

PRAKTIKUM IZ SOFTVERSKIH ALATA U ELEKTRONICI

2020/2021

Predrag Pejović

16.01.2021, 10:00

© 2021 Predrag Pejović, 

This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](#).

Linkovi na primere:

- ▶ OS1
- ▶ OS2
- ▶ L^AT_EX 1
- ▶ L^AT_EX 2
- ▶ L^AT_EX 3
- ▶ L^AT_EX 4
- ▶ Python 1
- ▶ Python 2
- ▶ Python 3
- ▶ GNU Octave
- ▶ SageMath

Alternativni linkovi na primere:

- ▶ OS1
- ▶ OS2
- ▶ \LaTeX 1
- ▶ \LaTeX 2
- ▶ \LaTeX 3
- ▶ \LaTeX 4
- ▶ Python 1
- ▶ Python 2
- ▶ Python 3
- ▶ GNU Octave
- ▶ SageMath

PRAKTIKUM IZ SOFTVERSKIH ALATA U ELEKTRONICI

2020



PRAKTIKUM IZ SOFTVERSKIH ALATA U ELEKTRONICI

2020

© Predrag Pejović, 

Lica (i ostali podaci o predmetu):

- ▶ dr Predrag Pejović, red. prof, peja@etf.rs, 102 levo,
<http://tnt.etf.rs/~peja>
<http://peja.freedombox.rocks/peja/>
- ▶ Strahinja Janković, viši laboratorijski inženjer, jankovics@etf.rs
- ▶ sajtovi predmeta (mirror):
 - ▶ <http://tnt.etf.rs/~oe4sae>
 - ▶ <http://peja.freedombox.rocks/PSAE/index.html>
- ▶ cilj: savladati primenu niza programa koji se koriste za svakodnevne poslove u elektronici (i ne samo elektronici ...)
- ▶ svi programi koji će biti obrađivani su slobodan softver (free software), legalno možete da ih koristite (i ne samo to) gde hoćete, kako hoćete, za šta hoćete, koliko hoćete, na kom računaru hoćete ... i nikome ne podnosite račune za to!
- ▶ literatura ... sve sa www, legalno, besplatno!
- ▶ zašto svake godine updated slajdovi?

Lica, dopuna, odakle ja u ovome?

- ▶ nekada davno, drugom polovinom prošloga veka ...
- ▶ ... postojalo je usmereno obrazovanje
- ▶ i OVROMTS „Veljko Vlahović“ ...
- ▶ koja se sada zove „Matematička gimnazija“
- ▶ postojala su zanimanja, ja sam bio **programer**
- ▶ ... neki od učenika su bili veoma entuzijastični u korišćenju računara ... oni koji to ne razumeju bi rekli „zavisnici“
- ▶ ulazio sam u programiranje i izlazio, iz raznih razloga ...
- ▶ dok nisam „definitivno“ izašao 1995 sa win 95 ...
- ▶ ... da bih se vratio 2008. sa Ubuntu 8.04 Hardy Heron
- ▶ i to još traje!
- ▶ korišćenje računara je uživanje! **ako nije — ne koristim!!!**

Prezentacije predmeta, (linkovi ljubičasti)

- ▶ engleski
- ▶ srpski, kraća verzija
- ▶ engleski, prezentacija i animacije
- ▶ srpski, prezentacija i animacije

Izazvalo dosta galame, kao i dosta marginalizacije, ignorisanja ...

Biće još reči o ovome ...

Šta se radiLO u predmetu, koji programi?

1. uvod (upravo slušate): organizacija nastave + (FS: tehnička, ekonomski i pravna pitanja, kako to uopšte postoji?) (≈ 1 w)
2. operativni sistem (GNU/Linux, Ubuntu), komandna linija (!), shell scripts, IDE, poneki program ... (≈ 2 w)
3. tekst, \LaTeX (≈ 3 w)
4. crtanje šema, XCircuit (≈ 1 w)
5. probni kolokvijum ... (= 1 w)
6. **prvi kolokvijum** ...
7. numerika, GNU Octave (≈ 1 w)
8. vizualizacija, gnuplot ($\approx (1 + \varepsilon)$ w)
9. simboličko računanje, wxMaxima (≈ 1 w)
10. **drugi kolokvijum** ...
11. Python, IPython, PyLab, SymPy (≈ 3 w)
12. **treći kolokvijum = ispit** ...

Šta se sada radi u predmetu, koji programi?

1. uvod (upravo slušate): organizacija nastave + (FS: tehnička, ekonomski i pravna pitanja, kako to uopšte postoji?) (≈ 1 w)
2. operativni sistem (GNU/Linux, Ubuntu), komandna linija (!), shell scripts, IDE, poneki program ... (≈ 2 w)
3. tekst, L^AT_EX (≈ 3 w)
4. crtanje šema, XCircuit (≈ 1 w)
5. probni kolokvijum ... (= 1 w)
6. prvi kolokvijum ...
7. Python, verzija 3, PyLab (≈ 3 w)
8. drugi kolokvijum ...
9. numerika, GNU Octave (≈ 1 w)
10. numerika, Julia (≈ 1 w)
11. simboličko računanje, SageMath (≈ 1 w)
12. treći kolokvijum = ispit ...

Evolucija računarstva i društvene nauke

- ▶ teorijski, počinje 1940s
- ▶ praktično, kod nas možda 1970s
- ▶ 1980s koristimo računar
- ▶ 1990s koristimo računar u svakodnevnom radu
- ▶ 2000s sve radimo pomoću računara
- ▶ 2010s potpuno zavisimo od računara
- ▶ 2020s? pogledajte oko sebe, tokom online predavanja ...
- ▶ **PC, mobile, cloud, phone, tablet, ...**
- ▶ nema više PC? za koga nema? vi **niste** prosečni korisnici!
- ▶ ubrzava se ...
- ▶ nove opasnosti: security i privacy!
- ▶ ako neko ima mogućnost da vas iskoristi, iskoristiće vas; budite **sposobni!**

Problemi u radu sa računarima . . .

- ▶ svaki program zahteva obuku . . .
- ▶ čitati uputstvo? dostupna literatura? rano odustajanje . . .
- ▶ najbrže i najbolje se uči ako neko pokaže (“hands-on approach”)
- ▶ pomoćnik-savetodavac dostupan?
- ▶ Google pomoćnik? Youtube pomoćnik?
- ▶ koji program odabratи? stvaranje zavisnosti (navike + old files)? ozbiljna odluka! “**vendor lock-in**”!
- ▶ da li računar pomaže ili odmaže? kada se vraća naučeno?
- ▶ strahovito brze promene, novi programi . . .
- ▶ prilagođavanje: psihologija, ergonomija, namere autora programa (vagon primer, pričali mi učesnici, prvo Rade Nikolić, potom Dušan Vuksan, primedbu dao Helmut Krauch) . . .
- ▶ ekstremne posledice: tehnofobija illi zavisnost
- ▶ cilj: **računar treba da služi nama, ne mi njemu**

Pravni i ekonomski problemi

- ▶ Pravni problemi
 - ▶ licenciranje (program koji ste kupili je vaš?)
 - ▶ **MORAMO biti 100% legalni!!!**
- ▶ Ekonomski problemi
 - ▶ troškovi (i ne samo to! ↓)
 - ▶ navike i zavisnost
 - ▶ problemi sa proprietary formatima (primer: bas bih voleo opet da pročitam svoj doktorat u elektronskoj formi)
 - ▶ bezbednost podataka!
 - ▶ privatnost!!!
 - ▶ način naplate ...
 - ▶ opravdanost naplate ...
 - ▶ “*vendor lock-in*” ...
 - ▶ free (slobodni) software?
 - ▶ besplatno (ako tako hoćete) i jos mnogo više!

Kako je nekada bilo . . . (“učiteljica života”)

- ▶ mainframe računari
- ▶ software i hardware se prodavali zajedno
- ▶ računari bili ekskluzivan proizvod, skup
- ▶ mnogo manje korisnika
- ▶ mnogo manje komunikacije između korisnika
- ▶ okruženje bilo uglavnom naučno ili visoko poslovno
- ▶ poslovne primene uz razvoj sopstvenih programa, custom design
- ▶ dominantna zaštita poslovna tajna
- ▶ ogromne mogućnosti, pps, strahovito brz razvoj
- ▶ društvene implikacije, nestanak i nastanak velikog broja poslova, promena sistema vrednosti, pojava mladih ljudi na uticajnim pozicijama . . .

Kako je danas? (posle ≈ 1985)

- ▶ PC revolucija (prošla; prošla?)
- ▶ jeftino, dostupno, (modularno (?))
- ▶ sklopivo, prilagodljivo korisniku (?)
- ▶ izaberete komponente (?), sklopite računar (?), šta dalje?
- ▶ softver postao zasebna komponenta računara, posebno se nabavlja (kupuje?) (upravo ovde naša priča počinje . . .)
- ▶ operativni sistem, početak
- ▶ korisnički softver, zavisno od primene računara
- ▶ softver je intelektualni, nematerijalni proizvod . . .
- ▶ nematerijalni proizvodi su **veoma** specifični,
upotreboom se ne troše!
- ▶ rezultat: monopol i „verski ratovi“
- ▶ ideologizacija i manipulacija . . .
- ▶ Samizdat: And Other Issues Regarding the 'Source' of Open Source Code

Specifičnosti „intelektualnih“ (nematerijalnih) proizvoda 1

- ▶ značajni troškovi razvoja, skoro svi troškovi su razvoj!
- ▶ proizvod se upotrebom ne troši, traje
- ▶ potražnja se spontano ne obnavlja
- ▶ za zaradu neophodan novi razvoj ili vremenski ograničena licenca ili naplata po korišćenju, **mesečni račun** (cloud!)
- ▶ misaoni eksperiment: softver koji se kvari
- ▶ prestaje „podrška“
- ▶ troškovi umnožavanja zanemarljivi
- ▶ troškovi distribucije zanemarljivi
- ▶ kako organizovati proizvodnju da bi se ostvario profit?
- ▶ karakteristike veoma slične stvaralaštву u nauci, oblast slična matematici ...
- ▶ postoje programeri koji vole da programiraju ...
- ▶ ... dok ne dođu GUI i dokumentacija!

Specifičnosti „intelektualnih“ (nematerijalnih) proizvoda 2

(navedene osobine se uglavnom odnose na softver, manje na ostale intelektualne proizvode)

- ▶ ogroman komercijalni potencijal
- ▶ potencijal za stvaranje zavisnosti (izuzetno opasno!)
- ▶ potencijal za ucenjivanje, posledica zavisnosti, “vendor lock-in”
- ▶ problemi vezani za privatnost: kontrola, pa izvor zarade!
- ▶ primeri:
 - ▶ Phil Zimmerman, Boulder, CO, 1991, PGP
 - ▶ Mark Shuttleworth (1973), Thawte, digital certificates and Internet security
 - ▶ Mark Shuttleworth, Canonical, 2004, biće još reći ...
- ▶ sjajna osnova za izgradnju “social machinery” ni oko čega ...
- ▶ u čemu ima mnogo interesa ... koji se odlučno brane!

„Intelektualna svojina“ i njena „zaštita“

pokušaj da ideje dobiju tretman materijalnih objekata, a time i svog vlasnika; za sada delimično uspešan, uprkos paradoksima forme „intelektualne svojine“:

1. **copyright**
2. **patent**
3. **trade secret**
4. **trademark**, za nas manje važan
5. ima još sličnih ↑ pojmove koji nisu u našoj zoni ...

bitno se razlikuju!!!

**suština: svojina nad objektima koji nisu materijalni,
parcelisanje Platonovog sveta ideja**

Copyright

- ▶ štiti konkretan proizvod, ne ideju
- ▶ dugo traje, i preko 100 godina, zavisno od pravnog sistema (jurisdikcije)
- ▶ produžava se
- ▶ širi se domen
- ▶ sve copyrightable automatski je copyrighted (Bern ...)
- ▶ copyright notice, "all rights reserved" (ne košta!)
- ▶ dominantna zaštita za software u nizu pravnih sistema ...
- ▶ pošto se software kopiranjem umnožava, „proizvodi“, što je još jedna besmislena generalizacija; proizveo je programer!
- ▶ **ne sprečava da neko napiše nov program iste funkcionalnosti!**

Patent

- ▶ štiti ideju
- ▶ "kratko" traje (10-20 godina)
- ▶ kratko? za sotver kratko?
- ▶ širi se domen (predmet, metod, nova primena, organizam, . . .)
- ▶ zahteva prijavu i odobrenje patenta, **košta!**
- ▶ David Pressman, "Patent it Yourself," Nolo Press
- ▶ offensive rights
- ▶ samo 10% patenata isplati troškove (????)
- ▶ kritike, 19. vek i ranije, "Patent Absurdity"
<http://patentabsurdity.com/>
- ▶ nečitki patenti, promenjen smisao
- ▶ pojam „očigledno“
- ▶ kako to funkcioniše u praksi: patent portfolio, trolovi i pravi korisnici patentnog sistema . . .

Trade Secret (poslovna tajna)

- ▶ ne košta
- ▶ traje dok traje tajna
- ▶ nikog ne sprečava da otkrije isto
- ▶ zaštita kroz NDA
- ▶ negde se može primeniti, negde ne
 - 1. primer: program u interpretativnom jeziku
 - 2. primer: proizvodnja hrane, recepti
- ▶ funkcioniše gde se nešto može sakriti ...

Softver, terminološke zabune

1. Free Software
2. Open Source Software
3. Freeware
4. Shareware

Napomena: slobodan softver \neq besplatan softver, to su dva potpuno različita pojma! biće još reči o ovome!!!

Napomena: “A license (American English) or licence (British English) is an official permission or permit to do, use, or own something (as well as the document of that permission or permit).”

Free Software, definicija; free = slobodan

- ▶ **Richard Mathew Stallman**
- ▶ free as in free speech, not free beer (slobodan / besplatan)
- ▶ uvek je neko „platio“ razvoj programa
- ▶ korektan prevod „slobodan softver“
- ▶ definicija (po RMS):
 - ▶ Freedom 0: da se izvršava program, bez ograničenja
 - ▶ Freedom 1: da se proučava i menja program:
potreban source code
 - ▶ Freedom 2: da se dele kopije programa
 - ▶ Freedom 3: da se dele modifikovane verzije programa
- ▶ kad licenca ovo dopušta software je free
- ▶ FSF, <http://www.fsf.org>
- ▶ FSFE, <http://fsfe.org/>
- ▶ osim praktičnih, pokreću i etička pitanja

Novo i neobično, . . .

- ▶ neprijatelji? brojni!
 1. interesni, racionalni
 2. ideološki, iracionalni (teško ih je razumeti i predvideti)
- ▶ korisnici? pitaju li se oni šta? znaju li oni šta?
- ▶ motivi contributors?
- ▶ treba li programeri da budu plaćeni?
- ▶ poslovni modeli? baš ozbiljno pitanje! indirektni!
- ▶ „pustite nas da radimo!“
- ▶ “social machinery”
- ▶ konkurenčija (lojalna, nelojalna?)
- ▶ **potencijal: jedna donacija uništi tržište!**
- ▶ prošlost kod nas (do ≈ 2003.): sve je bilo freeware, jako malo free software; srećom, promenilo se!

Podela softvera

- ▶ free software (slobodni softver)
 - ▶ licenca dopušta četiri slobode iz definicije
 - ▶ izvorni kod (source code) uvek dostupan
 - ▶ ima **više** licenci koje to omogućuju, dve grupe:
 1. **restriktivne**, copyleft: GNU GPL (koriste: GNU/Linux)
 2. **liberalne**, dopustive: BSD, MIT (koriste: BSD, Python, Spice)
 - ▶ biće još reči o licencama, a i rad na **PSSOH** ...
- ▶ proprietary software (vlasnički softver)
 - ▶ licenca ne dopušta bar jednu od navedene četiri slobode
 - ▶ izvorni kod može biti dostupan
 - ▶ izvorni kod obično nije dostupan, poslovna tajna
 - ▶ korisnik baš i ne zna šta program radi na njegovom računaru
 - ▶ maksimiranje profita utiče na software design, primeri ...
 - ▶ “**vendor lock-in**”, budite oprezni!
- ▶ postoji i dvojno licenciranje, ali to je “advanced topic” ...

Free Software

Da ponovimo, dopušta:

- ▶ Freedom 0: da se izvršava program, bez ograničenja
- ▶ Freedom 1: da se proučava i menja program:
potreban source code
- ▶ Freedom 2: da se dele kopije programa
- ▶ Freedom 3: da se dele modifikovane verzije programa

Open Source Software

- ▶ Bruce Perens i Eric Steven Raymond (ESR)
- ▶ "The Cathedral and the Bazaar"
- ▶ Open Source Initiative, www.opensource.org
- ▶ u osnovi, to je free software, iste licence, gde je razlika?
- ▶ ideja: popraviti marketing (ne zamerati se)
- ▶ podržao Linus Torvalds
- ▶ u početku uspeh (sistemu je bilo šta prihvatljivije od slobode)
- ▶ softver kome se zna izvorni kod MOŽE biti i proprietary!!!
Primer: FFTW, dvojno licenciranje, i slobodan i neslobodan
- ▶ gotovo uvek je (open source = free) software
- ▶ F(L)OSS (Free, (Libre), and/or Open Source Software)
- ▶ ozbiljan sukob sa FSF, ali koristili i dalje GPL
- ▶ danas je sukob prošlost, mada je ostalo FOSS i FLOSS

Freeware

- ▶ proprietary, ali besplatan softver
- ▶ besplatan iz nekog razloga
 - ▶ reklame
 - ▶ prikupljanje informacija
 - ▶ stvaranje navike kod korisnika
 - ▶ testiranje programa (besplatno)
 - ▶ testiranje tržišta
 - ▶ želja za sakrivanjem dela koda ili podataka, nečim motivisana; čime?
- ▶ obično nepoznat source code
- ▶ proučiti motive pre upotrebe, razmisliti
- ▶ primer: [LTspice](#), da ima za GNU/Linux možda bih i koristio!
- ▶ **shvatite i zapamtite:**
slobodan softver \neq besplatan softver!

Shareware

- ▶ problem kod reči “sharing”
- ▶ softver koji se slobodno distribuira
- ▶ još manje slobodan nego freeware
- ▶ obično služi za promociju programa
- ▶ ograničena funkcionalnost i/ili vremensko ograničenje

GNU

- ▶ 1983. ili 1984., GNU project, **RMS**, **MIT**
- ▶ recursive acronym, GNU's Not Unix
- ▶ da bude i **ostane** free
- ▶ nije baš bilo bez pomoći sa strane, prećutna podrška MIT ...
- ▶ kako da ostane free?
- ▶ izmišljen **copyleft** (jako važno! biće još reči ...)
- ▶ <http://copyleft.org/>
- ▶ zašto baš Unix?

Unix, UNIX (i C)

- ▶ Unix ...
- ▶ 1969, AT&T, Ken Thompson, Dennis Ritchie, Brian Kernighan, Douglas McIlroy, and Joe Ossanna
- ▶ portability
- ▶ modularity
- ▶ Unix philosophy
- ▶ vrlo povoljna osnova za GNU!
- ▶ C kompjajler i modul po modul, ...

Dennis Ritchie, #1

Following Ritchie's death, computer historian Paul E. Ceruzzi stated:

"Ritchie was under the radar. His name was not a household name at all, but . . . if you had a microscope and could look in a computer, you'd see his work everywhere inside."

Dennis Ritchie, #2

Views on computing

In an interview from 1999, Dennis Ritchie clarifies that he sees Linux and BSD operating systems as a continuation of the basis of the Unix operating system, and as derivatives of Unix:

"I think the Linux phenomenon is quite delightful, because it draws so strongly on the basis that Unix provided. Linux seems to be among the healthiest of the direct Unix derivatives, though there are also the various BSD systems as well as the more official offerings from the workstation and mainframe manufacturers."

POSIX

Linux

- ▶ do 1991. skoro sve gotovo
- ▶ nedostajao kernel OS
- ▶ RMS očekivao Hurd
- ▶ Hurd ne baš sjajno napredovao
- ▶ **Andy Tanenbaum, Minix**
- ▶ Linus Torvalds, University of Helsinki → Linux
- ▶ konačno sve gotovo, free OS
- ▶ počele distribucije, ...
- ▶ „nesporazumi oko imena“
- ▶ danas uglavnom GNU/Linux ili GNU+Linux

Ličnosti

- ▶ Richard Mathew Stallman (RMS)
- ▶ Eben Moglen
- ▶ Linus Torvalds
- ▶ (Larry Lessig, Creative Commons)
- ▶ Mark Shuttleworth

Licence ...

- ▶ Understanding Open Source and Free Software Licensing
- ▶ **GPL, General Public License, FSF, copyleft**, detaljno na sledećem slajdu ...
 1. GNU General Public License
 2. GNU Lesser General Public License
 3. GNU Affero General Public License
 4. GNU Free Documentation License
- ▶ liberalne, “permissive”, licence:
 1. BSD License,
 2. MIT License
 3. ...
- ▶ dvojno licenciranje moguće kod copyleft, FFTW
- ▶ CeCILL
- ▶ <http://www.gnu.org/licenses/license-list.html>
- ▶ PSSOH rad ...

Važno: Copyleft

- ▶ Copyright: "all rights reserved", Tiny BASIC
- ▶ Copyleft: "all rights reversed" ili "all wrongs reversed" ili "some wrongs reversed"
- ▶ cilj da se free software zaštiti, da ostane slobodan
- ▶ restriktivne licence
- ▶ „virusna licenca“
- ▶ Copyleft and the GNU General Public License:
A Comprehensive Tutorial and Guide

Značajne copyleft licence:

1. The GNU General Public License
2. The GNU Lesser General Public License
3. The GNU Affero General Public License
4. The GNU Free Documentation License

Spice, BSD licenca, original

http://en.wikipedia.org/wiki/BSD_licenses

7/17/2007

Spice is covered now covered by the BSD Copyright:

Copyright (c) 1985-1991 The Regents of the University of California.
All rights reserved.

Permission is hereby granted, without written agreement and without license or royalty fees, to use, copy, modify, and distribute this software and its documentation for any purpose, provided that the above copyright notice and the following two paragraphs appear in all copies of this software.

IN NO EVENT SHALL THE UNIVERSITY OF CALIFORNIA BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN IF THE UNIVERSITY OF CALIFORNIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE UNIVERSITY OF CALIFORNIA SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE SOFTWARE PROVIDED HEREUNDER IS ON AN "AS IS" BASIS, AND THE UNIVERSITY OF CALIFORNIA HAS NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

MIT license

http://en.wikipedia.org/wiki/MIT_License

Copyright (C) <year> <copyright holders>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

A Šta je sa hardverom?

- ▶ „otvorení hardver“
- ▶ Arduino: Creative Commons Attribution Share-Alike License
- ▶ ... kao i niz GPL licenci ...
- ▶ Interaction Design Institute Ivrea, 2001-2005
- ▶ “design files” su javno dostupni i slobodni za primenu; copyleft? licence?
- ▶ ozbiljna igra: RISC-V, UC Berkeley, Krste Asanović
- ▶ BSD licenca
- ▶ SiFive
- ▶ reakcija ARM-a
- ▶ vredi pomenuti i <https://www.raspberrypi.org/> i slične “single board computers” projekte
- ▶ WikiHouse

Cloud?

- ▶ There is no cloud: it's just someone else's computer!
- ▶ baš ništa novo, superkomputeri, Boulder 199x, $3 \leq x \leq 5$
- ▶ Who does that server really serve?
- ▶ baš dobar koncept: ni softver ni vaši podaci nisu kod vas!
- ▶ stvara se zavisnost, "vendor lock-in"
- ▶ nekome se isplati: isplati li se vama?
- ▶ Service as a Software Substitute (SaaS)
- ▶ svakako aktuelna tema ...
- ▶ ... posebno u doba online nastave!
- ▶ moje mišljenje: **odbaciti!**
- ▶ a da imate svoj lični cloud?
- ▶ a da imate svoj lični server?

