

# Razvoj programskog jezika JavaScript

Seminarski rad u okviru kursa  
Dizajn programskih jezika  
Matematički fakultet

Anka Stanković  
ankastankovic98@gmail.com

## Sažetak

Osnovna ideja rada je praćenje razvoja programskog jezika JavaScript. Cilj je bio napraviti razvojno stablo koje obuhvata programske jezike koji su imali uticali na razvoj JavaScript-a. Istaknute su najvažnije karakteristike programskih jezika koji čine razvojno stablo i navedene su osobine koje je JavaScript preuzeo od njih.

## Sadržaj

<b>1</b>	<b>Uvod</b>	<b>2</b>
<b>2</b>	<b>Osnovno o JavaScript-u</b>	<b>2</b>
<b>3</b>	<b>Razvojno stablo</b>	<b>4</b>
3.1	Java	4
3.2	Scheme	5
3.3	Perl	6
3.4	Self	6
3.5	Python	6
3.6	HyperTalk	7
<b>4</b>	<b>Zaključak</b>	<b>7</b>

# 1 Uvod

Od samog nastanka, internet predstavlja tržište koje zahteva konstantan razvoj i napredak. Nije mnogo prošlo od nastanka, a internet je postao sastavni deo svakog doma, a popularnost i mogućnosti koje pruža konstantno rastu. Za razliku od prvobitnih mogućnosti i osnovnih servisa koji su bili na raspolaganju korisnicima (mail i web pretraga) danas je na internetu ponudjen veliki broj različitih servisa. Iz dana u dan javljaju se nova kreativna rešenja koja pospešuju funkcionalnost i upotrebljivost weba. Prethodnih godina, napravljeni su mnogobrojni alati za programiranje weba. Razvoj interneta (web development) danas je jedno veoma široko i razgranato područje. U zavisnosti od kreativnosti i programerskih sposobnosti, svako ko je zainteresovan za ovu delatnost može naći svoje mesto u frontend ili backend programiranju. Jedan od glavnih alata koji se koristi na web stranicama je programski jezik **JavaScript**. O njegovoj popularnosti i korisnosti najviše govori šaljiva krilatica poznatija kao "*Etvudov zakon*". On kaže da sve što može biti napravljeno u JavaScriptu, na kraju će i biti napravljeno u JavaScriptu!

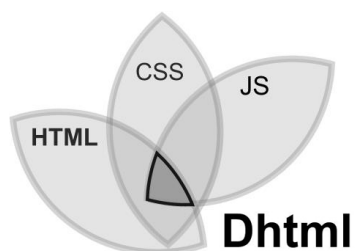


Slika 1: Logo programskog jezika JavaScript

## 2 Osnovno o JavaScript-u

*JavaScript* je objektno zasnovan skript jezik. Još od najranijih dana web-a, JavaScript je osnovna tehnologija koja omogućava interaktivan rad sa sadržajem. Spada u grupu multiparadigmatskih jezika, pa osim objektno orijentisane paradigme zastupljene su i imperativna paradigma, programiranje vođeno događajima i funkcionalno programiranje. Ovaj jezik je razvio Brendan Ajh (engl. *Brendan Eich*) pod pokroviteljstvom Netscape Communication korporacije 1995. godine. Razvoj jezika je podstaknut idejom o dinamičnijem web-u i potrebom HTML-a za "glue language"-om koji bi omogućio lako spajanje komponenti web stranice (slike, dodaci... ). Zajedno sa HTML-om i CSS-om, JavaScript čini DHTML (Dynamic HTML).

Po sintaksi, jezgro jezika je slično jezicima C, C++, Java. Slabo je tipiziran jezik. Koristi se kao client-side (klijent zasnovan) internet programski jezik, što znači da se njegova obrada vrši na korisnikovom računaru, u samom pregledaču čime se postiže veća brzina rada, web aplikacije odgovaraju brže na korisnikove akcije i samim tim se postiže veća efikasnost. JavaScript je možda jedan od jezika na kojem je najlakše započeti učenje programiranja jer se složeni koncepti koji su duboko ukorenjeni u jezik pojavljuju u naizgled izuzetno jednostavnim oblicima, lakim za korišćenje. Međutim ovaj jezik je složena i nijansirana kolekcija jezičkih mehanizama koji će, ako se ne proučavaju pažljivo, učiniti da stvarno razumevanje ne



Slika 2: DHTML

dosegnu čak ni najiskusniji JavaScript programeri. U tome leži paradoks JavaScripta, tj Ahilova peta tog jezika, veoma je teško u potpunosti ga savladati.

