

# Razvoj programskog jezika Python

Seminarski rad u okviru kursa  
Dizajn programskih jezika  
Matematički fakultet

Ana Nikolić  
nikoliceva06@gmail.com

20. decembar 2019

## Sažetak

U ovom radu prikazuju se osnove razvoja programskog jezika Python. Ukratko su opisani jezici koji su najviše uticali na njegov razvoj: ABC, Bourn Shell, C, Modula i Algol 68. Za svaki jezik date su osnovne informacije kao i način na koji je taj jezik uticao na osobine i razvoj jezika Python. Prikazano je razvojno stablo koje uključuje ove jezike.<sup>1</sup>

## Sadržaj

<b>1</b>	<b>Uvod</b>	<b>2</b>
<b>2</b>	<b>Osnovno o Pythonu</b>	<b>2</b>
<b>3</b>	<b>Razvojno stablo</b>	<b>3</b>
3.1	ABC . . . . .	3
3.2	C . . . . .	4
3.3	Modula-3 . . . . .	4
3.4	Smalltalk . . . . .	4
3.5	Algol-68 . . . . .	5
<b>4</b>	<b>Zaključak</b>	<b>5</b>
	<b>Literatura</b>	<b>5</b>

---

<sup>1</sup>Tekst u ovom radu je najvećim delom preuzet iz master rada Milene Dukanac [?].

# 1 Uvod

U oblasti razvoja programskih jezika, još devedestih godina, javila se potreba za programskim jezikom koji će nasuprot tradicionalnim biti praktičan za brže i lakše programiranje i bržu obradu velike količine podataka. Tako nastaje 1991. godine programski jezik *Python*. Za veoma kratko vreme postao je izuzetno popularan i danas ga veliki broj kompanija koristi za svoje projekte i softverska rešenja. Logo jezika Python prikazan je na slici 1.



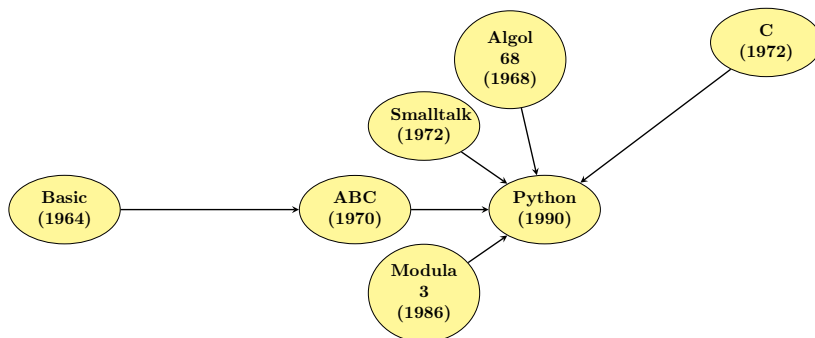
Slika 1: Logo programskog jezika Python

## 2 Osnovno o Pythonu

Python je interpretirani jezik opšte namene čiji je tvorca je Gvido van Rossum (engl. *Guido van Rossum*) [1]. Krajem 1980-ih je koncipiran kao naslednik jezika **ABC** [3], a prvi put je objavljen 1991. godine. Filozofija dizajna jezika Python naglašava čitljivost koda. Njegove jezičke konstrukcije i objektno-orijentisani pristup imaju za cilj da pomognu programerima da napišu jasan i logičan kôd za male i velike projekte. Python je dinamički tipiziran jezik i poseduje sistem za prikupljanje smeća (engl. *garbage collector*). Podržava više paradigmi programiranja uključujući proceduralno, objektno-orijentisano i funkcionalno programiranje. Python interpreteri su dostupni za mnoge operativne sisteme. Globalna zajednica programera razvija i održava referentnu implementaciju otvorenog koda **CPython**. Nefitna organizacija *The Python Software Foundation* upravlja resursima i usmerava ih za razvoj jezika Python i CPython. Jedna od osobina koje je Elixir nasledio od Python-a je podrška za dokumentaciju u vidu dokumentacionih stringova (engl. *docstrings*) koji omogućavaju povezivanje dokumentacije sa modulima, funkcijama, klasama, metodama. Python se danas koristi u velikim kompanijama, kao što su *Facebook*, *Netflix*, *Spotify* i *Google* [?].