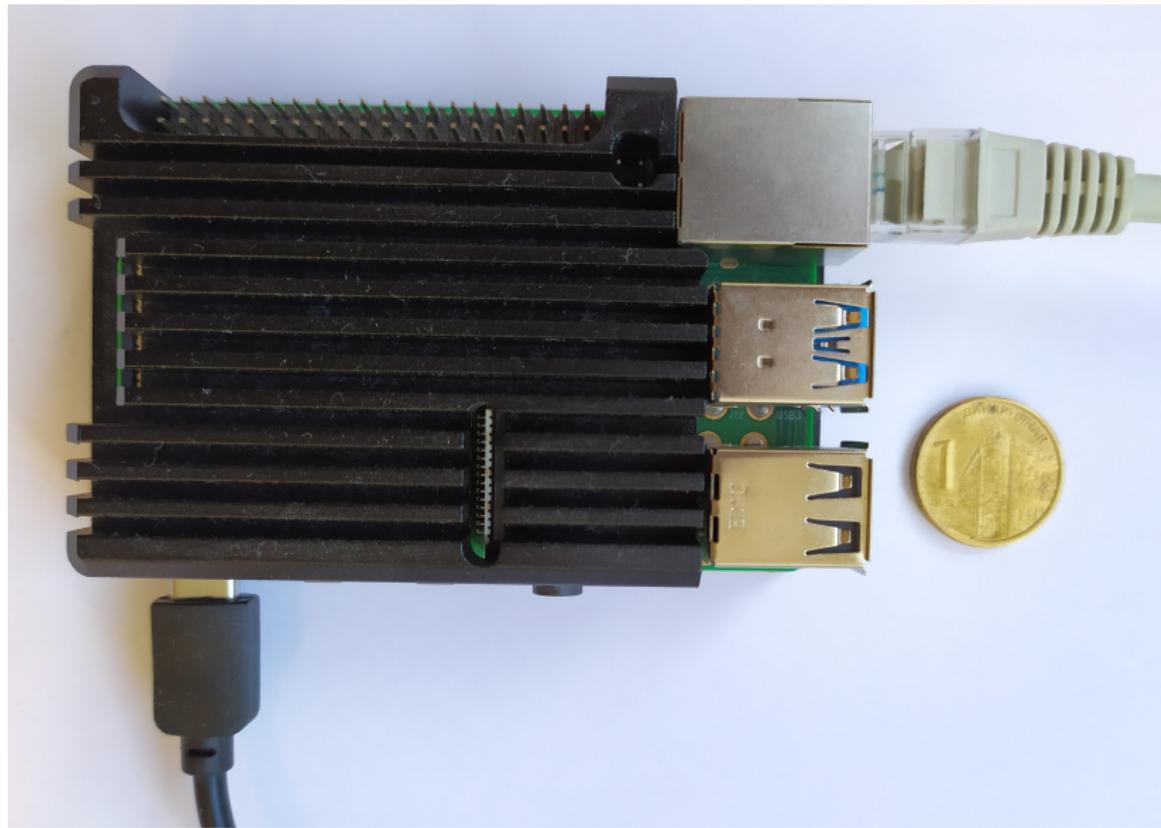
Lični serveri ...

- ▶ poput <http://peja.freedombox.rocks>
- ▶ advanced topic ...
- ▶ hot topic ...
- ▶ no time to cover ...
- ▶ just some links:
 1. <https://freedombox.org/>
 2. <https://yunohost.org/#/>
 3. <https://nextcloud.com/>
 4. <https://ownyourbits.com/nextcloudpi/>
 5. ...
- ▶ po meni, ovo je budućnost, evolucija fiksnog telefona ...
- ▶ za spretnije, <https://ubuntu.com/download/server> ...
- ▶ ... sada imate punu slobodu!

<http://peja.freedombox.rocks>



Modernija verzija, RPi 4, ...



Funkcionalnosti kućnog servera?

da ne prevodim, ovo mahom pričam na engleskom ...

1. e-mail server
2. personal cloud ...
3. personal communication server
4. personal library
5. home page and/or blog?
6. proxy server
7. vpn server
8. smart home
9. video surveillance
10. ...

pogledati šta sve ima <https://yunohost.org/#/>

Par reči o enkripciji ...

- ▶ koliko god se trudili da je onemoguće, nije moguće bez enkripcije, elektronska trgovina, e-everything
- ▶ backdoor ako nije free software? a, da, rekli su da nema!
- ▶ javni ključ
- ▶ Simon Singh, Knjiga o šiframa, The Code Book
- ▶ PGP, Pretty Good Privacy
- ▶ kombinacija:
 1. Thunderbird
 2. Pretty Easy Privacy
- ▶ Mailvelope za webmail?
- ▶ termin: “end-to-end encryption”
- ▶ termin: “transport encryption”

Organizacija predmeta

- ▶ sasvim praktičan predmet, lak; lak (?)
- ▶ savladati ((samostalno) korišćenje) niza programa
- ▶ organizacija nastave: radionica, "hands-on approach"
- ▶ ... ovo baš i ne ide online
- ▶ **Q: koliko ljudi za računarom?**
- ▶ **A: preferably 1!**
- ▶ ... eto socijalnog distanciranja!
- ▶ maksimum 20 studenata, najbolje 10, sa 4 m^2 po studentu rezultat je 0 studenata
- ▶ ocenjivanje: mali kolokvijumi, 3 kolokvijuma
- ▶ za koga predmet **nije**
- ▶ ako ste nekompatibilni, sada je trenutak za zamenu

Program predmeta, teme nekada

(ono što je realno moglo da se stigne prethodnih godina)

- ▶ Ubuntu, osnove i komandna linija, shell scripts
- ▶ C, Python, IDEs, kao ilustracija prethodnog
- ▶ \LaTeX
- ▶ XCircuit
- ▶ GNU Octave
- ▶ gnuplot
- ▶ Maxima, wxMaxima
- ▶ **Python**, PyLab (NumPy, SciPy, Matplotlib, IPython), SymPy

politika: dva alata za istu namenu kada to ima ikakvog smisla; kada znate dva, lako ćete naučiti treći; Jawaharlal Nehru i pismo

Program predmeta, teme sada

- ▶ **Ubuntu**, osnove i komandna linija, shell scripts
- ▶ C, Python, IDEs, kao ilustracija prethodnog
- ▶ **LATEX**
- ▶ **XCircuit**
- ▶ **Python**, PyLab (NumPy, SciPy, Matplotlib, IPython), SymPy
- ▶ **GNU Octave**
- ▶ **Julia**
- ▶ **SageMath**

Šta bi još moglo da se uvrsti u predmet?

- ▶ VirtualBox
- ▶ LibreOffice
- ▶ Scilab, Xcos
- ▶ Inkscape
- ▶ GIMP, ImageMagick
- ▶ Asymptote
- ▶ git
- ▶ Qucs
- ▶ ngspice
- ▶ KiCad
- ▶ R
- ▶ ...

Ispit, ocenjivanje

- ▶ kolokvijumi/ispiti sa literaturom
 1. dokumentacija 1/3 (1. kolokvijum)
 2. programiranje 1/3 (2. kolokvijum)
 3. matematika (scientific computing) 1/3 (3. kolokvijum = ispit)
- ▶ ne traži se memorisanje, već snalaženje
- ▶ na kolokvijumu/ispitu jedino nije dozvoljena saradnja među studentima
- ▶ Ocenjivanje po kolokvijumima:
 1. LaTeX (80) + XCircuit (20) + Beamer (20) (bonus)
 2. Python (50) + PyLab (50)
 3. Octave (20) + Julia (40) + SageMath (40)
- ▶ otvoreno za primedbe i sugestije (do određene granice) ...

Linkovi . . .

sajt predmeta:

<http://tnt.etf.bg.ac.rs/~oe4sae/>

alternativni sajt predmeta:

<http://peja.freedombox.rocks/PSAE/index.html>

edX Linux kurs (bez programa koje obrađuje PSAE):

<https://www.edx.org/course/linuxfoundationx/linuxfoundationx-lfs101x-introduction-1621>

HONOR CODE CERTIFICATE



Jerry Cooperstein, Ph. D.

Training Program Director
The Linux Foundation

Clyde Seepersad

General Manager, Training
The Linux Foundation

Predrag Pejovic

successfully completed and received a passing grade in

LFS101x: Introduction to Linux

a course of study offered by LinuxFoundationX, an online learning initiative of The Linux Foundation through edX.

HONOR CODE CERTIFICATE
Issued August 15th, 2014

Verify the authenticity of this certificate at
<https://verify.edx.org/cert/0b2944574c99408e8092adc07b1f9cc5>

Linkovi za zainteresovane . . .

Richard Matthew Stallman, TEDx, Geneva, 2014.

Eben Moglen, kako je sve počelo . . .

Eben Moglen, gde smo sada, 2013.

Revolution OS, dokumentarni film iz 2001.

O patentima . . .

promeniti ili ne promeniti, pitanje je sad?

week #1

— the end —

operativni sistem

— part 1 —

© Predrag Pejović, 

operativni sistem

- ▶ služi da bi pokretali i koristili programe
- ▶ win, macOS, GNU/Linux, BSD, Solaris, OpenIndiana, ...
- ▶ GNU/Linux, distribucije:
 - ▶ Ubuntu
 - ▶ Linux Mint
 - ▶ fedora
 - ▶ debian
 - ▶ openSUSE
 - ▶ KNOPPIX (sjajno za probu i popravke, live DVD, live USB disk, "live medium")
 - ▶ ... distrowatch
 - ▶ i jedna knjiga: [linux from scratch](#)
- ▶ Ubuntu ...
- ▶ ... i baš za hardline free-distros

Ubuntu

- ▶ www.ubuntu.com
- ▶ desktop, server, Kubuntu, Xubuntu, Lubuntu, Edubuntu, Ubuntu GNOME, Ubuntu MATE ... menja se ... (Gnubuntu, Ubuntu-libre, Gobuntu)
- ▶ dnload, narezati DVD ili napraviti USB startup disk
- ▶ narezati? InfraRecorder, <http://infrarecorder.org/>
- ▶ USB startup? na win <https://rufus.ie/>
- ▶ pre instalacije PROBATI, live DVD, live USB disk, "live medium"
- ▶ Ubuntu, instalacije
 - ▶ samo Ubuntu
 - ▶ dual boot (dva diska? pazite!)
 - ▶ wubi, inside windows, prošlost
 - ▶ **VirtualBox**, iso file, + some proprietary drivers ...

desktop environments

- ▶ kod windows samo win
- ▶ duga istorija pre toga, Xerox, Apple, ...
- ▶ kod GNU/Linux mnoštvo desktop environment mogućnosti
- ▶ popularni **GNOME**, **KDE**, **Xfce**, **LXDE**, **MATE**, **Cinnamon**, do skoro **Unity** ...
- ▶ svaki od desktop environments ima svoje motive, istoriju, izgled, ... **HIG**
- ▶ na kraju sve skoro isto, ali treba vremena da se to shvati ...
- ▶ možete da instalirate i štošta drugo: **GNOME**, **MATE**, **Cinnamon**, **Cairo**, ...
- ▶ sjajna zabava kada imate višak vremena

setting up the system . . .

- ▶ instalira se operativni sistem . . .
 - ▶ ... proprietary drivers (uglavnom graphichs i wlan) . . .
 - ▶ ... proprietary codecs . . . ogg? <http://xiph.org/>
- ▶ programi se instaliraju iz repository . . .
- ▶ za instalaciju potreban **jedan** disk ili USB flash drive i **Internet**
- ▶ za probu sjajno ako imate stari računar/laptop
- ▶ moje iskustvo sa instalacijama . . .
- ▶ a sada prepostavimo da je operativni sistem instaliran . . .
- ▶ to je pre online nastave bilo lako prepostaviti . . .
- ▶ sada instalirate Ubuntu 20.04 i treba da instalirate programe
- ▶ idemo korak po korak . . .

instaliranje programa . . .

1. Ununtu Software
2. Synaptic
3. komandna linija:
 - 3.1 sudo apt-get ...
 - 3.2 sudo apt ...
 - 3.3 sudo snap ...

desktop environment ... DEFAULT!

- ▶ zašto default?
- ▶ **insistiramo na defaults**
- ▶ customize sami ... na **SVOM** nalogu i/ili računaru
- ▶ **pitanje pristojnosti** ...
- ▶ do sada nije bilo problema ...
- ▶ svoje podatke ponesite na USB flash drive ... ako su vam važni
- ▶ ne ostavljajte podatke na računaru, počistite ...
- ▶ rekoh, **pitanje pristojnosti** ...

desktop environment ...

- ▶ sistem boot-ovan
- ▶ let's log in ...
- ▶ launcher (application switcher)
- ▶ top menu bar
- ▶ indicators ...
- ▶ activities ...
- ▶ applications ...
- ▶ Desktop

launcher . . .

- ▶ levi bar
- ▶ pokreće (lansira) aplikacije
- ▶ ostale aplikacije se dodaju (Add to Favorites) ili uklanaju (Remove from Favorites), desno dugme miša
- ▶ ostalim aplikacijama se lako menja redosled
- ▶ aplikacija se pokreće sa single click
- ▶ pokrenuta aplikacija: tačka (puna)
- ▶ vežba: Octave, $2 + 2$

Show Applications

- ▶ pretražuje računar, i applications i files
- ▶ pokreće se sa single click dole levo
- ▶ Frequent / All
- ▶ name search

vežba . . .

- ▶ start Software Updater
- ▶ update
- ▶ start Ubuntu Software Center
- ▶ install GeoGebra
- ▶ run GeoGebra
- ▶ nacrtajte elipsu i još ponešto
- ▶ Add to Favorites
- ▶ Remove from Favorites
- ▶ pokrenete GNU Octave . . .
- ▶ Alt-Tab menja aplikacije

get acquainted, startovanje programa, . . .

- ▶ updated?
- ▶ Octave, plot x^2 za $0 \leq x \leq 1$, . . . kao Matlab, isto!
- ▶ `octave`, poznato?
- ▶ `octave-cli`, za old-fashioned
- ▶ `ipython3`
- ▶ terminal, `ipython3`, `a=2`, `b=3`, `a^b`, `a**b`
- ▶ terminal, `jupyter notebook`
- ▶ gedit, iz Launcher-a
- ▶ terminal, `gnuplot`, plot $\sin(x)$
- ▶ terminal, `gedit`

top menu bar . . .

- ▶ levo, **Activities**
- ▶ sredina, time/date, notifications
- ▶ desno, system status and parameters
- ▶ sound
- ▶ Internet connection
- ▶ keyboard switcher
- ▶ settings
- ▶ . . .

system settings . . .

- ▶ System Settings
- ▶ pogledati Details
- ▶ u 18 **ne menjati** System Settings!
- ▶ rekoh, pitanje pristojnosti!

razlike . . .

- ▶ različiti desktop environments imaju različite nazine i različit raspored komandi
- ▶ primer: Linux Mint na Električnim merenjima
- ▶ ipak, ista funkcionalnost se ostvaruje
- ▶ potrebno kratko navikavanje i povremena potraga da se nešto pronađe
- ▶ pitanje ukusa o kojima nema smisla raspravljati . . .

Files . . .

- ▶ obično prva ikona, mada ne mora
- ▶ ime sve kaže
- ▶ dva prozora (taba), ctrl-t, može i više
- ▶ gase se klikom (desno je x, levo je ime)
- ▶ full path ctrl-l, izlaz sa esc
- ▶ ovo je otprilike sve za početak . . .

file system . . .

- ▶ Home Folder (Home Directory), ima svaki korisnik
- ▶ Ctrl/l, full path, Location, wheel-button
- ▶ subfolders, create your folder
- ▶ MSDOS: **folder**; UNIX, GNU/Linux: **directory** (mnogo ovakvih situacija)
- ▶ MSDOS: directory separator \; UNIX, GNU/Linux: directory separator /
- ▶ MSDOS: command option separator /; UNIX, GNU/Linux: command option separator -, --
- ▶ POSIX, UNIX, Linux, BSD, . . . jedan tree, vratiću se . . .
- ▶ win: svaka storage device ima tree, razlika!
- ▶ **moć navike!** (dobre i lose navike)
- ▶ komandna linija? već ste je sreli: terminal

komandna linija . . .

- ▶ Terminal (izlaz: exit ili Ctrl/d)
- ▶ Alt/Ctrl F1 do F6, tty1 do tty6, neki je vaš prozor sada je to F2
- ▶ eto nas na komandnoj liniji
- ▶ jako dobra knjiga:
<http://sourceforge.net/projects/linuxcommand/>
- ▶ ali 555 strana!
- ▶ lako se čita!!! (telefon, tablet, prevoz, . . .)

neke komande . . .

- ▶ `date`
- ▶ `cal`
- ▶ `cal -help`
- ▶ `ncal`
- ▶ `df`
- ▶ `clear`
- ▶ `free`
- ▶ od sada: terminal na medium, File Manager, home, medium
- ▶ ideja: da vidite šta komandna linija radi, kako fajlovi i direktorijumi nastaju i nestaju

vrlo osnovne komande, ls

- ▶ uvek prva komanda, ls, ekvivalent za dir iz MSDOS-a
- ▶ ls, ls -l, ls -a, ls -al, ls -lh
- ▶ boje govore o tipu fajla, zelena je .exe ili .com ili .BAT
- ▶ man ls, man -t ls, man -t ls >ls.ps
- ▶ ls
- ▶ ps2pdf ls.ps
- ▶ ls
- ▶ file ls.ps
- ▶ file ls.pdf
- ▶ evince ls.pdf
- ▶ evince ls.ps
- ▶ postoji i evince za win,
<http://projects.gnome.org/evince/>
- ▶ man convert ...

pokušaj prevar, cp, file

- ▶ ls ls.ps
- ▶ cp ls.ps lspdf.pdf
- ▶ ls lspdf.pdf
- ▶ file lspdf.pdf
- ▶ cp ls.pdf lsps.ps
- ▶ ls lsps.ps
- ▶ file lsps.ps

cat, more, less . . . biće još jednom

- ▶ cat ls.ps
- ▶ cat ls.ps ls.ps
- ▶ man cat
- ▶ ko je napisao cat?
- ▶ more ls.ps
- ▶ less ls.ps
- ▶ “less is more!”, autor

cat, gedit i redirekcija

- ▶ gedit proba
- ▶ sadržaj fajla po želji, npr dva reda teksta, save!!!
- ▶ cat proba proba proba proba proba > proba5.txt
- ▶ cat proba5.txt
- ▶ more proba5.txt
- ▶ less proba5.txt
- ▶ izlaz je q, ne esc!!!

počistite za sobom!

- ▶ ls
- ▶ rm ls.ps
- ▶ ls
- ▶ rm -i ls.pdf
- ▶ ls
- ▶ ^d
- ▶ shutdown mašine, potom ugasite monitore
- ▶ vratite stolice na mesto
- ▶ ponesete sve svoje stvari

operativni sistem

— part 2 —

© Predrag Pejović, 

zagrevanje . . .

- ▶ **update**
- ▶ komandna linija, nastavak . . . mnoštvo programa
(jezik se zaboravlja ako se ne govori)
- ▶ scripting, bash, sh
- ▶ malo uvoda u Python, IDLE, idle
- ▶ malo C
- ▶ (ni)malo IDE, Integrated Development Environment,
Code::Blocks, Eclipse (drugi to rade . . .)

help na komandnoj liniji . . .

- ▶ `man ls` (manual)
- ▶ `ls --help` (help)
- ▶ `info ls` (GNU info system)
- ▶ `apropos`, `man apropos`
- ▶ `whatis`, `whatis apropos` (vrlo kratak opis)
- ▶ `whereis`, `whereis apropos` (vrlo kratak opis)

synaptic

- ▶ još jedan način za instaliranje programa
- ▶ jako dobro, omogućava precizna podešavanja
- ▶ nije instaliran by default ...
- ▶ zahteva neko razumevanje ...
- ▶ pokrenete Synaptic Package Manager
- ▶ dopunite octave, modul octave-control
- ▶ pokrenete `octave`
- ▶ `h = tf([1], [1, 1])`
- ▶ `bode(h)`
- ▶ `h = tf([10], [1, 0, 100])`
- ▶ `bode(h)`
- ▶ `h = tf([10], [1, 0.1, 100])`
- ▶ `bode(h)`

file system, nastavak . . .

- ▶ bitno razlikit file system u odnosu na MS-DOS/win
- ▶ ne postoje volumes poput C:\, D:\, ...
- ▶ Ext 3, Ext 4, journal file system
- ▶ case sensitive, kada to može, ne kod FAT;
ovo pravi probleme kod kopiranja na USB flash,
File i file primer, preklapanje
- ▶ jedinstven file system, stablo počinje u root, /
- ▶ volumes se mount-uju: ubacite USB flash
- ▶ USB flash, CD, u /media/, pogledajte sa ctrl/l
- ▶ a sada kroz file system pomoću komandne linije . . .
- ▶ terminal na pola, File Manager na pola, . . .

directory system commands: pwd, mkdir, rmdir, cd . . .

- ▶ pwd
- ▶ mkdir <ime>
- ▶ ls
- ▶ cd <ime>
- ▶ pwd
- ▶ ls
- ▶ cd ..
- ▶ cd -
- ▶ cd ~
- ▶ cd /
- ▶ cd ~/<ime>
- ▶ cd
- ▶ cd <ime>
- ▶ pwd

redirekcija, cat, tac, head, tail, more, less . . .

- ▶ ls -lht /usr/bin > usrbin1.lst
- ▶ ls /usr/bin > usrbin2.lst
- ▶ ls -lht > ls.lst
- ▶ ls
- ▶ cat usrbin1.lst
- ▶ tac usrbin1.lst
- ▶ head usrbin1.lst
- ▶ tail usrbin2.lst
- ▶ cat usrbin1.lst usrbin2.lst
- ▶ cat usrbin1.lst usrbin2.lst > usrbin3.lst
- ▶ more usrbin3.lst
- ▶ less usrbin3.lst

pipe, grep . . .

- ▶ ls -lht /usr/bin | more
- ▶ ls -lht /usr/bin | less
- ▶ ls /usr/bin | more
- ▶ ls /usr/bin | less
- ▶ grep maxima usrbin1.lst
- ▶ grep maxima usrbin2.lst
- ▶ ls -lht /usr/bin | grep maxima

grep i regular expressions

- ▶ grep, globally search a regular expression and print
- ▶ regular expression
- ▶ knjiga, 197 strana
- ▶ Google Analytics, Regular Expressions

scripting ...

- ▶ pokrenuti editor, `gedit radi`
- ▶ View, Highlight Mode, sh
 - ↑ podešava context highlighting
- ▶ scripts su obično sa ekstenzijom sh
- ▶ mada ne mora ...
- ▶ videćete šta radi file

script file 1: radi

```
ls -lh /usr/bin > usrbin1.lst
ls /usr/bin > usrbin2.lst
ls -lh > ls.lst
mkdir lst
mv usrbin1.lst lst
mv usrbin2.lst lst
mv ls.lst lst
cd lst
ls -lh
cd ..
```

script file 2: briši

```
gedit brisi.sh
```

```
rm -i lst/*.lst
```

startovanje shell file kao data file

- ▶ **file radi**, koji je tip fajla?
- ▶ **ls -lh radi**, biće još reči o ovome
- ▶ **bash radi**
- ▶ pogledajte šta je urađeno!
- ▶ **bash brisi.sh**
- ▶ pogledajte šta je urađeno!
- ▶ **sh radi**
- ▶ pogledajte šta je urađeno!
- ▶ **sh brisi.sh**
- ▶ pogledajte šta je urađeno!

privileges . . .

