

**SVEUČILIŠTE U ZAGREBU**  
**FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA**

**Sustavi za praćenje i vođenje procesa**  
Seminarski rad

**ECMA skripte**

28. 05. 2007.

Ivan Miloš  
0036406642  
INE

# 1. Sadržaj

1. Sadržaj.....	2
2. Uvod.....	3
3. Povijest.....	5
4. Verzije i dijalekti.....	10
5. Zaključak.....	14

## 2. Uvod

*ECMA Script* je client-side skriptni programski jezik, koji svoju primjenu nalazi na Webu. U biti, riječ je o standardu koji postavlja *Ecma International*<sup>1</sup>. Dokument koji specificira jezik *Ecma Script* zove se *ECMA-262* i dostupan je na <http://www.ecma.ch/ecma1/STAND/ECMA-262.HTM>.

*Ecma International* je međunarodna, privatna organizacija sa sjedištem u Ženevi, čiji je cilj standardiziranje informacijskih i komunikacijskih sustava. U preko četrdeset godina svoga djelovanja aktivno je doprinjela standardizaciji informacijskih i telekomunikacijskih tehnologija na svjetskoj razini čemu svjedoči preko 370 objavljenih ECMA standarda i 90 tehničkih izvješća. Međutim, najbolji dokaz kvalitete rada ove organizacije su sljedeće činjenice: preko dvije trećine izdanja ove organizacije prihvaćena su kao međunarodni standardi, a među inim odgovorna je i za službene specifikacije: jezika *C#*, jezika *C++*, *CLI-a*<sup>2</sup>, te *Office Open XML* formata.<sup>3</sup>

Vrlo je interesantan ustroj same organizacije. Za razliku od nacionalnih standardizacijskih odbora, *ECMA* je članski orijentirana organizacija sa poslovnim pristupom standardiziranju. Takav pristup vodi do boljih rezultata u minimalnom vremenskom razdoblju zahvaljujući

---

<sup>1</sup> *ECMA* je akronim za *European Computer Manufacturers Association*, i to je bilo ime organizacije do 1994. god. kada je promijenjeno u *Ecma International*, te se više ne smatra akronimom, a kako bi se istaknula međunarodna orijentacija organizacije

<sup>2</sup> *Common Language Infrastructure* je otvorena specifikacija razvijena od strane Microsofta koja definira izvršni kod i runtime okolinu koja čini jezgru *MS .NET* frameworka. Takva definicija omogućava da se razni viši jezici koriste na različitim platformama bez potrebe da se pišu različite verzije za svaku specifičnu arhitekturu platforme. Implementacija *CLI-a* je *Common Language Runtime*.

<sup>3</sup> Od posljednje verzije *MS Office-a* dokumenti su dostupni i u *XML* formatu. Stvar je vrlo praktična budući da se sad, recimo umetnutoj slici, može fizički pristupiti sa diska, bez potrebe za otvaranjem dokumenta u *MS Office-u*, što uvelike olakšava posao aplikacijskim programerima

slabije birokratiziranom procesu razvoja fokusiranom na dobivanje rezultata konsenzusom.

Također, u svom *Ecma International Standard 370*, organizacija je udružila snage sa skandinavskom *IT Eco declaration*, kako bi postavila smjernice o informiranju korisnika u vezi poštovanja zaštite okoliša od strane proizvođača informacijskih i komunikacijskih tehnologija te potrošačke elektronike. *IT Eco declaration* sadrži sve ekološki relevantne informacije o proizvodu.

### 3. Povijest

Sve je počelo 1995.god. kada je u *Netscape Navigator*<sup>4</sup> implementirana podrška za *Java-applete*<sup>5</sup>. Pojavila se potreba za jednostavnim i moćnim skriptnim jezikom koji bi mogli koristiti web-dizajneri i programeri koji ne poznaju *Javu*<sup>6</sup>.

*Netscape Communications* zaposlio je Brendana Eichu kao glavnog i odgovornog inženjera za razvoj i implementaciju novog jezika. U skladu s nametnutim zahtjevom, Eich je odlučio da će novi jezik biti slabo obilježen<sup>7</sup> skriptni jezik, budući da takva svojstva, osim korisnicima, odgovaraju i okolini u kojoj će se izvoditi programi tj. skripte. Također, korisnici su morali biti u mogućnosti povezati elemente stranice kao što su forme, okviri<sup>8</sup> i slike bez *Java-bytecode* prevodioca i poznavanja principa objektno-orijentirane paradigme.

