

HTML5

U ovoj jedinici ćemo upoznati detalje HTML5 jezika i uočiti razlike u odnosu na prethodne verzije.

Novi standard

HTML5 je sledeća generacija HTML jezika. Zameniće starije verzije HTML-a, kao i xHTML-a koje se trenutno i dalje koriste.

Razvoj još uvek traje i finalna 5.0 verzija se očekuje u 2014. godini (ovo treba uzeti sa rezervom jer je podložno čestim promenama). Ipak, i pored toga što nije završen, veći deo se već sada može koristiti.

HTML5 donosi nove mogućnosti u radu, ali i standardizuje mnoge postojeće funkcije, koje se uveliko koriste i koje su već godinama prisutne, ali nisu nikad zvanično dokumentovane, tj. nisu našle put do pisanih standarda u prethodnim verzijama.

Kao i prethodnici, osmišljen je kao *cross-platform* i *cross-browser* solucija. Rad na svim sistemima i svim browserima jeste cilj. Ipak, za sada, podrška u browserima se znatno razlikuje, dok neke starije verzije određenih browsera (kao na primer Internet Explorer 8 i stariji) uopšte i ne prepoznaju HTML5 elemente bez pomoći sa strane.

Šta je u suštini HTML5?

Ovo može zvučati kao čudno pitanje, ali je primetna tendencija da se skoro sve nove tehnologije vezane za web dizajn i web programiranje (i uopšteno za web), svode pod kapu HTML verzije 5. Na primer, SVG (Scalable Vector Graphics) se pominje kao deo HTML5 iako je to nezavisna tehnologija stara više godina i uvedena u W3C standarde kao takva.

I sami W3C i WHATWG doprinose konfuziji zbog protivrečnih ili čak kontradiktornih izjava i objavama dokumenata u poslednje vreme.

Ipak, ono što je nepobitno je da će HTML5 sa svim svojim tehnologijama i mogućnostima doprineti boljem i naprednijem webu i da je on sledeći evolutivni korak razvoja.

Kako je nastao HTML5

1998. godine W3C je odlučio da obustavi dalji razvoj HTML jezika i svi resursi su prebačeni na razvoj nove verzije xHTML-a. Negde u to vreme, xHTML je postao de-facto standard za nove stranice i većina profesionalaca (barem onih na odlučujućim pozicijama) je verovala da je budućnost weba u XML standardima, odnosno u xHTML jeziku.

Nešto kasnije, počeo je razvoj xHTML-a verzije 2 koji je trebalo da napravi revolucionarni skok. Taj jezik, odnosno verzija je u startu imala je problem jer je predviđeno da ne bude

kompatibilna sa prethodnom, XHTML 1 verzijom (koja je i dalje u upotrebi), a sve zbog logičnijeg pristupa pisanju koda i boljem osnovnom dizajnu jezika. Vladalo je uverenje da će autori web sadržaja lako preći na nove standarde, odnosno nov način pisanja i rada sa kodom, ali u stvarnosti nije bilo baš tako, kao što je vreme pokazalo.

Uporedo sa „zvaničnim“ razvojem, mala grupa eksperata iz tima Opera browsera krenula je sa radom na novom standardu pod nazivom **Web Forms 2.0**, koji bi zadržao kompatibilnost sa starijim verzijama HTML i XHTML jezika. Ta pravila i ta specifikacija je kasnije postala osnova za HTML5.

U određenom trenutku razvoja, ime standarda je promenjeno u **Web Applications**. Ubrzo su im se priključili drugi koji su verovali u taj alternativni pravac. Tu su se našli i eksperti iz organizacija kao što su Apple, Mozilla i mnogih drugih. I danas se dokaz te saradnje više različitih kompanija može videti u live specifikaciji HTML-a¹: „Copyright 2004-2011 Apple Computer Inc, Mozilla Foundation, and Opera Software ASA.“

