

Osnove programiranja (Python)

Uvod u programiranje (Python)

Dr Milan Paroški mparoski@singidunum.ac.rs Univerzitet Singidunum 2024/2025



Sadržaj

- A.Uvod
- B. Programski jezici
- C. Proces razvoja programa
- D. Programski jezik Python



A. UVOD

- 1. Računanje i računari
- 2. Računarski program
- 3. Izvršavanje programa
- 4. Algoritam





Industrijske revolucije

Koliko ih je bilo?

- 1.Poljoprivreda
- 2.Industrija
- 3.Informatika

- 1. Mehanizacija proizvodnje parne mašine
- 2. Masovna proizvodnja struja
- 3.Informatika digitalna revolucija
- 4. Al, IoT, robotika ...









Uklanjanje barijera između ljudi i mašina: veštačka inteligencija, nanotehnologija, IoT (Internet of things), robotika, 3D štampači, biotehnologija, pametni gradovi i ostale tehnologije u razvoju.

Ove promene zahtevaju učešće i angažovanost ne samo pojedinaca i kompanija, već i država



Mnogi u svetu, nisu ostvarili koristi od prethodne 3 revolucije:

1/3 nema pristup pitkoj vodi,

1/6 nema električnu energiju

Četvrta industrijska revolucija: uz pametnu upotrebu će biti nosilac održivog razvoja budućnosti – kako na ekonomskom, tako na društvenom i ekološkom planu.



Easter Parades in New York City



Year 1900: One Motor Vehicle Year 1913: One Horse & Carriage







Slike sugerišu da je tržište vozila za samo 13 godina totalno promenjeno - prešlo sa skoro svih vrsta prevoza na prevoza na motorni pogon

Zgrade?

Konstrukcije od cigle i maltera izgledaju otprilike isto



A.1 Računanje i računari

Računanje u smislu *aritmetičkih operacija* je automatizovano raznim uređajima. Kojim?

Abakus, digitron....

Računar koji može da izvršava bilo kakva računanja, osmišljen je u XIX, a realizovan u XX veku

Šta je računar ili računska mašina?

programabilni uređaj koji automatizovano izvršava niz aritmetičkih ili logičkih operacija

Algoritam je?

niz instrukcija nezavisnih od računara

Program je?

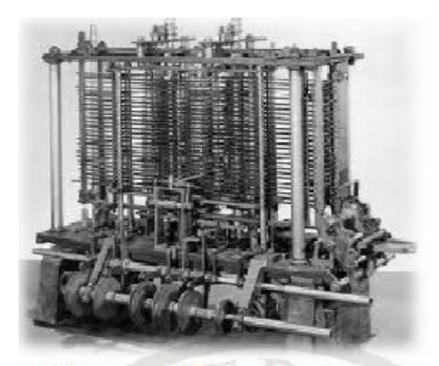
niz instrukcija za računar zapisan nizom simbola na osnovu algoritma





Prvi digitalni računar

- Matematičar Alan Turing izneo je ideju da se računarski program može upotrebiti za modeliranje ljudskog mišljenja (veštačka inteligencija)
- Prvi programabilni (mehanički) računar - početkom XIX veka -Čarls Bebidž, profesor matematike Univerziteta Kembridž
- Prvi program za računar Ada Bajron, njegov asistent, pa se smatra i prvim programerom
- Po njoj je nazvan poznati programski jezik Ada (1980)

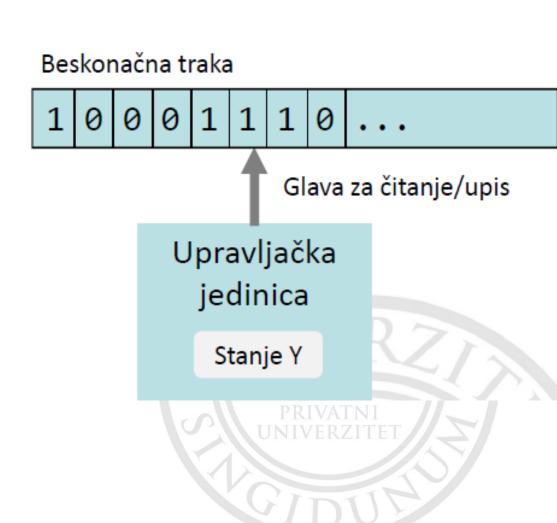


prvi programabilni računar Analitical Engine, XIX vek



Tjuringova mašina

Univerzalna *Tjuringova* mašina je teorijski uređaj koji manipuliše simbolima na beskonačnoj traci u skladu s tabelom pravila naziva se univerzalnom jer može da simulira svaki računarski algoritam





Šta je ovo?





Bušene kartice

Bušene kartice su prvi medij korišten za memorisanje podataka, i važan korak u razvoju računarstva.