Novostvoreni jezik prvotno je nazvan *Mocha*, zatim *LiveScript* (kako bi naglasio dinamičku prirodu jezika) da bi konačno bio preimenovan u *JavaScript*. Razlog za takvo ime bio je marketinške prirode, međutim ispostavio se kao velika greška budući da je to ime uzrokovalo i dan-danas uzrokuje zabune kod velikog broja korisnika koji miješaju *JavaScript* sa *Javom*. Kao takav implementiran je u *Netscape Navigator 2.0* beta verziju.

U prosincu 1995.god. *Netscape Communications* i *Sun Microsystems* zajednički su najavili novi jezik, nazivajući ga komplementom *Jave* i *HTML*-a.

---

<sup>4</sup> Jedan od popularnih web-browsera razvijen od strane *Netscape Communications*.

<sup>5</sup> Programi pisani u Javi namijenjeni izvršavanju u striktno ograničenoj okolini – web-browseru. Pružaju sofisticiranu podršku web-stranicama daleko moćniju od *DHTML*-a i *JavaScripta*.

<sup>6</sup> Popularan, jednostavan i izuzetno moćan objektno-orijentirani jezik, razvijen za upotrebu u raspodijeljenim okolinama i na Internetu. Razvijen od strane *Sun Microsystems*-a.

<sup>7</sup> eng. loosely-typed

<sup>8</sup> eng. form

*JavaScript* je vrlo brzo prihvaćen. U početku je velika većina napisanih skripti koristila samo da zamjenjuje slike u skladu sa korisničkim klikanjem miša. Takva situacija posljedica je tadašnjeg trenda povećanja interaktivnosti i sofisticiranijeg korisničkog sučelja nasuprot do tada, statičkom webu.

Zapanjujući uspjeh *JavaScript* zahvaljuje upravo svojoj jednostavnosti: nema potrebe za prevodiocem/prevođenjem, skripte se mogu copy-paste-ati u već postojeće HTML stranice bez potrebe za promjenom koda, ili sa vrlo malo promjena. Iskusni programeri pisali su skripte gotovo kao da je riječ o dječjoj igri. U početku je postojao nesrazmjer između moćnog *JavaScripta* i relativno slabog rendering-engine-a uzrokujući probleme sa bljeskanjem na korisničkoj strani pri izvršavanju.

Nedostaci *JavaScripta* bili su sljedeći: nedostatak odgovarajućeg razvojnog okruženja<sup>9</sup>, nedostatak pouzdanog debugger-a neovisnog o platformi na kojoj se izvodi, te s tim u uskoj sprezi, mogućnost testiranja isključivo u kontekstu web – browsera u kojem će se stranica gledati; potom nekoliko javno eksponiranih sigurnosnih nedostataka i nekoliko knjiga o *JavaScriptu* čija su ciljana skupina bili ne-programeri, uzrokovali su da se *JavaScript* podcijeni.

Naredna izdanja *Netscape Navigatora* uključivala su podršku za skriptno orijentiranu interakciju sa korisnicima, rigorozniji sigurnosni model itd. *Netscape-ov LiveWire*<sup>10</sup> predstavio je *JavaScript* kao server-side jezik sposoban za postavljenje upita bazama podataka i drugih naprednih mogućnosti, međutim server-side skriptiranje nikada nije dostiglo popularnost primjerice jednog *ASP-a*, *Perl-a* ili *Cold Fusion-a*. Također, takve su mogućnosti zanemarivali svi, izuzev *Netscape-ovih*

---

<sup>9</sup> eng. IDE – Integrated Development Environmet

<sup>10</sup> komponenta *Netscape Navigatora* koja implementira potrebnu funkcionalnost za razvoj web-stranica i uključuje *site manager* za održavanje *HTML* linkova, prevodioca za razvoj server-side *JavaScript* skripti te knjižnicu funkcija potrebnih za spajanje na razne baze podataka. Iako u početku odvojeni proizvod, uključen je u standardne alate *Netscape Enterprise Servera* od verzije 3.0

server korisnika. Sudbina *JavaScript*-a bila je da služi kao alat pomoću kojega su autori web-stranica kontrolirali i proširivali mogućnosti klijenta.