U 2006. godini W3C je ipak priznao da je prenaglio sa odlukom da razvija XHTML 2 i sa nametnutom primenom XML-a. Tim Berners Li je tada napisao: „Potrebno je da HTML evoluiramo postepeno. Nije uspeo pokušaj da se nateraju svi da odjednom pređu na XML, uključujući i korišćenje znaka navoda oko vrednosti u atributima i slash karaktera u praznim tagovima i namespace elementima“. W3C je tada napravio zaokret za 180°. Obnovljena je radna grupa za HTML (*HTML Working Group*) koja je predložila da gorepomenuta *WHATWG Web Applications* specifikacija postane osnova za buduću, narednu verziju HTML jezika. Razvoj XHTML 2 je postepeno izgubio na značaju i kompletno je obustavljen 2009, kada je postalo očigledno da je HTML5 „dobio bitku“ nad XHTML 2.

Ono što je zanimljivo je da su ostale dve posebne specifikacije HTML5 jezika. Jedna koju izdaje W3C (www.w3.org/TR/html5/), kao i druga, koju izdaje WHATWG grupa (<http://whatwg.org/html5>). Tu se konfuzija ne završava. I u okviru samih W3C i WHATWG grupa postoje i prihvaćene, ali i verzije u razvoju, kao i neki izdvojeni delovi (kao npr *geolokacija* koja nije našla mesto u WHATWG specifikaciji).

Da bi se još zakomplikovalo, pojavila se i XHTML5 verzija koja bi se upotrebljavala samo ukoliko je neophodno. Može se napraviti paralela između XHTML1 - HTML4 i XHTML5 - HTML5. Mi ćemo se koncentrisati samo na HTML5.

Pored svog tog konfuznog i trnovitog razvoja HTML5, dobili smo sada veoma napredan jezik i korisne funkcije koje su mahom potekle od samih korisnika i autora.

Filozofija koja stoji iza razvoja HTML5 (barem po W3C-u) se može naći na <http://www.w3.org/TR/html-design-principles/>. Sam proces nastanka HTML5 jezika sadrži i mnoge druge pojedinosti i detalje, kojima se nećemo baviti u ovom kursu. Upoznaćemo se sa osnovama i detaljima koji su nama bitni tokom pisanja koda i kreiranja sadržaja. Ukoliko Vas zanima više o HTML5, preporučujem odličnu knjigu Marka Pilgrima pod nazivom „Dive into HTML5“ koja se može naći i besplatno online na sledećoj adresi (na engleskom): <http://diveinto.html5doctor.com/>

Šta donosi HTML5

HTML5 donosi nove mogućnosti od kojih ćemo se baviti najvažnijima:

¹ <http://www.whatwg.org/specs/web-apps/current-work/> (strana je podložna promenama na dnevnom nivou)

- Tu su novi semantički elementi: <header>, <footer>, <section>...
- Canvas, dvodimenzionalna površina za crtanje koju kontroliramo JavaScriptom.
- Video koji se može embedovati (ugraditi) bez upotrebe plugin-ova.
- Geolocation (geolokacija), opcija koja omogućava da korisnici podele informaciju o svojoj lokaciji sa stranicom.
- Local storage, odnosno lokalno skladištenje podataka na računaru korisnika.
- Novi elementi i unapređenje HTML formi, kao i mnogi drugi koji se tek pojavljuju.

Osnovna struktura

Za razliku od ranijih verzija HTML i XHTML jezika, HTML5 je doneo dosta pojednostavljen doctype. Dovoljno je pisati:

```
<!doctype html>
```

Ukoliko uporedimo sa ranijim verzijama, jasno je da će svi autori moći da zapamte novi doctype i da nema potrebe za njegovim kopiranjem sa W3C sajta ili drugih lokacija. Ideja je maksimalno uprošćavanje uz održavanje standarda. Uočite nedostatak napomene o verziji HTML jezika.

Posle Doctype-a potrebno je postaviti i meta tag za encoding (pominjali smo ranije u 3. modulu). U slučaju naše HTML5 varijante, postavimo:

```
<!doctype html>
<meta charset=utf-8>
```

Opet, upoređujući sa prethodnim verzijama, vidimo određene promene. Ranije smo pisali: `<meta http-equiv="Content-Type" content="text/html" charset=utf-8">` a sada je i to uprošćeno. Može se koristiti i dalje ovaj duži način, ali i kraći sasvim regularno funkcioniše.