Od kartona su, a podaci se spremaju bušenjem rupa na određenom mestu na kartici. Svaka kartica ima definisanu najveću količinu podataka koju može primiti, a ona je zavisila od proizvođača.

Bušene kartice i trake bile su najjednostavnija forma digitalnog memorisanja podataka, sistem u kom rupica označava jedinicu, a njen nedostatak - nulu (ili obrnuto).

Rupice su bile standardnih dimenzija, kao i međuprostor između njih, da ih kompjuterski čitač može pravilno obraditi. Takve trake često su bile duge i po nekoliko metara.



Fon Nojmanova arhitektura računara (implementacija Tjuringove mašine)

Većina savremenih računara opšte namene zasniva se na modelu računara koji je zamislio Džon fon Nojman

Ulazno-izlazna jedinica

Ulazno-izlazna jedinica

Aritmetičko-logička jedinica

Memorijska jedinica

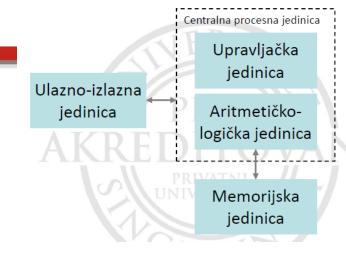
Koje su ulazne jed.?

Ulazne jedinice – Tastatura, miš, skener, mikrofon, CD

Izlazne jedinice Hardver računara – Displej ili video monitor – Štampač – Zvučnici, Ploter



Artimetičko-logička jedinica je sklop koji vrši osnovne aritmetičke operacije (sabiranje, oduzimanje i drugo), logičke operacije (I, ILI, NE) i upoređivanje (da li su neki sadržaji podudarni).



Upravljačka jedinica usmerava rad računara. Ona kontroliše komunikaciju i koordinaciju između ulazno / izlaznih uređaja. Upravljačka jedinica čita i interpretira instrukcije i određuje redosled za obradu podataka. Ona usmerava rad drugih jedinica. Svim resursima računara upravlja upravljačka jedinica).

Svrha programskog jezika je da ostvari komunikaciju između čoveka i računara.

Cilj je da čovek napiše tekst (program) koji će računar moći da izvrši.



HW, OS,SW?

Hardver (Hardware) – skup fizičkih komponenti računara (tastatura, miš, monitor,...)

Operativni sistem – omogućuje komunikaciju softvera i hardvera tj. Omogućuje funkcionisanje računara

Softver – skup programa za obavljanje određenih poslova



Osnovne funkcije računara

- 1. Prihvatanje ulaza: prihvatanje podataka iz spoljašnjeg sveta
- Obrada (procesiranje) podataka: obavljanje aritmetičkih ili logičkih operacija (donošenje odluka) nad podacima
- Formiranje izlaza: dobijanje informacija i slanje informacija u spoljašnji svet
- Memorisanje informacija: slanje i skladištenje informacija u memoriju računara



HARDWER

Memorije i memorijski uređaji –

Primarna memorija: RAM (Random Access Memory)

Sekundarna memorija:

Memorijski uređaji koji služe za dugotrajno skladište podataka?

Tvrdi diskovi (HD)

CD i DVD jedinice

Jedinice magnetnih traka



Hardver se ređe menja nego softver

Mogućnosti računara u najvećoj meri zavise od hardvera i njegovog kvaliteta (performanse)

Trka HW/SW





Informacije i računar

Informacija?

Komunikacija koja poseduje vrednost informisanja

Informacije se pojavljuju u mnogim oblicima –

Reči,

Brojevi,

slike

Audio,

video





Računarske informacije su digitalne

Bit ili binarna cifra

Najmanja jedinica informacija

Binarni brojni system

Može da ima samo dve vrednosti: 1 ili 0

Može da predstavlja brojeve, kodove ili naredbe

Bajt: skup od 8 bit



Binarni brojni sistem

- Korišćenjem dva simbola (0 i 1) svi brojevi mogu biti predstavljeni i nad njima se mogu obavljati proizvoljne aritmetičke operacije
- Bilo koji broj se može posmatrati kao komponente svojih pozicionih vrednosti od kojih je svaka stepena dva
- 19 se predstavlja binarno sa :00010011





Instrukcije programa su u binarnom zapisu

Sačuvani programi se smeštaju kao skupovi bita

Programske instrukcije su prezentovane u binarnoj notaciji kao

odgovarajući kodovi instrukcija





Osnovni nedostatak u binarnom predstavljanju brojeva je predugačak zapis broja

U računarskim sistemima najčešće se koristi heksadecimalni sistem predstavljanja brojeva

Heksadecimalni brojni sistem – ? cifara:

16

0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F (za svaku cifru 4 bita)

Primeri:

0100 0001 b = ?