- ▶ `ls -lhtr`
- ▶ i postoji nešto nalik na `-rw-r--r--` pa podaci o fajlu
- ▶ i nešto nalik na `drwxr-xr-x` pa podaci o direktorijumu
- ▶ prvi znak: `-` za fajl ili `d` za direktorijum
- ▶ `r` je read, `w` je write, `x` je execute (`ls`)
- ▶ tripleti `rwx` `rwx` `rwx`
 user group other
- ▶ svaka privilegija (`r`, `w` ili `x` je jedan bit)
- ▶ triplet daje oktalnu cifru, `rwx` je 7, `r--` je 4, `--x` je 1, ...
- ▶ privilegija se kodira kao trocifreni oktalni broj
- ▶ `rwx-----` je 700
- ▶ `rwxr--r--` je 744
- ▶ `rw-r--r--` je 644

chmod . . .

- ▶ ls -lh radi
- ▶ chmod <code> <file_name>
- ▶ chmod 700 radi
- ▶ ls -lh radi
- ▶ pozeleneo?
- ▶ drugi način
- ▶ chmod {u|g|o}{+|-}{r|w|x} <file_name>
- ▶ chmod u-w radi
- ▶ ls -lh radi
- ▶ chmod u+w radi
- ▶ ls -lh radi
- ▶ chmod go+x radi
- ▶ ls -lh radi
- ▶ chmod go-r radi
- ▶ radi
- ▶ ./radi

script file 3

```
gedit argumenti0.sh
```

```
#!/bin/bash
echo $0
echo $1
echo $2
echo $3
```

```
file argumenti0.sh
sh argumenti0.sh
sh argumenti0.sh a b c
sh argumenti0.sh d e
bash argumenti0.sh
./argumenti0.sh
```

#!

- ▶ #!
- ▶ promenite u argumenti0.sh prvi red u
#!/bin/sh
i save kao argumenti0a.sh
- ▶ promenite u argumenti0.sh prvi red u
#!/usr/bin/python
i save kao argumenti0b.sh
- ▶ obrišete prvi red i save kao argumenti0c.sh
- ▶ file argumenti0.sh
- ▶ file argumenti0a.sh
- ▶ file argumenti0b.sh
- ▶ file argumenti0c.sh

shell? malo istorije (za zainteresovane) . . .

- ▶ https://en.wikipedia.org/wiki/Unix_shell
- ▶ https://en.wikipedia.org/wiki/Thompson_shell
- ▶ https://en.wikipedia.org/wiki/Bourne_shell
- ▶ https://en.wikipedia.org/wiki/Stephen_R._Bourne
- ▶ https://en.wikipedia.org/wiki/Bash_%28Unix_shell%29
- ▶ i jedna knjiga:
<http://sourceforge.net/projects/linuxcommand/>

script file 4

gedit argumenti1.sh

```
#!/bin/bash
echo 0
echo $0
echo 1
echo $1
echo 2
echo $2
echo 3
echo $3
```

file argumenti1.sh
sh argumenti1.sh
.argumenti1.sh

script file 5

```
gedit argumenti2.sh
```

```
#!/bin/sh
echo 0: $0
echo 1: $1
echo 2: $2
echo 3: $3
```

```
file argumenti2.sh
sh argumenti2.sh
```

environment variables, PATH

- ▶ `printenv`
- ▶ `printenv | less`
- ▶ `printenv PATH` (prikaže vrednost promenljive PATH)
- ▶ `printenv PATH | grep home`
- ▶ `echo $PATH` (prikaže vrednost promenljive PATH)
- ▶ `$PATH` (shell pokušava da izvrši)

environment variables, PWD

- ▶ `pwd` (print working directory)
- ▶ `printenv PWD` (isto!)
- ▶ `echo $PWD` (isto!)
- ▶ `echo PWD` (ništa korisno)
- ▶ `$PWD` (pokuša da izvrši)
- ▶ `echo $pwd` (ništa)
- ▶ `echo $(pwd)` (ništa)
- ▶ `$(pwd)` (pokuša da izvrši)

proširimo PATH!

- ▶ echo \$PATH
- ▶ echo \$PATH:\$PWD
- ▶ PATH=\$PATH:\$PWD
- ▶ printenv PATH
- ▶ printenv PATH | grep home
- ▶ radi

čišćenje ... uvek!!!

- ▶ cd lst
- ▶ ls
- ▶ ls -1
- ▶ rm -i *
- ▶ ls
- ▶ cd ..
- ▶ rmdir lst

vaš prvi Python program, proba, bez .py . . .

- ▶ python, imate li instaliran python koji nije python3?
- ▶ za ovaj primer razlika je samo u ()

```
#!/usr/bin/python3

print()
print('Hello World!')
print()
print('MPFC')
print()
```

pokretanje, . . .

- ▶ `python3 proba`
- ▶ `chmod +x proba`
- ▶ `./proba`
- ▶ `chmod -x proba`
- ▶ `./proba`
- ▶ `python3 proba`
- ▶ `cp proba proba.py`
- ▶ `ls -lhtr`
- ▶ `idle` (Not IDE, Eric Idle?)
- ▶ File, Open, proba.py
- ▶ context highlighting
- ▶ Run, Check Module (Alt+X)
- ▶ Run, F5

vaš drugi Python program, f.py, in gedit . . .

```
#!/usr/bin/python3

print()
n = int(input('argument = '))
print()

f = 1
for i in range(n):
    f = f * (i + 1)

print(n, "!" "=", f)
print()
```

pokretanje . . .

- ▶ da probamo sva tri nacina, sa i bez .py?
- ▶ `ipython3`
- ▶ `run f.py`
- ▶ `ctrl/d ili ^d`
- ▶ vrednosti promenljivih ostaju u interpreteru
- ▶ nema problema sa velikim brojevima, dynamic typing
- ▶ a kako je počelo? ovako:
- ▶ `python3`
- ▶ `execfile('f.py')`
- ▶ `^d`

Zadatak: Fibonačijevi brojevi

- ▶ $f_n = f_{n-1} + f_{n-2}$, $f_1 = 1$, $f_2 = 1$
- ▶ lista od prvih n Fibonačijevih brojeva

Fibonačijevi brojevi, Python

```
#!/usr/bin/python3

n = int(input('duzina liste (int, >2): '))

print()

a = b = 1

print(16 * '-')

i = 1
print(repr(i).rjust(3) + ' ' + repr(a).rjust(10))
i = 2
print(repr(i).rjust(3) + ' ' + repr(b).rjust(10))

for i in range(2,n):
    a, b = b, a + b
    print(repr(i+1).rjust(3) + ' ' + repr(b).rjust(10))

print(16 * '-')
```

zadatak, C program, faktorijel

- ▶ Dennis MacAlistair Ritchie (Username: dmr, September 8, 1941 – October 8, 2011)
- ▶ "UNIX is very simple, it just needs a genius to understand its simplicity."
- ▶ "C is quirky, flawed, and an enormous success."
- ▶ zatraži argument sa **argument =**
- ▶ izračuna faktorijel i javi rezultat u formi **m! = n**
- ▶ **gedit faktorijel.c**
- ▶ pišete program ...

jedan C program . . .

```
#include <stdio.h>

int main()
{
    int i, n, f;

    printf("\nargument = ");
    scanf("%d", &n);

    f = 1;
    for (i=1; i<=n; i++) {f=f*i;}

    printf("\n%d! = %d\n\n", n, f);

    return 0;
}
```

GCC, prevodenje

- ▶ `gcc -o faktorijel faktorijel.c`
- ▶ `ls -lh`
- ▶ zeleni `faktorijel?`
- ▶ `./faktorijel`
- ▶ problemi sa velikim brojevima?

Fibonačijevi brojevi, C, simplified

```
#include <stdio.h>

int main(void)
{
    printf("\nn = ");
    int n;
    scanf("%d", &n);

    int i = 1, j = 1, k, c;
    printf("\n\n1\n1\n");

    for (c = 2; c <= n; c++)
    {
        k = i + j;
        i = j;
        j = k;
        printf("%d\n", k);
    }

    return 0;
}
```

korak napred, IDE, Code::Blocks

- ▶ IDE, Integrated Development Environment
- ▶ Eclipse, Code::Blocks i Geany popularni ... ovde C::B
- ▶ gdb, command line, sami
- ▶ kreirate projekat, Empty
- ▶ Debug/Release version
- ▶ Build
- ▶ watches
- ▶ breakpoints
- ▶ start, step into, shift-F7
- ▶ izvršavanje korak po korak, F7
- ▶ na kraju build Release version
- ▶ F7 ili button panel
- ▶ watch Local variables
- ▶ just to get acquainted ...

LATEX

— I deo —

© Predrag Pejović, 

Uvod 1

- ▶ TeX (ili TeX), sistem za pripremu dokumenata,
<http://en.wikipedia.org/wiki/TeX>
- ▶ Donald Knuth,
http://en.wikipedia.org/wiki/Donald_Knuth,
70's, 1978 tačno
- ▶ Free/Open-Source Software (FOSS)
- ▶ standard za slaganje teksta, posebno naučnog, knjige, formule,
import slika, čak i crtanje ...
- ▶ Computer Modern family of fonts,
http://en.wikipedia.org/wiki/Computer_Modern
- ▶ može sve da se uradi, ali ponekad previše slobode škodi, ...

Uvod 2

- ▶ \LaTeX (ili \TeX), sistem makroa za \TeX ,
<http://en.wikipedia.org/wiki/LaTeX>
- ▶ „a document markup language and document preparation system for the \TeX typesetting program“
- ▶ Markup? Lako ćete naučiti **html** ako znate \LaTeX !
- ▶ Leslie Lamport,
http://en.wikipedia.org/wiki/Leslie_Lamport, 80's
- ▶ \LaTeX je ono što se svakodnevno koristi
- ▶ ima još „nadogradnji“, **LyX**
- ▶ i dalje je sve free software
- ▶ i dalje svi dokumenti u ascii formatu, u slobodnom formatu, standardizovanom

Uvod 3

- ▶ suštinski, lako se uči
- ▶ zaista daje kvalitetan izgled dokumenata
- ▶ platforme na kojima je implementiran uglavnom free software
- ▶ sve što je potrebno postoji kao free software
- ▶ a potrebno je:
 - ▶ editor teksta
 - ▶ \LaTeX procesor

Uvod 4, Leslie Lamport

- ▶ “How \LaTeX changed the face of Mathematics”, an E-interview with Leslie Lamport, the author of \LaTeX
- ▶ Q: *Is \LaTeX hard to use?*
- ▶ A: *It's easy to use—if you're one of the 2% of the population who thinks logically and can read an instruction manual. The other 98% of the population would find it very hard or impossible to use.*

Uvod 5, Leslie Lamport

Even if there were some point to teaching kids such an esoteric system, it couldn't be done for the same reason that it's been impossible to raise the level of math and science in this country — namely, kids can't learn from teachers who don't know the subject well, and people who are good in math and science don't become grade-school teachers.

Uvod 6, Leslie Lamport

Q: *Three \LaTeX mistakes that people should stop making?*

A:

1. *Worrying too much about formatting and not enough about content.*
2. *Worrying too much about formatting and not enough about content.*
3. *Worrying too much about formatting and not enough about content.*

Editori za L^AT_EX

- ▶ ima ih raznih, Wikipedia, liista i poređenje
- ▶ neki free, neki ne
- ▶ koncentrišemo se na free, samo njih ćemo obraditi
- ▶ može bilo kakav editor, čak i gedit ima context highlighting za TeX, počećemo sa njim
- ▶ komforntiji za rad su specijalizovani editori, nalik na IDE za programanje
- ▶ od specijalizovanih, Texmaker, moj izbor, jako napredovao, sada se ustalio, gotov proizvod
- ▶ Texmaker je „isti“ i pod win i pod lin (GNU/Linux)
- ▶ valja pomenuti i TeXstudio
- ▶ pod win je popularan i TeXnicCenter

LATEX procesori

- ▶ pod lin [TeX Live](#), uglavnom, ima još ...
- ▶ pod win uglavnom [MikTeX](#) ...
- ▶ prevodi *.tex file u *.dvi ili direktno u *.pdf
- ▶ pod lin, komandna linija

`latex <fname>`

ili

`latex <fname>.tex`

rezultat .dvi

- ▶ za direktno prevodjenje u *.pdf

`pdflatex <fname>`

ili

`pdflatex <fname>.tex`

LATEX procesori

- ▶ dvi može da se gleda i štampa, [http://en.wikipedia.org/
wiki/Device_independent_file_format](http://en.wikipedia.org/wiki/Device_independent_file_format)
- ▶ malo staromodan, mada radi ...
- ▶ može da se konvertuje u razne druge formate korišćenjem free
software
 - ▶ dvips konvertuje dvi u ps
 - ▶ dvipdf, dvipdfm, dvipdfmx konvertuju dvi u pdf
 - ▶ dvipng konvertuje dvi u gif ili png
 - ▶ dvisvg, dvisvgm konvertuju dvi u svg
 - ▶ ...
- ▶ pravilo: što manje konverzija — to bolje

\LaTeX osobine

- ▶ obeležen tekst ([markup language](#))
- ▶ obeležavanjem se označava šta sa pojedinim delovima teksta treba raditi, šta je naslov, šta je naslov poglavlja, šta je numerisana lista, šta je nenumerisana lista, šta je jednacina . . .
- ▶ potrebno poznavati „komande“
- ▶ \LaTeX „težak“ za „prosečnog“ korisnika
- ▶ [WYSIWYM](#)
- ▶ [WYSIWYG](#) ([WYSIAYG](#)) i WYSIWYM konvergiraju?
- ▶ preambula (header) i (obeležen) tekst

LaTeX resursi

- ▶ puno resursa na www, Google your problem
- ▶ <http://www.tug.org/>
- ▶ <http://www.tug.org/ctan.html>
- ▶ <http://www.andy-roberts.net/misc/latex/>
- ▶ <http://en.wikibooks.org/wiki/LaTeX>
- ▶ <http://upload.wikimedia.org/wikipedia/commons/2/2d/LaTeX.pdf>, pdf verzija, 748 strana
- ▶ pročitati do sutra ...

LATEX resursi, za win ako baš hoćete to

- ▶ <http://miktex.org/>
- ▶ eventualno <http://www.ghostscript.com/>
- ▶ <http://www.texniccenter.org/>
- ▶ ili (po mom ukusu): <http://www.xmlmath.net/texmaker/>

LaTeX resursi

- ▶ pod lin sve već imate u repository, dovučete Texmaker i to je sve što treba da uradite?
- ▶ ne bas, ponekad mora još nesto da se dodovuče, Synaptic
- ▶ ili: `sudo apt-get install texlive-full`
- ▶ bitno je da vam je mreža pri ruci ako nemate sredjenu mašinu
- ▶ **po mom ukusu:** <http://en.wikibooks.org/wiki/LaTeX>, 748 strana
- ▶ **po mom ukusu:** <http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf>, 153 strane
- ▶ čitati uputstvo?
- ▶ ako baš mora; bolji hands-on approach!

„A sada, svi na svoje radne zadatke!“

Ako hoćete da imate ikakve koristi od ovog časa:

1. unosite primere svojeručno
2. radite i koncentrišite se na to što radite
3. ako vas mrzi da kucate (!), imate na sajtu primere, kako source files tako i output files koje treba da dobijete
4. NIKAKO ne radite copy/paste sa pdf, biće problema sa ' i još nekim specijalnim znacima; kopirajte sa source files
5. zovite kada negde zapnete
6. ako vas sve ovo mrzi i dosadno vam je, **idite.**

primer 1, da počnemo

gedit, View, Highlight Mode, LaTeX

gedit p1.tex

```
\documentclass{minimal}
```

```
\begin{document}
```

Prva proba za \LaTeX.

```
\end{document}
```

procesiranje na komandnoj liniji

- ▶ u direktorijumu ste gde je p1.tex
- ▶ kucate **pdflatex p1** (ekstenzija .tex se ovde podrazumeva)
- ▶ sve je gotovo (ako nema poruka o greškama)
- ▶ pogledate pdf file sa **evince p1.pdf**
- ▶ ovaj prozor možete da ostavite, automatski refresh (pod lin, doduše)
- ▶ ovako ćemo raditi neko vreme, sticanja navike i stava radi

primer 2, dijakritici

```
\documentclass{minimal}
```

```
\begin{document}
```

Prva proba za \LaTeX.

Šta će biti kada se pojave naši znaci?

```
\end{document}
```

primer 3, dijakritici sa input encoding

```
\documentclass{minimal}  
  
\usepackage[utf8]{inputenc}  
  
\begin{document}
```

Prva proba za \LaTeX.

Šta će biti kada se pojave naši znaci?

```
\end{document}
```

primer 4, dijakritici bez input encoding

```
\documentclass{minimal}
```

```
\begin{document}
```

Prva proba za \LaTeX.

Šta će biti kada se pojave naši znaci?

Možete ovako: \v{S}ta \'{c}e biti kada
se pojave na\v{s}i znaci?

```
\end{document}
```

značenje Tab i Enter

- ▶ Tab služi u formatiranju “source” teksta
- ▶ Tab se u izlazu ne vidi kao n*space
- ▶ zapravo, ne vidi se uopšte
- ▶ jedan Enter je separator kao space
- ▶ služi da se pregledno formatira “source”
- ▶ nov paragraf je prazan red, dva Enter
- ▶ **veoma** povoljno u praksi

primer 5, sreden input encoding

```
\documentclass{minimal}  
\usepackage[utf8]{inputenc}
```

```
\begin{document}
```

Prva proba za \LaTeX.

Šta će biti kada se pojave naši znaci?

A ostaje i: \v{S}ta \'{c}e biti kada se pojave na\v{s}i znaci?

```
\end{document}
```

primer 6, specijalni znaci ne rade

```
\documentclass{minimal}  
  
\usepackage[utf8]{inputenc}  
  
\begin{document}
```

Prva proba za \LaTeX.

Šta će biti kada se pojave naši znaci?

A ostaje i: \v{S}ta \'{c}e biti kada se pojave
na\v{s}i znaci?

A kada dodamo specijalne značke poput < i > ?

```
\end{document}
```

primer 7, specijalni znaci rade, sreden font encoding

```
\documentclass{minimal}  
  
\usepackage[utf8]{inputenc}  
\usepackage[T1]{fontenc}
```

```
\begin{document}
```

Prva proba za \LaTeX.

Šta će biti kada se pojave naši znaci?

A ostaje i: \v{S}ta \'{c}e biti kada se pojave
na\v{s}i znaci?

A kada dodamo specijalne znake poput < i > ?

```
\end{document}
```

jedna posredna konverzija, za primer

```
latex p7  
ls p7.dvi  
dvips p7  
ls p7.ps  
ps2pdf p7.ps  
ls p7.pdf  
evince p7.pdf
```

može i evince p7.ps kao i evince p7.dvi

ovo se može automatizovati

gedit napravi

```
#!/bin/sh
latex $1
dvips $1
ps2pdf $1.ps
evince $1.pdf
```

```
file napravi
chmod u+x napravi
ls napravi
./napravi p7
PATH=$PATH:$PWD
napravi p7
```

Texmaker

- ▶ od sada idemo u prijatnije okruženje, Texmaker
- ▶ pogledati Tools
- ▶ PDFLaTeX je F6
- ▶ View PDF je F7
- ▶ LaTeX je F2
- ▶ Dvi -> PS je F4
- ▶ PS -> PDF je F8
- ▶ Dvi -> PDF je F9
- ▶ context highlighting, spell check, ...

primer 8, Ћирилица у проблему

```
\documentclass{minimal}\usepackage[utf8]{inputenc}\usepackage[T1]{fontenc}\begin{document}
```

Prva proba za \LaTeX.

Šta će biti kada se pojave naši znaci?

A ostaje i: \v{S}ta \'{c}e biti kada se pojave
na\v{s}i znaci?

A kada dodamo specijalne značke poput < i > ?

A kako ide sa ћирилицом?

```
\end{document}
```

primer 9, Ћирилица

```
\documentclass{minimal}
```

```
\usepackage[utf8]{inputenc}  
\usepackage[T1, T2A]{fontenc}
```

```
\begin{document}
```

Prva proba za \LaTeX.

Šta će biti kada se pojave naši znaci?

Ostaje i: \'{c}e biti kada se pojave na\v{s}i znaci?

A kada dodamo specijalne značke poput < i > ?

A kako ide sa ћирилицом?

```
\end{document}
```

primer 10, package datetime, today

```
\documentclass{minimal}

\usepackage[utf8]{inputenc}
\usepackage[T1, T2A]{fontenc}
\usepackage{datetime}

\begin{document}
```

Ilustracija za packages:

```
\bigskip

Данас је \today.

\end{document}
```

primer 11, package babel

```
\documentclass{minimal}

\usepackage[utf8]{inputenc}
\usepackage[T1, T2A]{fontenc}
\usepackage[serbian]{babel}

\usepackage{datetime}
```

```
\begin{document}
```

Ilustracija za packages, sada je dodat babel, serbian:

```
\bigskip
```

Данас је \today.

```
\end{document}
```

primer 12, specijalni znaci

```
\begin{document}
```

А шта да радимо са специјалним знацима?

```
\bigskip
```

% ово је коментар, неће се видети у документу

ако треба у тексту \% он се пише као \textbackslash\%.

можда вам затреба \textasciitilde \% за ~

а можда и витичасте заграде \{ и \}

има тога још: \\$ _ \& \pounds{} \copyright{}
\textless{} \textbar{} \textgreater{}

```
\end{document}
```

primer 13, okruženje verbatim

```
\documentclass{minimal}
```

```
\usepackage[utf8]{inputenc}  
\usepackage[T1, T2A]{fontenc}  
\usepackage[serbian]{babel}
```

```
\begin{document}
```

Дођосмо до окружења. Прво радим окружење \verb-verbatim-

```
\begin{verbatim}
```

Ovde možete kucati šta god hoćete.

И како год хоћете.

Ништа се не форматира.