Pored toga, primećujemo u novoj verziji i drugačije pisanje samozatvarajućih tagova, vraćamo se na HTML pravila i kosa crta je nepotrebna na kraju (samozatvarajućih) tagova, mada nije greška pisati ako želite. Dalje, i sami atributi nad tagovima više ne zahtevaju znake navoda, ali opet, možete ih koristiti ako želite.

Ako bismo nastavili dalje i upisali još neke elemente, možemo dobiti ovakvu stranicu:

```
<!doctype html>
<meta charset=utf-8>
<title>HTML5 Hello world!</title>
<p>HTML5 test</p>
```

Ono što će Vas možda iznenaditi ili čak zaprepasiti je to da je ovaj kôd sasvim regularna i validna HTML5 stranica.

Ali ovde nedostaju osnovni tagovi, HTML, HEAD i BODY? Da, ovde smo izostavili te tagove jer, verovali ili ne, oni su *opcion*i u HTML5. Dakle, nismo u obavezi da ih postavljamo. Sami browseri prepoznaju delove stranice i „u letu“ dodaju osnovne tagove prilikom prikaza. Dakle, HTML, HEAD i BODY su i dalje potrebni, ali ih ne moramo mi postavljati, uradiće to browser za nas.

Ipak, svakako preporučujem da uvek postavljate pomenute tagove kako bi održali kompatibilnost sa starijim browserima, koje bi njihov nedostatak zbunio. Dakle, ako ih postavimo ne mogu da škode, a to znači da koriste. Sa tim na umu, naša HTML5 stranica, odnosno njen kôd bi izgledao:

```
<!doctype html>
<html>
  <head>
    <meta charset=utf-8>
    <title>HTML5 Hello world!</title>
  </head>
  <body>
    <p>HTML5 test</p>
  </body>
</html>
```

Zaključujemo, dakle, da se kod može umnogome pisati dosta kraće i lakše u HTML5, ali isto tako, regularno možemo koristiti i stariji, duži način. HTML5 će se snaći u obe varijante.

Novi semantički elementi

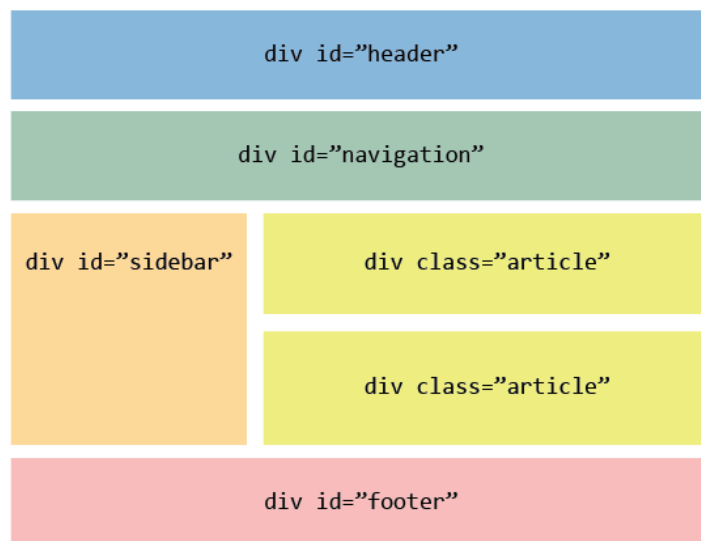
Kao što već znamo, div elementi su osnovni gradivni delovi praktično svih sajtova. Radi raspoređivanja u layout, koristimo razne ID i CLASS vrednosti nad njima. Pogledajmo primer jednog tipičnog sajta, odnosno njegov osnovni deo (u bodyju):

```
<div id="header">
  ...
</div>
<div id="navigation">
  ...
</div>
<div id="sidebar">
  ...
</div>
<div class="article">
  ...
</div>
<div class="article">
  ...
</div>
<div id="footer">
  ...
</div>
```

Dok bi CSS mogao biti:

```
#header {...}
#navigation {...}
#sidebar {float:left; width:20%;}
.article {float:right; width:80%;}
#footer {clear:both;}
```

A sam prikaz u browseru bi imao raspored kao na sledećoj skici:



„Klasičan" layout upotrebom div elemenata

Ovo je sasvim dobar pristup i čak i u HTML5 varijanti bi bilo regularno postaviti ovakvu stranicu.