41 h

1010 0111 b = ?

A7 h



Obeležavanje:

b – bit

B – bajt

Bajt = 8 b

- Kilobajt (KB) = $1024 B = 2^{10} B$

- Megabajt (MB) = $1024 \text{ KB} = 2^{10} \text{ KB}$

- Gigabajt (GB) = $1024 \text{ MB} = 2^{10} \text{ MB}$

- Terabajt (TB) = 1024 GB = 2^{10} GB

- Petabajt (PB) = $1024 \text{ TB} = 2^{10} \text{ TB}$

Primer: broj reči u Wordu

AKREDITOVAN

PRIVATNI Univerzitet



Podela računara

Personalni računari (monokorisnički)

Stoni

Prenosivi

Radne stanice

Tanki klijenti (specijalizovan softver ..)

Miniračunari

Više korisnika

Više procesora

Računari opšte namene "mainframe

Naučno tehnički problemi

Poslovna obrada...

Superračunari



Superračunari

ASCI White (Acceleretited Strategic Computing Iniciative)

Proizvođač IBM

Veličina – dva košarkaška terena

8192 procesora

512 servera RS/6000 sa po 16 procesora Power3- III)

6,7 terabajta memorije

12,3 triliona operacija (teraflop/s)

Predhodni naj: Deep blue

Pobeda 3,5:2,5 protiv Kasparova

50 Biliona pozicija za tri minuta.

200 miliona pokreta u sekundi.



Prva automatizovana obrada podataka?

Obrada podataka popisa stanovništva – 1884 - Herman Hollerith, osnivač IBM

Skupovi podataka o stanovnicima kodirani su na bušene papirne kartice; obrada podataka vršena je korišćenjem čitača bušenih kartica

Korišćenjem igala sa oprugom na vrhu je vršeno očitavanje prisustva ili odsustva rupica na kartici. Kada naidje na rupicu igla bi prolazila i tako ostvarivala električnu vezu koja pokreće brojač.

Obrada podataka o popisu stanovništva – 2,5 godine

1950 - prva poslovna obrada podataka na digitalnom elektronskom računaru - Univac 1

Za njega je razvijen prvi prevodilac za programski jezik (nazvan A1)



A.2 Računarski program

Računarski program je skup naredbi u nekom programskom jeziku za izvršenje određenog zadatka na računaru

Programski jezici su tako koncipirani da budu razumljivi čoveku koji programira

Programi koje kreiraju programeri nazivaju se *izvorni* programi ili izvorni kod (*source code*)

Procesor računara izvršava samo binarno kodirane instrukcije, prilagođene procesoru, koje su čoveku teško razumljive i nepraktične za izradu programa

Programi u binarnom jeziku procesora nazivaju se *mašinski* programi ili mašinski kod (*machine code*)

Da bi se izvorni program mogao izvršiti, potrebno ga je *prevesti* iz izvornog u mašinski kod



Svrha programskog jezika je da ostvari komunikaciju između čoveka i računara.

Cilj je da čovek napiše tekst (program) koji će računar moći da izvrši.

Prvi programski jezik je bio mašinski jezik računara.

Prvi programski jezici posle mašinskih jezika su asemblerski jezici. Ovako napisan program se prevodio i izvršavao na mašinskom jeziku. (assembly language).



A.3 Izvršavanje programa

Programi se mogu prevoditi pre? ili u toku izvršavanja? (kompilacija i interpretacija)

Kompajler ili programski prevodilac je računarski program koji transformiše programski kod višeg programskog jezika u mašinski jezik (prevodi jedan program u drugi)

Interpreter je računarski program koji u realnom vremenu izvršava izvorni kod napisan u nekom programskom jeziku.



A.4 Algoritam

- Algoritam je opis postupka za rešavanje nekog problema
- reč algoritmi nastala je od latinskog prevoda prezimena persijskog naučnika, astronoma i matematičara iz IX veka Al Horezmija
- Računarski algoritam je ?
 opis konačnog broja koraka za rešavanje nekog problema na računaru
- nezavisan je od programskog jezika
- Opis algoritma može biti
- 1. tekst u prirodnom jeziku (narativni opis)
- 2. strukturirani tekst (pseudokod)
- grafički prikaz, npr. dijagram toka (flowchart) ili UML dijagram



1.NARATIVNI OPIS

Algoritam kuvanja kafe?:

Uzeti džezvu i sipati vodu.

Uključiti ringlu.

Sačekati dok voda ne proključa.

Kad voda proključa, skinuti džezvu.

Staviti kafu u džezvu.

Po želji, dodati kašiku šećera.

Isključiti ringlu

Sipati kafu u šoljicu.



2.PSEUDO KOD

Pseudokod - metod opisivanje strukture nekog sistema.

