

Универзитет у Београду, Математички факултет

М А С Т Е Р Р А Д

тема:

**Коришћење специфичности HTML5 и CSS3, као и
JavaScript-а, у изради динамичких презентација**

ментор:

Др Душан Тошић

студент:

**Татјана Јакшић
бр. индекса: 1117/2011**

Београд, децембар 2012. године

1. Увод

1.1. Интернет

Интернет је глобална светска рачуарска мрежа на коју је прикључен огроман број рачунара. Иако се од свог настанка врло брзо проширио на свет трговине и пословања, било је потребно скоро 20 година да се наметне као технолошка иновација која непрекидно трансформише друштво и економију.

Развој Интернета започео је у САД у доба хладног рата 1969. године. Тада се јавила идеја прављења мреже рачунара која би обезбедила комуникацију војних лабораторија, владиних бироа и универзитета на којима су се одвијали бројни пројекти за потребе америчке војске. Развојем мреже руководила је Advanced Research Project Agency под надзором Министарства одбране САД, а мрежа је добила назив ARPANET. Састојала се од малих рачунара који су били повезани модемима. Они су међусобно комуницирали бит-серијском везом брзином од 50kb/s.



Слика 1. Универзитетски рачуарски центри у САД (мрежа ARPANET)

Мрежа је имала конфигурацију од 4 рачунара који су били смештени у универзитетским рачуарским центрима у САД-у.^[1]

Интернету се може приступити било где, бројним средствима, такође и путем мобилних уређаја. Образовни материјал за све нивое доступан је са Интернета. За образовање на даљину, помоћ око домаћих задатака, жељу за усвајањем нових знања, детаља у вези са одређеном темом, Интернет је од велике користи. Приступ образовним информацијама на било ком нивоу са било ког места никада није био лакши.

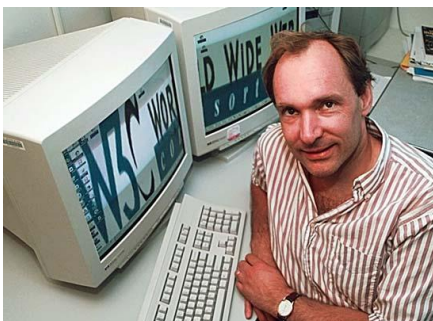
Интернет се код нас интензивније користи почев од 1996. године када је национална академска мрежа прикачена на Интернет преко провајдера ВеoTelNet-а.

1.2. World Wide Web (WWW)

Хипертекст се састоји од докумената који су међусобно повезани означеним појмовима или скупом речи. Када се у тексту помиње нека означена реч, хипертекст нам омогућава да пређемо на други документ који нам даје више информација о тој речи.^[2]

Могућност мултимедија да користе хипертекст посебно добија на значају ако су рачунари повезани на неку глобалну мрежу. За коришћење хипертекста дефинисан је сервис Веб (World Wide Web [WWW]). То је информациони систем базиран на хипертексту.

Британски инжењер и научник сер Тим Бернерс Ли, садашњи директор W3C (World Wide Web Consortium), написао је у марту 1989. предлог о томе шта би обухватао WWW.



Слика 2. сер Тим Бернерс Ли



Слика 3. Мрежни протокол



Слика 4. Ознака за Веб

Претпоставља се да Интернет данас користи више од милијарду људи, при чему их највише има у Северној Америци и на Далеком Истоку. Употребом бежичних технологија, дошло је и до пораста броја корисника и у мање развијеним земљама са лошијом инфраструктуром.

2. HTML

2.1. Историја и развој HTML-а

HTML је скраћеница од енглеских речи HyperText Markup Language. Он је званични језик веб страница познат под именом „хипертекстуални маркерски језик“. HTML није програмски језик, већ маркерски језик, који поставља тагове (ознаке) око текста да би интернет претраживач „знао“ како да на екрану прикаже одређени текст. HTML елементи су основа за прављење веб страница. Сврха веб претраживача јесте да прочита HTML документа и преведе их у видљиве или звучне веб странице.^[3]

Ознаке HTML-а су дате у стреличастим заградама, тј. знацима „мање“ и „веће“. Први таг је стартни таг (таг отварања), а други таг, уколико постоји, јесте завршни таг (таг затварања). Између два тага додаје се садржај који желимо да буде приказан на страници. На пример:

```
<b>Здраво свете! </b>
```

говори претраживачу веба да текст између два тага прикаже подебљано (bold). Постоје и тагови који немају завршни таг и називају се празни елементи. Такав је, на пример, таг ``. Према стандарду XHTML-а овакве ознаке нису коректне, па се морају завршавати са `/>`.

Атрибути представљају специфична својства ознаке. Они дефинишу њихово понашање и изглед. Налазе се само у почетним ознакама.

HTML омогућава додавање фотографија и других објеката и може се користити за креирање интерактивних форми. Омогућено је структурирање докумената додавањем тагова за наслове, листе, линкове, параграфе, цитате итд. Такође, и скриптови написани у другим језицима, рецимо JavaScript-у, могу бити додати у оквир HTML документа и на тај начин утицати на његово понашање.

1989. године физичар Тим Бернерс-Ли из CERN-а (Европске организације за нуклеарно истраживање), специфирао је HTML и написао софтвер за претраживач и сервер. 1991. године на интернету се спомиње први јавно доступан опис HTML-а. Зове се «Ознаке HTML-а». Тај опис се састоји од 20 елемената релативно једноставног дизајна HTML-а. Од тих 20, 13 елемената још увек постоји у оквиру HTML 4.^[2]

Тим Бернерс-Ли напустио је CERN и 1994. основао организацију World Wide Web Consortium, која је позната и као W3C. Она се бави стандардизацијом технологија коришћених на вебу.

Свака нова верзија HTML-а развијана је тако да буде читљива на свим претраживачима веба. Прва верзија објављена је 1993. године. Тада није било могуће додати слике у HTML документе. Прва именована верзија јесте верзија 2.0, али ни она није постала стандард. 1995. године објављена је верзија 3.0. Омогућено је додавање табела. Због различитих веб претраживача дошло је до дуплирања неких ознака, па је тако, рецимо, подебљани текст било могуће обележити са `` или ``.

HTML 4 је представљен децембра 1997. године. Настављено је са прихватањем ознака које су наметали произвођачи различитих веб претраживача. Истовремено, покренуто је и такозвано чишћење стандарда где су се неке ознаке проглашавале сувишним. Коначна верзија, HTML 4.01 представљена је децембра 1999. године.

HTML5 верзија је још увек у развоју.

2.2. Структура HTML документа

Основна структура HTML документа јесте универзална и иста је за све веб стране. Неки основни елементи заједнички за све HTML документе чине основу те структуре. У основне елементе спадају дефиниција типа документа (DTD-Document Type Definiton), главни елемент (заглавље) и тело елемента. Сви HTML документи састоје се из два примарна одељка: заглавља стране и тела. Све оно што корисник види и на шта делује налази се у телу. Одељак са заглављем стране садржи наслов и податке који нису видљиви за кориснике.

У првој линији сваког HTML документа је дефиниција типа документа (DTD). Он саопштава претраживачу да очекује HTML документ и због тога пре DTD-а не сме да постоји празна линија. У ранијим верзијама DTD је био знатно сложенији. У HTML5 верзији DTD се читава на следећи начин:

```
<!DOCTYPE html>
```

Метаознака се састоји од метаподатака, који представљају податке о веб страни. Обележава се са `<meta></meta>`. У HTML5 верзији само се наводи таг отварања и начин кодирања карактера: `<meta charset=utf-8>`.

Након DTD-а следи HTML елемент чија је ознака `<html></html>` и између ова два тага налазе се сви остали елементи HTML датотеке. Глава документа иде иза `<html>` елемента и наводи се на следећи начин: `<head>...</head>`. Ту се исписују информације о самом документу: наслов (који се јавља на насловној траци при врху претраживача) и било који стил или скрипт у документу. После `<head>` секције следи одељак са телом. Све што је видљиво на екрану, поред наслова, налази се у овом делу, између ознака: `<body>` и `</body>`. Коментар се наводи на следећи начин:

```
<! - -Ово је HTML коментар. - ->
```

2.3. HTML5

2.3.1. Историја и стандардизација

HTML5 је језик за структурирање и презентовање садржаја на интернету. То је пета верзија HTML стандарда и још увек је у развоју. Његов основни циљ јесте да побољша језик подршком за најновије мултимедијалне садржаје, а да притом буде разумљив и лако читљив. Направљен је тако да апсорбује не само HTML 4, већ и XHTML 1 и DOM 2 HTML (делимично JavaScript). Већина нових и популарних веб претраживача подржава HTML5, мада не све елементе, нити атрибуте за сада, па стога странице не изгледају исто код свих корисника.

Документи из HTML5 се записују коришћењем синтаксе HTML-а или XHTML-а. Он побољшава и рационализује ознаке за документа и уводи ознаке и апликациони програмски интерфејс за комплексне веб апликације. Због свих ових разлога, HTML5 је потенцијални кандидат за вишеплатформске мобилне апликације. Многе особине HTML5 омогућавају да се овај језик може користити и на уређајима мање снаге: смарт телефонима и таблетима.



Слика 5. Званични лого HTML5, од 1. априла 2011.

Након HTML 4.01. прекинут је развој HTML-а и W3C се оријентисао ка развоју XHTML-а. За разлику од HTML-а који је био флексибилнији и садржао низ непрецизности, XHTML је базиран на XML-у и знатно је строжи.

XHTML (*Extensible HyperText Markup Language*) је развијен да би проширио HTML и да би се повећала интероперабилност са другим форматима података. Разлике HTML-а и XHTML-а су следеће: код XHTML-а документи морају бити добро формиран, елементи морају бити ваљано угњеждени, имена тагова морају бити записана малим словима, сви

елементи морају бити затворени, празни елементи такође морају бити затворени, имена атрибута се пишу малим словима. Атрибути обавезно морају бити под наводницима. XHTML је скоро идентичан са HTML 4.01 и враћа HTML првобитној намени, а то је структурирање веб докумената. XHTML је базиран на XML-у (*Extensible Markup Language*). XHTML 1.0 постао је W3C препорука јануара 2000-те године. Развој XHTML-а трајао је скоро деценију, развијене су верзије 1.0, 1.1, 1.2, 2.0. У верзији 1.1 представљени су модули за формуларе и табеле. Додати су елементи за бољу подршку источно-азијских језика. Постао је препорука 29. јула 2008.

Због неслагања са концептом који је користио W3C, Web Hypertext Application Technology Working Group (WHATWG) започела је са радом на новом стандарду 2004. године. То је период када је једна група из W3C-а радила на развоју XHTML 2.0, а HTML 4.01. није био освежен од 2000.-те године. WHATWG применила је други приступ који се заснивао на слободи, а не на строгости. Почели су да развијају нову верзију HTML-а која је толерантна на грешке у презентацији. Главни мотив је био креирање платформе за динамичне веб апликације. Сматрали су да XHTML 2.0 није погодан за креирање интернет форума или онлајн продавница.

W3C је пре пар година практично признао да је њихов приступ био погрешан, расформирао је групу за развој XHTML-а и зауставио његов развој. Након тога прихватили су предлог WHATWG групе и сада раде заједно на развоју HTML5.

Иако је HTML5 био познат веб програмерима годинама, постао је главна тема априла 2010. када је бивши директор Apple-а објавио јавни чланак „Мишљење о Flash-у“ где закључује да „Adobe Flash није више неопходан да би се гледао видео или пратили било какви веб садржаји и да ће нови отворени садржаји, као што је HTML5, настали у ери мобилних телефона, победити.“ Ово је изазвало дебату у круговима стручњака за веб. Неки су предложили да, иако HTML5 пружа побољшану функционалност, програмери морају узети у обзир разлике у функционалности HTML5 и Flash-а. Почетком новембра 2011. године, Adobe компанија је изјавила да престаје са развојем Flash-а за мобилне телефоне и да ће преоријентисати своје снаге на развој алата коришћењем HTML5.

WHATWG је започео рад на спецификацији стандарда јуна 2004. године под именом “Веб апликације 1.0”. W3C је усвојио HTML5 спецификацију 2007. године, а јануара 2008. објавио Први радни нацрт спецификације. Спецификација је у току, а очекује се да ће тако и остати дуги низ година, иако ће делови HTML5 бити завршени и имплементирани у претраживачима пре него што целокупна спецификација достигне финални статус препоруке. HTML5 је достигао фазу кандидата за статус препоруке средином децембра 2012. године. Критеријум неопходан за W3C спецификацију је: “две потпуно комплетне и интероперабилне имплементације. W3C развија свеобухватан тест пакет за постизање широке интероперабилности за пуну спецификацију до 2014. године, која је сада циљни датум за статус препоруке. Ипак, многи делови спецификације су стабилни и могу бити имплементирани у производе.

