na blogu (engl. *blog post*), uputstvo, novinarska priča, strip ili video s transkriptom – sve se to savršeno uklapa u ovakvu definiciju. Manje intuitivno je zaključiti da se ova definicija odnosi i na pojedinačne poruke u veb programu za elektronsku poštu, mape i ponovno upotrebljive mini-aplikacije (engl. *widgets, gadgets*) za veb. Primetićete da je, kao u slučaju elementa `<nav>`, naslov deo samog članka, pa se umeće u element. Prema tome, ovakav kôd je neispravan:

```
<h1>My article</h1>
<article>
  <p>Blah blah</p>
</article>
```

Treba da glasi:

```
<article>
  <h1>My article</h1>
  <p>Blah blah</p>
</article>
```

Element `<article>` ima još mnogo zanimljivih aspekata a razmotićemo ih u sledećem poglavlju.

U čemu je stvar?

Moj premudri prijatelj Robin Berdžon napisao je: „Gotovo svako u poslu programiranja za veb slaže se s tim da je 'semantika super i pribavlja kolačiće', i to je verovatno tačno. Ali kada zagrebete malo dublje, postaje jasno da veoma malo ljudi može formulisati razlog za to.

„Uopšten odgovor je da 'menja namenu sadržaja.' Na prvi pogled, to je dobro objašnjenje, ali vrlo brzo ćete se upitati u šta to menja namenu sadržaja? Na primer, ako želite da se stranice prikazuju na malom ekranu (oblik menjanja namene), onda vam element `<nav>` ili `<footer>` govore da ti delovi nisu sadržaj i da se mogu odvojiti; ali ako vas zanimaju neka pravna pitanja, kopanje po elementu `<footer>` pomoću nekih heuristika neće koristiti...

„Mislim da bi HTML trebalo da doda samo elemente koji ili izražavaju funkcionalnost koja bi inače bila besmislena (na primer, `<canvas>`) ili obezbeđuju semantiku koja doprinosi promeni namene sadržaja, ali sa aspekta pregledanja veb lokacija.“ www.alistapart.com/comments/semanticsinhtml5?page=2#12

Kao što Robin sugeriše, uređaji s malim ekranom mogli bi da odseku oblasti bez sadržaja (ili da uvećaju oblasti s glavnim sadržajima). Dovoljan je dodir da se fokus premesti na element `nav`, `footer` ili `header`. Pretraživač bi veze u podnožju mogao da smatra manje važnim od veza na traci za navigaciju (engl. *navigation bar*). Biće mnogo primena u budućnosti koje ne možemo da naslutimo – ali sve zavise od nedvosmislenog pridruživanja značenja sadržaju, što je definicija semantičkog označavanja.

Sažetak

U ovom poglavlju predstavili smo HTML5 i DOCTYPE. Strukturirali smo glavne orijentire veb stranice pomoću elemenata `<header>`, `<footer>`, `<nav>`, `<aside>` i `<article>`, otkrivši vam korisničke agente s više semantike od besmislenog generičkog elementa `<div>` koji nije imao alternativu u jeziku HTML 4, i stilizovali smo nove elemente pomoću magije CSS-a.

Ukazali smo vam na fleksibilna sintakсна pravila kao što su ravnopravno korišćenje malih i velikih slova, citiranje i minimizacija atributa, izostavljanje podrazumevanih elemenata poput `head/body`, izostavljanje standardnih stvari poput `type="text/javascript"` i `type="text/css"` za oznake `<script>` i `<style>` i pokazali smo vam kako da ukrotite zver starih verzija Internet Explorera. Nije loše za jedno poglavlje, zar ne?