Koriste se reči slične ključnim rečima programskih jezika, ali bez poštovanja striktnih pravila i sintakse koju određeni programski jezik zahteva.

Obično je pseudokod mešavina nekog prirodnog jezika i programskog jezika.

Primer: pseudokod za sortiranje imena u abecedni redosled



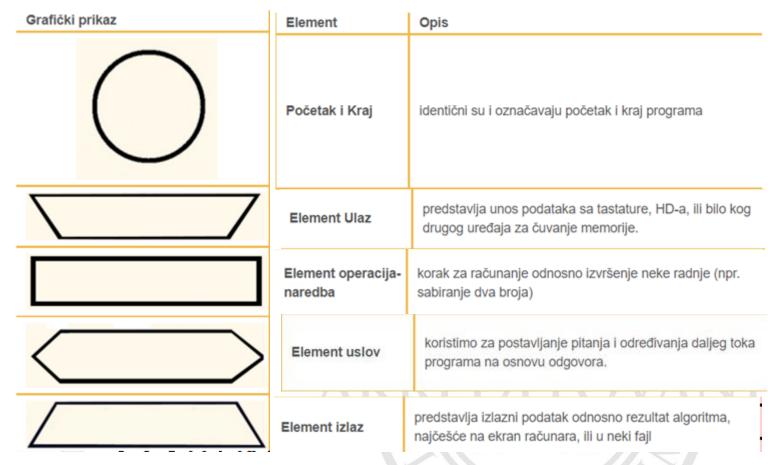
Unesi imena Do Postavi indikator na NE Pogledaj prvi par imena Do While imena nisu u redosledu do Begin Zameni imena Postavi indikator na DA End Pogledaj sledeći par imena Until nema više parova imena Until indikator je NE Odstampaj imena u redosledu





3. Grafički algoritam

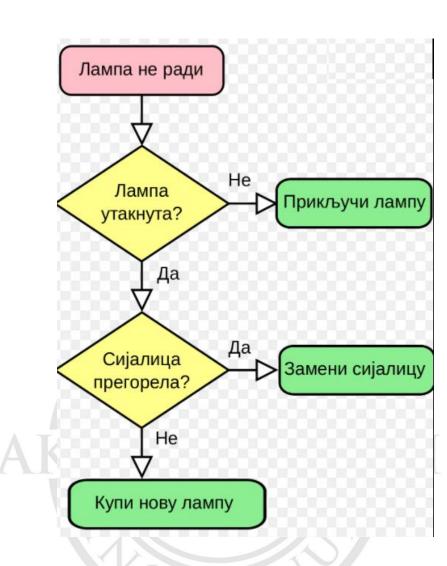
Za crtanje algoritama , koristimo sledeće elemente:





Elemente algoritma međusobno povezujemo sa pravom linijom.

Svaki algoritam mora imati svoj POČETAK i KRAJ.





Primer: Euklidov algoritam

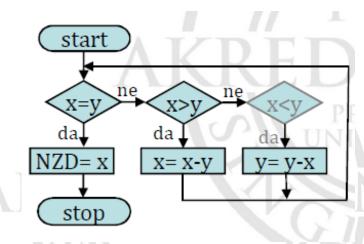
- Algoritam za računanje najmanjeg zajedničkog delioca dva pozitivna cela broja, NZD(x,y). Koji je NZD za 18 i 15?
- 1. Narativni opis:?

Deliti oba broja sa 1 pa sa 2 pa sa 3 itd i uzrti najveći zajednički

2. Pseudokod (strukturirani tekst):

Sve dok je x≠y, ako je x>y onda x = x-y, ako je x<y, onda y = y-x 3. Dijagram toka (*flowchart*)

> Za x=18 i y=15 NZD je 3 Probati...





B. Programski jezici

- 1. Pojam i razvoj
- 2. Izvršavanje programa
- 3. Vrste programskih jezika
- 4. Sintaksa i semantika programskih jezika



B.1 Pojam i razvoj Razvoj programskih jezika

Jedna od podela programskih jezika:

- Programski jezici za velike centralne računare
- FORTRAN, COBOL, Algol, Pascal
- Programski jezici za mini i mikroračunare
- C, BASIC , Visual Basic, Turbo Pascal
- Programski jezici za moderne računare u svetskoj mreži
- C++, Java, C#, PHP, Javascript, R, Python



Istorija programskih jezika

Postoji više hiljada programskih jezika. Ovde su neki od njih navedeni hronološkim redom:

Fortran (1954) C++ (1985)

Cobol (1960) Perl (1987)

Basic (1964) Python (1990)

C (1972) PHP (1994)

Smalltalk-80 (1980) Java (1996)



Oblasti primene programskih jezika

```
Opšta namena (C, Python, Java ...)
```

Matematika i tehnika (FORTRAN,...)