HTML5 претраживач ће бити флексибилан у руковању некоректном синтаксом. Дизајниран је тако да стари претраживачи могу безбедно игнорисати нове HTML5 конструкције. За разлику од HTML 4.01, HTML5 спецификација даје детаљна правила за лексичку анализу и парсирање, са намером да различити претраживачи производе исте резултате у случају некоректне синтаксе.^[2]



Слика 6. W3C лого

[2.3.2. Ознаке у HTML5 за структурирање садржаја стране](#)

HTML5 нуди нове ознаке којима се обезбеђује строго структуриран садржај страна. HTML5 дозвољава да маркирање буде везано за конструисање стране, док компоненте стила означавају како ће страна изгледати. Неке од нових ознака су `<video>`, `<audio>`, `<canvas>`. Омогућена је интеграција SVG (Scalable Vector Graphics) садржаја што замењује коришћење ознаке `<object>`. Ова својства су дизајнирана да би олакшала укључивање и руковање мултимедијалним и графичким садржајима на вебу, без потребе да се прибегне додавању plug-in-ова и програма написаних у различитим програмским језицима. Неки други елементи, као што су `<section>`, `<article>`, `<header>`, `<nav>`, дизајнирани су како би се обогатио семантички садржај докумената. Из истог разлога уведени су и нови атрибути, док су неки елементи и атрибути, за које се проценило да су застарели, уклоњени.

`<html>` елемент није обавезан елемент верзије HTML5. Глава и тело су такође необавезни. Међутим, помоћу њих се обезбеђује прегледност кода и одржавање добрих стандарда, те их је из тог разлога пожељно наводити. HTML5 не захтева појединачне или дупле наводнике око вредности атрибута, као што је то било потребно у претходним верзијама. Старији маркерски језици их захтевају, па зато није лоше наводити их. Иако су неке ознаке, атрибути или елементи застарели, то не значи да ће их претраживачи који подржавају HTML5 неправилно приказати. Застарелост значи само да су одређена ознака или атрибут замењени новом HTML5 ознаком, атрибутом или елементом, или је функција пренета у CSS.

У HTML 4.0 представљен је таг `<div>`. Помоћу ове ознаке дефинише се контејнер за чување информација и погодан је за дељење стране на подобласти, али се не обезбеђује добро структурирани оквир. То се постиже новим ознакама у HTML5 верзији.

Ознаке `<section>` `</section>` служе за тематско груписање садржаја у HTML документу. Садржај могу представљати поглавља, увод у неку тему, вести, контакт информације, ...^[4]

Пример:

```
<section class="block">
  <a href="brankov.html"></a>
  <h2><a href="brankov.html">Brankov most</a></h2>
  <p>Prvi, prelepi drumski most otvoren je 16. decembra 1934.
  godine. Prvobitno se zvao Most kralja Aleksandra... </p>
</section>
```

Елемент `<article>` `</article>` дели страну на одељке или ставке, као што су форуми, новински чланци, поруке на блоговима. За разлику од елемента `<section>` који тематски групише садржај, `<article>` служи за јасно раздвајање битних информација на страни, у одељку или у документу.^[3] Са овим маркером користе се глобални атрибутути.^[4] Од њих, нови су: `contenteditable|contextmenu|draggable|dropzone|hidden|spellcheck`.

Елементи `<nav>` `</nav>` представљају секцију текуће стране која је повезана са другим странама или са деловима у оквиру исте стране. То је одељак са навигационим линковима. Својства CSS-а могу директно да се примене помоћу ове ознаке. Такође се користе глобални атрибутути код овог тага.^[5]

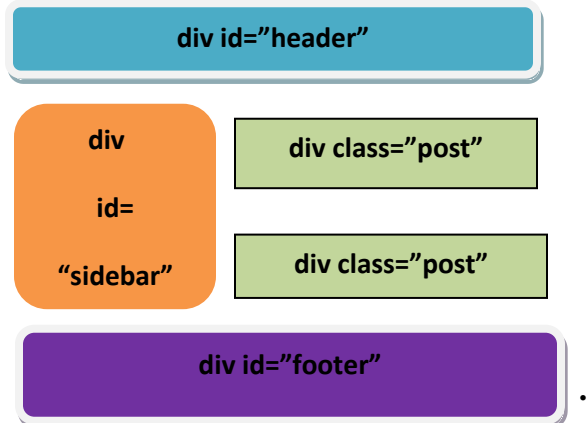
Елементи `<aside>` `</aside>` служе за прављење одељка за бочни стубац. Могу означавати и одељак где се смештају графички објекти. Подржавају глобалне атрибуте.^[3]

Заглавља стране за одељке и чланке направљена су помоћу елемента `<header>` `</header>`. То су одељци који се смештају на страну како бисмо дозволили уношење наслова. Као и други структурални HTML елементи, захтевају почетну и крајњу ознаку и могу имати сопствено форматирање.^[3]

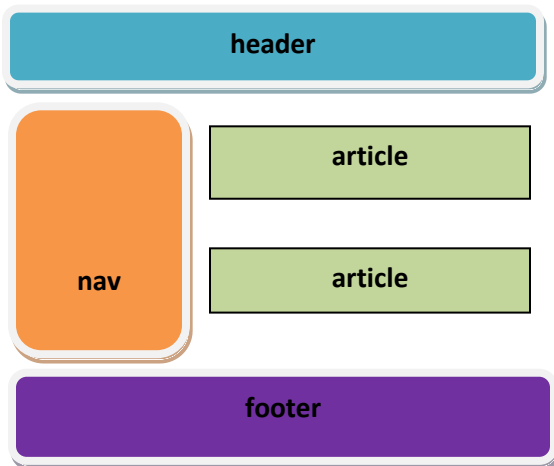
HTML5 омогућава да садржина наслова буде груписана. Наслови су груписани заједно са својим поднасловима или алтернативним насловима. За ово се користе ознаке `<hgroup>` `</hgroup>`.^[5]

Ознаке `<footer>` `</footer>` служе за израду одељка за подножје стране. У подножју стране се може наћи доста информација, али неке од уобичајених су: везе са информацијама о особи која је аутор сајта или са компанијом којој веб локација припада. Сваки садржај између почетног и крајњег тага биће приказан у одељку за подножје стране. Ако се ова ознака дода на страни у елемент `<body>`, реч је о подножју веб стране, ако се дода пре крајње ознаке елемента `<section>`, у питању је подножје стране одељка, а уколико се дода у елементе `<article>`, ради се о подножју стране чланка.^[3]

Нека је у верзији HTML 4.01 веб страна имала следећи изглед:



У верзији HTML5 та страна изгледа овако:



Елементи `<figure>` `</figure>` и `<figcaption>` `</figcaption>` такође су новина у HTML5. Први елемент служи за обележавање илустрација, дијаграма и фотографија. Може се користити за повезивање наслова заједно са неким уграђеним графичким садржајем или видеом. `<figure>` елемент користи се заједно са `<figcaption>` да би обезбедио натпис за садржај `<figure>` елемента. Елемент `<figcaption>` или мора да иде пре било које друге слике са садржајем, или мора да иде на крају, после свих слика са текстуалним садржајем. Сваки елемент `<figure>` може да има само један `<figcaption>`.^{[4] [5]}

Пример 1:

```
<figure>  
  
<figcaption class="naslov_slike"> Gejrir </figcaption>  
</figure>
```



Слика 7. Пример 1 у веб претраживачу Mozilla Firefox.

Уколико бисмо уметнули још један исти елемент, наредна фотографија заједно са натписом, аутоматски би се пребацила у следећи ред.

`<mark>` таг подржавају сви претраживачи. Уколико желимо да истакнемо нешто у тексту, онда користимо овај таг. ^[5]

Пример 2:

```

<section>
<article class="group3">
<h2><mark>O Avali</mark></h2>
<p>Avala je niska planina, 16,5 km ju&#382no od Beograda... .</p>
</article>
<article class="group3">
<h2><mark>Spomenik Neznamom junaku</mark></h2>
<p>Posle Prvog svetskog rata, lokalno stanovni&#353tvo... .</p>
</article>
.....
</section>
    
```



Слика 8. Пример 2 у веб претраживачу Mozilla Firefox.

Ознаке: `<menu>`, `<bdi>`, `<command>`, `<details>`, `<summary>`, `<meter>`, `<progress>`, `<time>`, `<wbr>`, `<ruby>`, `<rt>` и `<rp>` нове су у HTML5. ^[4]

2.3.3. HTML5 обрасци

Многе веб стране од својих корисника прикупљају информације које складиште или користе на другим веб странама. Сакупљене информације се преносе на сервер мреже ради даље обраде. HTML5 обезбеђује контроле на обрасцу које су у ранијим верзијама биле недоступне веб програмерима и дизајнерима. Као замену за то, морали су раније да користе скрипт језике који се уобичајено користе за обраду података.

HTML образац означен је тагом `<form>`. Он групише све остале елементе за образац. Поред глобалних атрибута и атрибута догађаја ^[6], подржава следеће атрибуте: `action`, `method` који има две вредности: `get` и `post`, `name`, `target`...

Нови атрибути у верзији HTML5 су:

- `autocomplete` – синтакса је:

`<form autocomplete="on|off">`, што значи да аутоматско попуњавање може бити укључено или искључено. Подразумевана вредност јесте 'on' и веб претраживач ће аутоматски попунити вредности које је корисник раније уносио. Уколико је изабрана опција `off`, неће бити аутоматског попуњавања и корисник ће за свако поље морати да уноси вредности. Могуће је имати аутоматско попуњавање за форму укључено, а искључено за појединачна поља, и обрнуто.

- `novalidate` – када је овај атрибут присутан, говори да подаци из обрасца не треба да буду потврђени када се пошаљу. Синтакса је:

`<form novalidate="novalidate">` или `<form novalidate>` или `<form novalidate="">`.^[4]

`<form>` таг може садржати један или више следећих елемената:

`<input>`, `<textarea>`, `<button>`, `<select>`, `<option>`, `<optgroup>`, `<fieldset>`, `<datalist>`, `<output>`, `<label>` ^[4]. Много нових атрибута уведено је за ове ознаке.

- `<input>`

У верзију HTML5 уведен је велики број нових вредности.

Синтакса је: `<input type="value" />`. Од старих вредности (`value`) и даље се користе:

- ✓ `button`, `checkbox`, `file`, `hidden`, `image`, `password`, `radio`, `reset`, `submit`, `text`.

У HTML5 уведене су и следеће вредности:

- ✓ `color` – дефинише бирач боја.
- ✓ `date` – дефинише контролу датума (година, месец и дан).
- ✓ `datetime` – дефинише датум и контролу времена (година, месец, дан, сат, минут, секунда, стотинка, засновано на UTC времену).

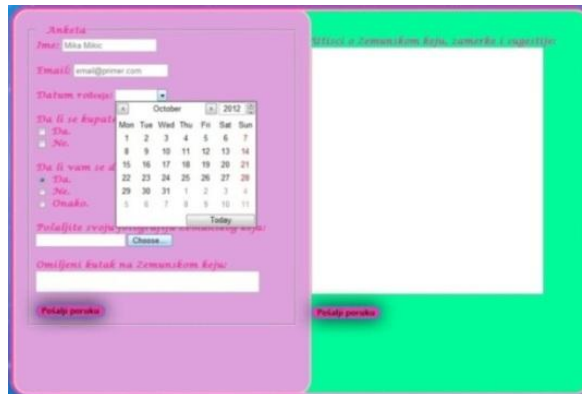
- ✓ **datetime-local** -- дефинише датум и контролу времена (година, месец, дан, сат, минут, секунда, стотинка, нема временске зоне).
- ✓ **email** – дефинише поље за email адресу.
- ✓ **month** – дефинише месец и контролу године (без временске зоне).
- ✓ **number** – дефинише поље за упис броја.
- ✓ **range** – дефинише контролу за упис броја чија тачна вредност није битна (као контрола слајдера).
- ✓ **search** – дефинише текстуално поље за упис стринга за претрагу.
- ✓ **tel** – дефинише поље за упис броја телефона. Још увек ову вредност не подржавају познатији веб претраживачи.
- ✓ **time** – дефинише контролу за упис времена (без временске зоне).
- ✓ **url** – дефинише поље за упис URL адресе.
- ✓ **week** – дефинише недељу и контролу године (без временске зоне).

Пример 3

```
<div class="anketa">
  <form action="mailto:user@gmail.com" method="post"
name="zkejanketa1" autocomplete="on">
  <fieldset>
  <legend>Anketa</legend>
  <label for="name">Име:</label>
    <input type="text" name="ime" required
placeholder="Mika Mikic" />
    <br/><br/>

    <label for="email">Email:</label>
    <input type="email" name="email" required
placeholder="email@primer.com" />
    <br/><br/>

    <label for="birthday">Datum ro&#273enja:</label>
    <input type="date" name="birthday" />
  <br/><br/>
    .....
    <input type="submit" value="Po&#353alji poruku"
id="button" />
  </fieldset>
</form>
</div> ...
```



Слика 9. Пример 3 у веб претраживачу Орега.