Ipak, vremenom je primećeno da se određene ID i CLASS vrednosti stalno ponavljaju na velikom broju sajtova. Na primer, veliki procenat sajtova sadrži `div` koji ima ID vrednost „footer”. Urađena su neka istraživanja^{2 3} o upotrebi ID i CLASS vrednosti, a koja su obuhvatila veliki broj sajtova, i pokazala su sledeće rezultate:

CLASS		ID	
popularnost	naziv	popularnost	naziv
1	footer	1	footer
2	menu	2	content
3	style1	3	header
4	msonormal	4	logo
5	text	5	container
6	content	6	main
7	title	7	table1
8	style2	8	menu
9	header	9	layer1
10	copyright	10	autonumber1
11	button	11	search
12	main	12	nav
13	style3	13	wrapper
14	small	14	top
15	nav	15	table2
16	clear	16	layer2

² <https://developers.google.com/webmasters/state-of-the-web/2005/classes>

³ <http://devfiles.myopera.com/articles/572/idlist-url.htm>

17	search	17	sidebar
18	style4	18	image1
19	logo	19	banner
20	body	20	navigation

Kada su se otklonile „smetnje” poput *msonormal* klase koja ostaje greškom u HTML stranicama prilikom kopiranja iz MS Word aplikacije ili *style1*, *autonumber1* i ostalih tog tipa (koje su rezultat lenjosti ili lošeg rada autora), dobijena je prilično jasna slika o tome šta autori najčešće koriste.

Vodeći se ovim i mnogim drugim istraživanjima, kreirani su novi strukturalni elementi semantičke prirode. Umesto da koristimo generički div sa ID ili CLASS, nastali su specifično krojeni elementi (tagovi) za određene namene. Na primer, nastali su <header>, <footer>, <nav> i mnogi drugi. Shodno tome, umesto da koristimo:

```
<div id="footer">
...
</div>
```

Možemo pisati jednostavnije:

```
<footer>
...
</footer>
```

U novoj HTML5 varijanti, ne samo da je kod kraći i pregledniji, već i obezbeđuje semantičko značenje elemenata, a to znači da browser i drugi alati mogu prepoznati namenu elementa.

Ukoliko bi se vratili na naš primer strane, ta ista stranica bi mogla biti u HTML5 zapisana:

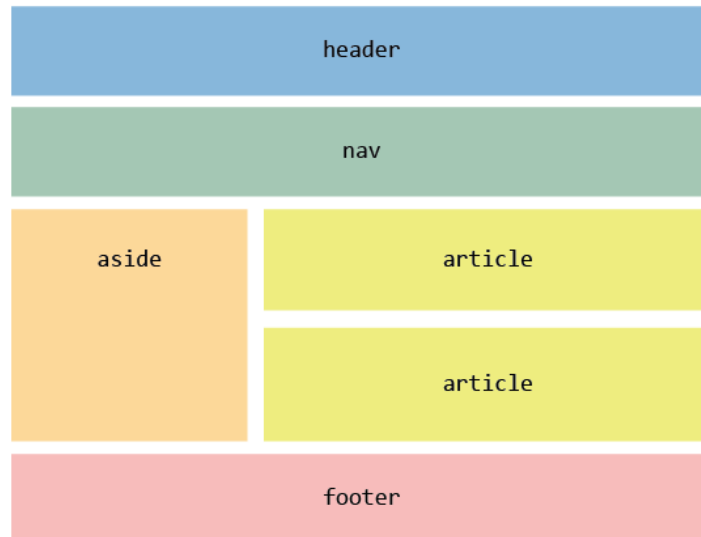
```
<header>
...
</header>
<nav>
...
</nav>
<aside>
...
</aside>
<article>
...
</article>
<article>
...
</article>
<article>
...
</article>
<footer>
...
</footer>
```

I u CSS-u:

```
header {...}
nav {...}
```

```
aside {float:left; width:20%;}  
article {float:right; width:80%;}  
footer {clear:both;}
```

i u browseru:



Layout postavljen novim HTML5 elementima

Vidimo da je kod uprošćen, nemamo ID i CLASS vrednosti, mada ih regularno možemo koristiti. Primećujemo da i u CSS kodu selektori ne ukazuju na ID vrednosti (nema # znaka), već direktno na elemente.