Poslovna obrada podataka (COBOL, ...)

Obuka u programiranju (Pascal, ...)

Veštačka inteligencija (LISP, PROLOG, ..)

Singidunum www.singidunum.ac.

B.3. Vrste programskih jezika (još podela...)

- Programski jezici niskog nivoa
- mašinski jezici (binarno kodirane instrukcije)
- asembleri (simbolički zapis mašinskih instrukcija i adresa)
- Programski jezici visokog nivoa
- Proceduralni programski jezici (za opis algoritama):
- C/C++, Java, C#, Python
- Neproceduralni jezici (ne opisuju način, već traženi rezultat):
- Prolog, SQL
- Proceduralni programski jezici mogu biti
- klasični : Pascal, C
- funkcionalni: LISP, F#
- objektno-orijentisani : C++, Java, C#, Python



Proceduralni jezik specificira kako ce nesto biti izvršeno, a neproceduralni šta ce biti izvršeno, ne ulazeći u detalje kako.

Masinski jezik je 100% proceduralan.

Proceduralni su svi jezici mašinski orijentisani(simbolicki i makro jezici).

Neproceduralni su problemsko i objektno orijentisani.



Implementacija programa u različitim programskim jezicima

- Primer programa koji samo ispisuje na ekran (konzolu) poruku "Pozdrav svima!"
- implementacija u tri različita
 savremena programska jezika
- program u jeziku Python je kraći i jednostavniji

```
C++
#include <iostream>
using namespace std;
int main() {
cout << "Pozdrav svima!" << endl;</pre>
Java
public class PozdravSvima {
public static void main(String[] args)
System.out.println("Pozdrav svima!");
Python
print("Pozdrav svima!")
```



B.4. Sintaksa i semantika programskih jezika

- Sintaksa jezika je ?
- skup znakova i prihvatljivih nizova tih znakova (sekvenci) u nekom jeziku
- Semantika nekog jezika bavi se?
 smislom sintaktički ispravnih nizova znakova
- smisao reči razmatra se u okviru istog jezika, npr. U
 kineskom jeziku reč "Hao" u latiničnoj transkripciji znači
 "dobro", dok u srpskom jeziku ta reč Hao nema smisla

	Srpski	Kineski (pinyin)	Kineski (kinesko pismo)
Sintaksa	Нао	Нао	好
Semantika	<i>besmisleno,</i> sintaktički neispravno	Dobro	UNDobro ZITET



Sintaksa je proučavanje strukture rečenice, dok je semantika proučavanje značenja jezika.

Sintaksa proučava skup pravila, načela i procesa koji vladaju strukturom rečenica na bilo kojem jeziku. Ovde se termin struktura rečenice odnosi na poredak reči. Značenje rečenice može zavisiti o redosledu reči.

2 primera u nastavku.

Jer on banana je jeo gladan.

Jeo je bananu jer je gladan.

Prvi primer nema smisla, ali ako pažljivo pogledate, on sadrži iste reči kao i drugi primer. Jedina razlika postoji u redosedu reči. Stoga je **redosled reči ključni element u sintaksi.**

Ali takodje, rečenica može biti sintaktički ispravna, ali nema smisla.



Bezbojne, zelene ideje besno spavaju.

Da li sintatički i semantički ispravna ili neispravna?

Iako gore navedena rečenica nema smisla, ona je sintaktički ispravna.

U ovoj rečenici možete primetiti da su pridevi, prilozi smešteni u ispravnom redosledu, a subjekt i glagol su u skladu jedan s drugim.



C. Proces razvoja programa

- 1. Algoritam
- 2. Programiranje (kodiranje)
- 3. Objektno-orijentisano programiranje
- 4. Testiranje



C.1 Algoritam

- Kako nastaje algoritam koji rešava neki problem?
- Dizajn algoritma je kreativni proces, koji se tipično sastoji od:
 - 1. razumevanja i definisanja problema
 - 2. izgradnje neke vrste modela problema
 - 3. izbora metoda rešavanja
 - 4. provere ispravnosti i preciznog opisa metoda rešavanja
- Algoritam se može implementirati u različitim programskim jezicima



C.2 Programiranje (kodiranje)

Programiranje je prevođenje algoritma u program u nekom programskom jeziku

Pri tome je algoritam rešenja nekog problema potrebno dokumentovati, npr. objasniti komentarima u kodu programa



C.3 Objektno-orijentisano programiranje

- Objektno-orijentisano programiranje je pristup programiranju koji rešenje nekog problema predstavlja pomoću *objekata* i opisa njihovih svojstava i ponašanja
- objekti međusobno komuniciraju slanjem i primanjem poruka
- Odgovara ljudskom načinu razmišljanja i predstavlja jedan od načina smanjenja složenosti obimnih softverskih programa