Слободно куцате \ ~ { }, нема никакво значење.

```
\end{verbatim}
```

```
\end{document}
```

primer 14, noindent

```
\begin{document}

Још мало форматирања. Обрадили смо \verb+\bigskip+.

Da видимо шта ради \verb|\noindent|.

\bigskip

\noindent Има \verb|\noindent|.

Нема \verb|\noindent|.

\noindent
Има \verb|\noindent|.

Нема \verb|\noindent|.

\end{document}
```

primer 15, itemize u problemu

```
\documentclass{minimal}

\usepackage[utf8]{inputenc}
\usepackage[T1, T2A]{fontenc}
\usepackage[serbian]{babel}

\begin{document}
```

Још мало окружења (environments). Листе су следећа тема.

```
\begin{itemize}
    \item prva tačka
    \item druga tačka
    \item treća tačka
\end{itemize}
```

Ovo baš i neće valjati.

```
\end{document}
```

primer 16, itemize

```
\documentclass{article}

\usepackage[utf8]{inputenc}
\usepackage[T1, T2A]{fontenc}
\usepackage[serbian]{babel}
```

```
\begin{document}
```

Још мало окружења (environments). Листе су следећа тема.

```
\begin{itemize}
    \item prva tačka
    \item druga tačka
    \item treća tačka
\end{itemize}
```

Ovo će valjati.

```
\end{document}
```

primer 17, enumerate

```
\documentclass{article}

\usepackage[utf8]{inputenc}
\usepackage[T1, T2A]{fontenc}
\usepackage[serbian]{babel}

\begin{document}

Још мало окружења (environments).
Нумерисане листе су следећа тема.

\begin{enumerate}
    \item prva tačka
    \item druga tačka
    \item treća tačka
\end{enumerate}

\end{document}
```

primer 18, veličina slova

```
\documentclass{article}

\usepackage[utf8]{inputenc}
\usepackage[T1, T2A]{fontenc}

\begin{document}

\noindent A сада мало играња са величином слова:

\Huge \noindent Ali ovo hoće! \\
\huge Ali ovo hoće! \\
\LARGE Ali ovo hoće! \\
\Large Ali ovo hoće! \\
\large Ali ovo hoće! \\
\normalsize Ali ovo hoće! \\
\small Ali ovo hoće! \\
\footnotesize Ali ovo hoće! \\
\scriptsize Ali ovo hoće! \\
\tiny Ali ovo hoće!

\end{document}
```

primer 19, margine i prazna strana

```
% margine na 2cm na a4 i da uklonimo broj strane  
  
\documentclass[a4paper,12pt]{article}  
  
\usepackage[utf8]{inputenc}  
\usepackage[T1, T2A]{fontenc}  
\usepackage[margin=2cm]{geometry}  
  
\begin{document}  
  
\pagestyle{empty} % ima i \thispagestyle{empty}  
  
\Huge \noindent Ali ovo hoće! \\  
\normalsize Ali ovo hoće! \\  
\tiny Ali ovo hoće!  
  
\end{document}
```

primer 20, kvazitabela

```
\begin{document}

\thispagestyle{empty}

\noindent А сада једна табела:

\noindent
\Huge          \verb|\Huge|          \\
\huge          \verb|\huge|          \\
\LARGE         \verb|\LARGE|         \\
\Large          \verb|\Large|          \\
\large          \verb|\large|          \\
\normalsize    \verb|\normalsize|    \\
\small          \verb|\small|          \\
\footnotesize  \verb|\footnotesize|  \\
\scriptsize    \verb|\scriptsize|    \\
\tiny          \verb|\tiny|          \\

\end{document}
```

primer 21: još malo o fontovima i pravim tabelama

```
\begin{document}

\begin{tabular}{|l|l|}

\hline
команда & ефекат \\

\hline
\verb|\textnormal| & \textnormal{Још мало о фонтовима (typefaces).} \\
\verb|\textit| & \textit{Још мало о фонтовима (typefaces).} \\
\verb|\underline| & \underline{Још мало о фонтовима (typefaces).} \\
\verb|\textbf| & \textbf{Још мало о фонтовима (typefaces).} \\
\verb|\emph| & \emph{Још мало о фонтовима (typefaces).} \\
\verb|\textrm| & \textrm{Још мало о фонтовима (typefaces).} \\
\verb|\textsf| & \textsf{Још мало о фонтовима (typefaces).} \\
\verb|\textsc| & \textsc{Још мало о фонтовима (typefaces).} \\
\verb|\textsl| & \textsl{Још мало о фонтовима (typefaces).} \\
\verb|\texttt| & \texttt{Још мало о фонтовима (typefaces).} \\

\hline
\end{tabular}

\end{document}
```

primer 22: gore, dole, levo, desno

```
\documentclass[a4paper,12pt]{article}

\usepackage[utf8]{inputenc}
\usepackage[T1, T2A]{fontenc}
\usepackage[margin=1cm]{geometry}

\begin{document}

\thispagestyle{empty}

горе лево \hfill горе десно

\vfill

доле лево \hfill доле десно

\end{document}
```

LATEX

— II deo —

© Predrag Pejović, 

struktura dokumenta, WYSIWYM

- ▶ zavisi od tipa dokumenta
- ▶ za article:

```
\section{...}  
\subsection{...}  
\subsubsection{...}  
\paragraph{...}  
\ subparagraph{...}
```

primer a, struktura dokumenta, 1

```
\documentclass[a4paper,12pt]{article}

\usepackage[utf8]{inputenc}
\usepackage[T1, T2A]{fontenc}
\usepackage[serbian]{babel}

\usepackage{datetime}
\usepackage[bottom=3cm,top=3cm,left=3cm,right=2cm]{geometry}

\title{Primer jednog struktuiranog dokumenta}
\author{Predrag Pejović}
\date{\today}
```

primer a, struktura dokumenta, 2

```
\begin{document}
```

```
\maketitle
```

```
\section*{Abstract}
```

Ovo je samo primer za odeljak koji nije numerisan, \\ * znači da se ne numeriše.

```
\section{Uvod}
```

U uvodu se piše šta se radi i zašto.

Važan deo uvoda je i pregled postojećih rezultata, to pomaže da se objasni zašto se nešto radi.

primer a, struktura dokumenta, 3

```
\section{Centralni deo 1}
```

Ovde ilustrujemo subsections.

```
\subsection{Prvi pododeljak}
```

Kao što rekoh, prvi pododeljak.

```
\subsection{Drugi pododeljak}
```

Kao što rekoh, drugi pododeljak.

```
\section{Centralni deo 2}
```

Nastavljamo sa izlaganjem \ldots

primer a, struktura dokumenta, 4

```
\section{Zaključak}
```

U zaključku se rezimira šta je urađeno i kako.

Poneko voli i da kaže šta bi još moglo da se uradi.

```
\section*{Literatura}
```

Ovaj odeljak se obično ne numeriše,
ali referenciranje je baš duga priča.

```
\end{document}
```

šta je cilj primera?

- ▶ kako se automatski pravi naslov
- ▶ boldujte naslov, `\textbf{...}`, radi
- ▶ obrisite datum, `\date{}`
- ▶ struktura dokumenta, numerisanje odeljaka
- ▶ preglednost dokumenta, bilo gde nov red
- ▶ prazan red znači nov paragraf
- ▶ nov red sa `\newline` ili `\\\`, bilo ranije
- ▶ podelite dokument po stranama sa `\newpage` i/ili `\clearpage`
- ▶ vertical space, `\vspace{...}`

primer b, doterivanje naslova

```
\title{\LARGE  
\textbf{Primer jednog struktuiranog dokumenta}}  
\author{\textsl{Predrag Pejović}}  
\date{}
```

primer b, brisanje broja strane

može (prva strana ili bilo koja specifična strana)

\thispagestyle{empty}

što je važno kada automatski generiše naslov

i/ili

\pagestyle{empty}

kad prođe naslov, odnosi se na strane osim prve

primer b, table of contents

iza naslova:

```
\vspace{3cm}  
\tableofcontents  
\newpage
```

nenumerisani items nisu u table of contents!

primer b, dodavanje items u table of contents

```
\newpage
```

```
\section*{Literatura}
```

```
\addcontentsline{toc}{section}{Literatura}
```

primer c, još malo environments, 1

```
\begin{document}

\begin{Huge}
    \begin{center}
        Ovde su centrirana velika slova.
    \end{center}
\end{Huge}

\begin{flushright}
    ovde su normalna slova, \\
    ali je sve poravnato \\
    na desnu stranu
\end{flushright}

\end{document}
```

primer d, još malo environments, 2

```
\begin{document}
```

```
\begin{flushleft}
```

Ovaj environment (okruženje) ukida poravnavanje desne ivice, a to je ponekad potrebno.

Primer su US patenti u postupku i neki pravni dokumenti za koje se veruje da su čitljiviji ako se ne poravnava desna ivica, mada manje lepo izgledaju. O ukusima ne vredi raspravljati, a u osnovi je sasvim svejedno.

```
\end{flushleft}
```

```
\end{document}
```

primer e, još malo environments, 3

```
\begin{document}

\begin{description}
    \item[Prvi] je onaj koji dolazi pre drugog
    \item[Drugi] je onaj koji dolazi posle prvog
    \item[Treći] je u ovom slučaju poslednji
\end{description}

\end{document}
```

primer f, malo o razmacima

```
\documentclass[12pt, a4paper]{article}
```

```
\usepackage[utf8]{inputenc}  
\usepackage[T1, T2A]{fontenc}
```

```
\begin{document}
```

Mi smo na ETF. Nismo na ETH.

Mi smo na ETF\@. Nismo na ETH. % dodaje razmak

```
\bigskip
```

Ono je Prof. Davić.

Ono je Prof.\ Davić. % smanjuje razmak

```
\end{document}
```

primer g, još malo o razmacima

```
\documentclass[12pt, a4paper]{article}
\usepackage[utf8]{inputenc}
\usepackage[T1, T2A]{fontenc}
```

```
\begin{document}
\frenchspacing
```

Mi smo na ETF. Nismo na ETH.

Mi smo na ETF@. Nismo na ETH.

```
\bigskip
```

Ono je Prof. Davić.

Ono je Prof.\ Davić.

```
\end{document}
```

primer h, jednačine, stilovi i subscript

```
\begin{document}
```

Numerisana jednačina:

```
\begin{equation}
    i_{123} = i_1 + i_{2} + i_3
\end{equation}
```

```
\[ i_{123} = i_1 + i_{2} + i_3 \]
```

a može biti i u tekstu \$i_{123} = i_1 + i_{2} + i_3\$

može i ovako \$\$ i_{123} = i_1 + i_{2} + i_3 \$\$
mada nije po standardu!

```
\end{document}
```

primer h, jednačine, superscript i grčka slova

Leonard Ojler je mnogo voleo jednačinu

```
\[ e^{i \pi} + 1 = 0 \]
```

```
\noindent
```

Grčka slova su: $\alpha, \beta, \gamma, \delta,$
 $\ldots \omega.$

Velika grčka slova su: $\Gamma, \Delta, \ldots \Omega.$

Svaki pristojan quick reference ima tabelu.

Većina editora, takođe.

primer h, označavanje objekata

```
Neka je  
\begin{equation}\label{prva}  
    e^{i \pi} + 1 = 0  
\end{equation}  
Mi bi to pisali  
\begin{equation}\label{druga}  
    e^{i \pi} + 1 = 0  
\end{equation}
```

Sada se treba pozivati na jednačinu (`\ref{prva}`) i jednačinu (`\ref{druga}`).\\%

A može i ovako: `\eqref{prva}` i `\eqref{druga}`!

Dobra praksa je da se u `\label` piše i tip objekta, da se ne pojave slika, tabela i/ili jednačina sa istim imenom. Primer: `\label{eq:prva}`

Sređivanje referenci zahteva dva prolaza kompjlera, pokrenete ga dva puta za redom.

primer h, korenje i razlomci

Korenje: $\sqrt{9}=3$, $\sqrt[3]{8}=2$.

\bigskip

Razlomci: $\frac{1}{2}$, $\displaystyle \frac{1}{2}$,
 $\frac{a}{b}$, $\frac{a+c}{b}$, može i $1/2$, a može i

\[\frac{1}{2} \]

\[\frac{a+b}{c} \]

Pogledajte package amsmath: `\dfrac{}{}` i `\tfrac{}{}`.

primer h, integrali, razmaci i sume

Gustina verovatnoće: $\int_{-\infty}^{\infty} p(x) dx = 1$.
Ili $[\int_{-\infty}^{\infty} p(x) dx = 1]$.

Razmaci:

\$ab\$

\$a \ , \ b\$

\$a \ : \ b\$

\$a \ ; \ b\$

\$a \ ! \ b\$

Jedno sumiranje: $\sum_{k=1}^n k = \frac{n(n+1)}{2}$. Ili
 $[\sum_{k=1}^n k = \frac{n(n+1)}{2}]$

primer h, o zagradama

\noindent

Malo o zagradama:

\[(\frac{a}{b}) \]

\[\left(\frac{a}{b} \right) \]

\[\big(\frac{a}{b} \big) \]

\[\Big(\frac{a}{b} \Big) \]

\[\bigg(\frac{a}{b} \bigg) \]

\[\Bigg(\frac{a}{b} \Bigg) \]

I još malo:

\[|x + j y| = \sqrt{x^2+y^2} \]

I još malo:

\[k \in \{1, 2, \dots n\} \]

primer h, funkcije, operatori

Funkcije se pišu u plain fontu (ima i priča oko razmaka), dosta ih je uneto i počinju sa `\`. Neke nisu (npr. `sgn`)! Mogu se dodefinisati, package amsmath, `\operatorname{operatorname}`.

```
\[ \sin(x+y)=\sin(x) \, , \cos(y)+\cos(x) \, , \sin(y) \]
\[ \sin^2 x + \cos^2 x = 1 \]
\[ \cosh^2 x - \sinh^2 x = 1 \]
\[ x \, , \operatorname{sgn}(x) = |x| \]
\[ \operatorname{h}(x)=1, \quad \text{if } x > 0 \]
\[ \operatorname{h}(x)=0, \quad \text{if } x < 0 \]
```

primer h, jednačine sa matricama

Matrice u jednačinama:

```
\[ \operatorname{h}(x) = \left\{ \begin{array}{lll} 0, & \text{if } x < 0 \\ \frac{1}{2}, & \text{if } x=0 \\ 1, & \text{if } x>0 \end{array} \right.
```

primer h, jedinice mere, plain font, amsmath, text

```
\usepackage[greek, serbian]{babel}
```

\noindent Jedinice mere se pišu u plain fontu, \\ razmaknute od „brojne vrednosti“.

```
\[ E = 10 \mbox{ V} \]
```

\[E=10 \text{ V} \] % daje potpuno isti output
ispravno je:

```
\[ E = 10 \, , \text{V} \]
```

```
\[ C = 100 \, , \mbox{\textgreek{m}}\text{F} \]
```

```
\[ t_0 = 1 \, , \text{\textgreek{s}} \]
```

%

Ovde ima problema sa grčkim slovima, mnogi rade:

%

```
\[ C=100 \, , \mu\text{F} \] % namerno je oko = ovako
```

```
\[ t_0= 1 \, , \mu\text{s} \] % namerno je oko = ovako
```

primer h, jedinice mere, upgreek

```
\usepackage{upgreek}
```

```
\[ C = 100 \, , \, \mu\text{F} \]
\[ t_0=1 \, , \, \mu\text{s} \]
```

```
\[ C=100 \, , \, \mbox{\textgreek{m}}\text{F} \]
\[ t_0 = 1 \, , \, \text{\textgreek{m}}\text{s} \]
```

```
\[ C = 100 \, , \, \mathrm{\mu F} \]
\[ t_0 = 1 \, , \, \mathrm{\mu s} \]
```

primer h, matrice

Matrice:

```
\[ \det  
\begin{bmatrix}  
a & b \\  
c & d \\  
\end{bmatrix}  
= a\,d - b\,c \]
```

Pogledajte još i okruženja: `matrix` i `pmatrix`.

primer h, amssymb package

- ▶ ima simbola koje je definisao AMS ...
- ▶ to je American Mathematical Society, nije Auto-Moto Savez
- ▶ usepackage{amssymb} i \usepackage{amsmath}
- ▶ na CTAN, <https://www.ctan.org>
- ▶ $v_X \triangleq \frac{dx}{dt}$ daje $v_X \triangleq \frac{dx}{dt}$
- ▶ \circlearrowright i \circlearrowleft su \circlearrowright i \circlearrowleft
- ▶ a ima i \leadsto što je \rightsquigarrow
- ▶ kao i \looparrowleft što je \leftarrowtail ...
- ▶ gde je levo, tu je i desno \looparrowright što je \looparrowright
- ▶ i još puno simbola u amssymb ...
- ▶ pogledati [\LaTeX symbols](#)

LATEX

— III deo —
+ XCircuit
+ some scripting

© Predrag Pejović, 

priprema slika

- ▶ vektorska i rasterska grafika, Wikipedia
- ▶ za crteže prednost ima vektorska grafika
- ▶ formati od interesa dvi, ps, eps (za latex) i pdf (za pdflatex)
- ▶ od značaja još i png i jpg, mogu da se importuju u pdflatex
- ▶ konverzija programima dvips, ps2eps, ps2pdf, pstoppdf i epstopdf, a ima i još
- ▶ nekih od ovih programa možda nema instaliranih na sistemu, instalirati
- ▶ od značaja:
 - ▶ **epstopdf**, konverzija eps slika u pdf
 - ▶ **pdfcrop**, obrezivanje pdf slika
 - ▶ **pdffonts**, prikazivanje fontova koji se koriste u dokumentu i njihovog statusa (embedded ili ne)
 - ▶ **convert**, konverzija formata slike
- ▶ pogledajte man za gornje programe

još malo o grafičkim formatima . . .

- ▶ https://en.wikipedia.org/wiki/Comparison_of_graphics_file_formats
- ▶ https://en.wikipedia.org/wiki/Device_independent_file_format
- ▶ <https://en.wikipedia.org/wiki/PostScript>
- ▶ https://en.wikipedia.org/wiki/Encapsulated_PostScript
- ▶ https://en.wikipedia.org/wiki/Portable_Document_Format
- ▶ <https://en.wikipedia.org/wiki/JPEG>
- ▶ https://en.wikipedia.org/wiki/Portable_Network_Graphics
- ▶ <https://en.wikipedia.org/wiki/GIF>
- ▶ https://en.wikipedia.org/wiki/Scalable_Vector_Graphics
- ▶ sa gubicima (jpg), manji file, jača kompresija!
- ▶ bez gubitaka (png, gif)
- ▶ proprietary versus free

sin in gnuplot, po ko zna koji put, ako već nemate sliku . . .

- ▶ komandna linija
- ▶ `gnuplot`
- ▶ `set terminal pdfcairo`
- ▶ `set output "slika.pdf"`
- ▶ `plot sin(x)`
- ▶ `set output`
- ▶ `set terminal epscairo`
- ▶ `set output "sin.eps"`
- ▶ `replot`
- ▶ `set output`
- ▶ `Ctrl/d`

slike i konverzije

- ▶ napravili `sin.eps`
- ▶ probajte `ps2pdf sin.eps`
- ▶ pogledate sa `evince sin.pdf`
- ▶ baš ružno!
- ▶ da probamo `pdftocrop sin.pdf`
- ▶ `evince sin-crop.pdf`
- ▶ nešto je urađeno, ali ne valja, nema ticks
- ▶ `rm sin.pdf`
- ▶ `epstopdf sin.eps`
- ▶ `evince sin.pdf`
- ▶ to je to
- ▶ pokušaji nisu bili besmisleni, videćete smisao kod šema

primer, unošenje slika, pdflatex

```
\documentclass[12pt,a4paper]{article}

\usepackage{graphicx} % package za unosenje slika
\usepackage[utf8]{inputenc}
\usepackage[T1,T2A]{fontenc}
\usepackage[serbian]{babel}

\begin{document}

Ovde ćemo ubaciti jednu sliku, sliku \ref{slika:sinus}.

\begin{figure}[h!]
    \centering
    \includegraphics{sin.pdf}

    \caption{Funkcija  $\sin \left( x \right)$ }
    \label{slika:sinus}
\end{figure}

\end{document}
```

komande za procesiranje i formati slika

- ▶ F6, \Leftrightarrow pdflatex
- ▶ F7, display pdf
- ▶ prvi prolaz, ??, još nije pohvatao refs
- ▶ drugi prolaz, sada je sve ok, slika ispravno numerisana
- ▶ F2, tex \rightarrow dvi, \Leftrightarrow latex
- ▶ F4, dvi \rightarrow ps, \Leftrightarrow dvips
- ▶ F8, ps \rightarrow pdf, \Leftrightarrow ps2pdf
- ▶ F7, display pdf
- ▶ latex "handle" samo ps i eps slike, konvertorvati
- ▶ pdflatex "handle" pdf, png, jpg i gif, ne ps/eps
- ▶ probati

primer, unošenje slika, latex

```
\documentclass[12pt,a4paper]{article}

\usepackage{graphicx} % package za unosenje slika
\usepackage[utf8]{inputenc}
\usepackage[T1,T2A]{fontenc}
\usepackage[serbian]{babel}

\begin{document}

Ovde ćemo ubaciti jednu sliku, sliku \ref{slika:sinus}.

\begin{figure}[h!]
    \centering
    \includegraphics{sin.eps}

    \caption{Funkcija  $\sin \left( x \right)$ }
    \label{slika:sinus}
\end{figure}

\end{document}
```

primer, unošenje slika, scaling

```
\documentclass[12pt,a4paper]{article}

\usepackage{graphicx} % package za unosenje slika
\usepackage[utf8]{inputenc}
\usepackage[T1,T2A]{fontenc}
\usepackage[serbian]{babel}

\begin{document}

Ovde ćemo ubaciti jednu sliku, sliku \ref{slika:sinus}.

\begin{figure}[h!]
    \centering
    \includegraphics[scale=1.5]{sin.pdf}

    \caption{Funkcija  $\sin(x)$ }
    \label{slika:sinus}
\end{figure}

\end{document}
```

primer, unošenje slika, rotate

```
\documentclass[12pt,a4paper]{article}

\usepackage{graphicx} % package za unosenje slika
\usepackage[utf8]{inputenc}
\usepackage[T1,T2A]{fontenc}
\usepackage[serbian]{babel}

\begin{document}

Ovde ćemo ubaciti jednu sliku, sliku \ref{slika:sinus}.

\begin{figure}[h!]
    \centering
    \includegraphics[angle=90, scale=1.5]{sin.pdf}

    \caption{Funkcija  $\sin(x)$ }
    \label{slika:sinus}
\end{figure}

\end{document}
```

primer, unošenje slika, position

```
\documentclass[12pt,a4paper]{article}

\usepackage{graphicx} % package za unosenje slika
\usepackage[utf8]{inputenc}
\usepackage[T1,T2A]{fontenc}
\usepackage[serbian]{babel}

\begin{document}

Ovde ćemo ubaciti jednu sliku, sliku \ref{slika:sinus}.

\begin{figure} % [h!] % [h] % [t] %[t!] % [b!] % [b]
    \centering
    \includegraphics[scale=1.5]{sin.pdf}

    \caption{Funkcija  $\sin \left( x \right)$ }
    \label{slika:sinus}
\end{figure}

\end{document}
```

primer, unošenje tabela

Ovde smo ubacili tabelu `\ref{tabela:ime}`.

```
\begin{table}[h!]
    \centering
    \caption{Primer jedne tabele} % ispod slike, iznad tabele

    \begin{tabular}{|c|cc|}
        \hline
        a & b & c \\
        \hline \hline
        d & e & f \\
        \hline
    \end{tabular}
    \label{tabela:ime}
\end{table}
```

primer, unošenje tabela, centriranje

Ovde smo ubacili tabelu \ref{tabela:ime}.

```
\begin{table}[h!]
    \centering
    \caption{Primer jedne tabele} % ispod slike, iznad tabele

    \begin{tabular}{r|c|l}
        \hline
        a & b & c \\
        \hline
        dddd & eeee & ffff \\
        \hline
    \end{tabular}
    \label{tabela:ime}
\end{table}
```

primer, unošenje tabela, phantoms

Ovde smo ubacili tabelu \ref{tabela:ime}.

```
\begin{table}[h!]
    \centering
    \caption{Primer jedne prazne tabele}
    % caption ide ispod slike, iznad tabele

    \begin{tabular}{|l|l|l|}
        \hline
        \phantom{XXXXX} & \phantom{XXXXX} & \phantom{XXXXX} \\
        \hline
        & \vphantom{\Huge{X}} & \\
        \hline
    \end{tabular}
    \label{tabela:ime}
\end{table}
```

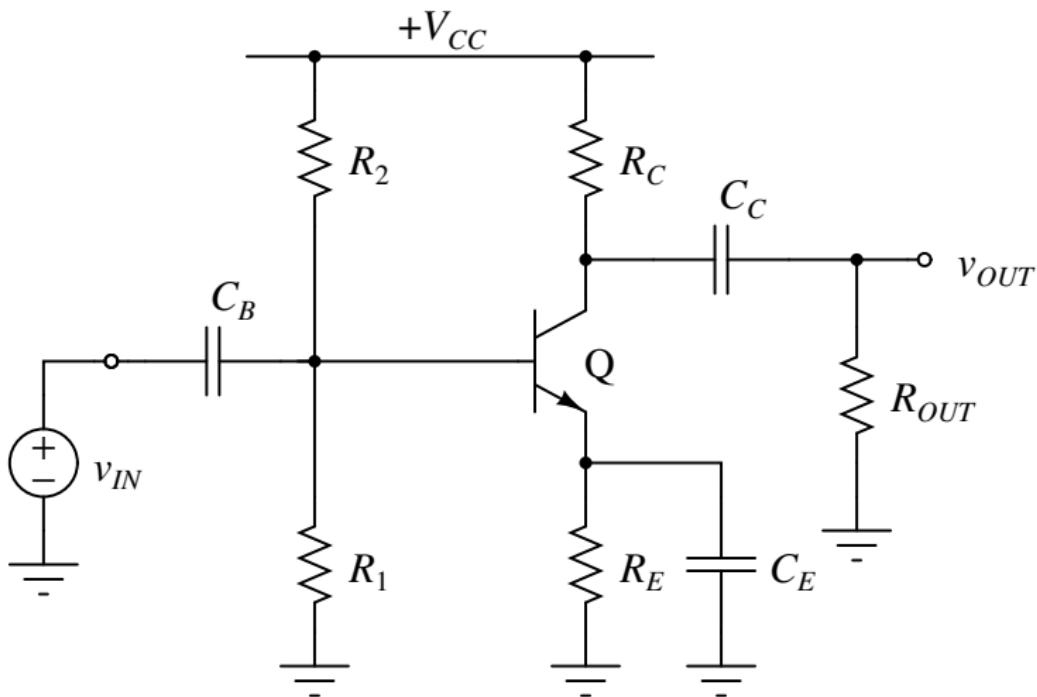
table tricks

- ▶ još mnogo toga se može uraditi sa tabelama
- ▶ korisno: <https://www.cl.uni-heidelberg.de/courses/ss19/wissenschreib/material/tableTricks.pdf>
- ▶ nije loše imati gornji pdf kod sebe, 16 strana samo
- ▶ pitanja stila, obratite pažnju, kod nas sve tabele kao rešetke, pogledajte šta drugi rade, često samo tanke horizontalne linije