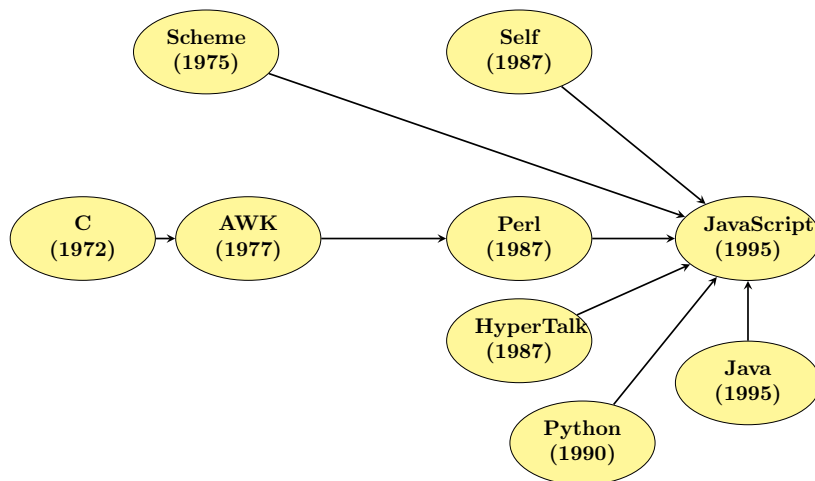
Iako je do pre par godina JavaScripta bio prisutan samo na webu, danas je situacija značajno drugačija. 2014. je proglašen jezikom godine. Danas je osim na webu prisutan i u razvoju mobilnih aplikacija i kao i u razvoju video igara. On je osmišljen da se koristi kao pomoć ostalim "velikim" programskim jezicima za kreiranje web aplikacija, međutim sa pojavom Node.js moguće je kreirati potpuno funkcionalne web aplikacije pomoću JavaScripta koje se brinu i za klijentsku i za serversku stranu.

ECMA je standardizovala jezik koji je u opštoj javnosti i literaturi poznat kao JavaScript. Međutim kako nije bilo moguće postići dogovor o zvaničnom imenu koji bi svima odgovarao, jezik opisan ECMA standardom na koji se misli kada se kaže JavaScript se u prema standardu naziva **ECMAScript**.

### 3 Razvojno stablo

Ideja o nastanku JavaScript-a je potekla od kompanije Netscape. Prvobitno je bilo zamišljeno da se napravi jezik koji će ugraditi programski jezik Scheme u njihov pregledač Navigator. U međuvremenu Netscape postiže se dogovor sa kompanijom Sun Microsystems o uključivanju Jave u pregledač. Ideja je bila da se **Java** stavi na raspolaganje profesionalcima za pisanje komponenti, dok bi Mocha (to je bio radni naziv projekta) bila dostupna za manje zadatke. Drugim rečima, Mocha je trebao da bude Javin skript pratilac. Jezik se prvi put pojavio u Netscap-ovom Navigatoru septembra 1995. pod imenom LiveScript, da bi u decembru bio preimenovan u JavaScript.

Na slici 3 može se videti razvojno stablo jezika JavaScript.



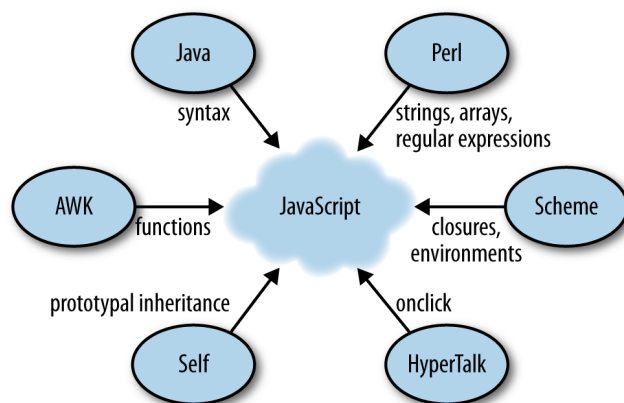
Slika 3: Razvojno stablo jezika JavaScript

#### 3.1 Java

Java je programski jezik opšte namene. Razvila ga je kompanija Sun Microsystems. Prva verzija ovog programa objavljena je 1995. godine. Mnogi koncepti Jave zasnovani su na jeziku Oberon. Takodje je i formalno uveden koncept *klasa* iz objektno-orijentisane paradigme. Java je dizajnirana tako da je nezavisna od platforme i sa pojednostavljenim upravljanjem memorijom. U stvaranju jezika Java oslonjeno je na pet osnovnih ciljeva:

1. Mora biti jednostavan, objektno orijentisan, prepoznatljiv.
2. Mora biti temeljan i siguran.
3. Mora biti arhitektski neutralan i prenosiv.
4. Mora da se izvršava sa visokim performansama.
5. Mora biti interpretiran, praćen i dinamičan.