### 3 Razvojno stablo

Na nastanak jezika Python je najviše uticao programski jezik **ABC**. Pri njegovom kreiranju značajnu ulogu u smislu sintakse imao je programski jezik **C**, a iz jezika kao što su **Modula 3**, **Algol 68** i **Smalltalk** je preuzeo mnoge koncepte. Razvojno stablo jezika Python može se videti na slici 2.



Slika 2: Razvojno stablo jezika Python

#### 3.1 ABC

ABC je imperativni jezik opšte namene razvijen na Institutu za matematiku i informatiku, u Holandiji, a njegovi tvorci su bili Leo Džjurts, Lambert Mertend i Stiven Pemberton (engl. *Leo Geurts, Lambert Meertens, i Steven Pemberton*). Naime, nastao je oko 1975. godine i prvobitni naziv mu je bio B (koincidencijom isto ime je nosio i prethodnik programskog jezika C). Zamišljen je kao zamena za Basic, kao jezik ali i okruženje pogodno za učenje, što je i njihova zajednička karakteristika. Međutim, ABC je drugačije prirode: mnogo ekspresivniji i praćen korisnim programskim okruženjem. Tu ekspresivnost ali i jednostavnost kao i mali broj koncepata, Python nasleđuje kao jednu od glavnih karakteristika jezika. ABC je dinamički tipiziran jezik, pa ne dozvoljava deklaracije tipova podataka. Za razliku od većine dinamički tipiziranih jezika, ABC je i jako tipiziran, što znači da svi elementi neke strukture moraju biti istog tipa. Python je dinamičko tipiziranje nasledio od ABC-a, a što se tiče jakog tipiziranja ono je usvojeno od verzije Python 3, tj. verzije pre ove nisu podrzavale pomenut koncept. Jos nešto što karakterise Python je to što blokove koda grupise na osnovu uvlačenja, a to je jos jedan koncept koji je nasledio od ABC-a. Takođe, nedostaci koje je posedovao ABC, to da je u suštini bio zatvoren sistem sa postojanjem samo standardnog ulaza i izlaza, bez mogućnosti redirekcije ulaza/izlaza na/u neki fajl, rešeni su kod Pythona. Jasno je da ovaj jezik ima veliki uticaj na programski jezik Python. Naime, Gvido van Rosum, tvorac Pythona, je bio uključen u razvoj programskog jezika ABC osamdesetih godina. [2].

## 3.2 C

C proceduralni programski jezik opšte namene. Prvobitno ga razvija Dennis Ritchie u Belovim laboratorijama, između 1972. i 1973, no značajan doprinos nastanku C-a dali su Ken Thompson kao autor programskog jezika B i Martin Richards, autor programskog jezika BCPL. Programski jezik C dosta se menjao tokom godina te je u više navrata neformalno i formalno standardiziran. Prva važnija verzija poznata je pod nazivom "KR C", što je engl. skracenica prezimena dvojice autora najpoznatijeg C priručnika "The C Programming Language", a to su Brian Kernighan i Dennis Ritchie. Tokom 1980-ih, C je postepeno stekao popularnost. Postao je jedan od najčešće korišćenih programskih jezika. Ono što je Python nasledio od C-a je sintaksa. Pre svega, misli se na ključne reči. Takođe prioritet operatora je isti.

## 3.3 Modula-3

Modula-3 je programski jezik zamišljen kao naslednik nadograđene verzije Modula-2 poznate kao Modula-2+. Iako je imao uticaja u istraživačkim krugovima (uticao na dizajn jezika kao što su Java, C i Python), u industriji nije široko prihvaćen. Dizajnirali su ga Luca Cardelli, James Donahue, Lucille Glassman, Mick Jordan, Bill Kalsov i Greg Nelson, krajem 1980-ih. Glavne karakteristike Modula-3 su jednostavnost i sigurnost. Modula-3 imala je za cilj da nastavi tradiciju bezbednosti tipova koja je karakterisala Pascal, istovremeno uvodeći neke nove koncepte tada potrebne. Konkretno, Modula-3 je dodala podršku za generičko programiranje, multithreading, rukovanje izuzecima, sakupljač smeća, objektno orijentisani koncept i eksplicitno označavanje nesigurnog koda. Cilj dizajna Modula-3 bio je jezik koji u sasvim osnovnim oblicima implementira najvažnije karakteristike savremenih imperativnih jezika. Tako su hipotetički opasni i komplikovani koncepti poput višestrukog nasljeđivanja i preopterećenja metoda izostavljeni. Mnogi od pomenutih koncepata je Python nasledio od Module 3: module, sakupljač smeća, rukovanje izuzecima. Python je od Module 3 pokupio saupljač smeća, što i potvrđuje da je Python lak za korišćenje- nije potrebno da programer eksplicitno oslobađa memoriju (kao npr u C-u) i ne mora da brine o zarobljenoj memoriji i defragmentaciji iste. Još jedan značajan koncept koji dele ova dva jezika su moduli. Koncept modula zapravo je preuzet iz Module 3. Nekađ nam program postaje duži, i hoćemo ga podeliti u nekoliko datoteka radi lakšeg održavanja ili želelimo da koristimo funkciju koju smo napisali u nekoliko programa bez kopiranja njene definicije u svaki program. Da bi to podržao, Python ima način da definicije stavi u datoteku i koristi ih u skriptu. Takva se datoteka naziva modul; definicije iz modula mogu se uvesti u druge module ili u glavni modul.