X Circuit

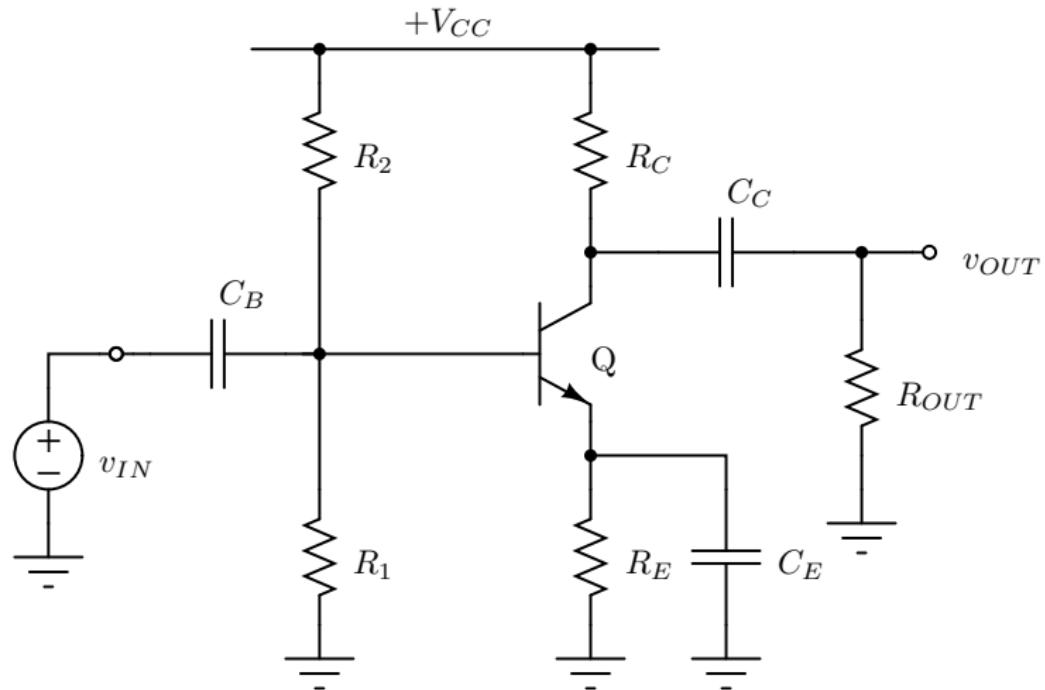
- ▶ program za crtanje električnih šema
- ▶ dominantan gde je od značaja izgled
- ▶ može da crta i lakše tehničke crteže
- ▶ malo neobičan interface
- ▶ malo neobična instalacija, synaptic potreban
- ▶ pravi PostScript files i sve radi na tom nivou
- ▶ konvrsija u pdf sa **epstopdf**,
fajl koji pravi X Circuit je .eps iako je ekstenzija .ps
- ▶ postoji i **ps2pdf**, razlike ...
- ▶ moguće je uneti i **LATEX** lettering
- ▶ pokreće se iz komandne linije sa **xcircuit**
- ▶ manual: <http://opencircuitdesign.com/xcircuit/>
- ▶ mora da se pokaže, da probate, nezgodno za opis rečima, ...
- ▶ **radimo primer zajedno, posle vi sami, ja pomažem!**

zadatak: nacrtati i importovati u pdflatex



ime fajla: **ce1.ps**, convert to **ce1.pdf**, **epstopdf ce1.ps**

LATEX lettering, nacrtati i importovati u pdflatex



container file, ce2.tex

```
\documentclass[a4paper,12pt]{article}

\usepackage{graphicx}

\usepackage[margin=1cm]{geometry}

\pagestyle{empty}

\begin{document}

\input{ce2-raw.tex}

\end{document}
```

procesiranje, varijanta 1

- ▶ `latex ce2`
- ▶ `dvips ce2`
- ▶ `ps2eps -l -f ce2.ps`
`-l` je za loose format, 1 pt margine,
`-f` je za force za overwrite
- ▶ ovde stajete ako koristite latex;
za pdflatex: `epstopdf ce2.eps`
- ▶ `evince ce2.pdf` ili `evince ce2.eps`

procesiranje, varijanta 2

- ▶ `latex ce2`
- ▶ `dvips ce2`
- ▶ do sada je isto kao pre, imamo ps file na celoj strani;
sada pocinju razlike zato da napravimo copy:
`cp ce2.ps ce2-alt.ps`
- ▶ `ps2pdf ce2-alt.ps`, pogledajte pdf
- ▶ `pdftcrop ce2-alt`
- ▶ `evince ce2-alt-crop.pdf`

procesiranje, script 1

napravite `mf.sh`

```
# make figure  
latex $1  
dvips $1  
ps2eps -l -f $1.ps  
epstopdf $1.eps
```

`sh mf.sh ce2`

generic container file, genfig.tex

```
\documentclass[a4paper,12pt]{article}

\usepackage{graphicx}

\usepackage[margin=1cm]{geometry}

\pagestyle{empty}

\begin{document}

\input{\jobname-raw.tex}

\end{document}
```

procesiranje, script 2

napravite `genmf.sh`

```
# make figure, generic
cp genfig.tex $1.tex
latex $1
dvips $1
ps2eps -l -f $1.ps
epstopdf $1.eps
```

```
sh genmf.sh ce2
sh genmf.sh proba
```

procesiranje, script 3

napravite `genmf1.sh`, druga linija nema prekid, ima samo dve linije!

```
# make figure, generic  
cp genfig.tex $1.tex && latex $1 && dvips $1 &&  
ps2eps -l -f $1.ps && epstopdf $1.eps  
  
sh genmf1.sh ce2  
sh genmf1.sh proba
```

procesiranje, script 4

napravite **gmf.sh**, preglednost i estetika, primena \

```
# make figure, generic
cp genfig.tex $1.tex \
&& latex $1 \
&& dvips $1 \
&& ps2eps -l -f $1.ps \
&& epstopdf $1.eps
```

sh gmf.sh ce2

sh gmf.sh proba

procesiranje, script 5

napravite `gmfc.sh`, čišćenje za sobom

```
# make figure, cleanup
cp genfig.tex $1.tex \
&& latex $1 \
&& dvips $1 \
&& rm $1.tex \
&& rm $1.dvi \
&& rm $1.aux \
&& rm $1.log \
&& ps2eps -l -f $1.ps \
&& rm $1.ps \
&& epstopdf $1.eps \
&& rm $1.eps
```

`sh gmfc.sh ce2`

`sh gmfc.sh proba`

LATEX
— IV deo —
Referenciranje i Beamer

© Predrag Pejović, 

Referenciranje?

- ▶ pozivanje na prethodne publikacije, **važno!**
- ▶ praktično standardizovan format
- ▶ najčešće se lista referenci uređuje po redosledu pozivanja
- ▶ poziv na referencu je broj u uglastoj zagradi, npr. [1]
- ▶ u posebnom odeljku (section) „Literatura“ (“References”) mora biti navedeno šta je [1]
- ▶ potrebno je navesti **sve** podatke kako bi citirana referenca mogla biti jednoznačno određena
- ▶ često časopisi specificiraju strog format reference
- ▶ poseban format za knjige, radove u časopisima i radove na konferencijama

Referenciranje u L^AT_EX-u

- ▶ L^AT_EX ima sistem za automatsko numerisanje referenci, **izuzetno** važno u praksi, omogućava lage izmene dokumenta
- ▶ baza referenci važna stvar, L^AT_EX ima nekoliko načina za unos referenci; ovde samo najjednostavniji
- ▶ za pravljenje baze referenci pogledajte package BibTeX, jako korisno ako pišete neki veći rad

standardna preambula

```
\documentclass[a4paper,12pt]{article}

\usepackage[T1, T2A]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage[serbian]{babel}

\usepackage{graphicx}
\usepackage{amsmath}
\usepackage{xcolor}

\usepackage[margin=2cm]{geometry}

\title{Referenciranje}
\author{Predrag Pejović}
\date{\today}
```

tekst

```
\begin{document}
```

```
\maketitle
```

```
\section{Uvod}
```

Ovde ću citirati \cite{ND}.

```
\section{Zaključak}
```

Ovde citiram referencu reda radi, \cite{rr}.

reference

```
\begin{thebibliography}{[X]}

\bibitem{ND}
M.~Depenbrock, C.~Niermann,
“A new 12-pulse rectifier with
line-side interphase transformer
and nearly sinusoidal line currents,”
\emph{Proceedings of the Power Electronics
and Motion Control Conference},
Budapest, 1990, pp.~374--378.

\bibitem{rr}
Ovo je referenca reda radi.

\end{thebibliography}

\end{document}
```

detalji, dashes

- ▶ <http://en.wikipedia.org/wiki/Dash>
- ▶ “hyphen”, -, push-pull
- ▶ “en-dash”, --, pp. 17–45
- ▶ “em-dash”, ---, IEEE Transactions on Circuits and Systems—Fundamental Theory and Applications
- ▶ “minus”, \$-\$, $3 - 2 = 1$
- ▶ da lih uporedimo: -, –, —, —
- ▶ vidite li razlike?

A sada nečto važno ...

Beamer

beamer, uvod 1

- ▶ još jedna klasa dokumenata za L^AT_EX
- ▶ popularan za pravljenje prezentacija
- ▶ rezultat je pdf file
- ▶ nema ništa od proprietary formata
- ▶ po svemu sudeći pdf će dugo moći da se čita, recycle stare prezentacije, . . .
- ▶ Berkeley, EPFL, . . .
- ▶ na „nemačkom“ je projektor beamer
- ▶ nije jedino rešenje, ali ubedljivo najzastupljenije

beamer, resursi

- ▶ user guide, 247 strana,
<http://www.ctan.org/tex-archive/macros/latex/contrib/beamer/doc/beameruserguide.pdf>
- ▶ primeri (jako važno!) http://www.informatik.uni-freiburg.de/~frank/ENG/latex-course/latex-course-3/latex-course-3_en.html
- ▶ preporučujem: <http://www.tug.org/pracjourn/2005-4/mertz/mertz.pdf>,
17 strana
- ▶ Q: Kako to obično funkcioniše?
- ▶ A: Primer koji vam se dopada dovučete i modifikujete.

beamer, ciljevi

- ▶ osnovna prezentacija, frame (128mm x 96mm)
- ▶ dinamika prezentacije (pause, onslide)
- ▶ liste (itemized i enumerated)
- ▶ jednačine
- ▶ slike
- ▶ boje
- ▶ dve kolone
- ▶ handouts
- ▶ „teme“
- ▶ nema previše alata za sakrivanje nedostatka sadržaja u prezentaciji

beamer, header

- ▶ ne treba graphicx
- ▶ ne treba xcolor, color, ...

napomena

%\end{frame}

značí:

\end{frame}

dok se ne reši bug!

beamer, tipičan header

```
\documentclass{beamer}

% početak: može, ali ne mora
\usetheme{default} % {Copenhagen} {Goettingen}
\setbeamercolor{covered}{transparent} % ovo je važno za dinamiku
\usecolortheme{default} % {seahorse}{rose}{wolverine}{infolines}
% kraj: može, ali ne mora

\usepackage[utf8]{inputenc}
\usepackage[T1, T2A]{fontenc}
\usepackage[serbian]{babel}

\usepackage{datetime}

\title[prezentacija]{\huge prezentacija}
\date{\today}
```

beamer, naslov, prvi slajd, dorađivaćemo

```
\begin{document}  
  
\begin{frame}  
  
    \titlepage  
  
%\end{frame}  
  
\end{document}
```

dinamika običnog teksta

```
\section{dinamika običnog teksta}

\begin{frame}{jednostavna dinamika}

\pause
prva \\
\pause
druga \\
\pause
treća

%\end{frame}
```

složena dinamika

```
\begin{frame}{složena dinamika}

\onslide<2->{prva} \\
\onslide<3-3>{druga} \\
\onslide<2-2>{opet prva, kratko} \\
\onslide<4->{treća} \\

%\end{frame}
```

itemized list

```
\section{itemized list} % nece se odmah videti  
  
\begin{frame}{nabranje}  
  
    \begin{itemize}  
        \item <2-> prva stavka  
        \item <3-> druga stavka  
        \item <4-> treća stavka  
        \item <2-> još jedna prva stavka  
    \end{itemize}  
  
\end{frame}
```

hide-unhide

```
\begin{frame}{hide-unhide}

\begin{itemize}
\item <2-2> prva stavka
\item <3-3> druga stavka
\item <4-4> treća stavka
\end{itemize}

%\end{frame}
```

za lenje

```
\begin{frame}{za lenje}

\begin{itemize}
\item <+-> jedan
\item <+-> dva
\item <+-> tri
\end{itemize}

%\end{frame}
```

numerisane liste

```
\section{enumerated list i jednačine}

\begin{frame}{numerisana lista}

\begin{enumerate}
    \item <2-> prvi
    \item <3-> drugi
    \item <4-> treći
\end{enumerate}

%\end{frame}
```

progresivno boldovanje

```
\begin{frame}{progresivno boldovanje}

\begin{enumerate}
\item \textbf{prvi}
\item \textbf{drugi}
\item \textbf{treći}
\end{enumerate}

%\end{frame}
```

jednačine

```
begin{frame}{malo matematike}
```

Pitagorina teorema glasi

```
\pause
```

```
\bigskip
```

$$c^2 = a^2 + b^2$$

```
\bigskip
```

```
\pause
```

znate li dokaz?

```
%\end{frame}
```

potpuno sakrivanje

```
\begin{frame}{malo matematike, Pitagorin stil}

\setbeamercovered{invisible}
Pitagorina teorema glasi

\pause
\bigskip

$c^2 = a^2 + b^2$


\bigskip
\pause

znate li dokaz?
\setbeamercovered{dynamic} % ili {transparent}

%\end{frame}
```

detour: da napravimo sliku, ipython3

komandna linija, terminal:

```
ipython3 --pylab
dm = 360
deg = linspace(- dm, dm, 2 * dm + 1)
y = cos(radians(deg - 45))
plot(deg, y)
xlim(- 360, 360)
xticks([-360, -270, -180, -90, 0, 90, 180, 270, 360])
ylim(- 1.5, 1.5)
xlabel('x [deg]')
ylabel('y = cos(x - pi/4)')
```

save, **slika.png**

ubacivanje slike

```
\section{mogu slike}

\begin{frame}{slika}

\pause

često je potrebno ubaciti neku sliku, npr. png sliku

\pause

\includegraphics[scale=0.35]{slika.png}

\pause

što je odavno poznata funkcija

%\end{frame}
```

sakrivanje slike

```
\begin{frame}{slika}

\setbeamercovered{invisible}

\onslide<2->{\često je potrebno ubaciti neku sliku}

\onslide<3->{\includegraphics[scale=0.3]{slika.png}}

\onslide<4->{\što je odavno poznata funkcija}

\setbeamercovered{dynamic}

%\end{frame}
```

bojena slova

```
\section{boje}

\begin{frame}{bojena slova}

\begin{Large}
\textcolor{red}{crvena} \\
\pause
\textcolor{blue}{plava} \\
\pause
\textcolor{cyan}{kako se ova zove?} \\
\pause
\textcolor{magenta}{a ova?} \\
\pause
\textcolor{yellow}{žuta} \\
\pause
\textcolor{green}{zelena}
\end{Large}

%\end{frame}
```

bojena podloga

```
\beamertemplatesolidbackgroundcolor{teal!20}

\begin{frame}{\textcolor{teal}{bojena pozadina}}

\begin{Huge}
\textcolor{teal}{sa tamnim slovima}
\end{Huge}

%\end{frame}

\beamertemplatesolidbackgroundcolor{white}
```

u dve kolone

```
\section{dve kolone}

\begin{frame}{dve kolone}

\begin{columns}
\column{5cm}
\onslide <2-> gore levo \\
\onslide <3-> dole levo \\

\column{5cm}
\onslide <4-> gore desno \\
\onslide <5-> dole desno \\
\end{columns}

%\end{frame}
```

handouts

```
\documentclass[handout]{beamer}
```

rešava posao, ali krupno

```
\usepackage{pgfpages}
\pgfpagesuselayout{2 on 1}[a4paper, border shrink=5mm]
\pgfpagesuselayout{4 on 1}[a4paper, landscape, border shrink=5mm]
\pgfpagesuselayout{8 on 1}[a4paper, border shrink=2mm]
```

ima još puno toga, odlični web resursi!

teme

- ▶ „teme“, stilovi, ...
- ▶ `\usefonttheme{serif}`
- ▶ tema default na Goettingen
- ▶ sections sada dobijaju smisao
- ▶ tema na Copenhagen
- ▶ zašto li je ovo popularno?
- ▶ možete se igrati sa colortheme
- ▶ primeri: http://www.informatik.uni-freiburg.de/~frank/ENG/latex-course/latex-course-3/latex-course-3_en.html
- ▶ vodite računa o vremenu!
- ▶ ovo će biti bonus deo (30%) prvog kolokvijuma!

Python, I deo

© Predrag Pejović, 

Python?

- ▶ programski jezik
- ▶ Wikipedia:
 - ▶ “Python is a general-purpose, high-level programming language whose design philosophy emphasizes code readability. Python claims to “[combine] remarkable power with very clear syntax”, and its standard library is large and comprehensive. Its use of indentation for block delimiters is unique among popular programming languages.”
 - ▶ “The reference implementation of Python (CPython) is free and open source software and has a community-based development model, as do all or nearly all of its alternative implementations. CPython is managed by the non-profit Python Software Foundation.”

Python??

- ▶ interpreter, scripting language
- ▶ po tome nalik na BASIC (nekada), Octave, ...
- ▶ nema kompilacije i linkovanja, vrlo brze probe
- ▶ sporije od C-a
- ▶ ali se dobro povezuje sa C-om
- ▶ jako moćne i raznovrsne biblioteke (pySerial, numpy, matplotlib, sympy, ...)
- ▶ jednostavna sintaksa
- ▶ opšta namena
- ▶ free!!!
- ▶ jako dobro podržan, razvija se, rasprostranjen
- ▶ Google, Youtube, ...
- ▶ svaka distribucija GNU/Linux-a ga ima

Python???

- ▶ Guido van Rossum, December 1989
- ▶ masovno se uči kao prvi programski jezik: MIT, CU Boulder,
...
- ▶ radi pod raznovrsnim platformama, sve koje se kod nas sreću obuhvaćene
- ▶ vrlo objektno orijentisan, mada ne mora da se koristi
- ▶ vrlo moćni tipovi podataka
- ▶ lako se prave novi tipovi podataka

Python, kako nabaviti? GNU/Linux

GNU/Linux:

- ▶ već ima interpreter, sigurno
- ▶ provera: komandna linija, `python3` nekad bilo `python`
`Python 3.8.5 (default, Jul 28 2020, 12:59:40)
[GCC 9.3.0] on linux`
`Type "help", "copyright", "credits" or "license" for
more information.`
- ▶ nešto valja dovući iz repository (za Python 3!):
 - ▶ IDLE
 - ▶ IPython
 - ▶ numpy
 - ▶ scipy
 - ▶ matplotlib
 - ▶ pylab (sve prethodno)
 - ▶ python-serial
 - ▶ Sympy
 - ▶ Spyder
 - ▶ ...

Python, kako nabaviti? win

Windows:

- ▶ <http://python.org/>
- ▶ dovucete, instalirate
- ▶ za win je IDLE included
- ▶ ostalo?
 - ▶ ili <https://pypi.org/project/pip/>
 - ▶ ili <https://www.anaconda.com/products/individual>
 - ▶ ipython+numpy+scipy+matplotlib+R+...
 - ▶ pySerial, SourceForge,
<https://pypi.python.org/pypi/pyserial>
 - ▶ Sympy, <http://sympy.org/en/index.html>
 - ▶ Spyder, <https://pypi.python.org/pypi/spyder>
 - ▶ ...

Python, 2 ili 3? 3!

- ▶ rešeno, verzija 2 više „nije podržana“
- ▶ forking, 3 je „nov“ jezik
- ▶ verzija 3 nema backward compatibility
- ▶ nisu prevelike razlike (print, input, range, za početak)
- ▶ problem sa već napisanim programima
- ▶ problem ako se oslanjate na već postojeće programe
- ▶ koristim numpy, matplotlib, ... pylab
- ▶ bilo neočekivano brzo prebačeno
- ▶ verzija 3 je aktuelna, nju predajem
- ▶ **python3**

```
Python 3.8.5 (default, Jul 28 2020, 12:59:40)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
```

Python, dokumentacija

- ▶ <http://www.python.org/doc/>, sve što treba
- ▶ <http://ocw.mit.edu/>, Python kursevi, ima više
- ▶ isto, edX
- ▶ <https://openbookproject.net/thinkcs/python/english3e/>
- ▶ <https://greenteapress.com/wp/think-python-2e/>
- ▶ još mnogo free resursa, realno je samo #1 potrebno
- ▶ izbor ostalog izlistan na sajtu predmeta

Python, dokumentacija, realno

Ako ne učite programiranje, već programski jezik:

- ▶ <http://www.python.org/doc/>
- ▶ A4, pdf, .zip, 15.2 MB
- ▶ Python 3.9.1rc1, December 03, 2020
- ▶ tutorial.pdf, Python Tutorial, 147 strana
- ▶ reference.pdf, The Python Language Reference, 168 strana
- ▶ library.pdf, The Python Library Reference, 2084 strane
- ▶ ↑ ovde je suština uspeha

valja pomenuti i ...

1. **PyCharm**, <https://www.jetbrains.com/pycharm/>
 - ▶ popularno ...
 - ▶ Editions:
 - 1.1 Professional (proprietary!)
 - 1.2 Community ("Open Source")
 - ▶ ne koristim ga, nekompetentan za komentare
2. **Thonny**, <https://thonny.org/>
 - ▶ ima ga Raspbian, RaspberryPi
 - ▶ ne koristim, samo pogledao, ...

Python, počinjemo, kalkulator

Pokrenete IDLE ili ipython, kako god znate (kom. lin., dash, ...)

osnovne operacije:

2+2

2-3

2*3

od verzije 3 nema iznenađenja:

3/4*100

3.0/4.0*100.0

3.0/4*100

3./4*100

Python, mislili ste da je sa deljenjem gotovo?