*Microsoft* je odgovorio izdavanjem svog *VBScript*-a koji je bio sposoban «kačiti» se na ugrađene komponente, iako je ta funkcionalnost prvotno bila ograničena na platformu – Windowse, gdje su takve komponente po prvi put i implementirane. Također, sredinom 1996. god., izdali su jezik vrlo sličan *JavaScript*-u, *JScript*. *Internet Explorer* 3.0, inače sjajno tehničko postignuće sa stajališta podrške tada novim *W3C*<sup>11</sup> standardima; kao primjerice *Cascading Stylesheet* tehnologijom, ipak je za jednu reviziju zaostajao za *Navigator*-ovim *JavaScript*-om; koji je omogućavao zamjenu slika, prestigao je *Netscape Navigator* tek od verzije 3.02. *JScript* uveo je funkcije za dobavljanje datuma koje nisu bile podložne Y2K bugu, za razliku od tadašnjeg *JavaScript*-a.

Mnogi web-dizajneri jednostavno su u svojim skriptama provjeravali da li je browser *Netscape Navigator*, te su ignorirali *Internet Explorer* kao izgubljen slučaj, baš kao što su rani CGI-programeri, ispitujući varijablu okoline browsera<sup>12</sup> tražili string «Mozilla», što je rezultiralo besmislenom konvencijom (koja, nažalost, vrijedi i danas) da se praktički svi značajniji browseri pri slanju HTTP i inih zahtjeva predstavljaju kao «Mozilla», nakon čega slijedi «compatible: <ime\_browsera>»! Ukratko, *JavaScript* je zaradio reputaciju, ne potpuno zasluženu, zapreke Webu temeljenom na platformno-neovisnom standardima.

Uvođenje *Internet Explorer*-a 3.0 i njegov nesretni nedostatak polja `document.images`<sup>13</sup> (to je ujedno i prva pojava nekompatibilnosti browsera) natjerali su *Sun* i *Netscape* da standardiziraju jezik uz pomoć *ECMA* organizacije krajem 1996. god. Gotovo istovremeno je i *Microsoft* podnio zahtjev za standardizacijom *JScript* specifikacije. Godine 1997.

---

<sup>11</sup> *W3C-World Wide Web Consortium* – organizacija koja se bavi iskorištavanjem potencijala *WWW*-a do krajnjih granica razvijanjem interoperabilnih tehnologija (specifikacija, smjernica, programske podrške i alata)

<sup>12</sup> eng. user agent environment variable

<sup>13</sup> u ovom polju nalaze se sve slike koje su na trenutnoj stranici, a dostupne su ili preko indeksa polja ili preko svog imena

*ECMA* je izbacila prvu verziju *ECMA-262* standarda, koji je do 1998. verificirao i ISO<sup>14</sup>.

Taj standard bavio se, međutim, samo jezgrenom funkcionalnošću jezika, ali je, upravo zbog standardiziranosti, bio vrlo moćan. Donešeno je i kompromisno ime novog jezika *ECMA Script*, za koje je autor Brendan Eich rekao: »ECMAScript je neželjeno kompromisno ime koje zvuči kao kožna bolest.»

U međuvremenu, *Netscape* i *Microsoft* su uveli novu 4.0 generaciju web-browsera, gdje je svaki browser imao vlastitu implementaciju vlasničkog DOM-modela<sup>15</sup>. To je izazvalo pravu pomutnju – pojavile su se mnogobrojne implementacije potrebne kako bi se osigurala kompatibilnost raznih web-browsera. Postojale su knjižnice koje su osiguravale da se *Netscape Navigator* ponaša kao *Internet Explorer*, i obrnuto. Zatim, izrađene su knjižnice koje su nekompatibilni kod «zamotavale» u konzistentni API<sup>16</sup>, ali je uskoro postalo bolno evidentno da je najmanji zajednički nazivnik potreban za istinsku neovisnost o platformi, prenisko na nivou apstrakcije za sve, osim za najosnovnije aplikacije, tako da su mnogi Windows korisnici, u potpunosti napustili *Netscape Navigator* u korist *Internet Explorer*-a.

I dok je bjesnio rat browsera, *Microsoft*, *Netscape Communications* i mnogi drugi radili su sa *W3C*-om na doista univerzalnom DOM-u, a koji bi bio kompatibilan (koliko je to moguće) sa starim modelom<sup>17</sup>. Želja da se Web pretvori u platformu inspiriranu *SGML*<sup>18</sup>-om, gdje struktura dokumenta enkodira semantiku, ali ne i prezentaciju podataka, pogonjena

---

<sup>14</sup> International Standards Organization

<sup>15</sup> eng. Document Object Model je platformno i jezično neutralno sučelje koje omogućava programima i skriptama dinamički pristup sadržaju, strukturi i izgledu web-dokumenata. Također dozvoljava obradu dokumenata i umetanje rezultata obrade natrag u dokument.