Java se izvršava korišćenjem Java virtuelne mašine - **JVM** (engl. *Java VirtualMachine*), koji predstavlja apstraktnu mašinu koja postoji samo u



Slika 4: Programski jezici koji su uticali na JavaScript

memoriji. Java je programski jezik koji se prevodi i interpretira. Java izvorni kod (\*.java fajl) nakon kompajliranja, kompajlerom javac, se prevodi u izvorni kod bajtkod (\*.class fajl). Formirani bajtkod je isti za sve arhitekture računara i predstavlja instrukcije za Java Virtuelnu mašinu. Bajtkod u Java virtuelnoj mašini, java interpretatorom se dešifruje i interpretira u izvršni, kod pogodan za izvršavanje na računaru, tj odgovarajućem hardveru.

Java je glavni uzor za sinatksu JavaScripta. Iz Jave je nasledjena i podela na primitivne i objektne tipove, kao i konstuktor za datume (Date constructor).

### 3.2 Scheme

Scheme (srp. *Skim*) je programski jezik razvijen 1970-tih u MIT-ovim laboratorijama, prvobitno namenjen za istraživanja i podučavanje studenata. Scheme su 1975. godine predstavili Gerald J. Sussman i Guy L. Steele, njegovi tvorci, serijom memoranduma/spisa koji su sada poznati kao "*Lambda papiri*". Nastao je pod uticajem Algol-60, imperativnog jezika i Lispa, funkcionalnog jezika.

Scheme prati filozofiju minimalističkog dizajna definisanjem malog standardnog jezgra jezika (primitivnih konstrukata), ali sa moćnim alatima za proširenje jezika. Zbog svoje kompaktnosti i elegancije, Scheme je programski jezik koji se koristi u raznovrsne namene. Međutim, zbog svoje minimalističke filozofije i standarda, nastale su i raznovrsne implementacije i nadogradnje jezika, što dovodi do nekompatibilnosti kodova pisanih u različitim implementacijama. Razilaženja u implementacijama su mnogobrojna, toliko da Scheme-ov upravni odbor naziva Scheme "najnekompatibilnijim programskim jezikom na svetu", pritom govoreći radije o Scheme-u kao o kolekciji dijalekata umesto jedinstvenom jeziku.

Scheme je razlog zbog kog JavaScript funkcije tretira kao "primarne" objekte (**first-class function**). Naime funkcije se tretiraju kao promenljive i mogu se prosledjivati kao argumenti drugih funkcija. Takođe, iz Schema je nasledjena i ideja closure funkcija. Closure su funkcije koje imaju pristup promenljivima koje se nalaze u domenu druge funkcije. Ovo se najčešće postiže ugradjivanjem funkcije unutar druge funkcije (čime se definiše "scope chain"), nakon čega closure dobija sposobnost da zapamti

referencu na promenjive iz roditeljskog domena.

### 3.3 Perl

Perl je veoma koristan programski jezik koji je postao standardan na gotovo svim verzijama Unix-a. Ovo je interpretiran jezik nezavistan od platforme. Njegova semantika je slična onoj u C-u, ali je u Perlu rad sa tekstom i regularnim izrazima na znatno višem nivou. Gotovo svaki iole jednostavniji, zadatak se može isprogramirati nekom perl scriptom. Perl je, takodje, veoma koristan kada je potrebno pretražiti velike količine teksta i unutar teksta uraditi dosta uporedjvanja. Istovremeno je veoma koristan i kao script jezik za web.

Skraćenica PERL potiče od Practical Extraction and Report Language koja precizno objašnjava najjače osobine perla - Practical za praktičnost tj. brže pisanje programa nego u programskom jeziku C, Extraction za izdvajanje i analizu datoteka i podataka, Report za generisanje izlaznih podataka Language za programski jezik.

Mnoge funkcionalnosti u programskom jeziku Perl su pozajmljene od drugih programskih jezika uključujući C i AWK. U osnovi Perl je proceduralniji jezik, C-olike sintakse. Iz AWK je nasledio princip "asocijalnih nizova". AWK je inspirisao i funkcije u JavaScriptu, ključna refunction je preuzeta baš iz njega.

Perl je ispirisao rukovanje stringovima, nizovima i regularnim izrazima u JavaScript-u.

### 3.4 Self

Self je objektno orijentisan programski jezik koji se zasniva na konceptu prototipova. Nastao je kao dijalekat programskog jezika SmallTalk. Korišćen je kao eksperimentalni sistem za testiranje jezika razvijanih 80-tih godina prošlog veka. Poslednja verzija Selfa objavljena je u maju 2017. godine. Veliki deo razvoja programskog jezika Self se odvijao u kompaniji Sun Microsystems, pa su tehnike razvijene za Self kasnije implementirane u virtuelnu mašinu programskog jezika Java (HotSpot).