10//3

10.0//3.0

10.0//3.0

-10.0//3.0

-10//3

Python, type i celobrojno deljenje

```
help(type)
type
type(3)
type(3.0)
type(3.)
type(10//3)
type(10.0//3)
type(10/3.)
type(10./3.)
```

Python, stepenovanje

2^3

3^2

3^3

10^10

2**3

2 ** 3

3 ** 2

10 ** 10

type(10**10)

3**64

type(3**3)

type(3**64)

Python, ostatak pri deljenju

10%3

11%3

12%3

t=54+12

print(t)

type(t)

s=t//60

m=t%60

print(s)

print(m)

print(s, m)

print('proteklo je', s, 'sat i', m, 'minuta')

Python, operatori poređenja

2 == 2

2==2

3 == 2

2 != 3

2 != 2

#2 <> 2

#2 <> 3

2 > 3

2 < 3

2 >= 1

2 >= 2

2 >= 3

2 <= 1

2 <= 2

2 <= 3

Python, logičke operacije, ;, \ i

```
type(True); type(False)
a = True
b = False
type(a)
a and b    # logicko i
not a       # logicko ne
a and a
a or not a
a or (not a)
a or \
not b      # ovako se nastavlja red
```

Python, zapisi brojeva

#012

0o12

0012

0x35

0X35

0b11

0B11

Python, konverzija zapisa brojeva

```
oct(10)
```

```
hex(53)
```

```
bin(3)
```

Python, da raščistimo \wedge , operacije nad bitima

```
a = 0b0101
```

```
a
```

```
b = 0b0011
```

```
b
```

```
a & b
```

```
bin(a & b)
```

```
bin(a | b)
```

```
bin(a ^ b)
```

```
bin(0)
```

```
bin(~0)
```

```
bin(2)
```

```
bin(~2)
```

```
~2
```

```
2 << 1
```

```
2 << 4
```

```
32 >> 2
```

```
3 >> 1
```

Python, a sada nesto sasvim drugačije: kompleksni brojevi

```
#j*j  
1j*1j  
2J * 2J  
type(1J)  
abs(3+4j)  
complex(1,2)  
a = 2 + 3j  
type(a)  
a.real  
a.imag  
a.conjugate()  
a * a.conjugate()  
del a  
#type(a)
```

Python, malo ozbiljnija matematika, moduli

```
sin(1)
import math
type(math)
dir(math)
help(math)
help(math.sin)
math.sin(1)
math.e
math.pi
math.sin(math.pi/2)
#math.exp(math.pi*1j)+1
math.cos(math.pi) + 1j * math.sin(math.pi) + 1
```

Python, namespaces

```
del math
import math as m
m.sin(m.pi / 4) ** 2
m.exp(1) - m.e
del m
from math import *
sin(pi / 4) ** 2
exp(1) - e
e
e = 32
e
pi
pi = 14
pi
```

Python, assignment operators

```
a = 1
a += 1
print(a)
a *= 2
print(a)
a /= 2
print(a)
a -= 4
print(a)
a **= 3
a %= 3
print(a)
-8 / 3
a = 11.0
a //= 3
print(a)
```

Python, funkcije

```
def pdv(x):  
    return x * 1.20
```

```
type(pdv)  
pdv(100)  
pdv(150)
```

Python, funkcije, help

```
def pdv(x):
    'ovo je funkcija koja racuna pdv'
    return x * 1.20

pdv(100)
help(pdv)
```

Python, funkcije, help u više redova

```
def pdv(x):
    '''ovo je funkcija koja racuna pdv
    a pdv je porez na dodatu vrednost'''
    return x * 1.20

pdv(100)
help(pdv)
```

Python, funkcije, opcioni argumenti

```
def pdv(x, stopa = 20):  
    return x*(1 + stopa/100)
```

```
pdv(100)  
pdv(150)
```

```
def pdv(x, stopa = 20):  
    return x * (1 + stopa / 100.)
```

```
pdv(100)  
pdv(150)  
pdv(100, stopa = 23)  
pdv(100, 23)
```

```
del pdv  
pdv(10)
```

Python, kontrola toka

```
def parnost(n):
    if n // 2 * 2 == n:
        print('paran')
    else:
        print('neparan')
```

```
parnost(4)
parnost(5)
parnost(4.2)
parnost(5.1)
```

Python, ispitivanje tipa

```
def parnost(n):
    if type(n) != "<type 'int'>":
        print('argument nije ceo broj')
        return
    if n // 2 * 2 == n:
        print('paran')
    else:
        print('neparan')
```

```
parnost(4.2)
```

```
parnost(4)
```

```
parnost(3)
```

```
type(4)
```

```
type(type(4))
```

```
type("<type 'int'>")
```

Python, ispitivanje tipa, sada radi

```
def parnost(n):
    if str(type(n)) != "<type 'int'>":
        print('argument nije ceo broj')
        return
    if n // 2 * 2 == n:
        print('paran')
    else:
        print('neparan')
```

```
parnost(4.2)
```

```
parnost(4)
```

```
parnost(3)
```

Python, ispitivanje tipa, može i ovako

```
def parnost(n):
    if type(n) != type(1):
        print('argument nije ceo broj')
        return
    if n // 2 * 2 == n:
        print('paran')
    else:
        print('neparan')
```

```
parnost(4.2)
parnost(4)
parnost(3)
parnost(4.)
```

Python, konverzije tipova i još ponešto

```
del parnost
int(-4.2)
int(4.2)
#long(_)
float(_)
float(5)
divmod(10, 3)
divmod(12, 3)
pow(2, 8)
2**8
str(float(2**8))
```

Python, liste

```
a = [1, 2, 5, 6]
type(a)
a[0]
a[1]
a[2]
a[3]
a[4]
a[-1]
a[-2]
a[-3]
a[-4]
a[-5]
print(a)
len(a)
```

Python, liste, slicing and mutability

```
a[1:3]
a[1 : 2]
a[1 : - 2]
a[2 : ]
a[:2]
a[:-2]
a[3] = 7
print(a)
```

Python, liste, dodavanje i brisanje elemenata

```
a + 9  
a + [9]  
a = a + [9]  
len(a)  
del a[(len(a) - 1)]  
print(a)  
del a[1]  
print(a)  
len(a)
```

Python, liste, metodi append i extend

```
a = [1, 2, 3, 4]
a.append(5)
print(a)
b = [6, 7]
a.append(b)
print(a)
len(a)
del a[5]
a.extend(b)
print(a)
len(a)
del a[5:]
print(a)
```

Python, liste, range

```
a = range(5)
len(a)
print(a)
type(a)
a = list(range(5))
len(a)
print(a)
type(a)
a = tuple(range(5))
len(a)
print(a)
type(a)
a = range(4, 10)
len(a)
print(a)
a = range(3, 10, 2)
print(a)
a = range(10, 0, -2)
print(a)
```

Python, stack

```
a = []
type(a)
a.append(1)
a.append(2)
a.append(3)
a.pop()
a.pop()
print(a)
a = list(range(10))
a.pop(3)
print(a)
```

Python, liste, insert

```
a = list(range(10))
a.insert(3, 4)
print(a)
a.insert(0, 1)
print(a)
a.insert(len(a), 'kraj')
print(a)
```

Python, liste, reverse, sort

```
a = list(range(10))
a.reverse()
print(a)
a.reverse()
print(a)
a = [3, 4, 2, 1]
a.sort()
print(a)
```

Python, liste, brojanje i brisanje

```
a = [3, 2, 3, 1, 4, 3, 2, 2, 5, 2]
a.count(2)
a.count(3)
a.remove(3)
a.count(3)
print(a)
a.remove(3)
print(a)
a.remove(3)
print(a)
#a.remove(3)
```

Python, in operator

```
3 in a  
4 in a  
a.remove(4)  
4 in a
```

Python, liste, index metod

```
print(a)
a.index(2)
a.index(5)
a.index(1)
a.index(3)
```

Python, aliases

```
a = 3
b = a
a is b
a == b
id(a)
id(b)
help(id)
b += 1
a == b
a is b
id(a)
id(b)
```

Python, aliases with lists

```
a = [1, 2, 3]
b = a
a is b
a == b
b[1] = 0
a == b
print(a)
a is b
c = a[:]
c == a
c is a
c[1] = 2
c == a
print(c)
print(a)
```

Python, matrice

```
a = [[1, 2], [3, 4]]  
len(a)  
len(a[1])  
print(a[1][1])  
print(a[0][0])  
a[0, 0]
```

Python, inicijalizacija nizova

```
a = []
print(a)
a = [0] * 10
print(a)
a = [[1] * 3] * 3
print(a)
```

Python, for petlja

```
a = range(10)
for i in a:
    print(i + 1, '/', len(a))
```

ili

```
a = list(range(10))
for i in a:
    print(i + 1, '/', len(a))
```

Python, for petlja, over string, “iterable”

```
a = 'neobicno bas'  
for znak in a:  
    print(znak)
```

Python, if-else

```
a = 'abradabra'  
b = ''  
for znak in a:  
    if znak != 'a':  
        b += znak  
    else:  
        b += '_'  
print(b)
```

Python, if-elif-else

```
a = 'abradabra'  
b = ''  
for znak in a:  
    if znak == 'a':  
        b += '_'  
    elif znak == 'k':  
        b += '*'  
    else:  
        b += znak  
print(b)
```

Python, II deo

© Predrag Pejović, 

jedna knjiga . . .

**How to Think Like a Computer Scientist
Learning with Python**

Allen Downey, Jeffrey Elkner, Chris Meyers

Green Tea Press, Wellesley, Massachusetts

... i jedan citat:

strana 68(90):

... One of the characteristics of algorithms is that they do not require any intelligence to carry out. They are mechanical processes in which each step follows from the last according to a simple set of rules.

In our opinion, it is embarrassing that humans spend so much time in school learning to execute algorithms that, quite literally, require no intelligence.

On the other hand, the process of designing algorithms is interesting, intellectually challenging, and a central part of what we call programming.

Python, dosta kalkulatora: programiranje!

- ▶ pokrenete IDLE
- ▶ File, New File (Ctrl + N)
- ▶ kucate program
- ▶ sintaksna provera: Run, Check Module (Alt + X)
- ▶ start programa: Run Module (F5)
- ▶ u IDLE je run `exec(open('filename.p').read())`
- ▶ u ipython3 radi i `run filename.p`

Python, for petlja, continue

```
a = []
for i in range(100):
    if i % 4 != 0:
        a.append(i)
        continue
    print('izbacen', i)
print
print(a)
```

Python, for petlja, da dodamo i break

```
a = []
for i in range(100):
    if i % 4 != 0:
        a.append(i)
        continue
    print('izbacen', i)
    if (i+1) % 77 == 0:
        break
print
print(a)
```

Python 3, input

```
x = input('neki podatak: ')
print(x, type(x))

while(True):
    x = input('neki podatak: ')
    print(x, type(x))

while(True):
    x = int(input('neki ceo broj: '))
    print(x, type(x))

while(True):
    x = input('neki ceo broj: ')
    try:
        y = int(x)
        print(y, type(y))
    except:
        print(x, 'pogrešan podatak')
```

Python 2, input

```
x = input('unesi neki podatak: ')
print type(x)
print x
```

```
probati: 1, 3.0, 'podatak'
```

Python 2, raw_input

```
x = raw_input('unesi neki podatak: ')
print type(x)
print x
```

```
probati: 1, 3.0, 'podatak'
```

Python, formatted print

calculator mode, again

```
print('{}')
print('{}'.format('Pera'))
print('{}, {}'.format('Mika', 'Laza'))
print('{0}, {1}'.format('Mika', 'Laza'))
print('{1}, {0}'.format('Mika', 'Laza'))
print('{1}, {1}'.format('Mika', 'Laza'))
```

Python, formatted print, brojevi

```
print('p = {}, q = {}'.format(3, 4))
print('p = {}, q = {}'.format(3.0, 4))
print('p = {1}, q = {0}'.format(3, 4))
import math
print('pi = {:.2f}'.format(math.pi))
print('pi = {:.5f}'.format(math.pi))
print('pi = {:10.5f}'.format(math.pi))
print('n = {:10d}'.format(100))
print('n = {:4d}'.format(100))
print('n = {:3d}'.format(100))
print('n = {:1d}'.format(100))
```

ima još mogućnosti, pogledajte Tutorial
postoji i formatiranje sa %, staro
koristiti str.format() metod za nove programe!

Python, stringovi, još ponešto

```
a = 'Petar'  
b = 'Marko'  
len(a); len(b)  
a + b  
a * 3  
3 * a  
(a + 2 * b) * 3  
a - b  
a == a  
a == b  
a < b  
b < a  
'1' < '2'  
'1' < 2
```

Python, nije sve u matematici (?)

```
import string as s
dir(s)
help(s)
s.ascii_letters
s.printable
s.uppercase
s.digits
s.octdigits
```

Python, string methods

del s

```
'Pera'.upper()  
'Pera'.lower()  
'Pera'.center(20)  
'Pera'.isalpha()  
'Pera6'.isalpha()
```

```
'kako da ne'.capitalize()  
'kako da ne'.title()  
'kako da ne'.split()
```

i tako dalje, manual for string methods, pogledajte pre nego što
reinvent

Python, OŠ, 1

Vreme je za zadatak:

- ▶ ulazni podatak je broj, $n < 20$ (npr.)
- ▶ treba odštampati LEPU (formatiranu) tablicu množenja

Polako, sada imamo preča posla ...

Python, files, pisanje

```
f = open('proba.txt', 'w')
print(f)
type(f)
f
f.write('prvi red\n')
f.write('drugi red\n')
f.write('treci red\n')
f.close # metod, obavezne()
f.close()
type(f)
f
```

pogledajte fajl proba.txt

Python, files, čitanje

```
f = open('proba.txt', 'r')
print(f)
f
type(f)
a = f.read()
a
print(a)
f.close()
f
```

Python, files, čitanje po 10 bajtova

```
f = open('proba.txt', 'r')
f.read(10)
f.read(10)
f.read(10)
f.read(10)
f.close()
```

Python, files, čitanje po redovima

```
f = open('proba.txt', 'r')
f.readline()
f.readline()
f.readline()
f.readline()
f.close()
```

Python, OŠ, 2

Vreme je za zadatak:

- ▶ ulazni podatak je broj, $n < 20$ (npr.)
- ▶ treba formirati LEPU (formatiranu) tablicu množenja
- ▶ i zapisati je u .txt file

Polako, još uvek imamo preča posla ...

Python, OOP

OOP, Python's middle name

- ▶ korisnik definiše svoje tipove podataka, klase
- ▶ nad definisanim podacima definiše funkcije, tzv. metode, već ih videli na delu
- ▶ čisto i pregledno
- ▶ može, a ne mora
- ▶ šteta propustiti

Python, OOP, konvencije za dalje ...

- ▶ pokrenete IDLE, tu je **interpreter**
- ▶ otvorite **editor**: File, New Window, ...
- ▶ ono što kucate u interpreteru je **crveno**
- ▶ ono što kucate u editoru je **plavo**
- ▶ kucate "as is", kreativnost za sada nije na ceni
- ▶ file neka se zove `vector.py`

Python, class

kucate u editoru, kada završite proverite sintaksu i izvršite,
to su Alt+X i F5

```
class Vector:  
    pass
```

Python, class

```
type(Vector)
a = Vector()
type(a)
print(a)
a.x = 1
a.y = 2.0
a.z = 3j
type(a.x)
type(a.y)
type(a.z)
print(a.x, a.y, a.z)
del a
type(a.x)
```

Python, class help

```
help(Vector)
```

```
class Vector:  
    """Vector(x, y, z)
```

```
    klasa koja definise vektore  
    i operacije nad njima"""
```

note: nema više **pass!**

```
help(Vector)
```

Python, inicijalizacija, dodajemo na class, help string ostaje

```
class Vector:  
    . . .  
    def __init__(self, x, y, z):  
        self.x = x  
        self.y = y  
        self.z = z  
  
i = Vector(1, 0, 0)
```

Python, inicijalizacija, dodajemo na class, help string ostaje

inicijalizacija i se dodaje u class file, vector.py pazite na indent,
inicijalizacija i class isto!

4 argumenta?

```
type(i)
print(i)
print(i.x, i.y, i.z)
```

```
x = Vector()
```

Python, inicijalizacija, default

```
class Vector:  
    . . .  
    def __init__(self, x = 0, y = 0, z = 0):  
        self.x = x  
        self.y = y  
        self.z = z  
  
i = Vector(1, 0, 0)  
j = Vector(0, 1, 0)  
k = Vector(0, 0, 1)  
  
x = Vector()  
print(x.x, x.y, x.z)  
print(i.x, i.y, i.z)  
help(Vector)
```

Python, string method, lepo prikazivanje instance

dodajete samo funkciju:

```
class Vector:  
    . . .  
    def __str__(self):  
        return 'Vector(' + str(self.x) + ', ' + \  
               str(self.y) + ', ' + str(self.z) + ')'  
  
help(Vector)  
print(i)  
print(j)  
a = Vector(2, 2, 0)  
print(a)
```

Python, operator overloading, sabiranje vektora

```
class Vector:  
    . . .  
    def __add__(self, other):  
        return Vector(self.x + other.x, self.y + \  
                      other.y, self.z + other.z)  
  
help(Vector)  
print(i + j)  
print(i + j + k)  
  
print(Vector.__add__(i, j))  
  
print(i.__add__(j))
```

Python, operator overloading, oduzimanje vektora

```
class Vector:  
    . . .  
    def __sub__(self, other):  
        return Vector(self.x - other.x, \  
                      self.y - other.y, self.z - other.z)  
  
help(Vector)  
print(i - j)  
print(i - j - k)  
  
print(Vector.__sub__(k, j))  
print(k.__sub__(j))
```

Python, da zakomplikujemo malo . . .

u interpreteru:

```
def sv(a, b):  
    return a + b
```

```
print(sv(i, j))  
print(sv(3, 4))  
sv  
type(sv)
```

```
del sv
```

```
sv = Vector.__add__  
sv  
type(sv)
```

```
print(sv(i, j))  
print(sv(3, 4))
```

Python, opeet operator overloading, negativni vektor

```
class Vector:  
    . . .  
    def __neg__(self):  
        return Vector(-self.x, -self.y, -self.z)  
  
help(Vector)  
print(-i)  
print(-(-i))  
  
print(Vector.__neg__(k))  
print(k.__neg__())
```

Python, štampanje kao metod

```
class Vector:  
    . . .  
    def pr(self):  
        print(self)  
  
help(Vector)  
help(Vector.pr)  
i.pr  
i.pr()  
Vector.pr(i)  
k.__neg__.pr()
```

Python, absolutna vrednost

```
class Vector:  
    . . .  
    def absval(self):  
        return math.sqrt(self.x**2 + \  
                         self.y**2 + self.z**2)  
  
help(Vector)  
help(Vector.absval)  
i.absval  
i.absval()  
import math  
i.absval()  
Vector.absval(i)  
a = Vector(2, 2, 0)  
a.absval(); Vector.absval(a); a.absval()**2  
a.x
```

Python, vektorski proizvod

```
class Vector:  
    . . .  
    def vecprod(self, other):  
        x = self.y * other.z - self.z * other.y  
        y = self.x * other.z - self.z * other.x  
        z = self.x * other.y - self.y * other.x  
        return Vector(x, y, z)  
  
print(Vector.vecprod(i, j))  
print(Vector.vecprod(j, i))  
print(i.vecprod(j))  
print(j.vecprod(i))  
i.vecprod(j).pr()  
j.vecprod(i).pr()
```

Python, skalarni (dot) proizvod

```
class Vector:  
    . . .  
    def dotprod(a, b):  
        dp = a.x * b.x + a.y * b.y + a.z * b.z  
        return dp  
  
print(Vector.dotprod(i, j))  
print(Vector.dotprod(j, i))  
a = Vector(2, 2, 0)  
print(a.dotprod(j))  
print(a.dotprod(k))  
a.dotprod(k).pr()  
k.dotprod(i).pr() # zašto ne radi?
```

Python, proizvod množenja skalarom

```
class Vector:  
    . . .  
    def scalprod(self, other):  
        return Vector(self.x * other, \  
                      self.y * other, self.z * other)  
  
print(Vector.scalprod(i, 3))  
print(Vector.scalprod(3, i))  
print(k.scalprod(3))  
print(3.scalprod(k))  
j.scalprod(3).pr()  
3.scalprod(i).pr()  
a = 3  
a.scalprod(i).pr()
```

Python, overloaded *

```
class Vector:  
    . . .  
    def __mul__(self, other):  
        return Vector.dotprod(self, other)  
  
print(i * 3)  
print(i * i)  
print(i * j)  
print(i * k)  
print(i.__mul__(i))  
(i * i).pr()  
i * i  
i * j
```

Python, overoverloaded *

```
class Vector:  
    . . .  
    def __rmul__(self, other):  
        return Vector.scalprod(self, other)  
  
print(3 * i)  
print(i * 3)  
print(i * j)  
print(i * i)  
print(i.__rmul__(3))  
i.__rmul__(3).pr()
```

Python, a sada poređenje, 1

```
class Vector:  
    . . .  
    def __cmp__(self, other):  
        a = self.absval()  
        b = other.absval()  
        if a > b:  
            return 1  
        elif a < b:  
            return -1  
        else:  
            return 0
```

Python, a sada poređenje, 2

```
a = Vector(2, 2, 2)
a > i
a == i
a < i
i > i
i == i
i != i
i < i
i == j
i == k
```

I ovo je kraj priče o vektorima, nema više red i blue!

Python, class and object variables

```
class Brojac:  
    broj = 0  
    def __init__(self):  
        Brojac.broj += 1  
        print('sada ih ima', Brojac.broj)  
    def __del__(self):  
        Brojac.broj -= 1  
        print('sada ih ima', Brojac.broj)
```

Python, class and object variables, primena

```
a = Brojac()
b = Brojac()
c = Brojac()
d = Brojac()
del c
a = Brojac()      # pise, pa brise
del a
del d
del b
```

Python, a sada nešto sasvim različito, map

```
x = range(11)
print(x)
def sq(a):
    return a**2

y = map(sq, x)
print(y)
```

Python, a sada nešto neočekivano

```
help([])  
help(())  
help({})  
a = (1, 2, 3, 4, 5)  
a[0]  
a[3]  
a[2] = 5
```

Python, dictionary

```
t = {'peja': 313, 'laza': 312, 'dule': 311}
print(t)
print(t['peja'])
t['peja'] = 314
print(t)
del t['peja']
print(t)
'dule' in t
'peja' in t
t.keys()
t['peja'] = 313
print(t)
```

Python, sets

```
a = 'abracadabra'  
b = 'alacazam'  
x = set(a)  
print(x)  
y = set(b)  
print(y)  
x - y  
y - x  
x | y  
x & y  
x ^ y
```

Python, exceptions, 0

```
def reciproc(x):
    try:
        return 1 / x
    except:
        print('Houston, we have a problem!')

reciproc(3)
reciproc(3.)
reciproc(0)
reciproc(0.)
reciproc(1 + 1j)
reciproc('a ovo?')
```

Python, exceptions, 1

```
def reciproc(x):
    try:
        return 1 / x
    except ZeroDivisionError:
        print('deliš nulom!')
```

```
reciproc(3)
reciproc(3.)
reciproc(0)
reciproc(0.)
reciproc(1 + 1j)
reciproc('a ovo?')
```

Python, exceptions, 2

```
def reciproc(x):
    try:
        return 1 / x
    except ZeroDivisionError:
        print('deliš nulom!')
    except TypeError:
        print('nije dobar broj')

recipro(3)
recipro(3.)
recipro(0)
recipro(0.)
recipro(1 + 1j)
recipro('a ovo?')
```

Python, exceptions, 3

- ▶ “built-in” exceptions?
- ▶ <http://docs.python.org/3/library/exceptions.html>
- ▶ ovo je samo početak ...
- ▶ možete sami da definišete exception ...
- ▶ ili da raise exception ...
- ▶ ako vas zanima

<http://docs.python.org/3/tutorial/errors.html>

Python, start programa iz shell-a

`python3 ime_programa.py`

ili

`./ime_programa.py`

akko je PRVA linija programa:

`#! /usr/bin/python3`

i ako je urađeno

`chmod +x ime_programa.py`

čega se sećate, naravno; da izbegnete `./` dopunite `PATH`

Python, rendering

```
print('\n\n\n')
```

```
print(r'\n\n\n')
```

```
print('\\n\\n\\n')
```

Python, nije kraj, ovo je početak!