Нови атрибути су:

- **autocomplete** -- ово поље одређује да ли input треба да има омогућено аутоматско попуњавање. Оно омогућава претраживачу да предвиди вредност на основу претходних које је унео корисник. Функционише уз следеће <input> типове: text, search, url, telephone, email, password, datepickers, range и color. Синтакса је: <input autocomplete="on|off" />.
- **formaction** – овај атрибут преписује action атрибут <form> елемента.
- **formnovalidate** – преписује novalidate атрибут <form> елемента. Функционише уз следеће <input> типове: text, search, url, telephone, email, password, date pickers, range, color.
- **placeholder** – представља малу помоћ при упису очекиване вредности у поље. То је обично неки пример вредности или кратак опис. Та вредност је приказана у пољу када је празно, а нестане када поље добије фокус. Ради уз следеће типове input-а: text, search, url, tel, email и password. Синтакса је: <input placeholder="text" />.
- **required** – казује да поље мора бити попуњено пре него што се проследи образац. Функционише уз следеће типове input-а: text, search, url, tel, email, password, date pickers, number, checkbox, radio и file. Синтакса: <input required="required" /> или <input required/> или <input required=""/>. [4]

Тагови у вези са формуларима који се и даље користе у HTML5 су: <textarea>, <button>, <select>, <option>, <optgroup>, <fieldset> и <label>, а нови су: <datalist>, <output> и <keygen>. [4] [6]

2.3.4. *HTML5 мултимедијални елементи*

У последњих неколико година аудио и видео садржаји постају све доступнији. Данас лако свако може поставити видео и аудио снимке на локације као што су: Youtube, Vimeo, Facebook, MySpace...

Нови тагови који знатно побољшавају подршку мултимедијалним садржајима су: `<audio>`, `<video>`, `<source>`, `<embed>`, `<track>`. Раније се у ове сврхе користио `<object>` таг, али ти кодови су били доста неуредни. Проблем су представљали и `plug-in`-ови. Наиме, корисник није имао одговарајућу верзију, или није имао права да инсталира, или није знао како. Данас, са HTML5 нису покривени сви случајеви `plug-in`-ова, али знатан напредак је постигнут. На пример, заштита од копирања је једна област којом се HTML5 није бавио, с обзиром да је стандард заснован на отворености.

`<audio>`

Дефинише звук, у музици или другим аудио токовима. Подржавају га сви познатији претраживачи. Било какав текст написан између овог тага и његовог тага затварања биће приказан само у претраживачима који не подржавају аудио. Поред глобалних атрибута и атрибута догађаја^[4], подржава и следеће:

- **autoplay**: када је овај атрибут присутан, аудио се стартује када је спреман и траје без престанка. Синтакса је: `<audio autoplay="autoplay">` или `<audio autoplay>` или `<audio autoplay="">`.
- **controls**: означава да аудио контроле треба да буду присутне. Оне подразумевају следеће:
 - ✓ Play
 - ✓ Pause
 - ✓ Seeking
 - ✓ VolumeСинтакса је: `<audio controls="controls">` или `<audio controls>` или `<audio controls="">`.
- **loop**: означава да ће се аудио понављати изнова сваки пут пошто се заврши. Ова опција није подржана у Firefox-у и Opera-и.
- **preload**: овај атрибут не подржавају претраживачи Internet Explorer и Opera. Одређује да ли треба и како треба да се аудио учита када се страница стартује, по мишљењу аутора, а засновано на икуству корисника. Овај атрибут се игнорише уколико је присутан атрибут `autoplay`. Вредности су:

auto: аутор сматра да претраживач треба да пусти цео аудио фајл када се страница учита.

metadata: аутор сматра да претраживач треба да прикаже метаподатке када се страница учита.

none: аутор сматра да претраживач не треба да пусти аудио када се страница учита.

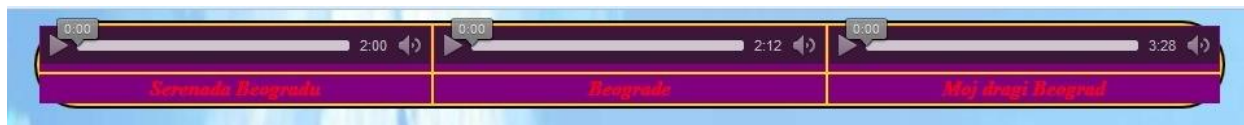
- **src**: специфира URL аудио фајла. Уколико фајл има екстензију .ogg, радиће у Firefox-у, Opera-и и Chrome-у, док је за пуштање аудио фајла у претраживачима Internet Explorer и Safari потребна екстензија .mp3. Екстензију .wav подржавају сви претраживачи сем Internet Explorer-а. Да би радило у свим претраживачима, користи се **<source>** таг и онда претраживач користи први формат који му је познат.

<source> таг се може користити и у оквиру **<video>** тага. Ово је нови таг у HTML5. Његови атрибути су:

- **media**: специфира тип медијског извора. Претраживач може искористити овај атрибут да одлучи да ли може да пусти фајл или не. Ако не може, онда може да одлучи да га не скине. Још увек није подржан у популарнијим веб претраживачима. Синтакса: **<source media="value">**.
- **src**: специфира URL адресу медија фајла.
- **type**: специфира одређене вредности:
 - за видео: video/ogg, video/mp4, video/webm,
 - а за аудио: audio/ogg, audio/mpeg. ^{[4][5][7]}

Пример 4:

```
<section>
<table class="muzika">
<tr>
<td>
<audio controls="controls" preload="none">
<source src="serenadabg.ogg" type="audio/ogg" />
<source src="serenadabg.mp3" type="audio/mp3" />
<source src="serenadabg.wav" type="audio/wav" />
  Vas pretrazivac ne podrzava audio element.
</audio>
</td>
.....
<tr>
<td>Serenada Beogradu</td>
.....
</tr>
</table>
</section>
```

Слика 10. Пример 4 у веб претраживачу Mozilla Firefox.

- <video>

Служи за постављање видео снимака. Нов је елемент у HTML5. Као и код <audio> тага, подржава атрибуте `autoplay`, `controls`, `loop`, `preload`, `src`. Controls укључују следеће: Play, Pause, Seeking, Volume, Fullscreen toggle, Captions/Subtitles (када је доступно), Track (када је доступно). Видео са екстензијом .ogg радиће у претраживачима Firefox, Opera и Chrome, док они са екстензијом .mp4 ће радити у претраживачима Internet Explorer и Safari. Такође, овај таг подржава још и атрибуте: `width`, `height`, `muted`, `poster`.^{[4][5]}

2.3.5. <canvas> маг

Овај елемент служи за исцртавање графике. За сада не пружа све могућности као Flash. <canvas> таг је контејнер за графику, за цртање мора се користити скрипт. Canvas је правоугаона област где се сваки пиксел контролише коришћењем JavaScript-а. Овај таг користи се за цртање облика, пуњење облика бојама, стварање градијента и дезена, рендеровање текста, копирање слика, видео слика итд., манипулисање пикселима итд. Сваки текст написан између <canvas> и </canvas> биће приказан у веб претраживачима који не подржавају ове тагове. HTML5 има уграђени објекат за canvas, то је 'getContext("2d")' објекат. Овај објекат има своје методе за цртање линија, кутија, кругова итд.^{[4][5][7]}

3. CSS

3.1. Карактеристике, историја и развој CSS-а

CSS представља скраћеницу енглеских речи Cascading Style Sheets, што означава каскадне листе стилова. HTML обезбеђује структуру странице, а CSS је језик форматирања који омогућава дефинисање изгледа веб странице.

CSS је, пре свега, дизајниран управо из разлога да раздвоји садржај документа од начина на који ће бити представљен (фонт, боја, величина...). Ово раздвајање доприноси доступности садржају, флексибилности, контроли спецификације карактеристика презентације, смањује комплексност и омогућава да већи број HTML страна буде формиран на исти начин. CSS јесте текстуални документ који садржи инструкције за форматирање. Њих користе веб претраживачи. Инструкције се примењују на елементе који су обухваћени скупом инструкција, при чему није битно где се ти елементи појављују на веб страни.

CSS одређује шему приоритета која одлучује која правила за стилове важе уколико се деси да постоји више правила за један елемент. Ово су CSS каскаде. Тако инструкције у вези са једним слојем стављају се ван снаге у слоју који је један корак нижи. Из овог разлога, форматирање се каскадно помера надоле по приоритету.

CSS омогућава форматирање на три различита нивоа. Први ниво представља екстерну листу стилова. То је засебни текстуални документ који се налази на серверу мреже и на ту листу се HTML датотеке позивају. Други ниво поставља инструкције за форматирање у елементе `<style>` у оквиру одељка `<head>`. Овај ниво ради на странама у којима су укључени стилови. Уколико дође до конфликта, ставиће ван снаге инструкције из екстерних листа стилова. Трећи ниво, најнижи, јесте редни стил или стил за редове. Они се налазе у атрибуту `style` почетне ознаке елемента који тај атрибут формира. Претраживач тим редом и тумачи инструкције за форматирање.

Роберт Каијо је желео да одвоји структуру од презентације. Идеалан начин био је дати кориснику различите опције и преносити три различите врсте листа стилова: један за штампање, један за презентацију на екрану и један за карактеристике едитора. Да би се побољшале могућности веб презентације, W3C-у је представљено девет различитих језика. Од тога два су изабрана као темељ за оно што данас представља CSS: Cascading HTML Style Sheets (CHSS) и Stream-based StyleSheet Proposal (SSP). Први од два предложио је Норвежанин Хакон Виум Ли октобра 1994. године, док је Берт Бос радио на претраживачу Argo који је користио његов језик листа стилова SSP. Њих двојица су радили заједно на развоју стандарда. Слово Н је уклоњено из имена из разлога што је означавало HTML, а ове листе стилова су могле бити употребљене и за друге маркерске

језике осим HTML-а. Лијев предлог био је представљен у Чикагу 1994. на конференцији "Mosaic and the Web"^[2], а идуће године поново, заједно са Босом. Они су имали примарну улогу техничког особља овог пројекта заједно са Томасом Рердоном из Microsoft-а. Августа 1996. Netscape Communication корпорација представила је алтернативне листе стилова JavaScript Style Sheets (JSSS). Ипак, ово никад није завршено и застарело је. До краја 1996. CSS је био спреман да постане и званичан, а препорука CSS ниво 1 објављена је у децембру.

CSS1: ова спецификација објављена је децембра 1996. Међу њеним могућностима су:

- боја текста, позадина и других елемената
- својства фонта као што су типографско писмо и наглашавање
- текстуални атрибути као што су размак између речи, слова и линије текста
- поравнање текста, слика и других елемената
- маргине, границе, 'padding' (простор између границе елемента и садржаја елемента) и позиционирање већине елемената
- јединствена идентификација и генеричка класификација група атрибута.

CSS2: ова верзија је објављена као Препорука у мају 1998. Овде је укључен и број нових могућности као што су апсолутно, релативно и фиксно позиционирање елемената и z-index, концепти медијских типова, подршка за звучне листе стилова и двосмеран текст и нова својства фонта као што су сенке.

CSS 2.1: овом верзијом исправљене су грешке у CSS2, уклоњене слабо подржане или не у потпуности интероперабилне карактеристике и додате су већ имплементирани екстензије претраживача. Да би био у складу са W3C процесом за стандардизовање техничких спецификација, CSS 2.1 је наизменично више година постизао статус Радног нацрта и кандидата за Препоруку. Кандидат за Препоруку први пут постао је 25. фебруара 2004. године, али је враћен у статус Радног нацрта 13. јуна 2005. за даље разматрање. Статуси су се још неколико пута мењали у међувремену да би коначно статус Препоруке био објављен 7. јуна 2011.^[2]

3.2. CSS3

Рад на CSS3 започео је још у време објављивања CSS2. Најранији CSS3 нацрти били су објављени јуна 1999. Оно што је специфично за CSS3, за разлику од претходних верзија, јесте начин на који су спецификације понуђене и расположиве произвођачима претраживача. Док је у ранијим верзијама била неопходна цела спецификација за одобрење, у CSS3 понуђени су модули. Они представљају неколико одвојених докумената. Сваки модул додаје нове функције или проширује својства дефинисана у CSS2. Новембра 2011. већ постоји преко 50 модула објављених од стране CSS радне групе. Од тога три су добила статус Препоруке током 2011. године.

Друге промене обухватају већу флексибилност позадина и оквира, колона са садржајем и већу могућност приликом штампања мултимедијалних страна.^[2]



Слика 11. Званични лого CSS3.



Слика 12. Након Wium Lie



Слика 13. Bert Bos

3.2.1. Синтакса CSS-а

Коментаре у CSS-у претраживач потпуно игнорише. Означавају се на следећи начин: `/* */`, између звездица налази се коментар.

CSS се састоји од два дела: селектора и од једне или више декларација. Селектор је HTML елемент који желимо да форматирамо. Декларација се састоји од атрибута који мењамо и вредности.