C.4 Testiranje

Proces prevođenja algoritma u kod određenog programskog jezika nije jednoznačan i podložan je greškama

Greške mogu biti

- sintaksne greške ili greške u načinu pisanja naredbi programskog jezika, koje otkriva prevodilac
- semantičke greške (logičke, greške u smislu), koje treba da otkrije programer testiranjem



Istorijat: prva računarska greška

- Elektromehanički računar *Harvard Mark II* je 1947. godine otkazao
- Otkriven je uzrok: između kontakata jednog od relea ušla je mušica (moljac), eng. bug
- termin "bug" je, kao naziv za sitne tehničke greške i probleme, poznat još od XIX veka
- povezan je s istoimenim stvarnim uzrokom
- mušica se danas čuva u muzeju, zajedno s dnevničkim zapisom o događaju
- Otud popularni naziv debugging za proces pronalaženja I uklanjanja grešaka u programima



D. Jezik Python

- 1. Istorijat
- 2. Upotreba (zašto Python)
- 3. Instalacija
- 4. Osnovni elementi jezika Python
- 5. Mali primeri
- 6. Velike Web aplikacije



D.1 Istorijat

- Autor programskog jezika i prevodioca je holandski programer Guido van Rossum, ranih 1990-tih
- naziv je dobio po poznatoj britanskoj humorističkoj TV seriji "Leteći cirkus Montija Pajtona" (Monty Python's Flying Circus)
- razvojno okruženje IDLE takođe nosi ime člana grupe (Eric Idle)
- Verzije
- Python 1 1994.
- Python 2 2000.
- Python 3 2008.
- Python 3.4 2014.
- Python 3.5 2015.
- Python 3.6 2016.
- Python 3.6.2 2017.
- Python 3.7.4
- Python 3.83.10, 3.11





Pomalo šaljivi principi (The Zen of Python)

- Lepo je bolje nego ružno
- Eksplicitno je bolje nego implicitno
- Jednostavno je bolje nego složeno
- Složeno je bolje nego komplikovano
- Ravno je bolje nego ugnježdeno
- Retko je bolje nego gusto
- Čitljivost je najvažnija
- Posebni slučajevi nisu dovoljno posebni da krše pravila
- Ipak je praktičnost važnija od čistote
- Greške nikad ne treba prihvatiti ćutke
- ... osim ako se eksplicitno ne ućuti

- Kod se pojavi dvosmislenost, ne treba pogađati
- Trebalo bi da postoji jedan poželjno i samo jedan - očigledan način da se nešto uradi
- ... mada taj način ne mora da bude vidljiv na prvi pogled, osim ako niste Holanđanin
- Sada je bolje nego nikada
 ... mada je nikada često bolje nego upravo sada
- Ako je implementaciju teško objasniti, ideja je loša
- Ako je implementaciju lako objasniti, mora da je ideja dobra
- Prostori imena su sjajna ideja hajde da ih napravimo još!

Komanda:import this



3.čas





D.2 Upotreba (zašto Python)

Prema godišnjoj analizi udruženja inženjera elektrotehnike i elektronike (*Institute of Electrical and Electronics Engineers*, IEEE), jezik Python je **trenutno najpopularniji programski jezik**

Najvažniji razlozi su:

- jednostavna sintaksa i veoma čitljiv kod
- istovremeno velike programske mogućnosti i široka upotreba
- koriste ga poznate velike kompanije i organizacije, kao što su YouTube, Google, Yahoo i NASA
- jezik Python je besplatan i ima dobru podršku www.python.org

Razvojem upravlja neprofitna organizacija *Python Software Foundation*. Sponzori razvoja su npr. *Google, Intel, Microsoft, O'Reilly Media* i *Red Hat*



D.3 Instalacija

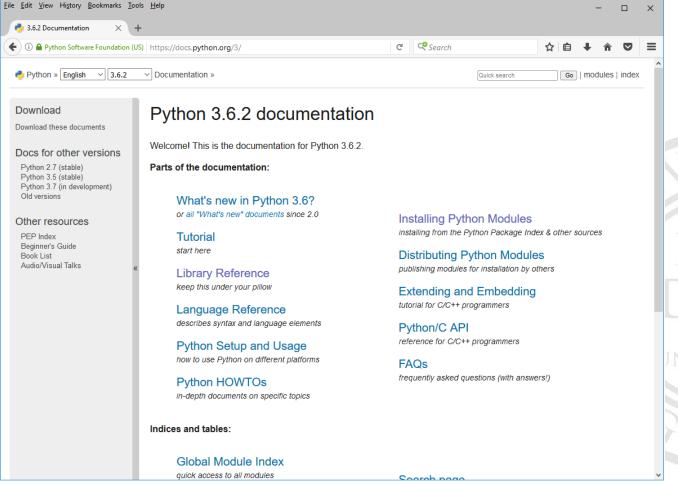
- Koristiće se aktuelna verzija
- Zavisno od vrste računara i operativnog sistema, preuzima se i pokreće odgovarajući instalacioni program