- ▶ ima još mnogo toga, ovo je samo osnova
- ▶ pre svega biblioteke, moduli ...
- ▶ mnogo toga je gotovo, vaša je primena!
- ▶ glue language!
- ▶ ostaju numpy, matplotlib, ipython, pylab, sympy
- ▶ a sada: tablica množenja

Python, III deo: PyLab i Jupyter

© Predrag Pejović, 

Šta je PyLab?

- ▶ environment koji cine
 - ▶ NumPy
 - ▶ SciPy
 - ▶ Matplotlib
 - ▶ IPython
- ▶ kako se instalira pylab?
- ▶ Ubuntu: Software Center ili Synaptic
- ▶ win: <https://www.anaconda.com/products/individual>

IDE? Spyder!

- ▶ potrebno?
- ▶ kako kome, kako za šta ...
- ▶ IPython3 meni sasvim dobar ...
- ▶ ... postoji i Spyder
- ▶ <https://pypi.python.org/pypi/spyder>
- ▶ ako nema dovucite iz repository ...
- ▶ ima i pod win, Anaconda

NumPy

- ▶ <http://www.numpy.org/>
- ▶ osnovna biblioteka za numerički zahtevne Python primene, sadrži:
 1. N-dimensional array object
 2. array slicing methods
 3. array reshaping methods
- i module za:
 1. basic linear algebra functions
 2. basic Fourier transforms
 3. advanced random number capabilities
- ▶ pokrenete IDLE
- ▶ `import numpy`
- ▶ `dir(numpy)`
- ▶ `help(numpy)`
- ▶ `del numpy`

NumPy

- ▶ obradićemo, sve, naravno, imamo vremena, ...
- ▶ evolutivno nastao, „haotično“ iz Numeric i NumArray
- ▶ nema načina da se zapamti i nije „zauvek“
- ▶ potreban nov način učenja i snalaženja
- ▶ izbeći reinventing
- ▶ masovno korišćenje help-a i primera
- ▶ Matlab-Python-R

SciPy

- ▶ <http://www.scipy.org/>
- ▶ scientific lib za Python, zavisi od NumPy
- ▶ nekoliko modula in a single package, kao i NumPy
- ▶ moduli za:
 1. statistics
 2. optimization
 3. numerical integration
 4. linear algebra
 5. Fourier transforms
 6. signal processing
 7. image processing
 8. ODE solvers
 9. special functions

SciPy

- ▶ i dalje u IDLE
- ▶ `import scipy`
- ▶ `dir(scipy)`
- ▶ `help(scipy)`
- ▶ `del scipy`
- ▶ dobro razmislite pre nego što krenete u reinvent!
- ▶ ja ovo ne mogu da zapamtim, sto zapamtim zaboravim
- ▶ pomenuh li nov način učenja?
- ▶ kako organizovati informacije, previše ih je (i kratko traju)
- ▶ “Numerical Recipes”

matplotlib

- ▶ <http://matplotlib.sourceforge.net/>, sjajan sajt
- ▶ package sa dugim nizom modula
- ▶ jako dobro se vidi struktuiran namespace, package.modul
- ▶ ono što ima sjajno je, a sada ima skoro sve
- ▶ galerija i primeri
- ▶ uputstvo, 3.2.1, pdf, 2598 strana, 19.03.2020.
- ▶ sintaksa vrlo liči na Matlab i Octave

IPython i Jupyter

- ▶ <http://ipython.org/>
- ▶ <https://jupyter.org/>
- ▶ interaktivni Python environment
- ▶ autocompletion by tab
- ▶ doteruje komande da budu shvaćene
- ▶ od mnogo mogućnosti: store, history, logging, ...
- ▶ Jupyter, isto samo u web browser okruženju!
- ▶ web server, može da se izvršava na nekoj drugoj mašini

PyLab

- ▶ sve prethodno zgodno spakovano
- ▶ sređen namespace da ne mora puno dot notation
- ▶ sintaksa jako liči na Octave/Matlab
- ▶ počinjemo, komandna linija, terminal
- ▶ `ipython3 --pylab`

PyLab, IPython, osnovno

```
help()  
?  
%quickref  
a = 3  
b = 'string'  
lista = [1, 2, 3]  
li<tab>  
whos
```

PyLab, IPython, store

```
store a
store b
store lista
del a
del b
a
b
store -r
a
b
store -z
store
```

Pylab, matematika, konstante

```
pi  
e  
j  
1j  
exp(1j * pi)  
math.exp(1j * pi)  
help(exp)  
help(math.exp)  
exp?  
math.exp?  
e**(1j * pi) + 1
```

Pylab, matrice 1

```
a = array([[1, 2], [3, 4]])  
a  
a.size  
a.shape  
a.ndim  
a.dtype  
a.dtype.name  
a.itemsize  
a.transpose()  
transpose(a)  
a.T
```

Pylab, matrice 2

```
det(a)
eig(a)
b = eig(a)
type(b)
len(b)
type(b[0])
type(b[1])
c, d = b
c.size
c.shape
c.ndim
c.dtype.name
c.itemsize
```

Pylab, matrice 3, inverzija i množenje

```
aa = inv(a)
aa
aa.dtype.name
aa * a
a * aa
dot(a, aa)
dot(aa, a)
x = arange(10)
x
print(x)
dot(x, x)
```

Sve operacije su **elementwise!!!**

Velika razlika u odnosu na **Octave!!!**

Pylab, gde je dot, tu je cross

```
i = array([1, 0, 0])
j = array([0, 1, 0])
print(dot(i, j), dot(j, i))
print(cross(i, j))
print(cross(j, i))
cross?
source(cross)
```

Poznato od nekud? Rekoh da već ima, ...

Teško je naći potpuno nov primer ...

Reinventing problem! Google pomaže puno!

Pylab, rang!!!

```
a = array([[1, 1], [1, 1]])
rank(a)      # ???
help(rank)    # sad jasno?
linalg.matrix_rank(a)
```

Pazite!!!

Nažalost, lično iskustvo, ne tuđe!

Pylab, inicializacija, neke posebne matrice

```
I = eye(3)
print(I)
nule = zeros(4)
print(nule)
zeros?
zeros((4, 2))
zeros(4, 2)      # pazite!
zeros((4, 2), dtype = int)
zeros((4, 2), dtype = complex)
ones([3, 4])
empty([6, 4])
empty?
```

Pylab, još elementwise operacija

```
a + 1  
a * a  
a**3  
a / 2  
a / 2.
```

Pazite na / kod Python 2!

Pylab, reshape

```
help(reshape)
print(a)
a.reshape(1, 4)
a.reshape(4, 1)
reshape(a, (1, 4))
reshape(a, (4, 1))
a.reshape(2, 3)
a.reshape(4)
a.reshape((4, ))
reshape(a, 4)
reshape(a, (4, ))
```

Pylab, in place array operations

```
xx = arange(20)
print(xx)
xx += 1
print(xx)
xx = xx.reshape(4, 5)
print(xx)
xx -= 1
print(xx)
xx *= 2
print(xx)
```

Pylab, transpose, revisited

```
help(transpose)
xx = arange(5) + 1
print(xx)
xx.shape
xx.ndim
xx = xx.transpose()
xx.shape
print(xx)
xx = xx.reshape(1, 5)
xx.ndim
print(xx)
xx.shape
xx = xx.transpose()
xx.shape
print(xx)
```

Pylab, eig, revisited

```
a = eye(2)
print(a)
b = eig(a)
type(b)
len(b)
c, d = b
print(c)
print(d)
a[0, 1] = 1
print(a)
b = eig(a)
c, d = b
print(c)
print(d)
```

Pylab, plot 1

```
x = arange(10) + 0.5  
y = x**2  
plot(x, y)  
plot(x, y, 'r+')
```

xlabel('x')
ylabel('y')
title('y = x**2')

pretty matlabeće, ali nema hold on
save kako hocete, odaberete format
u IDLE za ovo je bilo potrebno ion() (zaglavljivanje!)
pogledati show() i ion() help

close() ili close('all')

Pylab, plot 2

```
close('all')
phi = linspace(0, 2 * pi, 1000)
x = 2 * cos(phi)
y = 2 * sin(phi)
plot(x, y)
axis([-3, 3, -3, 3])
axis('equal')
axis([-3, 3, -3, 3])
axis([-4, 4, -3, 3])
title('kružnica')

axis?
```

Pylab, plot 3

```
close('all')

help(figure)

figure(figsize = (6, 6))
plot(x, y)
axis('equal')
axis([-3, 3, -3, 3])
title('kruznica')
```

Pazite na redosled 'equal' i [-3, 3, -3, 3]!!!

Pylab, plot 4, histograms

```
x = rand(10000)
plot(x)
close()
rand?
```

```
hist(x, 100)
close()
```

```
hist?
```

Pylab, plot 5, normal distribution

```
x = randn(10000)
plot(x)
close()
```

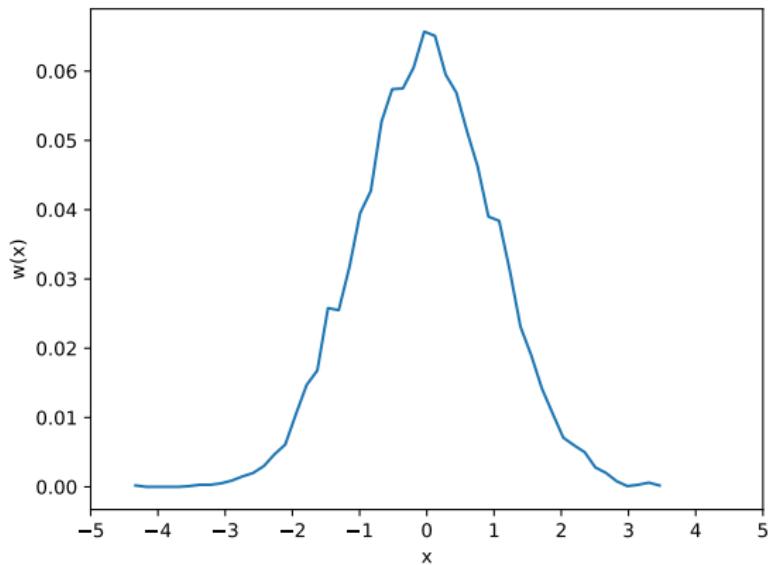
randn?

```
hist(x, 100)
close()
```

```
t = hist(x, 50)
type(t)
len(t)
len(t[0])
len(t[1])
```

Pylab, plot 6

```
y = t[0] / 10000.0    # pazite kod /!
x = t[1]
len(y)
len(x)
xx = (x[0 : len(x)-1] + x[1 : len(x)]) / 2
len(xx)
close()
plot(xx, y)
xlabel('x')
ylabel('w(x)')
xlim(-5, 5)
xticks(linspace(-5, 5, 11))
sum(y)
help(savefig)
savefig('slika') # potrazite slika.png
savefig('slika.pdf') # potrazite slika.pdf
```



Pylab, jedan script, dat.py

```
from pylab import *

deg = linspace(0, 4*360, 4*360 + 1)
wt = radians(deg)

f = exp(- wt / 2 / pi * 0.5) * sin(wt)
fp = exp(- wt / 2 / pi * 0.5)
fm = -fp

dat = array([deg, wt, f, fp, fm]).transpose()

np.save('deg.npy', deg)
np.save('f.npy', f)
np.save('fp.npy', fp)

np.savetxt('dat.txt', dat, fmt='%.4f')
```

Pylab, run, #1

na komandnoj liniji:

```
python3 dat.py  
more dat.txt  
less dat.txt  
ls *.npy  
ipython3 --pylab
```

Pylab, run, #2 i #3

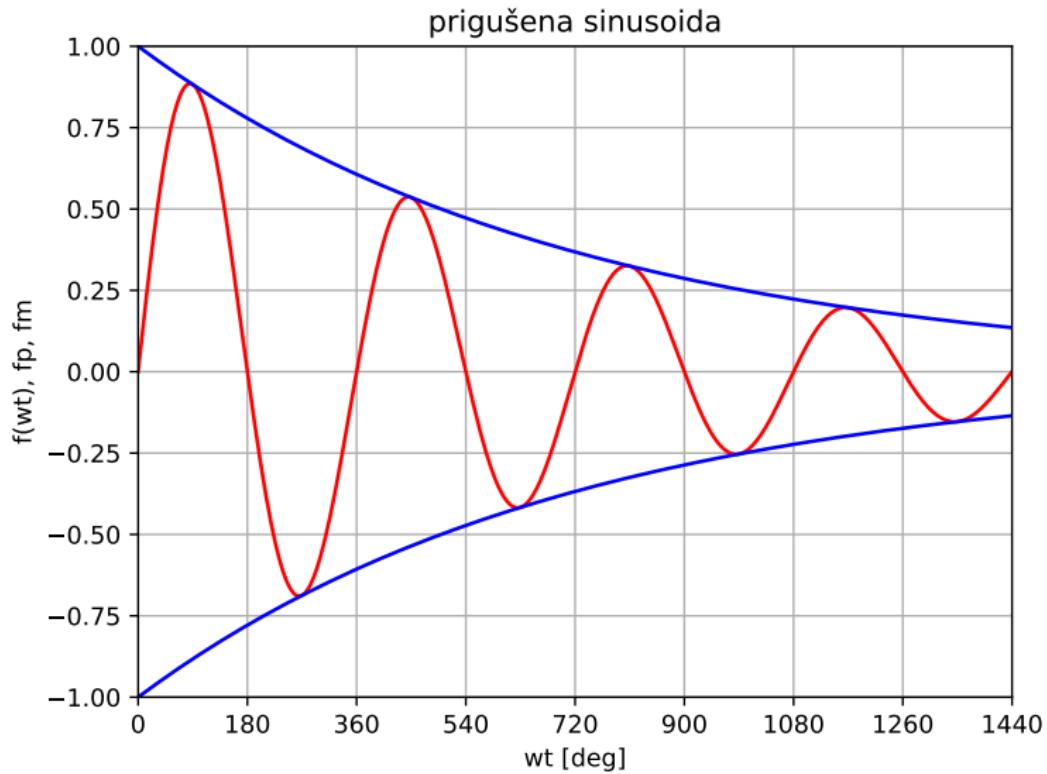
inside IPython:

```
run dat.py  
exec(open('dat.py').read())
```

execfile radi i kod IDLE

Pylab, after the run

```
np.save?  
np.savetxt?  
np.load?  
deg = np.load('deg.npy')  
f = np.load('f.npy')  
fp = np.load('fp.npy')  
fm = -fp  
plot(deg, f, 'r')  
plot(deg, fp, 'b')  
plot(deg, fm, 'b')  
xlim(0, 360 * 4)  
xticks(arange(0, 360 * 4 + 1, 180))  
xlabel('wt [deg]')  
ylabel('f(wt)')  
grid(True)  
savefig('datslik.pdf')  
close()
```

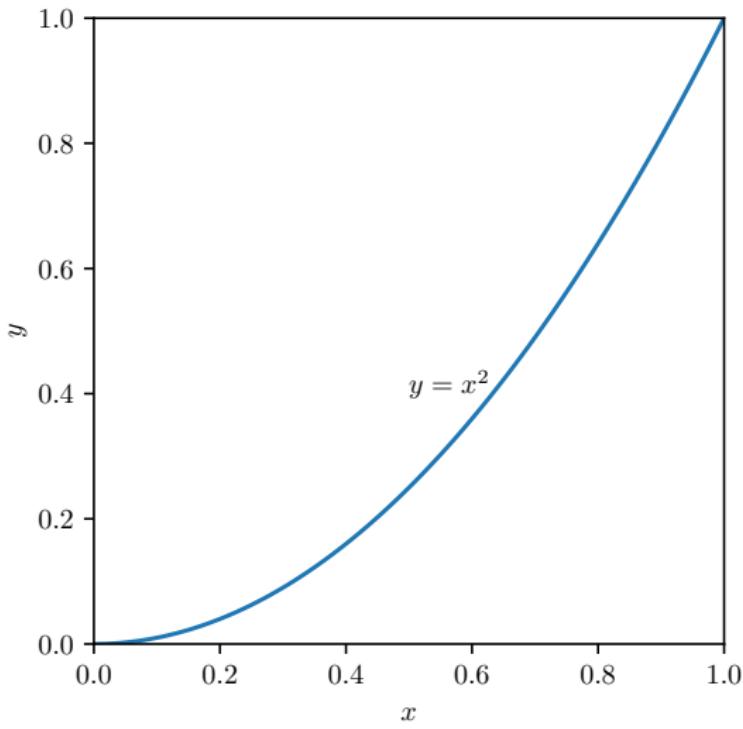


Pylab, L^AT_EX, estetika

```
help(rc)

close('all')
x = linspace(0, 1, 101)
y = x**2
rc('text', usetex = True)
rc('font', family = 'serif')
figure(0, figsize = (6, 4))
plot(x, y)
xlabel(r'$x$')
ylabel(r'$y$')
text(0.5, 0.4, r'$y = x^2$')
savefig('kvadrat.pdf', bbox_inches = 'tight')
```

kvadrat.pdf



Pylab, L^AT_EX, „standardna“ „preamble“

```
rc('text', usetex = True)
rc('font', family = 'serif')
rc('font', size = 12)
rcParams['text.latex.preamble']=[r'\usepackage{amsmath}']
```

Pylab, linear algebra, 0th part

hoću da rešim sistem jednačina:

$$x + y = 3$$

$$x - y = 1$$

```
a = array([[1, 1], [1, -1]])
print(a)
b = array([[3], [1]])
print(b)
x = solve(a, b)
print(x)
b = array([3, 1])
print(b)
x = solve(a, b)
print(x)
```

Pylab, linear algebra, 1st part

randn?

linalg.lstsq?

```
x = linspace(0, 2, 201)
```

```
y = x / 2
```

```
xe = linspace(0, 2, 21)
```

```
ye = xe / 2 + randn(21) * 0.1
```

```
A = array([xe, ones(len(xe))]).transpose()
```

```
t = linalg.lstsq(A, ye, rcond = None)
```

```
print(t)
```

```
type(t)
```

```
len(t)
```

```
a = t[0][0]
```

```
b = t[0][1]
```

Pylab, linear algebra, 2nd part

```
close('all')

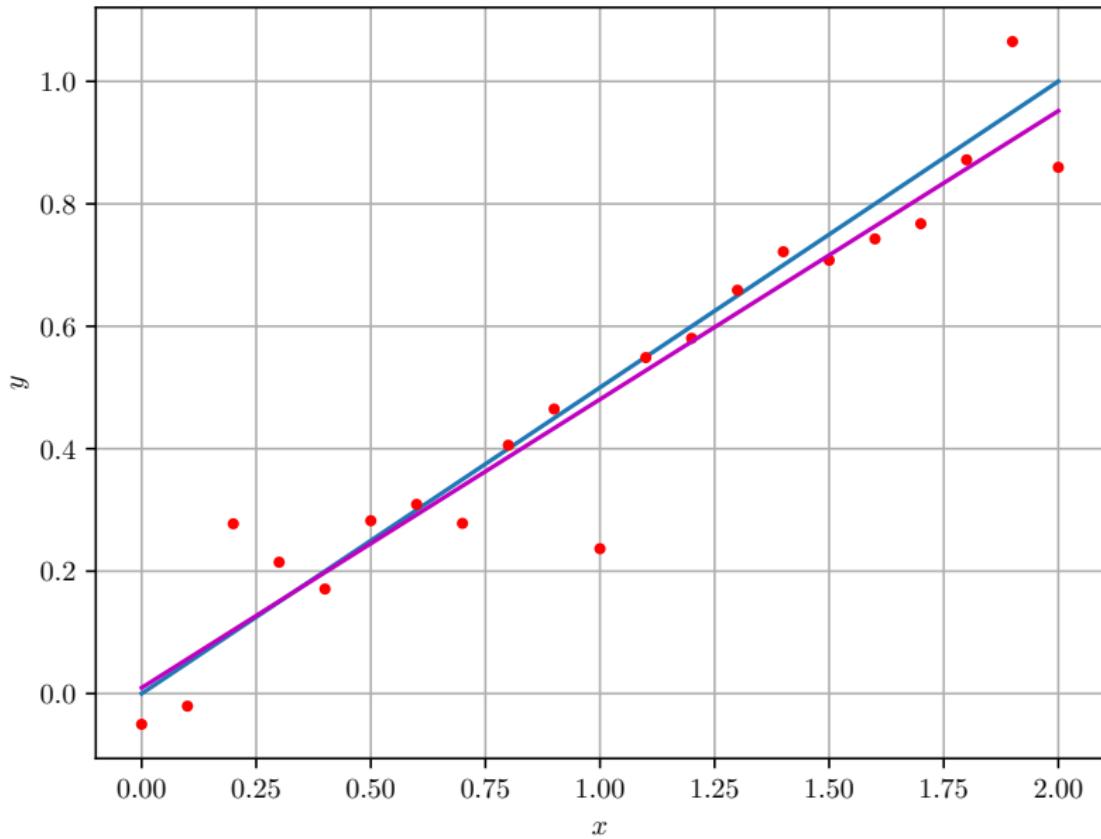
rc('text', usetex=True)
rc('font', family='serif', size='16')

plot(x, y, 'b')
plot(xe, ye, 'r.')

yfit = a * x + b
plot(x, yfit, 'm')
xlabel(r'$x$')
ylabel(r'$y$')
savefig('fitovanje.pdf')
```

fitovanje.pdf

fitovanje krive, najmanji kvadrati



Jupyter

- ▶ „notebook“ okruženje za računanje
- ▶ lako dokumentovanje rezultata
- ▶ nastao iz ipython, language agnostic
- ▶ radi i sa python i sa julia i sa sage i sa ...
- ▶ pokrene web server, sve se radi kroz browser
- ▶ popularno i za cloud applications
- ▶ <https://jupyter.org/>
- ▶ https://en.wikipedia.org/wiki/Project_Jupyter
- ▶ srećete se sa ovim još ...
- ▶ za sada samo primer ...

GNU Octave

© Predrag Pejović, 

Numerical Computing

- ▶ vrlo često u tehnici
- ▶ retko stvarni problemi imaju closed-form solution
- ▶ čak i kad imaju, pitanje upotrebljivosti
- ▶ mnogo detalja — numerički pristup
- ▶ u početku tretirano kao numerologija ...
- ▶ kolika je „saznajna vrednost“ nekog rešenja i kako se meri?

Numerical Computation, Tools #1: Matlab

- ▶ diminira(o) Matlab
- ▶ Wikipedia, http://en.wikipedia.org/wiki/Comparison_of_numerical_analysis_software, 26.12.2020:
 - ▶ “Numerical computation and simulation with extended 2D/3D visualization. Emphasis on vectorised processing.”
 - ▶ “Proprietary”
 - ▶ 2014: “\$2150 (commercial) / \$99 (student)”
 - ▶ 2014: \$300 manje nego pre tri godine, isto kao prošle i pretprešte godine
 - ▶ 2015: “\$2650 (standard) / \$625 (education)”
 - ▶ 2017: “\$2650 (standard) / \$625 (education)”
 - ▶ 2018: “\$2150 (standard) / \$500 (education) / \$149 (home) / \$49 (student)”
 - ▶ 2020: “\$2150 (standard) / \$500 (education) / \$149 (home) / \$49 (student)”
- ▶ osnovni tip podatka: n-dimenziona matrica
- ▶ extended 2D/3D visualization ...

Numerical Computation, Tools #2: GNU Octave

- ▶ GNU Octave
 - ▶ “General numerical computing package **with lots of extension modules**. Syntax mostly compatible with MATLAB”
 - ▶ “GPL”
 - ▶ “Free”
 - ▶ John W. Eaton
 - ▶ 1988
- ▶ <https://www.gnu.org/software/octave/>
- ▶ **uglavnom** sintaksa kao za Matlab, ali ima i nadgradnje i „podgradnje“
- ▶ nadgradnje: `i++`, `++i`, `i--`, `#`, do petlja, ...
- ▶ najveća razlika packages (toolboxes), mada rastu brzo!

red je da pomenemo (i ulinkujemo) i neke alternative ...