Због лепшег исписа, претходно је могуће записати и на следећи начин:

```
h1
{
color:blue;
font-size:12px;
}
```

Сваки HTML елемент има атрибуте `class` и `id` који могу тако да се подесе да форматирање помоћу CSS-а може лако да се изврши и то за изабрану групу елемената. Први служи за одређивање стила за групу елемената. Селектор је дефинисан са `."`. Други служи за одређивање стила за један елемент. Селектор је дефинисан са `"#"`.

```
<head>
<style type="text/css">
.centar
{
text-align:center;
}
#sadrzaj {
background-color: #F5DEB3;
border:1px solid transparent;
-moz-border-radius: 10px;
border-radius:10px;
}
</style>
</head>
```

Претходан код јесте интерна листа стилова. Inline стил је уколико је оваквог типа: `<p style="color:yellow; margin-left:15px"> ovo je paragraf.</p>`

Екстерна листа стилова је идеална када се стил примењује на више страна. Тада сваку страну за коју желимо да има један исти начин форматирања повежемо са листом стилова преко <link> тага коришћењем кода који може овако да изгледа:

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

Оваква листа стилова мора бити написана у неком текст едитору, сачувана са екстензијом .css (овде, конкретно, mojstil.css) и не сме садржати ниједан HTML таг.

3.2.2. Специфичности CSS3
















CSS3 је раздвојен у модуле. Неки од најбитнијих CSS3 модула су:

- селектори
- модел области
- позадине и оквири
- текстуални ефекти
- 2D/3D трансформације
- анимације
- више колона
- кориснички интерфејс.

а) CSS3 оквири

Са верзијом CSS3 могу се креирати заобљене ивице, додавати сенке оквирима и користити слике као оквири, све то без коришћења програма као што је Photoshop.

Табела 1. Подршка веб претраживача за атрибуте border-radius, box-shadow и border-image.

Атрибут	Подршка претраживача				
border-radius					
box-shadow					
border-image					

CSS3 заобљене ивице

У CSS2 верзији за заобљивање ивица користиле су се различите слике за сваки ћошак. У CSS3 верзији, то је доста олакшано. За то се користи border-radius.

Опција border-radius property подржана је у IE9+, Firefox 4+, Chrome, Safari 5+ и Опера-и. Синтакса је:

border-radius: 1-4 length|% / 1-4 length|%; при чему:

length: дефинише облик ћошкова

%: дефинише облик ћошкова у процентима. ^[4]

Пример 5:

У .html фајлу:

```
<head>
<link rel="stylesheet" href="styles.css" type="text/css" media="screen" />
</head>

<body>
.....
    <section id="intro">
        <p>"Od svega &#353to &#269ovek u &#382ivotnom nagonu podi&#382e i
gradi, ni&#353ta nije u mojim o&#269ima bolje i vrednije od mostova."
        <br/> --Ivo Andri&#263 --
    </p>
        
    </section>
.....
</body>
```

У styles.css фајлу:

```
#intro {
    position: relative;
    margin-top: 15px;
    padding: 40px;
    background: #6600CC;
    -webkit-background-size: 100%;
    -o-background-size: 100%;
    border:1px solid transparent;
    -moz-border-radius: 10px;
    -webkit-border-radius: 10px;
    border-radius:10px;
}
```



Слика 14. Пример 5 у веб претраживачу Mozilla Firefox.

CSS3 сенка области

box-shadow property подржан је у IE9+, Firefox 4, Chrome, Opera и Safari 5.1.1.

Синтакса је:

box-shadow: h-shadow v-shadow blur spread color inset; при чему:

h-shadow – је неопходан и означава позицију хоризонталне сенке. Негативне вредности су дозвољене.

v-shadow – је неопходан и означава позицију вертикалне сенке. Негативне вредности су дозвољене.

blur – је опциона вредност. Величина која представља удаљеност замућености од објекта.

spread – је опциона вредност. Представља величину сенке.

color – је опциона вредност. Означава боју сенке.

inset – је опциона вредност. Мења сенку од спољње сенке до унутрашње. ^[4]

Пример 6:

у .html фајлу:

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

```
<body>
```

```
.....
<article class="prvi">
<cite>

"До#353ав#353и, на#273оh најкрасније место од давнине, превелики град
Београд, који је по слу#269ају разру#353ен и запустео,
саздах га и посветих Богоматери..."<br/> Despot Stefan Lazarevi#263 </cite>
</article>
```

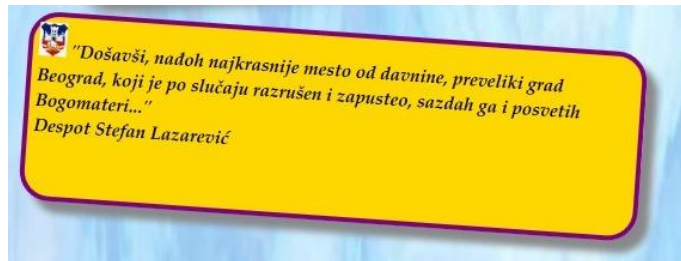
```
.....
</body>
```

у style.css фајлу:

```
article.prvi
{
.....
-moz-box-shadow: 10px 10px 5px #888888; /* Firefox 3.6 i ranije verzije */
```



```
-webkit-box-shadow: 10px 10px 5px #888888; /* safari */  
box-shadow: 10px 10px 5px #888888;  
}.....  
}
```



Слика 15. Пример 6 у веб претраживачу Mozilla Firefox.

CSS3 слика као оквир

border-image није подржан у већим претраживачима. Firefox, Opera, Safari и Chrome подржавају алтернативне верзије. То су:

за Firefox: -moz-border-image, за Opera-у: -o-border-image, за Chrome, Safari: -webkit-border-image.

border-image подразумева следећу синтаксу:

border-image: *source slice width outset repeat;*

border-image-slice: одређује унутрашње офсете. Вредности су:

number: представља пикселе на слици (уколико је растерска) или координате векора (уколико је векторска).

%: проценти одговарају величини слике: дужина за хоризонталне офсете, ширина за вертикалне офсете.

fill: чини да средишњи део border-image буде сачуван.

border-image-width: овај атрибут није подржан. Вредности су:

number: представља више одговарајућих border-width.

%: односи се на величину области граничне слике: ширина области за хоризонталне офсете, висина за вертикалне офсете.

auto: уколико је наведено, ширина је унутрашња ширина или висина одговарајућег дела слике.

border-image-outset: представља износ за који област граничне слике превазилази граничну област.

border-image-repeat: није подржан у модерним веб претраживачима. Користи се border-image уместо. Синтакса је: **border-image-repeat: stretch|repeat|round;** где је:

stretch: слика је раширена тако да покрије целокупну област.

repeat: слика се понавља да би попунила област.

round: слика се понавља да би попунила област. Ако се не попуни област уз помоћ целих понављања слика, слика се рескалира да би цела стала. ^[4]

Пример 7:

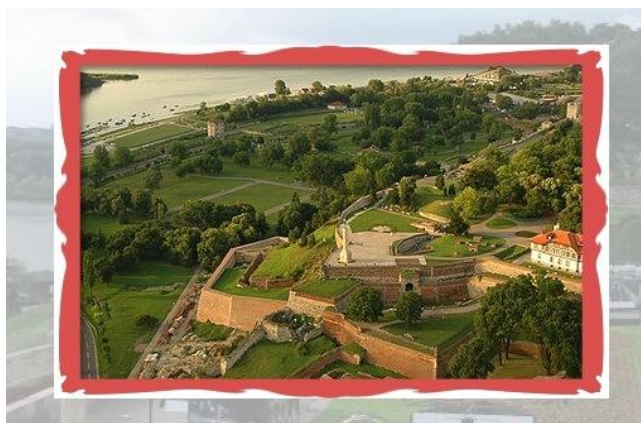
У .html фајлу:

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

```
<body>
.....
<div class="ram">
</div>
.....
</body>
```

У kalemegdan.css фајлу:

```
.ram{
.....
width:380px;
height:230px;
background:url(slike/tvrđjava.jpg) center center no-repeat;
border-width: 25px 30px;
-moz-border-image: url(slike/frame2.png) 25 30 stretch;
-webkit-border-image: url(slike/frame2.png) 25 30 stretch;
-o-border-image: url(slike/frame2.png) 25 30 stretch;
-ms-border-image: url(slike/frame2.png) 25 30 stretch;
border-image: url(slike/frame2.png) 25 30 stretch;
}
```



Слика 16. Пример 7 у веб претраживачу Mozilla Firefox.

б) CSS3 позадине

background-size: одређује величину позадинске слике. Пре CSS3 величина је била одређена стварном величином слике. Овде је могуће одредити величину слике, што омогућава поновно коришћење слике у различитим контекстима. Могуће је изразити величину слике у процентима. Синтакса је:

background-size: *length|percentage|cover|contain*; где:

length: поставља висину и ширину позадинске слике. Прва вредност поставља ширину, друга вредност поставља висину. Ако је само једна вредност дата, друга је постављена на аутоматску.

percentage: поставља висину и ширину позадинске слике родитељског елемента у процентима. Прва вредност поставља ширину, друга поставља висину. Ако је само једна вредност дата, друга је постављена на аутоматску.

cover: слика се скалира на најмању вредност тако да и ширина и висина могу да стану у област са садржајем.

contain: слика се скалира на највећу вредност тако да и ширина и висина могу да стану у област са садржајем. ^[4]

Пример 8:

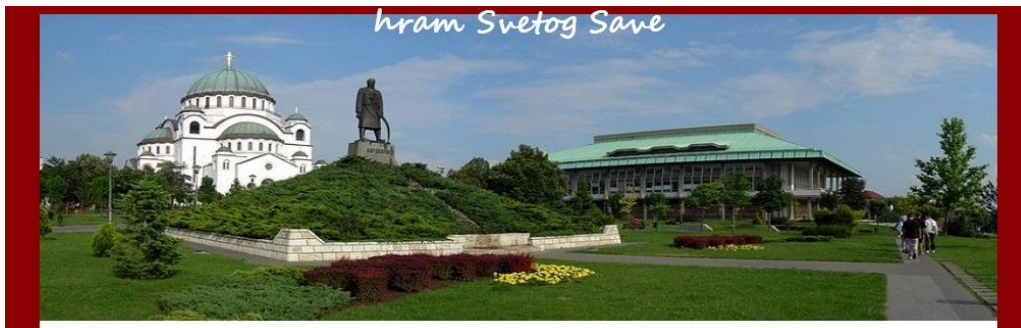
у .html фајлу:

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

```
<body>
.....
<header>
hram Svetog Save
</header>
.....
</body>
```

у hram.css фајлу:

```
header {
width: 940px;
height: 300px;
margin: 0 auto;
background: #000 url(ms_slike/hramwide.jpg) no-repeat left top;
background-size: 940px 300px;
.....
}
```



Слика 17. Пример 8 у веб претраживачу Mozilla Firefox.

в) Ефекти са текстом

CSS3 омогућава доста нових опција у вези са текстом. Једна од њих је `text-shadow`.

Служи за додавање сенке тексту. Подржана је у свим модерним претраживачима осим у Internet Explorer-у. Синтакса је:

`text-shadow: h-shadow v-shadow blur color;` при чему је:

`h-shadow`: неопходно. Представља позицију хоризонталне сенке. Негативне вредности су дозвољене.

`v-shadow`: неопходно. Представља позицију вертикалне сенке. Негативне вредности су дозвољене.

`blur`: опционо. Величина која представља удаљеност замућености од објекта.

`color`: опционо. Боја сенке. ^[4]

Пример 9:

у .html фајлу:

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

```
<body>
```

```
.....
<header id="naslov">
Trgovi Beograda
</header>
```

```
.....
</body>
```

у trgovi.css фајлу:

```
#naslov{
text-shadow: 6px 8px 3px #b72d23, -2px -7px 2px salmon;
font-size: 30px;
font-weight:bold;
font-style:italic;
font-family:Lucida Handwriting;
text-align:center;
}
```

Trgoví Beograda

Слика 18. Пример 9 у веб претраживачу Mozilla Firefox.

Г) CSS3 фонтови

Нова верзија CSS-а, CSS3, од изузетне је важности за фонтове. Сада је могуће користити фонтове који нам се допадају при креирању веб сајта или апликације.

Фонт који желимо да користимо поставимо на веб сервер и он се аутоматски скида кориснику. Наши фонтови дефинишу се у оквиру `@font-face` правила које је подржано у свим модерним претраживачима. Firefox, Chrome, Safari и Opera подржавају фонтове типа `.ttf` (True Type Fonts) и `.otf` (Open Type Fonts).

Internet Explorer 9+ подржава `@font-face` правило, али само подржава фонт типа `.eot` (Embedded OpenType), док Internet Explorer 8 и раније верзије не подржавају `@font-face` правило.

Пример 10:

У `.html` фајлу:

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

```
<body>
```

```
.....
<header id="naslov">
Zemunski kej
</header>
.....
</body>
```

У `dkej.css` фајлу:

```
@font-face
{
font-family:ActionIs;
src: url('ActionIs.ttf'),
      url('ActionIs.eot') format("opentype"); /*IE*/
}
#naslov
{
text-align:center;
text-decoration:italic;
font-family:ActionIs;
font-weight:bold;
color:#ff3399;
font-size:50px;
text-shadow: 2px 2px white;
}
```



Слика 19. Пример 10 у веб претраживачу Mozilla Firefox.