Ipak, i novi elementi podržavaju ID i CLASS kako bi ih dodatno distancirali ako je to potrebno, tako da možemo, na primer, postavljati:

```
<article id="x">  
...  
</article>  
<article id="y">  
...  
</article>
```

Pored svega navedenog, da ne bi bilo zabune sad u početku, razjasnićemo neke osnovne detalje:

- CSS ostaje veoma značajan. Sami elementi ne podrazumevaju float niti bilo šta slično. Ponašaju se, što se tiče pozicioniranja, isto kao klasični div.
- Div elementi se i dalje mogu regularno koristiti i nikako nije cilj da se div kompletno ukloni, već da ostane za sve elemente koji ne spadaju pod novouvedene.

Semantika i novi elementi

Novi HTML5 elementi ne samo da skraćuju kôd i njegovo pisanje, već i doprinose semantici. Ukoliko se fokusiramo samo na vizuelni aspekt, svi ti novi elementi se ponašaju kao div-ovi i nema nikakve razlike. Ali sa druge strane, oni doprinose semantici strane, odnosno definišu svrhu i namenu delova strane. Browseri, roboti pretraživača (Google, Bing..) i razni drugi softveri mogu ih analizirati i shodno nameni ih drugačije tretirati.

Najčešći novi gradivni elementi su:

- **<section>** element postavlja deo stranice, odnosno dokumenta. Može obuhvatiti više istih elemenata (npr <article>) koji imaju sličnu temu ili namenu. Može se koristiti i za grupisanje drugih elemenata u celine. Od novih elemenata, neki autori ga smatraju najbližim starijem divu.
- **<article>** element se ponaša kao okvir za bilo koji deo strane koji može stajati nezavisno od ostatka sadržaja. Namenjen je za članke, blog unose, forum postove i svaki drugi nezavistan deo sadržaja. Mogu se ređati jedan za drugim, a čak mogu biti i jedan u drugom (ukoliko se npr. pojavljuju komentari za određeni post koji je article, i sami komentari mogu biti article). Često je u početku zbunjujuća razlika između <section> i <article>.
- **<nav>** element se koristi za unos i postavljanje osnovnih navigacionih elemenata stranice (sajta). Još uvek traje polemika i rasprava da li treba koristiti nav za sve pozicije sa linkovima na stranici ili samo za glavnu navigaciju. Neko (uglavnom) prihvaćeno pravilo je da treba nav koristiti za veće blokove navigacije. Na primer, ako u footeru postavimo samo linkove za pravila korišćenja, pravne informacije i početnu stranu, to ne treba postavljati u nav. Ali ako u istom footeru imamo kompletno ponovljene glavne linkove, glavnu navigaciju sajta, onda je prikladno upotrebiti nav.
- **<aside>** element ima dve namene, u zavisnosti da li je unutar <article> elementa ili ne:
 - Ukoliko je unutar <article> elementa, sadrži informacije koje su povezane sa njim, ali nisu od presudne važnosti. Na primer, citati uz tekst i slično.
 - Ukoliko je van <article> elementa, odnosi se na celu stranu. Često se koristi kao glavni sidebar. Tu se mogu naći reklame, forme za kontakt, linkovi ka drugim preporučenim stranama i slično.
- **<hgroup>** predstavlja zaglavlje sekcije. Koristi se za grupisanje više h1-h6 naslova u celinu. Ne treba postavljati ništa osim ovih tagova unutar njega. Na primer, koristi se ako imamo glavni naslov i podnaslov uz njega (h1 i h2). Oko ovog elementa se pojavljuju neke kontraverze, i neki autori osporavaju primenu. Čak je bio i uklonjen iz predloga HTML5 jezika neko vreme, ali je vraćen.
- **<header>** element se koristi za definisanje individualnog headera određenog dela/elementa stranice (npr <article> ili <section>) ili mnogo češće za definisanje glavnog headera strane. Ukoliko je unutar dela strane, odvaja zaglavlje tog dela i u njemu naslov, odnosno <hgroup> ukoliko je potrebno i sl. Ukoliko je direktno na stranici, postavlja header sajta koji sadrži logo, pretragu, navigaciju i sl.
- **<footer>** je sličan headeru, samo sa drugačijom namenom. Dok header postavlja zaglavlje, footer postavlja završni deo elementa ili strane. Ukoliko je unutar dela strane, može sadržati ime autora teksta, povezane stranice i sl. Ukoliko je direktno na stranici, može sadržati razne elemente koji su odmaćeni u dnu sajtova (pravne informacije, linkove ka mapi sajta i pravilima korišćenja, ponovljenu navigaciju i sl).
- **<figure>** ima osnovnu namenu da uokviri dodatni sadržaj vezan za tekst (deo strane). Na primer, za postavljanje slika, video materijala, grafikona i slično. Unutar njega, može se postaviti i **<figcaption>** pored samog sadržaja, kao njegov opis.