<sup>16</sup> eng. Application Programming Interface

<sup>17</sup> DOM level-0

<sup>18</sup> eng. Standard Generalized Markup Language – jezik za opisivanje strukture dokumenata na najapstraktnijoj razini



relativnom jednostavnošću moćnog XML<sup>19</sup>-a, dovela je do još jednog sloja apstrakcije i nekoliko godina nepotpunih i nekompatibilnih implementacija.

.

---

<sup>19</sup> eng. Extensible Markup Language

## 4. Verzije i dijalekti

Kao što je već rečeno *ECMAScript* je client-side skriptni jezik. Dakle, to je jezik koji se izvršava na strani klijenta, ne prevodi se u izvršnu datoteku, već ga instalirani interpretator, svaki put kada se pokrene, prevodi i izvršava, naredbu po naredbu.

Također, jezik je slabo obilježen<sup>20</sup>, što će reći da korisnik pri deklaraciji varijable ne mora navesti tip varijable, što, naravno, koliko god olakšava upotrebu jezika, toliko ju i otežava; ukoliko programer nije discipliniran<sup>21</sup>. Provjera ispravnosti obilježja je, dakle, dinamička, točnije radi se o patkasto obilježenosti<sup>22</sup>, u doslovnom prijevodu. Po tom principu semantiku objekta određuju sučelja i varijable objekta.

Jezik implementira funkcionalnu, imperativnu i objektno-orijentiranu paradigmu. Funkcionalna paradigma tretira računanje kao evaluaciju matematičkih funkcija, dok imperativna paradigma vrši računanje naredbama koje mijenjaju stanje programskih varijabli (može se povući paralela sa imperativnim načinom govora kod prirodnih jezika). Objektno-orijentirana paradigma problem razlaže na skup objekata koji međusobno komuniciraju porukama, pritom strogo poštujući (u teoretskom slučaju) princip enkapsulacije, te se stavlja programeru na raspolaganje principe nasljeđivanja i polimorfizma što u konačnici rezultira modularnošću napisanog koda (write once – use everywhere).

Ekstenzija datoteke sa izvornim kodom je «.es», a MIME<sup>23</sup> tip glasi «application/ecma». Slijede izdanja/verzije ECMA-262 specifikacije:

---

<sup>20</sup> eng. loosely typed

<sup>21</sup> pretpostavimo sljedeću situaciju: var x=2;var y=»13»;var z=x+y; Rezultat je, u općem slučaju, nedefiniran: hoće li to biti cijeli broj 15 ili string 213?

<sup>22</sup> eng. duck typing

<sup>23</sup> eng. Multipurpose Internet Mail Extensions

Izdanje	Datum izdavanja	Razlike u odnosu na prethodno izdanje
1	lipanj 1997	urednik Guy L. Steele, Jr.
2	lipanj 1998	usklađivanje specifikacije sa ISO/IEC 16262 standardom, urednik Mike Colishaw
3	prosinac 1999	uveđeni regularni izrazi, bolji rad sa stringovima, nove kontrolne naredbe, try/catch exception handling, uže definicije grešaka, formatiranje numeričkog izlaza itd. urednik Mike Cowlishaw.
4	-	vidjeti Zaključak

Tablica 1. izdanja/verzije *ECMA-262* specifikacije

U lipnju 2004. *ECMA International* izdala je *ECMA-357* standard poznatiji kao *E4X<sup>24</sup>*, koji definira nativnu XML podršku za *ECMA Script*, podržavajući pristup XML dokumentu u obliku koji imitira XML sintaksu. Cilj ovog standarda je pružiti alternativno DOM sučelje radi jednostavnijeg baratanja XML-om – koji se po ovom standardu tretira kao primitivni tip podatka, a ne kao, dosada, objekt.

*ECMA Script* podržan je u mnogim aplikacijama, naročito web-browserima, gdje se obično naziva *JavaScript*. Dijalekti obično uključuju vlastite, različite standardne knjižnice, od kojih su neke standardizirane odvojeno – kao što je W3C-ov DOM. Neke implementacije, poput *ActionScript*-a, imaju potpuno drukčiji skup knjižnica. Navedeno za posljedicu ima da aplikacije pisane u jednom dijalektu *ECMA Scripta*, najvjerojatnije neće raditi u drugom dijalektu, osim ako su dizajnirane da budu kompatibilne.