Дескриптори фонта:






font-family: за вредност узима неко име. Неопходан дескриптор. Дефинише име фонта.

src: за вредност узима неки URL. Неопходан дескриптор. Дефинише URL фонт фајла.

д) CSS3 2D трансформације

Трансформација је ефекат који омогућава да елемент мења облик, величину, позицију. У CSS3 подржане су 2D и 3D трансформације.

Табела 2. Префикси потребни за подршку веб претраживача код 2D трансформације.

Атрибут	Подршка претраживача				
transform					

Атрибути који се користе код 2D, а исто тако и код 3D трансформација су следећи:

transform: примењује 2D или 3D трансформацију на елемент. Синтакса је следећа:

transform: none | *методе трансформације*.

transform-origin: дозвољава промену позиције трансформисаних елемената. 2D трансформисани елемент може променити x и y осу елемента. 3D трансформисани елемент може променити z осу елемента. Користи се заједно са transform. Синтакса је следећа:

transform-origin: *x-оса y-оса z-оса*;

Табела 3. Вредности за transform-origin.

Вредност	Опис
x оса	Дефинише где се налази поглед на x осу. Могуће вредности су: left, center, right, <i>length</i> , %.
y оса	Дефинише где се налази поглед на y осу. Могуће вредности су: top, center, bottom, <i>length</i> , %.
z оса	Дефинише где се налази поглед на z осу. Могућа вредност је: <i>length</i> .

Табела 4. Методе 2D трансформације.

Функција	Опис
matrix(n, n, n, n, n, n)	Дефинише 2D трансформацију користећи матрицу од 6 вредности.
translate(x, y)	Дефинише 2D транслацију померајући елемент дуж X и Y оса.
translateX(n)	Дефинише 2D транслацију померајући елемент дуж X осе.
translateY(n)	Дефинише 2D транслацију померајући елемент дуж Y осе.
scale(x,y)	Дефинише 2D скалирање мењајући елементима ширину и висину.
scaleX(n)	Дефинише 2D скалирање мењајући елементима ширину.
scaleY(n)	Дефинише 2D скалирање мењајући елементима висину.
rotate(angle)	Дефинише 2D ротацију са углом задатим као параметром.
skew(x-angle, y-angle)	Дефинише 2D трансформацију искошења дуж X и Y оса.
skewX(angle)	Дефинише 2D трансформацију искошења дуж X осе.
skewY(angle)	Дефинише 2D трансформацију искошења дуж Y осе.

Пример11:

у .html фајлу:

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

<body>

.....

```
<article class="drugi">
```

```
<cite>
```

```

```

```
"Beograd ne voli da se slika. Mrzi ga da pozira. Mrda se...."<br/> Момо Капор
```

```
</cite>
```

```
</article>
```

.....

```
</body>
```

у style.css фајлу:

```
article.drugi
```

```
{
```

.....

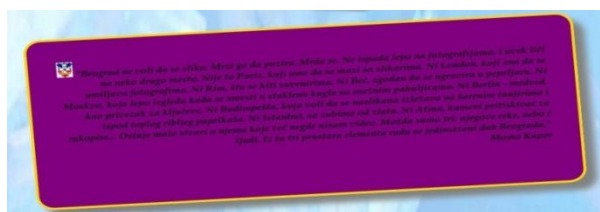
```
-moz-transform:rotate(-4deg);
```

```
-webkit-transform:rotate(-4deg);
```

```
-o-transform:rotate(-4deg);
```

```
transform:rotate(-4deg);
```

```
}
```



Слика 20. Пример 11 у веб претраживачу Mozilla Firefox.

ђ) CSS3 3D трансформације

Internet Explorer, Firefox и Opera још увек не подржавају 3D методе трансформације. Chrome и Safari захтевају префикс `-webkit-`.^[4]

е) CSS3 транзиције

Транзиције су ефекти који дозвољавају да се елемент постепено мења из једног стила у други. Internet Explorer још увек не подржава ову опцију, док остали модерни веб претраживачи захтевају коришћење префикса. Тако Firefox 4 користи префикс `-moz-`, Chrome и Safari `-webkit-`, Opera `-o-`.

Атрибут `transition` је скраћеница следећих: `transition-property`, `transition-duration`, `transition-timing-function` и `transition-delay`. Мора бити специфициран `transition-duration` да би транзиција имала ефекта. Иначе је подразумевана вредност: `all 0 ease 0`.

transition-property: одређује име CSS атрибута за који је предвиђен транзициони ефекат. Дешава се обично када корисник мишем прелази преко елемента. Синтакса је:

transition-property: `none|all|property`; Уколико је изабрана прва вредност ниједан атрибут неће добити транзициони ефекат, уколико је изабрана друга, онда ће сви добити. Трећа вредност дефинише листу атрибута одвојених зарезима код којих се примењује ефекат.

transition-duration: одређује колико секунди или милисекунди је потребно да се изврши транзициони ефекат. Подразумевана вредност је 0, што значи да нема ефекта.

transition-timing-function: одређује брзину криве транзиционог ефекта. Дозвољава ефекту да мења брзину за време извршавања. Вредности су:

linear: представља ефекат који има исту брзину од почетка до краја.

ease: представља ефекат који има спор старт, затим иде брзо, потом се споро завршава.

ease-in: представља ефекат са спорим стартом.

ease-out: представља ефекат са спорим крајем.

ease-in-out: представља ефекат са спорим стартом и крајем.

cubic-bezier(*n,n,n,n*): дефинишу се сопствене вредности у овој функцији. Могуће вредности су од 0 до 1.

transition-delay: одређује када ће ефекат почети да се извршава. Наводи се број секунди или милисекунди које су потребне да прођу док ефекат не почне да се извршава.^[4]

Пример12:

У .html фајлу:

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

```
<body>
```

```
.....
<div id="zamena">


</div>
```

```
.....
</body>
```

У dkej.css фајлу:

```
#zamena {
.....
}
#zamena img {
    position:absolute;
    left:0;
    -webkit-transition: opacity 1s ease-in-out;
    -moz-transition: opacity 1s ease-in-out;
    -o-transition: opacity 1s ease-in-out;
    -ms-transition: opacity 1s ease-in-out;
    transition: opacity 1s ease-in-out;
}

#zamena img.top:hover {
    opacity:0;
}
```

При преласку мишем преко прве фотографије, приказује се друга.



Слика 21. Пример 12 у веб претраживачу Mozilla Firefox.

ж) CSS3 анимација

Да би се креирале анимације у CSS3, користи се `@keyframes`. То је место где се дефинише анимација. Internet Explorer и Opera још увек не подржавају ово, Firefox захтева префикс `-moz-`, а Safari и Chrome `-webkit-`. Рецимо, уколико правимо анимацију и хоћемо да она ради у Firefox-у, пишемо следећи код:

```
@-moz-keyframes мојаАнимација /*Firefox*/
{
from {background: red;}
to {background: violet;}
}
```

У делу у којем се дефинише стил наводимо атрибут:

`-moz-animation: мојаАнимација 5s; /* Firefox */` где се наводе име анимације и њено трајање.

Код анимација увек се наводи када ће она започети и докле ће се извршавати и за то служе кључне речи “from” и “to”. Такође, могу се користити и проценти. 0% представља почетак анимације, а 100% њен крај. Треба увек дефинисати и 0% и 100% због боље подршке претраживача. У делу након имена анимације у оквиру `@keyframes` правила следе CSS атрибути које желимо да мењамо за време анимације.

Атрибут `animation` је скраћеница за `animation-name`, `animation-duration`, `animation-timing-function`, `animation-delay`, `animation-iteration-count` и `animation-direction`. Подразумевана вредност је `none 0 ease 0 1 normal`. Увек треба навести `animation-duration`, у супротном вредност је 0 и анимација неће бити извршена. Синтакса је ***animation: name duration timing-function delay iteration-count direction;*** где:

animation-name: *keyframename|none*; јесте атрибут у којем се наводи име анимације задато у оквиру `@keyframes`, а у супротном неће бити анимације.

animation-duration: *time*; одређује колико је потребно секунди или милисекунди да анимација изврши један круг. Уколико је вредност 0, неће бити анимације.

animation-timing-function: одређује брзину криве анимације. Брзина криве представља време које анимација користи да изврши промену из једног сета стилова у други. Подразумевана вредност је `ease`. Користи се математичка функција, Безијерова крива, која одређује брзину криве. Вредности су исте као код транзиционих ефеката: `linear`, `ease`, `ease-in`, `ease-out`, `ease-in-out`, `cubic-bezier(n,n,n,n)`.

animation-delay: дефинише када ће анимација почети са извршавањем што је изражено у секундама или милисекундама. Негативне вредности су такође дозвољене. Подразумевана вредност је 0. Дефинише број секунди или милисекунди које се чекају пре него што анимација почне.

animation-iteration-count: дефинише колико пута ће се анимација поновити. Вредност може бити неки број и то означава број понављања анимације. Подразумевана вредност је 1.

Друга могућа вредност је `infinite` и значи да се анимација извршава бесконачно много пута.

`animation-direction`: дефинише да ли анимација треба да се извршава уназад. Уколико се анимација извршава само једном, ово неће имати ефекта. Вредности су:

`normal`: ово је подразумевана вредност и анимација ће нормално бити приказана.

`alternate`: анимација ће бити приказана нормално приликом сваког непарног приказивања и уназад приликом сваког парног приказивања.

`animation-play-state`: `paused|running`; одређује да ли анимација траје или је паузирана. ^[4]

Пример13:

У `.html` фајлу:

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

<body>
.....
<div id="box">
<ul id="slider">
<li id="1" >
<p><span>Table sa spiskom i pravcem beamskih &#269etvrti. </span></p>
</li>
.....
</ul>
<ul id="thumb">
<li><a href="#1"></a></li>
.....
</ul>
</div>
.....
</body>
```

У `ulice.css` фајлу:

```
@-webkit-keyframes fadeIn {
0% {
opacity: 0;
}
50% {
opacity: 0.7;
}
100% {
opacity: 1;
}
}

#box {
```

```
.....
-webkit-animation-name: fadeIn;
  -webkit-animation-duration: 3s;
  -webkit-animation-iteration-count: 1;
  -webkit-animation-delay: 0s;
}

/* Animacija za :target sliku. */

@-webkit-keyframes moveTarget {
0% {
left:-700px;
}

100% {
left:0px;
}
}

ul#slider li:target {
-webkit-animation-name: moveTarget;
-webkit-animation-duration: .5s;
-webkit-animation-iteration-count: 1;
top:0px;
left: 0px;
z-index: 10;
}

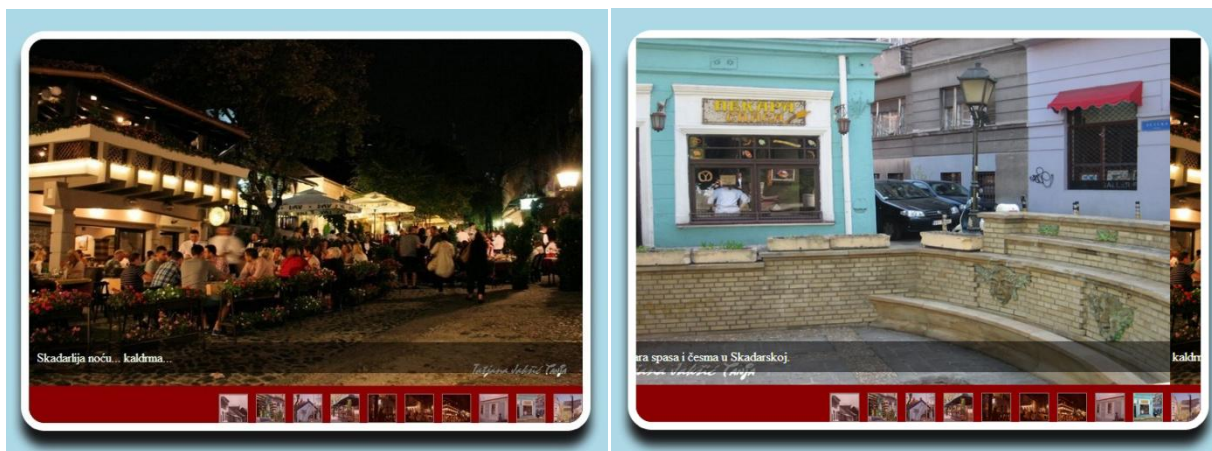
/*
Animacija za trenutnu sliku.
*/

@-webkit-keyframes moveTrenutnu{
0% {
left:0px;

}
50% {
left:700px;