Često postavljano pitanje koje se može čuti u poslednje vreme je: „Kada i u kojoj situaciji koristiti jedan ili drugi HTML5 element, i da li ih uopšte koristiti ili se vratiti na dobro poznati DIV?“ Možda Vam za sada sve ovo deluje konfuzno jer su neki elementi slični (npr aside i figure), ali svaki od njih ima svoju namenu.

U video materijalu uz ovu lekciju ćemo pojasniti detaljnije ove elemente.

Kao dodatni materijal, uz ovu jedinicu priložen je odličan grafikon koji je preuzet sa sajta <http://html5doctor.com> i koji Vam može pomoći da odredite koji je element potrebno, ali i poželjno upotrebiti.

Prepoznavanje elemenata

Iako je HTML5 budućnost weba, veliki broj korisnika još uvek koristi starije browsere koji ne podržavaju nove tehnike. Možda se sada pitate: „Kako da koristimo HTML5 pošto ga stariji browseri koji su još u upotrebi ne podržavaju?“ Samo pitanje je već pogrešno, jer smo već pomenuli da HTML5 nije jedna, objedinjena tehnika. Umesto toga, **HTML5 je kombinacija, skup više različitih, individualnih tehnika i funkcija**. Ne možemo govoriti o *podršci za HTML5*, već o podršci za pojedine mogućnosti, kao što su video, canvas, geolokacija i tako dalje. Ipak, bez obzira na tumačenje, ostaje problem podrške u browserima kao što su Internet Explorer 7 i 8, Firefox 2 i sl.

Kada browser prikazuje HTML stranicu, on konstruiše *Document Object Model* ili skraćeno DOM. DOM je kolekcija objekata koji predstavljaju HTML elemente na stranici. Svaki element, svaki <p>, <div> ili je predstavljen u DOM-u kao poseban objekat (postoje i globalni objekti, kao što su *Window* i *Document*). Svi DOM objekti dele određena pravila, ali se i razlikuju međusobno. Ukoliko browser podržava HTML5 pravila, HTML5 elementi će biti prepoznati i posedovaće potrebne osobine.

Svi ovi novi layout elementi su u osnovi block tipa, kao i njihov najbliži rođak, dobro poznati div. Noviji browseri i njihovi DOM-ovi prepoznaju te elemente, postavljaju ih kao block i dopuštaju CSS stilizaciju nad njima. Sa druge strane, stariji browseri ih ne prepoznaju. Ukoliko novi element nije prepoznat, on se tretira po default vrednostima, postaje inline tip i nemoguće ga je stilizovati i u principu koristiti ga na bilo koji valjan način.