## 3.4 Smalltalk

Smalltalk je objektno-orijentisan, dinamički tipiziran programski jezik koji podržava refleksiju. Jedan od mnogih jezika koji je baziran na jeziku Simula. Rezultat je višegodišnjeg proučavanja softvera koji je pogodan za pravljenje visoko-funkcionalne i interaktivne komunikacije operativnim sistemima personalnih računara. Jedan njegov deo stvoren je za obrazovne svrhe. Razvila ga je grupa istraživača iz kompanije Xerox PARC na čelu sa Alanom Kayom. Smalltalk ima mnogo objektno-orijentisanih naslednika, a jedan od programskih jezika koji je usvojio ovaj koncept bio je i Python.

```
1000 array=["Hello world","HI"]
1001 for i in array:
1002     print i
```

Listing 1: Python

```
1000 public class Test{
1001     public static void main(String
1002         args [])
1003     {
1004         String array={"Hello world","HI"};
1005         for(String i: array):
1006             System.out.print(i)
1007     }
1008 }
```

Listing 2: Java

Slika 3: Kôd za ispisivanje niza u jezicima Python i Java

### 3.5 Algol-68

ALGOL 68 (Algorithmic Language) je imperativni programski jezik koji je zamišljen kao naslednik programskog jezika ALGOL 60, dizajniran s ciljem mnogo šireg obima primene i rigoroznije definisane sintakse i semantike. Zbog složenost sintakse jezika, upotreba je bila donekle ograničena, ali našao je upotrebu posebno u Velikoj Britaniji, gde je bio popularan na ICL mašinama, i u naučnim i nastavnim okvirima. Ipak, doprinosi ALGOL-a 68 u oblasti računarske nauke bili su veliki, opsežni i trajni, mada su mnogi od tih priloga javno identifikovani tek kada su se ponovo pojavili u kasnije razvijenim programskim jezicima. Mnogi jezici su razvijeni posebno kao odgovor na uočenu složenost jezika, od kojih je najistaknutiji bio Pascal, ili su bili ponovni primeni za određene uloge, poput Ade. Mnogi jezici iz 1970-ih prate svoj dizajn posebno do ALGOL 68, birajući neke karakteristike, a odbacujući druge koje su smatrane previše složenim ili van dosega za dane uloge. Većina savremenih jezika prati barem neke svoje sintakse bilo na C ili Pascal, i tako direktno ili indirektno na ALGOL 68. Jedna od konkretnih karakteristika koju je Python pozamio od Algol-a je slice operator tj to što ga možemo primeniti na stringovne promenljive.

## 4 Zaključak

U ovom tekstu ukratko su predstavljene osnove razvoja programskog jezika Python. Prikazani su jezici koji su najviše uticali na njegov nastanak i razvoj, i prikazano je njegovo elementarno razvojno stablo. Za dodatne informacije i praćenje daljeg razvoja, kao početnu odrednicu najbolje je koristiti zvaničnu stranu programskog jezika Python [1].

## Literatura

- [1] Python. Zvanična stranica programskog jezika Python. <https://www.python.org/>.
- [2] Guido van Rossum. Guido personal home page. *Coder Stories*, 2019.
- [3] Rolf Zwart, April 2018. <https://reinout.vanrees.org/weblog/2018/04/25/origin-of-python-abc.html>.