Značajan je jer su kroz njega razvijeni koncepti objektno orijentisanog programiranja i just-in-time kompilacije. Za razliku od ostalih jezika, Self dopušta objektima da naslede stanja i da dinamički menjaju obrasce nasledjivanja. Kada je u pitanju jezik Self, veći značaj ima njegov uticaj na ostale jezike nego on sam kao jezik.

Objekti u JavaScript-u imaju ugrađeni mehanizam nasleđivanja kroz takozvano prototipsko nasleđivanje, to je nasledjeno baš od Self-a. Prototipsko nasleđivanje se zasniva na delegiranju. Svaki objekat ima svoja svojstva i metode, ali i posebnu vezu ka roditeljskom objektu kodji onda predstavlja njegov prototip od koga nasleđuje svojstva i metode.

### 3.5 Python

Python je interpretirani, interaktivni, objektno orijentisani programski jezik visokog nivoa, koji poseduje dinamičku semantiku. Stvorio ga je Gvido van Rosum (Guido van Rossum) 1991. godine, te spada u grupu novih programskih jezika. Njaveći uticaj na razvoj ovog jezika imao je jezik ABC, jer je tvorac radio na tom projektu pre početka razvoja Pythona. Ime je dobio po kultnoj britanskoj komediji „Monty Python”. Sintaksa jezika Python omogućava pisanje veoma preglednih programa. Za

programski jezik Python razvijen je veliki broj standardnih modula koji omogućavaju efikasan rad u mnogim oblastima. Većina ovih modula prenosiva je na različite platforme što omogućava da kompletni programi bez prilagođavanja (u većini slučajeva) pravilno rade na različitim mašinama i pod različitim operativnim sistemima.

Python je jezik široke primene. Bogat je bibliotekama i to mu omogućava da nađe primenu u mnogim oblastima. On je jedan od najkorisnijih jezika za data science i veštačku inteligenciju.

JavaScript je iz Pythona usvojio iterator protokol, plitki generator-korutine i način rada sa nizovima i generator izraza.

### 3.6 HyperTalk

HyperTalk je viši, proceduralni programski jezik. Napisao ga je Amerikanac Bill Atkinson 1987. Korišćen je u okviru kompanije Apple. Ovo je skript jezik programskog jezika HyperCard namenjen programerima početnicima. Sintaksa mu podseća na govorni engleski jezik, dok je logička struktura slična onoj u Pascal-u. Jezik je objektno-orijentisane prirode i povezan je sa HyperCard interfejsom. Spada u grupu slabo tipiziranih jezika. Ne postoje klase i strukture u tradicionalnom smislu, njihovo mesto zauzima posebna vrsta niski (štrings") koje sadrže listu objekata (items) razdvojenih određenim karakterom.

HyperTalk je uticao na to kako se JavaScript integriše u HTML. To je dovelo do HTML oznaka koje imaju attribute za rukovanje događajima kao na primer onclick.

## 4 Zaključak

Veoma česta greška je poistovećivanje JavaScripta sa programskom jezikom Java. Međutim, među ovim jezicima postoje značajne razlike. Java je jezik koji podržava klasično objektno-orijentisano programiranje, zasnovano na klasama i nasleđivanju. Sa druge strane, JavaScript je jezik koji je zasnovan na objektima, ali nema zaista "pravo" objektno-orijentisano programiranje. A druga bitna razlika je u tome što je Java strogo tipiziran jezik, dok je JavaScript dinamički tipizirana.

Iako je izvorno JavaScript bio zamisljen kao mali jezik namenjen za pomoć drugim "velikim" jezicima u izradi web stranica, danas je JavaScript mnogo više od toga. Domen primene je proširen i moguće je pisati i samostalno JavaScript aplikacije. U kombinaciji HTML5 i JavaScript-a ima dobru podršku za rad sa grafikom pa je popularna u razvoju video igara. JavaScript se koristi i za pisanje mobilnih aplikacija.

## Literatura

- [1] Naučite JavaScript  
*Kyle Simpson.* [link](#)
- [2] Perl  
*Perl-Referentni priručnik, The President and Fellows of Harvard Collegeperl*
- [3] JavaScript  
*WEBnSTUDY* [link](#)
- [4] Python  
[link](#)
- [5] Selfe  
[link](#)
- [6] HyperTalk  
*"Apple Ponders Standardizing on HyperTalk"*
- [7] Scheme  
[link](#)
- [8] Java  
[link](#)
- [9] JavaScript  
[link](#)