Jedno od rešenja je upotreba **HTML5shiv** skripte. Može se preuzeti sa sledeće adrese: <http://code.google.com/p/html5shiv/>. Ova skripta je jednostavno JavaScript rešenje koje primorava starije IE browsere (pre verzije 9) da prepoznaju osnovne gradivne elemente iz HTML5.

Dovoljno je učitati HTML5shiv JS fajl i automatski će biti inicijalizovan. Važno je pozvati ga unutar HEAD dela strane, pre samog bodyja, kako bi bio učitao pre nego što IE krene sa obrađivanjem sadržaja.

```
<script src="html5shiv.js"></script>
```

Može biti pre ili posle CSS-a, ali je radi bržeg rada preporuka da ide posle.

Iako je gornji red kôda dovoljan i regularno će funkcionisati, on ima jednu manu, a to je što će se učitavati za sve browsere, pa čak i za one koji podržavaju HTML5. Da bismo izbegli

nepotrebno opterećivanje stranica, možemo postaviti kondicionalni komentar i ceo kod za HTML5shiv bi izgledao ovako:

```
<!--[if lt IE 9]>
  <script src="html5shiv.js"></script>
<![endif]-->
```

Sada će se ovaj js fajl pozvati samo ako je u pitanju InternetExplorer i to samo ako se radi o verziji pre 9. Ostali browseri će ga jednostavno preskočiti.

Modernizr

Modernizr (nije greška u kucanju, namerno nedostaje slovo e) je JavaScript rešenje koje problemu browsera pristupa na malo drugačiji način. Može se naći na zvaničnoj stranici: <http://modernizr.com/>

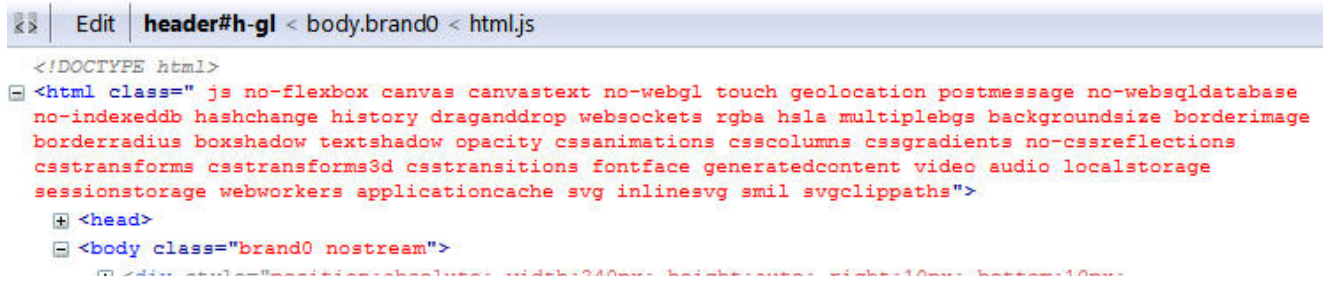
Prilikom prikaza stranica, Modernizr testira i prepoznaje koji su to elementi podržani od strane browsera i u trenutku prikaza menja kôd, tj. postavlja klase (class vrednosti) nad HTML tagom stranice. Osim toga, od verzije 1.5 ka novijima, uključuje funkcionalnost HTML5shiv-a, tako da njega možemo izostaviti ukoliko koristimo Modernizr.

Pogledajmo na primeru, kako bi bilo jasnije:

```
1 <!DOCTYPE HTML>
2 <!--[if lt IE 7 ]> <html class="ie6"> <![endif]-->
3 <!--[if IE 7 ]> <html class="ie7"> <![endif]-->
4 <!--[if IE 8 ]> <html class="ie8"> <![endif]-->
5 <!--[if IE 9 ]> <html class="ie9"> <![endif]-->
6 <!--[if (gt IE 9) !(IE)]><!--> <html class=""> <!--<![endif]-->
7 <head>
8   <meta charset="utf-8">
9   <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
10
11   <title>LINKeduTV</title>
12   <meta name="description" content="">
13   <meta name="viewport" content="width=device-width; initial-scale=1.0; maximum-scale=1.0;" />
14
15   <script src="js/jquery-1.7.1.min.js" type="text/javascript"></script>
16   <script src="js/jquery-ui.min.js" type="text/javascript"></script>
17
18   <script src="js/jquery.flexslider.js"></script>
19   <script src="js/modernizr-2.5.3.min.js"></script>
20   <script src="js/jquery.easing.1.3.js" type="text/javascript"></script>
21   <script src="js/jquery.mousewheel.min.js" type="text/javascript"></script>
22   <script src="js/jquery.mCustomScrollbar.js" type="text/javascript"></script>
23   <script src="js/customScrollBarMobile.js" type="text/javascript"></script>
```