}
100% {
left:-700px;
z-index: 5;
}
}

ul#slider li:not(:target) {
-webkit-animation-name: moveTrenutnu;
-webkit-animation-duration: 1.5s;
-webkit-animation-iteration-count: 1;
top:0px;
left: 0px;
}
}
```



Слика 22. Пример 13 у веб претраживачу; галерија; анимација преласка са једне слике на другу.

3) CSS3 колоне

Internet Explorer не подржава више колоне, док Firefox захтева префикс `-moz-`, Chrome и Safari `-webkit-`, а Opera подржава у потпуности.

column-count: одређује у колико колоне треба да буде подељен елемент.

column-gap: дефинише простор између колоне. Ако постоји `column-rule`, појавиће се у средини тог простора. Синтакса је: `column-count: length|normal;`. Прва вредност представља дужину простора између колоне, док је друга подразумевана и износи `1em`, односно отприлике `10px`.

column-rule: служи за постављање ширине, стила и боје у простор између колоне. Синтакса је следећа: `column-rule: column-rule-width column-rule-style column-rule-color;`. Подразумевана вредност је `medium none black`.

column-rule-color: вредност је нека боја која ће се користити за `column-rule`.

column-rule-style: дефинише стил за `column-rule`. Вредности су: `none|hidden|dotted|dashed|solid|double|groove|ridge|inset|outset;`

column-rule-width: одређује ширину простора између колоне. Вредности су: `thin|medium|thick|length;`

column-span: одређује преко колико колоне елемент треба да се простире. Синтакса је: `column-span: 1|all;`

column-width: одређује ширину колоне. Вредност може бити неки број, а подразумевана је `“auto”` при чему претраживач одређује ширину колоне. ^[4]

Пример 14:

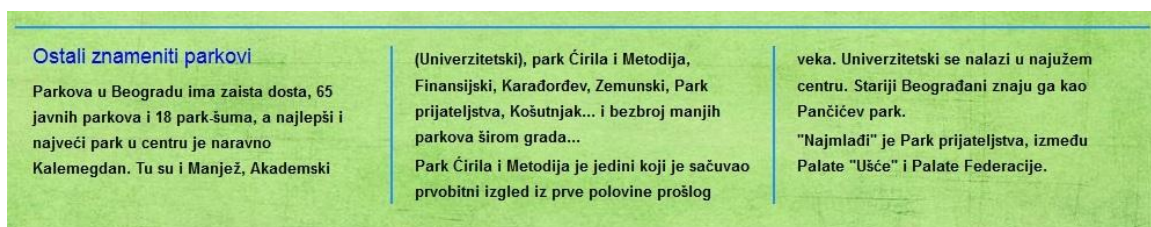
У .html фајлу:

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

<body>
.....
<section class="kolone">
<h3>Ostali znameniti parkovi</h3>
<article>
<p>Parkova u Beogradu ima zaista dosta, 65 javnih parkova... </p>
<p>Park Ćirila i Metodija je jedini koji je sačuvao prvobitni
izgled...</p>
<p>"Najmlađi" je Park prijateljstva... </p>
</article>
</section>
.....
</body>
```

У parkovi.css фајлу:

```
.kolone {
.....
-moz-column-count: 3; /* Firefox */
-webkit-column-count: 3; /* Safari and Chrome */
column-count: 3;
-moz-column-gap: 40px; /* Firefox */
-webkit-column-gap: 40px; /* Safari i Chrome */
column-gap: 40px;
-moz-column-rule: 2px solid dodgerblue; /* Firefox */
-webkit-column-rule: 2px solid dodgerblue; /* Safari i Chrome */
column-rule: 2px solid dodgerblue;
}
```



Слика 23. Пример 14 у веб претраживачу Mozilla Firefox.

4. JavaScript

4.1. Карактеристике, историја и развој JavaScript –а

JavaScript је скрипт језик који је динамичан, слабо типизиран и има првокласне функције. Подржава објектно-оријентисане, императивне и функционалне стилове програмирања. Синтакса JavaScript-а настала је под утицајем програмског језика C. Скриптови се извршавају без претходне компилације. Како се веб развијао, расле су и аспирације креатора. Дизајниран је да би се додала динамичност и интерактивност на HTML стране и обично је директно уграђен у HTML стране.



Слика 24. Незванични лого JavaScript-а и Брендан Ајк.

JavaScript је оригинално развио Брендан Ајк, програмер из Пенсилваније, сада један од водећих у компанији Mozilla. Тада је добио назив Mocha, а затим је преименован у LiveScript. Од свог настанка, прилично је еволуирао. Java, програмски језик за тржиште је развила компанија Sun Microsystems и управо у слично време добијао је доста публицијетета. Када је Netscape прерадио претраживач и направио нову верзију Navigator 2 септембра 1995. године, искористио је маркетиншки трик и преименовао LiveScript у JavaScript 4. децембра 1995., надајући се да ће преузети део славе програмског језика Java. Microsoft је увидео популарност овог скрипт језика, па су у тој компанији направили сопствени језик који у великој мери ради као JavaScript. Назван је JScript. ^{[2] [8]}

4.2. Веза између HTML5, CSS3 и JavaScript-а

JavaScript додаје интерактивност на веб странице. Коришћењем JavaScript-а омогућава се много лепши и узбудљивији утисак приликом прегледања веб страница. Свакако, то доприноси повећању радозналости корисника и може бити корисно, како креатору веб сајта, тако и самом кориснику. Одлична структура у HTML5, дизајн осмишљен у CSS3 и интерактивност и динамика у JavaScript-у омогућавају изузетан изглед веб странице.

JavaScript допушта прављење корисничког интерфејса, дајући корисницима повратне информације док прегледају страницу. Рецимо, допушта разне реакције када корисник прелази мишем преко неких локација на веб страници. Такође, JavaScript омогућава да се притиском дугмета изврши нека функција. Може се користити за проверу уноса података у обрасце, а уколико обрасци захтевају израчунавања, она се обављају на страни клијента без икакве обраде на страни сервера. Код попуњавања образаца не мора се чекати да се попуне сва поља, па онда вршити проверу, она је могућа и након сваког уноса. JavaScript пружа могућност генерисања сатова, календара пошто има скуп временских и датумских функција. JavaScript се користи да поставља и чита такозване колачиће, мале фајлове који користе податке у вези са коришћењем сајтова. Може се контролисати када је последњи пут посећен сајт, које је рекламе корисник видео итд. Има велику улогу приликом онлајн куповине, па се уз помоћ JavaScriptа додају артикли, процесира се формулар и шаљу се наруџбине. JavaScript може избацивати одређена упозорења уколико, рецимо, дође до погрешног уноса приликом попуњавања формулара. Користи се и за безбедност на интернету кад је реч о уношењу шифре на сајтовима који захтевају логовање.



Слика 25. Симболичан приказ везе између HTML5, CSS3 и JavaScript-а.

4.3. Основна својства JavaScript-а

Скриптови се могу поставити између ознака <head>и </head>или између <body>и </body>.

Следећи део кода:

```
<script type="text/javascript">
document.write("Zdravo, svete!");
</script>
```

исписаће на страници текст „Здраво, свете!“.

Неки веб претраживачи не подржавају JavaScript, па се убацују одређени коментари који сакривају скрипт од таквих претраживача. Претходни пример треба да изгледа овако:

```
<script type="text/javascript">
<!--
document.write("Zdravo, svete!");
//-->
</script>
```

Коментари који стају у један ред пишу се са // иза којих следи коментар, а дужи се наводе између /* и */.

Да би се убацили специјални знакови, као што су апостроф, наводници, знак за нови ред, таб, коса црта, испред њих се додаје знак за косу црту, односно \.

Може се користити и екстерни JavaScript фајл који се уметне у оквиру <head> и </head> тагова, а то се записује овако:

```
<script type="text/javascript" src="xxx.js"></script>.
```

Варијабле у JavaScript језику означавају се са кључном речи var.

Користе се аритметички оператори, оператори доделе, оператори поређења и логички оператори. Такође, у употреби су условне изјаве **if**, **if... else**, **if... else if... else**, **switch**, петље **for**, **while**, кључне речи **break** и **continue**.

Користе се и прозорчићи узбуне, потврђивања и прозорчићи са пољем за унос. Код првог синтакса је: `alert("неки текст");`, код другог `confirm("неки текст");`, а код трећег `prompt("неки текст", "подразумевана вредност");`.^[8]

4.4. jQuery

jQuery је позната библиотека JavaScript-а која знатно олакшава програмирање у JavaScript-у.

Додаје се на страницу на следећи начин:

```
<head>
<script type="text/javascript" src="jquery.js"></script>
</head>
```

Синтакса је: `$(selector).action()` при чему долар јесте знак који дефинише jQuery, (`selector`) служи да се пронађу HTML елементи, док је `action()` део где се задаје одређена акција која треба да се испуни. ^[9]

Примери:

`$(this).hide()` – сакрива тренутни елемент

`$("p").hide()` – сакрива све параграфе

`$("p.test").hide()` – сакрива све параграфе где је `class="test"`

`$("#test").hide()` – сакрива елемент где је `id="test"`

Све методе се налазе унутар следећег кода:

```
$(document).ready(function(){
});
```

То је урађено да би се спречило извршавање jQuery кода пре него што је документ спреман.

Селектори

jQuery селектори омогућавају да се селекује и манипулише HTML елементима. Они су потребни у сваком тренутку када се користи jQuery. Они омогућавају да се из HTML документа добије одређени елемент/атрибут. CSS селектори су подржани.

5. Опис презентације и начин коришћења

Примена HTML5 и CSS3, као и JavaScript-а приказане су у оквиру веб презентације са називом “*Топ дестинације у Београду*”. Презентација је оптимизирана за Firefox, Opera-у, Google Chrome, Internet Explorer 9, Safari. Internet Explorer 9 најмање подржава нове могућности HTML5 и CSS3. Најбољи преглед је у Mozilla Firefox-у, у резолуцији 1366px x 768px. Код је куцан у Notepad++-у.

Почетна страна је “bg.html”. Одмах на врху стране може се уочити примена нових `<audio>` и `<source>` елемената (странице 14., 15. и 16., Пример 4). Песме о Београду (Серенада Београду, Београде, Мој драги Београд) су у формату .mp3, .wav и .ogg, јер претраживачи подржавају различите формате.



Слика 26. Аудио и главни мени на почетној страни презентације у веб претраживачу Mozilla Firefox.

Главни мени је дат у оквиру табеле са 2 реда и 5 колона. У свакој ћелији је слика са линком који води на страницу о датој топ дестинацији. Тако су у првом реду:

Тврђава, Тргови Београда, Улице Београда, Ада Циганлија, Земунски кеј, док су у другом реду: Савски кеј, Храм Св.Саве, Паркови Београда, Мостови Београда, Авала.

Наслови су дати у оквиру `<header>` тага (страница 8.). Испод се налазе цитати о Београду у оквиру `<section>` тага (страница 8.), док се сваки цитат налази у оквиру `<article>` тага (страница 8.). Испод је табела са линковима ка корисним званичним сајтовима: Туристичка организација Београда, Туристичка организација Србије, сајт града Београда, сајт са виртуелним шетњама кроз град Београд, Београд 2020., затим: Belgrade Greeters, Аеродром Никола Тесла, Београдска арена, Београдски сајам, ПланПлус.

На дну сваке стране налазе се линкови ка главној страни, као и име аутора презентације, и то све у оквиру `<footer>` тага (страница 8.).

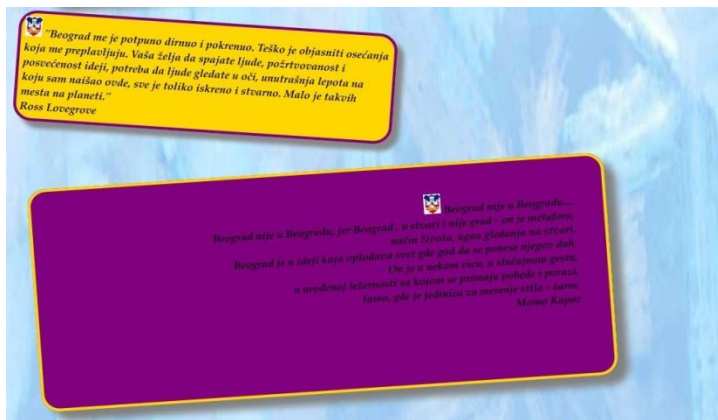
Што се CSS3 тиче, користи се ново својство на овој страни, а то је `background-size` са вредностима `100% 100%` (странице 26. и 27.).

Такође, `@font-face` (странице 28. и 29.), где су дефинисани фонтови Jockerman и DisneyFont који се користе у насловима одељака и линкова и у потпису аутора.

`border-image` (странице 24. и 25.) се јавља код тага `<table class="centar">`. У оквиру ове табеле налазе се линкови ка званичним сајтовима. Синтакса је:

```
table.centar
{
.....
-moz-border-image: url(slike/border.png) 15 15 15 15 stretch stretch;
-webkit-border-image: url(slike/border.png) 15 15 15 15 stretch stretch;
-o-border-image: url(slike/border.png) 15 15 15 15 stretch stretch;
border-image: url(slike/border.png) 15 15 15 15 stretch stretch;
}
```

На више места употребљен је `border-radius` са вредношћу `25px` (странице 22. и 23.), а код сваког цитата има и `box-shadow` (странице 23. и 24., Пример 6), као и `transform:rotate(4deg)` и `transform:rotate(-4deg)` (странице 29. и 30., Пример 11).



Слика 27. Цитати на почетној страни презентације у веб претраживачу Mozilla Firefox.



Слика 28. Табела са линковима ка званичним сајтовима на почетној страни презентације, у веб претраживачу Mozilla Firefox.