Online dokumentacija

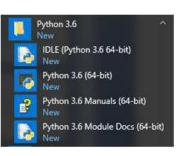
3.12 Documentation (python.org)







Interaktivni editor/interpreter IDLE



Pokretanje interpretera/editora U instalaciju jezika je uključen i interaktivni editor interpreter IDLE koji ćemo koristiti za razvoj programa

Interaktivni unos i izvršavanje pojedinačnih naredbi



Napomena: Python Tools for Visual Studio

Postoji besplatna podrška za razvoj programa u jeziku Python korišćenjem razvojnog okruženja Microsoft *Visual Studio* putem dodatka *Python Tools for Visual Studio*

https://www.visualstudio.com/vs/python/





D.4 Osnovni elementi jezika Python

- Sintaksa
- Naredbe (primeri)
- Standardne biblioteke
- Verzije





Sintaksa

- Program u jeziku Python je niz naredbi, čiji delimiter je kraj linije teksta
- Posebnu ulogu ima uvlačenje (indentacija), koja takođe služi kao delimiter u složenim višelinijskim naredbama
- U naredbama se razlikuju mala i velika slova (*case sensitive*), tako da se npr. naredbe print i Print razlikuju

```
>>> Print(a)
Traceback (most recent call last):
  File "<pyshell#5>", line 1, in <module>
    Print(a)
NameError: name 'Print' is not defined
```



Naredbe (primeri)

- Komentar
- Promenljive i izrazi
- Osnovni tipovi i strukture podataka
- Polja i neuređene liste
- Interaktivni ulaz-izlaz





Komentar

Komentari su važan deo programskog koda (dokumentacija)

```
>>> # Komentar u jednoj liniji
```

```
>>> """
```

Komentar u više linija

111111

```
>>> print(a) # ovo je stampa na ekran 10>>>
```

""" Ово је пример коментара у више редова. У нашем програму ћемо штампати Здраво Индијо. Овде ћемо научити о изјави за штампање.
Штампање """



Promenljive i izrazi

 Dodela vrednosti promenljivoj određuje njen tip (novi red je nova naredba). Kako dodeliti x broj2?

```
>>> x = 2 a kako dodeliti promenljivoj tekst Marko?
```

tip vrednosti promenljive može se ispitati funkcijom type()

• Vrednost promenljive može se i izračunati pomoću izraza,

Kako pomnoziti x sa 2?

$$>>> x = 2*x$$

>>> pozdrav = "Zdravo " + ime

'Zdravo Marko'

```
a=10 # aaa
print(a)
print(type(a))
```

Probati: a=10.1 a=,,Pera" a=True

a = 3 + 2i



Osnovni tipovi i strukture podataka (stringovi, liste i n-torke)

• Nizovi znakova (string), pozicija znaka od 0

>>> ime = 'Milan'

Kako odstampati prva 2 slova reci Milan?

>>> s = ime[0:2]

ime = "Milan"
s = ime[0:2]
print(s)

Rezultat je=Mi



Sta ce se odstampati?

ime = "Milan"

s = ime[1:3]

print(s)





Liste i n-torke: promenljivi i fiksni nizovi *različitih* podataka

$$x = [11,12,13]$$

Sta se dobije sa print(x[1])?

$$z = (1,2,3)$$





>>> x = [1,2,3]

Polja i neuređene liste

>>> print(ul)

Ugnježdene liste (dinamička polja)

```
>>> y = ['jedan', 'dva', 'tri']
                                        [[1, 2, 3], ['jedan', 'dva', 'tri']]
>>> ul = [x,y]

    Ugnježdene n-torke

>>> sn = ((1,2,3),('jedan', 'dva', 'tri'))
>>> print(sn)
((1,2,3),('jedan', 'dva', 'tri'))

    Neuredene liste ili rečnici (dictionaries)

stud = {'210':'Jovan','220':'Marija','245':'Ivan'}
print(stud['220'])
'Marija'
```



Interaktivni ulaz-izlaz

 Program naredbom input postavlja pitanje korisniku i čita odgovor, npr.

>>> ime = input("Kako se zovete?")

Kako se zovete? Jovana





Standardne biblioteke

• Standardna biblioteka se sastoji od velikog broja ugrađenih modula realizovanih u jezicima C i Python

https://docs.python.org/3/library/

- npr. matematičke funkcije, funkcije za rad s fajlovima, pristup operativnom sistemu, itd.
- Postoji ogroman broj dodatnih komponeti, programa, modula, paketa i aplikacija za jezik Python, koji se stalno dopunjava

https://pypi.python.org/pypi

a=sqrt(89)

print(a)

Šta je rezultat?