---

<sup>24</sup> *ECMA Script for XML*

Aplikacija	Dijalekt	Verzija	Korespodentna specifikacija
<i>Gecko</i> -browseri i drugi programi koji koriste <i>SpiderMonkey</i> , uključujući <i>Mozilla Firefox</i>	<i>JavaScript</i>	1.7	ECMA-262, v. 3
<i>Internet Explorer</i>	<i>JScript</i>	5.7	ECMA-262, v. 3
<i>Opera</i>	<i>ECMAScript</i> , sa proširenjima za <i>JavaScript</i> and <i>JScript</i>	1.3/1.5	ECMA-262, v. 3
<i>KHTML</i> -browseri, uključujući <i>KDE</i> -ov <i>Konqueror</i> i <i>Apple</i> -ov <i>Safari</i>	<i>JavaScript</i>	1.5	ECMA-262
<i>Microsoft .NET Framework</i>	<i>JScript .NET</i>	8.0	ECMA-262, v. 3
<i>Adobe Flash</i> i <i>Adobe Flex</i>	<i>ActionScript</i>	2 3	ECMA-262, v. 3 ECMA-262, v. 4
<i>Adobe Acrobat</i>	<i>JavaScript</i>	1.5	ECMA-262, v. 3
Skriptni jezik opće namjene	<i>DMDScript</i>	1.06	ECMA-262
<i>OpenLaszlo</i> platforma	<i>JavaScript</i>	1.4	ECMA-262, v. 3
<i>iCab</i>	<i>InScript</i>	3.22	ECMA-262, v. 3
<i>Samba 4</i> i ugrađeni serveri	ugrađeni <i>JavaScript</i>	-	ECMA-262

Tablica 2. korespondencije verzija i dijalekata

JavaScript	JScript	ECMA Script
1.0 (Netscape 2.0, 3.1996.)	1.0 (IE 3.0 , 8.1996.)	
1.1 (Netscape 3.0, 8.1996.)	2.0 (IE 3.0 1.1997.)	
1.2 (Netscape 4.0, 6.1997.)		
1.3 (Netscape 4.5, 10.1998)	3.0 (IE 4.0, 10.1997.)	izdanje 1 (7. 1997) izdanje 2 (7. 1998)
1.4 (samo Netscape Server)	4.0 (Visual Studio 6)	
	5.0 (IE 5.0, 3.1999.)	
	5.1 (IE 5.01)	
1.5 (Netscape 6.0, 11.2000; i kasnija Netscape i Mozilla izdanja)	5.5 (IE 5.5, 7.2000.)	izdanje 3 (12.1999.)
	5.6 (IE 6.0, 10.2001.)	
1.6 (Gecko 1.8, Firefox 1.5, 11. 2005.)		izdanje 3. uz dodatke
1.7 (Gecko 1.8.1, Firefox 2, 10. 2006.)		izdanje 3. uz dodatke
	JScript .NET (ASP.NET)	JScript .NET je navodno dizajniran uz sudjelovanje ECMA članova
JavaScript 2.0		izdanje 4.

Tablica3. korespondencija vezija *JavaScript*-a i *JScript*-a sa odgovarajućim standardom

## 5. ZAKLJUČAK

Četvrto izdanje *ECMA-262* je prva velika nadogradnja *ECMA Scripta* nakon trećeg izdanja iz 1999. Novija verzija jezika je kompatibilna sa trećim izdanjem dodajući nove mogućnosti:

- Razredi
- Paketi i prostori imena
- Opcionalno statičko obilježavanje<sup>25</sup>
- Generatori i iteratori
- JSON<sup>26</sup> enkodiranje/dekodiranje

ECMAScript 4 namjerava pružiti bolju podršku «programiranju naveliko»<sup>27</sup> i dozvoljava programerima da žrtvuju dinamičnost skripti za performanse.

JavaScript se razvio u nevjerojatno jak koktel najboljih mogućnosti *Perl*-a (asocijativna polja, slabo-obilježeni tipovi varijabli i regularni izrazi), *C*-a, *C++*-a i *Java* (čista, blokovski-parsirana sintaksa, objekti i klase, visoko razvijene datumske funkcije, matematičke i string knjižnice). Ovo su samo neke od postojećih mogućnosti, a sve to kombinirano sa snagom *W3C DOM*-a u posljednjim izdanjima browsera (*Mozilla*, *Netscape 6*, *IE6/Windows* i *IE5/Mac*).

Nakon burnog početka, možemo konstatirati da je *JavaScript* zavladao webom.

---

<sup>25</sup> eng. static typing

<sup>26</sup> eng. JavaScript Object Notation – lagani format za razmjenu podataka

<sup>27</sup> programiranje od strane velikih grupa ljudi ili manjih grupa ljudi na duža vremenska razdoblja