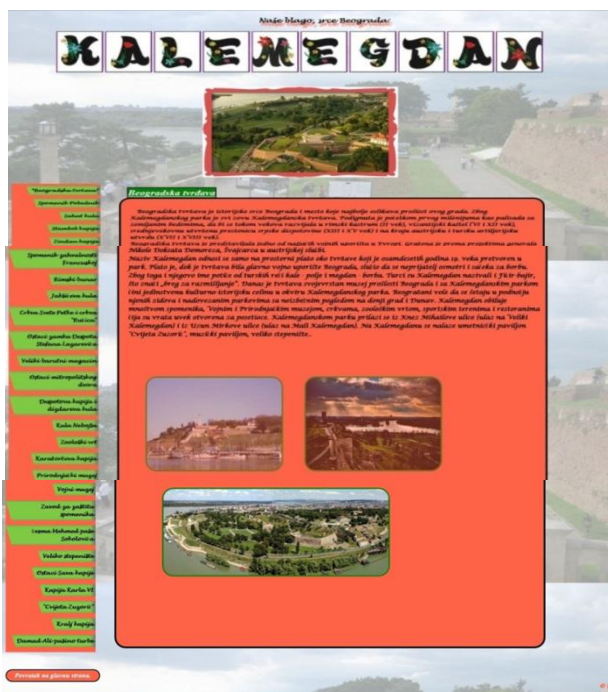
Први линк – **Тврђава**, води на страницу о Калемегданској тврђави и парку. Поред тагова као што су <header>, <section>, <article>, <footer> који су углавном на свакој страни, овде се налази и таг <nav> (страница 8.) где је листа линкова ка разним деловима тврђаве. Код тог менија користе се следећа својства и вредности CSS3: **box-shadow: inset -7px 0 5px rgba(114,114,114, 0.8);** (странице 23. и 24.) и **transition: all 0.2s ease-in-out;** (странице 31. и 32.) којом се постојећа својства преласком мишем преко изабраног линка у менију замењују следећим:

```
ul.vertikalnimeni li a:hover
{padding-right: 30px;
color: black;
background: rgb(153,249,75);
-moz-box-shadow: inset -3px 0 2px rgba(114,114,114, 0.8);
-webkit-box-shadow: inset -3px 0 5px rgba(114,114,114, 0.8);
box-shadow: inset -3px 0 5px rgba(114,114,114, 0.8);}
```

На тај начин линк се продужи улево, боја се промени у другу врсту зелене и другачија је сенка.

У тексту при врху користи се следећи CSS3 код: **text-shadow:4px 10px 2px tomato;** (странице 27. и 28.), а код главне слике испод слова К-А-Л-Е-М-Е-Г-Д-А-Н, користи се код за додавање рама (Пример 7). Постављен је и **background-size:1366px 768px;**.

Слике у садржају имају opacity=0.6, али преласком мишем преко, мењају им се дужина и ширина (уколико нису оне код којих је ширина 500px, а висина 200px), а opacity постаје 1.0, заправо слика постаје потпуно видљива. Заобљене ивице (border-radius) примењују се на више места.



Слика 29. Страница о београдској тврђави у веб претраживачу Mozilla Firefox.

Тргови Београда је други линк на главној страни. Ту се налази кружни мени са линковима на странице о трговима. Кружни мени је постигнут коришћењем `border-radius: 400px;`. За наслов користи се `text-shadow` (странице 27. и 28., Пример 9). На свакој страни налази се текст са описом трга, такође и фотографије. Већина фотографија у веб презентацији преузета је са интернета, а личне фотографије садрже потпис.



Слика 30. Врх странице о трговима Београда у веб претраживачу Mozilla Firefox.

Преласком мишем преко фотографије, излази текст са њеним описом. Користи се: `transition: all 0.5s ease-in-out;`. Појављује се сенка на ивицама фотографије коришћењем `box-shadow: 15px 15px 15px rgba(0, 0, 0, 0.5);`. Фотографија се тада и мало повећа због `transform: scale(1.05, 1.05);`. Код описа сваке слике `z-index` се поставља на 1 (мањи је од `z-index`-а слике који је 2) да би био сакривен. Позадина је црна, али је њена видљивост 80%. Овде је `border-radius: 0 0 8px 8px;`, `box-shadow: 0 0 6px rgba(0, 0, 0, 0.8)`, а `transition: all 0.5s ease 0.5s;` при чему се користи `transform: translate(0, 100%);` док је видљивост 100%.

Вредности за `translate` и `border-radius` су различите у зависности од тога на којој страни желимо да се појављује опис фотографије. Ако опис треба да буде десно, онда је `border-radius: 0 8px 8px 0;` и при преласку мишем `transform: translate(100%, 0);`, ако треба лево онда је `border-radius: 8px 0 0 8px;` и `transform: translate(-100%, 0);`, а ако је потребно да иде на горе онда је `border-radius: 8px 8px 0 0;` и `transform: translate(0, -100%);`, у супротном важи дефинисано.



Слика 31. Опис слике на страници о трговима у веб претраживачу Mozilla Firefox.

Улице Београда су следећи линк на главној страни. У менију се налазе неке од главних улица. Скадарлија је подразумевано селектована улица. На свакој страни налази се опис улице и галерија фотографија која се понаша као slide-show у Chrome-у и Safari-ју. Она садржи и описе фотографија. Кликом на мале иконе у галерији, отвара се већа слика, као и њен опис у дну фотографије. Овде је ново дефинисање анимације (странице 33., 34., 35. и 36., Пример 13).



Слика 32. Галерија на страници о улицама Београда у веб претраживачу Google Chrome.

У наслову се користи сенка текста, тј. `text-shadow: 5px 5px 5px #d50509;`

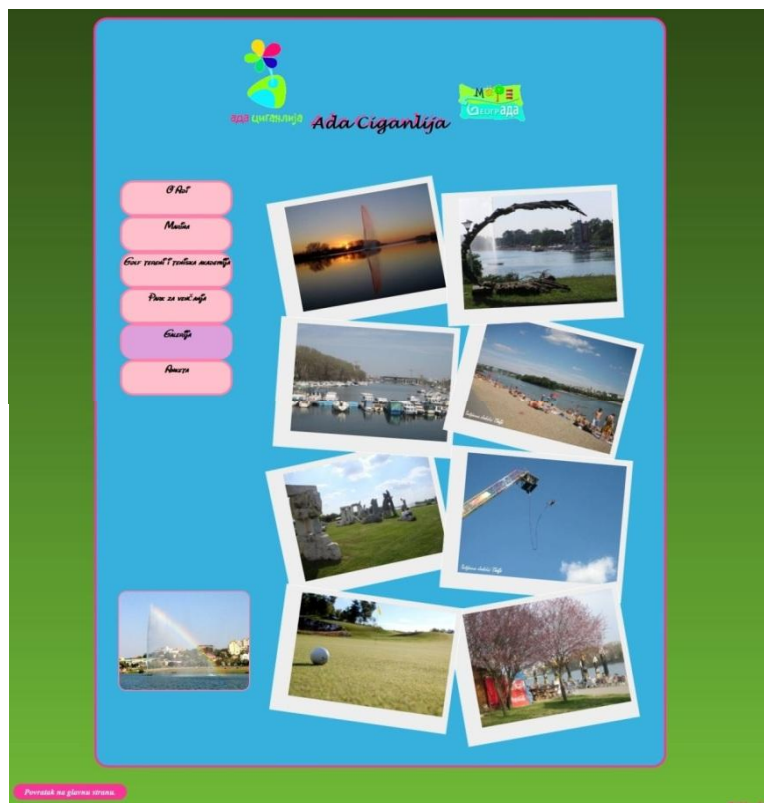
Ulice Beograda

Слика 33. Коришћење сенке текста на страници о улицама Београда.

Превлачењем миша преко табова у менију чује се звук, звиждук. То је уметнут скрипт, написан у JavaScript-у. Изражено је коришћење `border-radius`-а и `box-shadow`-а.

Четврти линк јесте **Ада Циганлија**. У свом менију има 6 делова: О Ади, Марина, Голф терени и тениска академија, Парк за венчања, Галерија и Анкета. Галерија се заснива на коришћењу `-moz-transform:rotate(неки степен)` и коришћењу z-index-а. Превлачењем мишем преко слика, ивице се померају. Кликом на сваку од фотографија, отвара се страна са датом фотографијом у пуној величини. Фотографија и њен наслов налазе се у оквиру `<figure>` тага, при чему је наслов у `<figcaption>` тагу (странице 9. и 10., Пример 1). Анкета се шаље на мој е-mail. Од нових опција овде су `autocomplete`, `required placeholder` и `input type=email` (странице 11., 12. и 13.).

На више места искоришћен је `border-radius` (код менија, анкете и сличице у доњем левом углу, као и код линка за повратак на главну страну). `box-shadow:0 0 30px #111;` користи се за сенку код дугмета на анкети.



Слика 34. Изглед странице Галерија Аде Циганлије у веб претраживачу.

Пета област је **Земунски кеј**. Испод наслова налази се слика у раму. Преласком мишем преко ње јавља се друга фотографија кеја. Испод тога налази се мени који садржи следеће линкове: Земунски кеј, Гардош, Галерија, Анкета. Преласком мишем преко ових кружних табова добија се, редом: инспирација за многе; симбол Земуна; фотографије кеја; утисци, сугестије. За промену две слике кључна ствар је `transition: opacity 1s ease-in-out;` (Пример 12) и `opacity:0` при преласку мишем;, док је за анимацију кружног менија који ротира и мења боју кључно: `transition: all 400ms linear;` и `transform: rotate(360deg);`.

За сличице у оквиру кружног таба захвалан је фонт WebSymbolsRegular. Странице о Земунском кеју и Гардошу садрже описе истих. Галерија садржи 17 фотографија. Користи се `transition: all linear 500ms;` при чему се мења `z-index`. Кликком на број фотографије дешава се дата транзиција и прелази се на нову фотографију. У анкети ново је `autocomplete="on"`, `required placeholder`, затим `input type="email"` и `input type="date"` (који још увек не функционише како треба у Mozilla Firefox-у) (странице 11., 12. и 13., Пример 3). У оквиру једног питања у анкети, при селектовању одговора 'Да' или 'Не', одговор се мало помери удесно захваљујући употреби:

```
input[type="checkbox"]:checked
{ transition: width 1s ease; },
input[type="checkbox"]:checked
{ width: 30px; } .
```

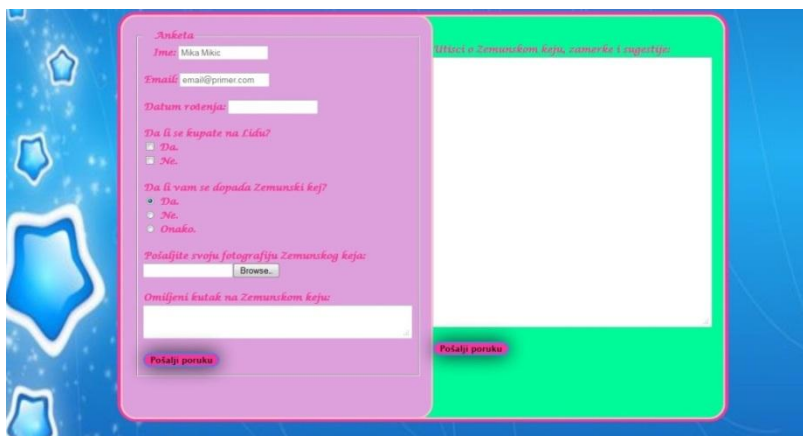
Карактеристично је коришћење `border-radius`-а, а за наслов задужено је својство `text-shadow` и уметнути фонт (28. и 29. страна, Пример 10).



Слика 35. Изглед главне странице о Земунском кеју у веб претраживачу Mozilla Firefox.

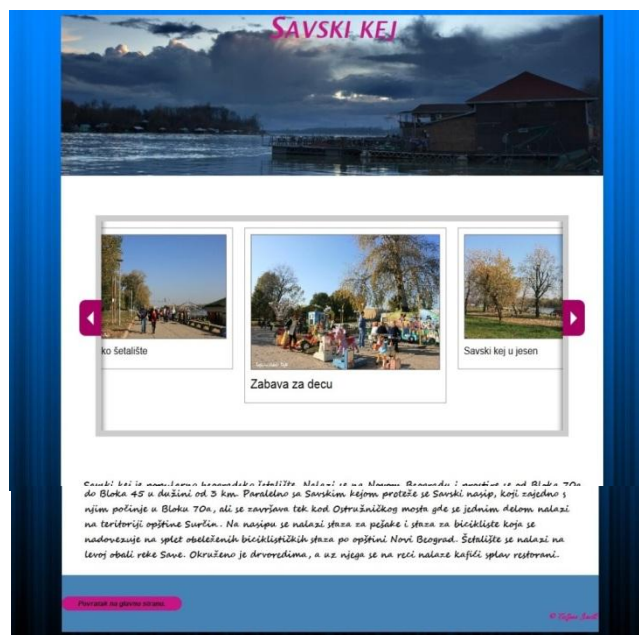


Слика 36. Галерија Земунског кеја у веб претраживачу Mozilla Firefox.