Šta treba dodati?

import math

a=math.sqrt(81)

print(a)





Verzije

- U široj upotrebi su dve verzije jezika
- Verzija 2 (nasleđene aplikacije i biblioteke programa)
- do 2008. godine, najnovija verzija 2.7.13 od 2016.
- Verzija 3 (aktuelna)
- Postoji programski alat koji automatizovano prevodi programe napisane u sintaksi verzije 2 u sintaksu koju koristi verzija 3 jezika Python



D.5 Mali primeri

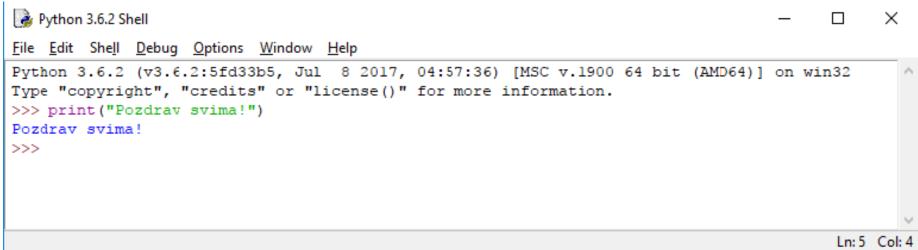
• Primer 1: Pozdrav

• Primer 2: Putovanje





Primer 1: Pozdrav







Primer 2: Putovanje

 Program za računanje vremena trajanja putovanja t na osnovu prosečne brzine kretanja vozila v i udaljenosti d između polazne lokacije i odredišta:

t = d/v

- Vrednost promenljivih d i v korisnik unosi interaktivno, na zahtev programa
- Algoritam na kome se program zasniva može se izložiti narativno:
- 1. učitaj brzinu v (u km/h) i udaljenost d (u km)
- 2. izračunaj vreme trajanja putovanja t = d/v
- 3. prikaži rezultat t



Realizacija algoritma u jeziku Python

Program za računanje vremena trajanja putovanja na osnovu brzine kretanja i udaljenosti odredišta:

Ovako treba da izgleda:

Unesite brzinu (km/h): 120

Unesite udaljenost (km): 210

Uz brzinu 120 km/h, potrebno je

1.75 sati putovanja da se pređe 210.0 km.

Pritisni taster Enter za kraj



Programi se čuvaju na fajlovima koji u svom nazivu imaju sufiks ".py". Izvršavaju se kao jedna celina:

vreme.py

```
brzina=input("unesite brzinu:")
brzina=int(brzina)
udalj=input("unesite udaljenost:")
udalj=float(udalj)
vreme=udalj/brzina
print(brzina,udalj,vreme)
```



Potrebna je konverzija promenljivih
Brzina=int(brzina)
Udaljenost=float(udaljenost) ili int(udaljenost)

Ovo je neophodno jer karakteri koji se unose sa tastature, učitavaju se kao niz znakova pa ih treba pretvoriti u brojeve



D.6 Poznate aplikacije

- Poznate aplikacije napisane u jeziku Python su, npr.
- Blender3D softver za 3D animacije, igre i video montažu
- Drobox sajt za Veb hosting fajlova
- Firefox jedan od najpoznatiih Veb čitača (browser)
- Google App Engine mnoge komponente najpoznatijeg pretraživača i platforme za razvoj Veb aplikacija
- Instagram društvena mreža za deljenje fotografija
- Pinterest društvena mreža za deljenje fotografija, videa i drugih materijala
- Quora poznati sajt i mreža za postavljanje pitanja
- Yahoo Maps određeni servisi softvera za rad s mapama
- Youtube najpoznatiji sajt za deljenje i prikaz videa



Literatura

- 1. Miškovic V., *Osnove programiranja Python*, Univerzitet Singidunum, 2017 (3.6)
- 2. Liang D., *Introduction to Programming Using Python*, Pearson Education, 2013 (3.2)
- 3. Dierbach C., Introduction to Computer Science Using Python: A Computational Problem-Solving Focus, John Wiley & Sons, 2013 (3.1)
- 4. Hetland M. L., *Beginning Python: From Novice to Professional*, 3rd Ed, Apress, 2017 (3.5)
- 5. Downey A. B., Think Python, 2nd Ed, O'Reilly Media, 2016 (3.4)
- 6. Python www.python.org
- 7. Python 3 tutorijel https://docs.python.org/3.4/tutorial/index.html
- 8. Learn python3 in Y Minutes https://learnxinyminutes.com/docs/python3/
- 9. Stanford CS41 Python http://stanfordpython.com/