Prikaz kôda jedne HTML5 stranice

Na slici je prikazan početak kôda jedne HTML5 stranice. Vidimo, među ostalim redovima, i HTML tag prvi vrhu, kao i pozivanje JavaScript fajla Modernizra (oba označena crvenom). Prilikom izvršavanja, prepoznaće se HTML5 mogućnosti browsera, i ispis će biti ovakav:



```
header#h-gl < body.brand0 < html.js
<!DOCTYPE html>
<html class=" js no-flexbox canvas canvastext no-webgl touch geolocation postmessage no-websqldatabase
no-indexeddb hashchange history draganddrop websockets rgba hsla multiplebgs backgroundsize borderimage
borderradius boxshadow textshadow opacity cssanimations csscolumns cssgradients no-cssreflections
csstransforms csstransforms3d csstransitions fontface generatedcontent video audio localstorage
sessionstorage webworkers applicationcache svg inlinesvg smil svgclippaths">
  <head>
  <body class="brand0 nostream">
```

Rezultat upotrebe Modernizra u Firefoxu 15

Iako je ovo ista stranica, odnosno prikazan kôd tokom razvoja i u trenutku prikaza, vidimo da je Modernizr dodao, dopisao veliki broj klasa nad HTML elementom. Podsećam, ukoliko se koristi IE verzije pre 9, Modernizr će srediti i detalje vezano za osnovne elemente.

Ali zašto su ove klase nad HTML tagom bitne? Znamo da klase koristimo za CSS stilizaciju, otkud sve one ovde nad bodyjem? Bitne su jer nam omogućavaju drugačiju stilizaciju kroz CSS za različite browsere (i podrške za HTML5 mogućnosti). Na primer, možemo postaviti:

```
body.boxshadow #nekiElement {
  border:1px solid #ccc;
  box-shadow: 1px 2px 2px 0px #000000;
}

body.no-boxshadow #nekiElement {
  border:3px solid #ccc;
}
```

Prvi opis će biti primenjen nad elementom **#nekiElement** u browserima koji podržavaju *boxshadow*. Ukoliko je suprotna situacija, primeniće se drugi opis. Na ovaj način, stariji browser (koji nema podršku za *boxshadow*) će prikazati samo border, dok će noviji prikazati senku kako treba. Modernizr nam je pomogao da razgraničimo to.

Nove mogućnosti linkovanja

U dosadašnjim verzijama, postavljanje block elemenata unutar `<a>` (link) tagova je smatrano nepravilnim. U HTML5, možemo postavljati link oko celih delova stranica i to na sledeći način:

```
<a href="o-nama.html">
  <article>
    ...
  </article>
</a>
```

Ovako smo ceo `<article>` pretvorili u link. Takva situacija se često sretala na ranijim sajtovima i dovodila je do kršenja pravila jezika (kada je `div` unutar `a`). Sada je to promenjeno i olakšan je rad.

Pitanje u jedinici

Da li je HTML unazad kompatibilan (da li se raniji kod može koristiti u okviru HTML5 dokumenta)?

- **Da.**
- Ne.

Najvažnije iz lekcije

- HTML5 je nadolazeći standard, ali već dobrim delom prihvaćen i u upotrebi.
- HTML5 donosi mnogo novih detalja i elemenata, ali i standardizuje i pojednostavljuje neke ranije detalje.
- HTML5 je evolutivni pomak i sledeći korak u razvoju, nasuprot napuštenom XHTML2 koji je predstavljao revolucionarni korak razvoja.