Слика 37. Анкета о Земунском кеју у веб претраживачу Mozilla Firefox.

Страна **Савски кеј** садржи краћи текст о кеју и галерију фотографија урађену у jQuery-ју (41. страна). Неке фотографије садрже и линк ка званичном сајту неког од сплавова на Савском кеју.



Слика 38. Страна о Савском кеју у веб претраживачу Mozilla Firefox.

Страница о храму Светог Саве садржи једну издужену слику у наслову стране и три колоне где свака изнад наслова и кратког описа има и фотографију. Око фотографија користи се `border-image:url(ms_slike/redborder.jpg) 20 20 round;`. У оквиру заглавља наслов је анимиран и константно прелази из беле у жуту боју. Употребљена је анимација:

```
@-moz-keyframes mojaAnimacija /*Firefox*/
```

```
{from {color: white;}
```

```
to {color: yellow;}} и -moz-animation: mojaAnimacija 5s ease 1.5s infinite; у .css фајлу за header.
```

Колоне су: Архитектура, Изградња и Галерија. Колоне садрже линк уколико желимо више да прочитамо о одређеној теми. Атрибут `background-size` описан је у Примеру 8 (26. и 27. страна). У галерији је употребљено својство `transition: all .3s ease-out;`, где се при преласку мишем преко слике користи `transform: scale(1.2);` и слика се зумира.



Слика 39. Изглед главне странице о храму Светог Саве у веб претраживачу Mozilla Firefox.

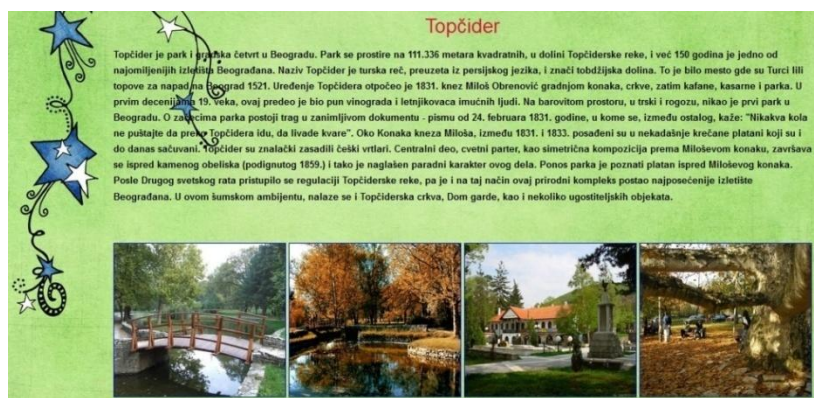


Слика 40. Изглед галерије храма Светог Саве у веб претраживачу Mozilla Firefox.

Паркови Београда приказују три прелепа парка: Ташмајдан, Топчидерски парк и Пионирски (Дворски) парк. Остали паркови су набројани, а неки су иначе већ поменути и описани у оквиру других страница. О сваком од ова три парка може се више прочитати. Оно што је ново овде јесте коришћење `column-count: 3`; за дефинисање три колоне. Такође `column-gap: 40px`; за размак између колона и `column-rule: 2px solid dodgerblue`; за ширину, стил и боју линије која раздваја колоне (странице 36. и 37., Пример 14).



Слика 41. Изглед главне странице о парковима Београда у веб претраживачу Mozilla Firefox.



Слика 42. Детаљ са странице о Топчидерском парку у веб претраживачу Mozilla Firefox.

Мостови Београда садрже на свакој страни на врху слику датог моста и цитат нашег нобеловца Иве Андрића. На почетној страни налази се осам подеока: Бранков мост, Газела, Савски мост, Панчевачки мост у горњем реду и Железнички мостови преко Саве, Остружнички друмски и железнички мост, Мост на Ади, Мост Обреновац-Сурчин, у доњем реду. У оквиру footer-а може се ући и на страну о Кинеском мосту пријатељства.

Од елемената поред већ поменутих нових, овде се користи и `<aside>` (страна 8.). О сваком мосту се може више прочитати. Информације прати и одговарајућа фотографија, а са стране су сличице осталих мостова, које служе као пречица за брже бирање преосталих страна о мостовима.

Користи се и скрипт за једнаке колоне, а типично је коришћење `border-radius`-а (стране 22. и 23., Пример 5).

Скрипт:

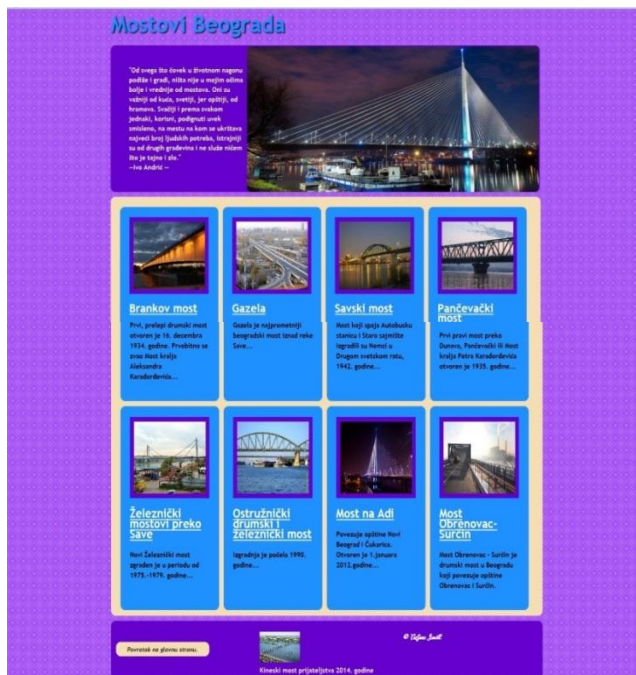
```
function smartColumns() {
    $("ul.column").css({ 'width' : "100%" });

    var colWrap = $("ul.column").width(); // Узима се ширина реда.
    var colNum = Math.floor(colWrap / 200); // Испитује се колико колона од
    200px може да стане у ред и онда се заокружује на цео број.
    var colFixed = Math.floor(colWrap / colNum); // Узима се ширина реда и
    дели бројем колона које може да садржи, потом се заокружује на цео број. Ова
    вредност ће бити тачна ширина нове колона.

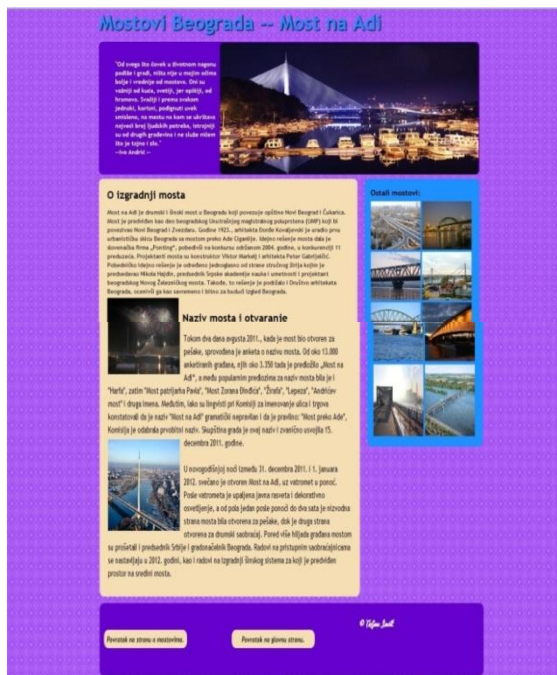
    $("ul.column").css({ 'width' : colWrap }); // Поставља се тачна ширина
    реда у пикселима уместо коришћења %.
    $("ul.column li").css({ 'width' : colFixed }); // Поставља се тачна
    ширина нове колона.
}
```

`smartColumns();` // када се страница учита, стартује се функција.

```
$(window).resize(function () {
    smartColumns();
});
```



Слика 43. Изглед главне странице о мостовима Београда у веб претраживачу Mozilla Firefox.



Слика 44. Страна о Мосту на Ади у веб претраживачу Mozilla Firefox.

Страна о **Авали** не садржи линкове на друге стране. Овде се користи нови HTML5 таг `<mark>` (10. страна, Пример 2), па су њиме означени наслови чланака. Испод је мала галерија. Фотографије имају оквир. Фонтови у оквиру `@font-face-a` су `BoycottRegular` и `WDScript (DisneyFont)`. У галерији је употребљено својство `transition: all 1s ease-out;`, где се при преласку мишем преко слике користи `transform: rotate(360deg);` и слика се ротира за 360 степени.

Код за линију испод наслова је `background: linear-gradient(left, #00aeff 1%, #71fc00 15%, #ffff00 30%, #fc7a00 50%, #cc00a6 70%, #ff0084 85%, #ff0000 100%);`^[10].



Слика 45. Део стране о Авали у веб претраживачу Mozilla Firefox.



Слика 46. Галерија на страни о Авали у веб претраживачу Mozilla Firefox.

6. Закључак

HTML5 је врло моћан језик за структурирање података, пун нових функција, техника за оптимизацију кода и много других ствари које су нам од велике користи. Најочигледнија корист уграђена у HTML5 су бројне API могућности које отварају будућност креирања веб апликација. Најбољи део HTML5 је дефинитивно његова “offline” способност чувања података. Програми попут Thunderbird-а, Outlook-а или Gmail-а омогућавају прегледања старих података током периода док корисник није повезан на интернет. Са HTML5 верзијом, постоји иста функционалност, али у претраживачу. Ово је први озбиљан корак према премошћивању јаза између десктоп-а и веб-а, и отвара врата за будућност веб апликација. У HTML5 поједностављено је прецизно утврђивање локације тако што је омогућен приступ и коришћење GPS API-а, дозвољено је креирање различитих графичких елемената, анимација и игрица за 2D пројекције на екрану. Омогућен је директан приступ мултимедијалним садржајима у оквиру веб странице и директан приступ сликама, аудио фајловима, email пошиљкама и другим документима са мобилног уређаја директно из претраживача. Сада је могуће користити много комплексније форме за унос података са уграђеним системом за проверу њихове валидности директно из самог претраживача.

Услед технолошког развоја мобилне апликације еволуирају у свој следећи облик: мобилне веб апликације. Власници свих смарт телефона ће моћи да користе HTML5 апликације. Тиме је флексибилност ових апликација знатно већа.

Предност јесте и количина новца које је потребно издвојити за развој оваквих апликација. Далеко је повољније развити мобилну веб апликацију која ће функционисати на свим платформама смарт телефона, него посебно развијати апликације за Android, iPhone, Blackberry, Windows итд. Углавном сви смарт телефони који се данас могу наћи на тржишту поседују интернет претраживаче који подржавају HTML5, JavaScript и CSS3. Ово омогућава компанијама да развијају своје мобилне сајтове са напредним графичким садржајима и са богатим мултимедијалним функционалностима које остављају веома јак визуални утисак.

HTML5 и CSS3 користиће се за изградњу јачих, богатијих веб-апликација у годинама које долазе.



Слика 47. Лого HTML5 и лого CSS3.

Садржај:

1. Увод	1
1.1. Интернет	1
1.2. World Wide Web (WWW)	2
2. HTML	3
2.1. Историја и развој HTML-а	3
2.2. Структура HTML документа	4
2.3. HTML5	5
2.3.1. <i>Историја и стандардизација</i>	5
2.3.2. <i>Ознаке у HTML5 за структурирање садржаја стране</i>	7
2.3.3. <i>HTML5 обрасци</i>	11
2.3.4. <i>HTML5 мултимедијални елементи</i>	14
2.3.5. <i><canvas> таг</i>	16
3. CSS	17
3.1. Карактеристике, историја и развој CSS-а	17
3.2. CSS3	19
3.2.1. <i>Синтакса CSS-а</i>	20
3.2.2. <i>Специфичности CSS3</i>	21
4. JavaScript	38
4.1. Карактеристике, историја и развој JavaScript -а	38
4.2. Веза између HTML5, CSS3 и JavaScript-а	39
4.3. Основна својства JavaScript-а	40
4.4. jQuery	41
5. Опис презентације и начин коришћења	42
6. Закључак	54

Референце:

- [1] *Рачунарство и информатика за IV разред гимназије -- Миодраг Стојановић; Завод за уџбенике и наставна средства, Београд, 2004.*
- [2] www.wikipedia.org
- [3] *“Бриљантно: HTML5 и CSS3” – Josh Hill, James A. Brannen; CET i Portalibris, 2011.*
- [4] <http://www.w3schools.com/>
- [5] *“Introducing HTML5” – Bruce Lawson, Remy Sharp; New Riders, 2012.*
- [6] *“Dive Into HTML5” – Mark Pilgrim; O’Reilly Media, 2010.*
- [7] www.tutorialspoint.com
- [8] *“JavaScript za World Wide Web” – Tom Negrino, Dori Smith; CET, 2005.*
- [9] <http://docs.jquery.com/Tutorials>
- [10] <https://developer.mozilla.org/en-US/docs/CSS/linear-gradient>