- ▶ Scilab, još jedan Matlab-like, French ...
- ▶ FreeMat (prilično neaktivran poslednjih godina)
- ▶ Python, PyLab ... prednosti modernog jezika
- ▶ R, statistika
- ▶ Julia, jako brzo, vrlo budućnost ...
- ▶ ...

Odakle se nabavlja GNU Octave?

- ▶ GNU/Linux, Ubuntu:
 - ▶ repository, Ubuntu Software Center
 - ▶ GNU Octave
 - ▶ Synaptic Package Manager
 - ▶ search “octave”
 - ▶ toolboxes, dodatna dokumentacija, ...
- ▶ win:
 - ▶ nije **bio** kao pod GNU/Linux
 - ▶ ima toolboxes
 - ▶ sada: [link](#)
- ▶ Da li vam je potreban GUI? Ima sada i to, by default!
(na **moju** žalost)

okruženja

da navedem tri:

1. `octave-cli`, command line interface
2. `octave`, GUI, od verzije 4
3. `jupyter-notebook`, mora da se instalira `octave_kernel` koristeći pip3 sa komandne linije

zbog izgrađenih navika na predavanjima uglavnom `octave-cli` pokrenemo sva tri okruženja i vidimo verzije

`ver`

literatura, dokumentacija

- ▶ dokumentacija koja ide uz GNU Octave, odlična, tu je sve što je realno potrebno
- ▶ <https://www.gnu.org/software/octave/octave.pdf>,
1121 strana!
- ▶ tutorials na www, mnogo
- ▶ izbor linkova, <http://tnt.etf.rs/~oe4sae/>
- ▶ napomena 2018: **GNU Octave 4.4.1, 1060 strana!!!**
- ▶ napomena 2020: **GNU Octave 6.1.0, 1121 strana!!!**

pocinjemo, GNU Octave kao kalkulator

2+2

2 + 2 # može space okolo, preporuka

2-3

2*3

3/4

3.0/4.0

10/3

-10/3

2^3

3^2

3^3

10^10 % sve su to floats!

3^64

3^640 # još nije Inf

Inf i NaN

```
3^6400 # Infinity, Inf  
5 / Inf  
5 * Inf  
0 / 3  
3 / 0  
0 / 0 # Not a Number, NaN  
4 * NaN  
5 / NaN
```

promenljive, ; i ans

```
a = 3
a = 4; # odziv suppressed
a
disp(a) # uočite razliku, nema a = . . .
b = 5
a + b
ans # kod Python-a je ovo _
```

```
clear all
```

```
a
```

```
3 + ...
2
```

kompleksni brojevi

čist run, dok niste predefinisali i i j

i^2

j^2

$1 + j$

$3 + 4j$

$\text{abs}(3 + 4j)$

$\text{arg}(1 + j)$

$180 / \pi * \text{arg}(1 + j)$

strukture podataka

```
tacka1.x = 5  
tacka1.y = 7  
tacka2.x = 1  
tacka2.y = -1  
tacka1 + tacka2  
tacka1.x + tacka2.x  
# slab overloading
```

ima i podršku za objektno orijentisano programiranje

stringovi

```
a = 'Pera'  
b = "Mika"  
a + b # čudan string?  
strcat(a, b)  
a == b  
a(1)  
a(3)  
b(1:3)  
b(1 : 3)  
b (1 : 3)
```

help!

```
help(floor)
help floor    # short description
doc floor    # documentation, long description
# izlazak sa q
help ceil
doc ceil
# izlazak sa q
help
doc
```

relacioni operatori

2 < 3

3 < 2

2 <= 2

2 >= 3

2 ~= 2

2 != 2 % malo proširenje u odnosu na Matlab

2 == 2 % pazite se, često pravi bugs!!!

logički operatori

1 & 0

1 & 1

0 & 0

0 | 0

1 | 0

1 | 1

~1

!1

~0

!0

logički operatori, nad nizovima

```
a = [0 1 0 1]
b = [0, 0, 1, 1]
a & b
a | b
!a
~b
a' & b'
a' | b'
[a' !a']
[a' b' a' & b']
[a' b' a' | b']
```

operatori inkrementiranja

c = 0

++c

c++

c

c--

c

--c

c

matrice

```
a = [1, 2; 3, 4]
a = [1 2; 3 4] # ovo ne može u Python-u
a = [1 2
3 4] # ni ovo

b = inv(a)
a * b
b * a
a .* b
b .* a
a ./ a
a / a
inv(a) * a
a / b
a * a
```

matrice, indeksiranje i transpose, 1

```
a = 1 : 5
a(1) # razlika u odnosu na Python!!!
a(5)
a(2 : 4)
a(-1)
length(a)
size(a)
b = a'
b(4)
length(b)
size(b)
```

matrice, indeksiranje i transpose, 2

```
c = [1 2 3; 4 5 6]
length(c)
size(c, 1)
size(c, 2)
c(2, 2)
c(2, :)
c(:, 2)
c(2 : 4) # pazite se, bugs!!!
c
```

neke posebne matrice

eye(3)
zeros(4)
zeros(4, 1)
zeros(1, 4)
ones(5)
ones(1, 5)

scripts, Gauss na loš način

```
gedit gauss.m
```

View / Highlight Mode / Octave

```
# Gauss, na los nacin

n = input('do koliko? ');

s = 0;

for i = 1 : n
    s += i;
end

printf('sum = %d\n', s)
```

scripts, Gauss na loš način, run

```
ls  
ls gauss.m  
help gauss  
gauss  
100
```

moglo je i ovako

```
a = 1 : 100    # ili a = 1 : 100;  
sum(a)  
# ili  
a = ones(1, 100)  
b = cumsum(a)  
sum(b)  
mean(a)  
mean(b)
```

save, prvi deo

```
clear all  
a = 3  
b = 5  
string = 'string'  
x = [1 2; 3 4]  
save ws # ceo workspace!
```

```
clear all  
load ws
```

a sada uradite

gedit ws

sa komandne linije, van octave, ili double click na ws

load, prvi deo

a

b

string

clear all

a

b

string

load ws

a

b

string

funkcije sa nizovima

```
deg = 0 : 360;
length(deg)
wt = deg * pi / 180;
s = sin(wt);
s2 = s .* s;
rmss = sqrt(mean(s2))
rmss - 1 / sqrt(2)
c = cos(wt);
c2 = c.^2;
rmsc = sqrt(mean(c2(1 : length(c2) - 1)))
rmsc - 1 / sqrt(2)
rmsc = sqrt(mean(c2(1 : end - 1)))
rmsc - 1 / sqrt(2)
```

plot, 2d

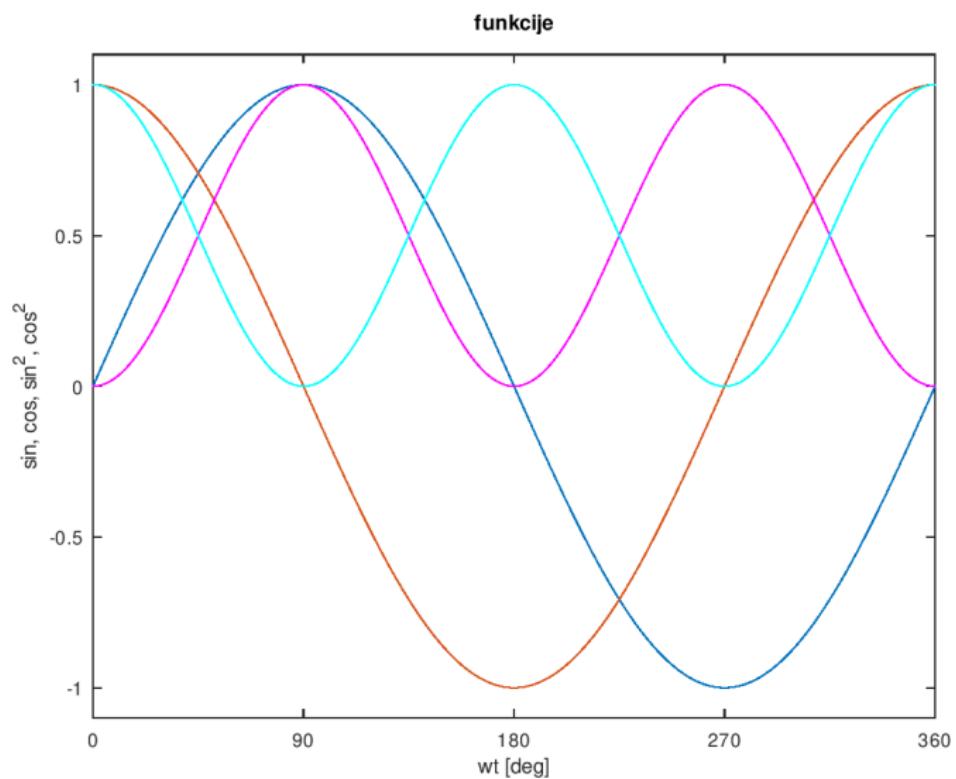
```
plot(deg, s)
plot(deg, c)
hold on
plot(deg, s, 'r')
close all
plot(deg, s, deg, c)
hold on
plot(deg, s2, 'm', deg, c2, 'c')
axis([0 360 -1 1]) # mora , in Python
set(gca, 'XTick', 0 : 90 : 360)
xlabel('wt [deg]')
ylabel('sin, cos, sin^2, cos^2')
title('funkcije')
```

plot, 2d

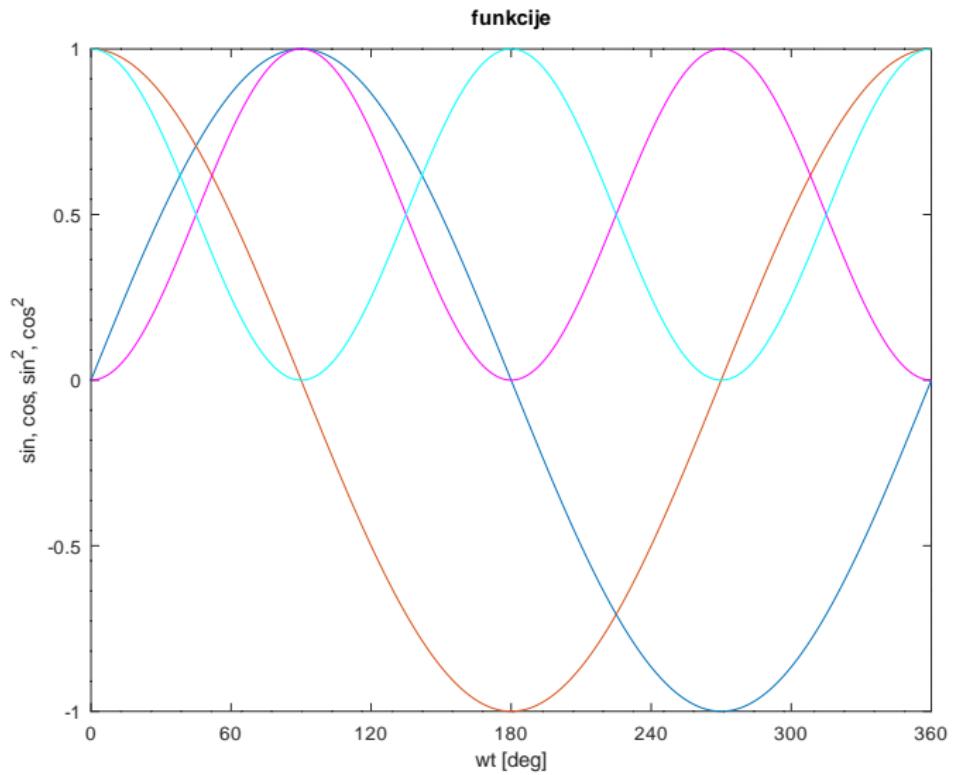
```
help figure
doc figure
help print
doc print
print('slika.png', '-dpng')
print('slika.pdf', '-dpdf')
print('slika.eps', '-deps')
print('slika.jpg', '-djpeg')
print('slika.svg', '-dsvg')
```

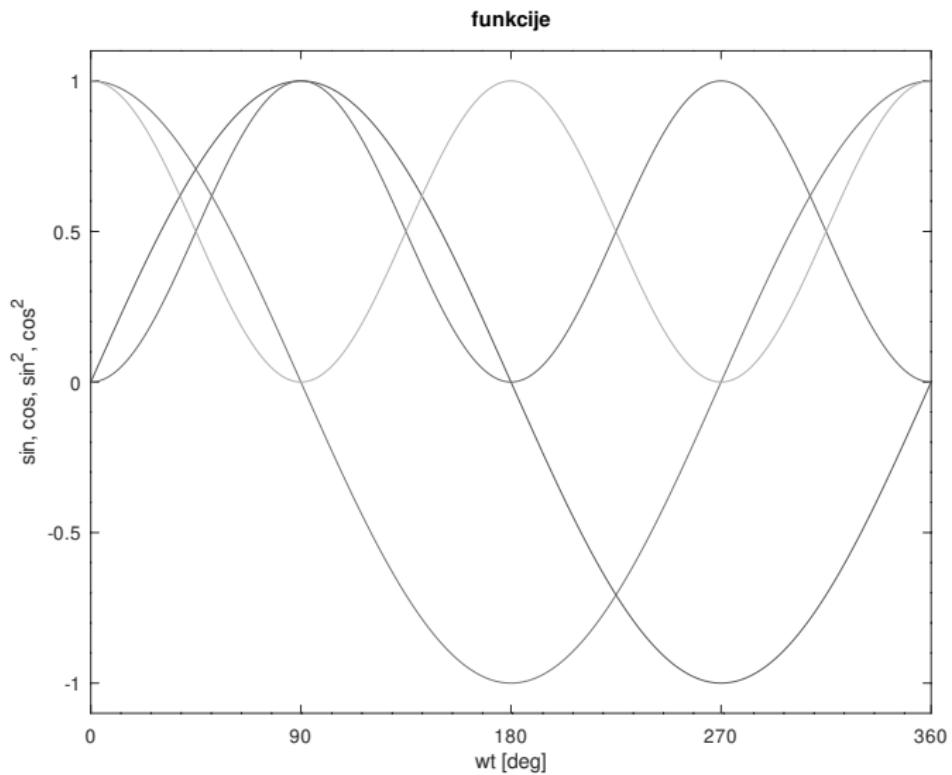
pogledajte png, pdf, eps, jpg i svg sliku, različite u odnosu na ekran, a i međusobno!

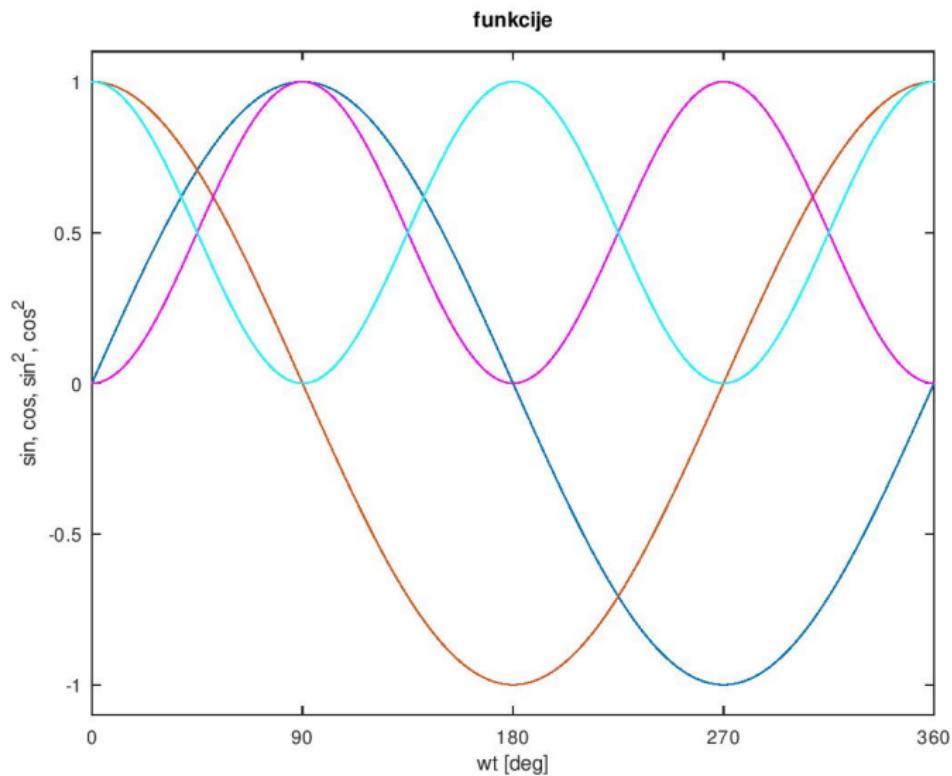
svaki drajver specifičan, kao kod gnuplot-a, što je do verzije 4.0 bio default graphics renderer, a sada optional



slika.pdf, cropped







save i load, drugi deo

whos

```
data = [deg' wt' s' c' s2' c2'];
```

```
data
```

```
size(data)
```

```
save data data
```

```
clear all
```

```
data
```

```
load data
```

```
data
```

```
gedit data
```

subplots

```
# frekvencijske karakteristike, realan pol u 1

close all

w = logspace(-2, 2, 401);
s = j * w;

H = 1 ./ (1 + s);

h = 20 * log10(abs(H));
phi = 180 / pi * arg(H);

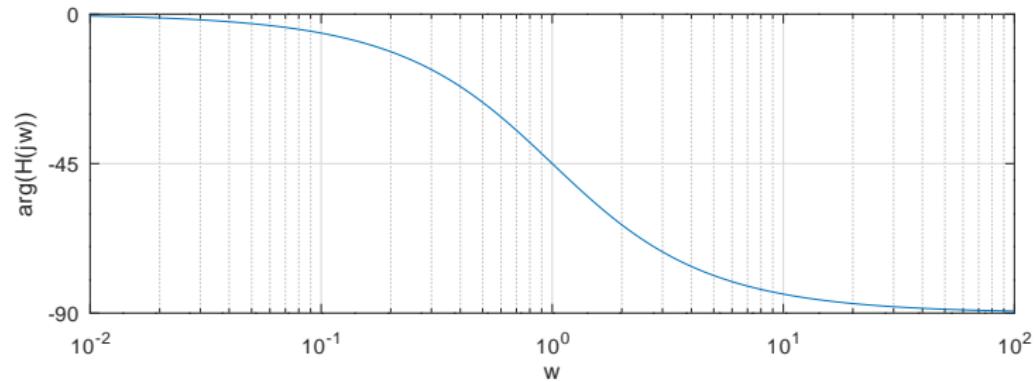
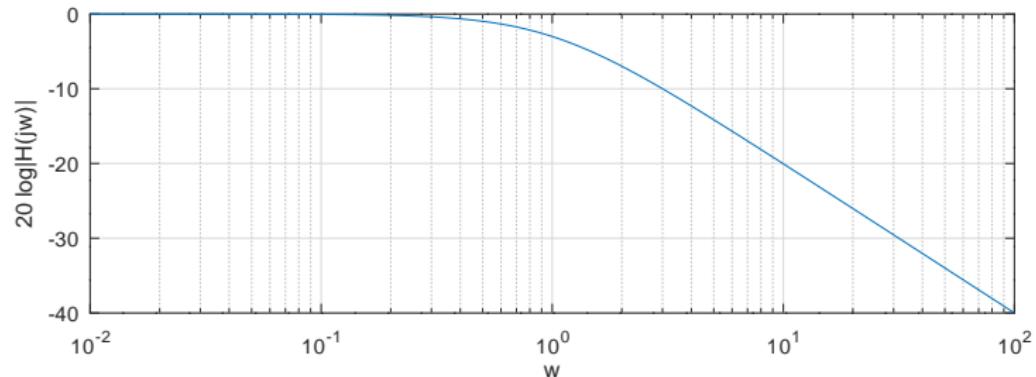
figure(1)

subplot(2, 1, 1)
semilogx(w, h)
axis([0.01, 100, -40, 0])
xlabel('w')
ylabel('20 log|H(jw)|')
grid()

subplot(2, 1, 2)
semilogx(w, phi)
axis([0.01, 100, -90, 0])
set(gca, 'YTick', [-90, -45, 0])
xlabel('w')
ylabel('arg(H(jw))')
grid()

print('rp1.pdf', '-dpdf')
```

rp.pdf, cropped



packages

ono što Matlab zove toolboxes
rastu i napreduju!
pravi trenutak je ovde!

```
pkg load control  
H = tf([1], [1 1])  
bode(H)
```

ima još packages!
menjaju se!
... synaptic

linearne jednačine

hoću da rešim sistem jednačina:

$$x + y = 3$$

$$x - y = 1$$

```
a = [1, 1; 1, -1]
```

```
b = [3; 1]
```

```
b = [3 1]'
```

```
a \ b # linear least squares
```

```
inv(a) * b
```

```
cond(a) # condition number
```

```
rcond(a)
```

```
rref(a) # reduced row echelon form
```

if

```
i = 1;  
if i == 1  
    disp(1)  
endif
```

```
if i == 1  
    disp(1)  
else  
    disp(0)  
endif
```

if-elseif-else

```
i = 2;  
if i == 1  
    disp(1)  
elseif i == 2  
    disp(2)  
else  
    disp(0)  
endif
```

```
i = 2;  
if i == 1  
    disp(1)  
elseif i == 2  
    disp(2)  
endif
```

switch

```
i = 2;  
if i == 1  
    disp(1)  
elseif i == 2  
    disp(2)  
endif
```

```
switch i  
case 1  
    disp(1)  
case 2  
    disp(2)  
case 3  
    disp(3)  
otherwise  
    disp(0)  
endswitch
```

funkcije

napravimo pdv.m, a tribute to a colleague

```
function y = pdv(x)
    #ovo je jednostavna funkcija koja računa pdv
    #
    #poziva se sa pdv(x) i vraća 1.2 * x

    y = x * 1.20
endfunction
```

```
pdv(100)
pdv(0)
pdv(200.0)
pdv([1 2 3 4 5])
pdv('wrong?')
help pdv
```

for petlja

```
for i = 1 : 100
    disp(i)
endfor
```

```
for i = [1,2,3,5,3,2,6,7,9,1,10]
    disp(i)
endfor
```

```
for i = 'pera mika laza zika'
    disp(i)
endfor
```

while petlja

```
i = 0;  
while (i <= 100)  
    disp(i++)  
endwhile
```

while petlja

```
i = 0;  
while (i <= 100)  
    disp(i++)  
endwhile
```

break

```
i = 0;  
while (i <= 100)  
    if i > 73  
        break  
    endif  
    disp(i++)  
endwhile
```

continue

```
# los program!
i = 0;
while (i <= 100)
    i++;
    if rem(i, 2) == 0
        continue
    endif
    disp(i)
endwhile

# bolje je ovako:
i = 0;
while (i <= 100)
    i++;
    if rem(i, 2) != 0
        disp(i)
    endif
endwhile
```

do petlja

```
i = 0;  
do  
    i++;  
    disp(i)  
until i == 100
```

```
i = 0;  
do  
    i++;  
    disp(i)  
until (i == 100)
```

i još mnogo toga ...

- ▶ GNU Octave je jako bogat funkcijama!
- ▶ sjajan reference manual, 1121 strana!
- ▶ ovo uputstvo će biti dopunjavano, proširivano ...
- ▶ toolboxes, packages ...
- ▶ jezik raste, mogućnosti rastu!
- ▶ a sada? koristi se, ali Python, Julia, ...
- ▶ mora da se predaje, Matlab je uzrok
- ▶ mnogo toga ima, prioritet?
- ▶ od kada je ovaj predmet započet, 2010. godine, mnogo toga je nastalo!

SageMath

© Predrag Pejović, 

SageMath, uvod 1: šta je to CAS?

- ▶ treba predavati u prvoj godini studija (još jedna jeretička misao)
- ▶ symbolic computation, algebraic computation, computer algebra ...
- ▶ nekad smatrano za umnu delatnost ...
- ▶ u osnovi manipulacija stringova po utvrđenim pravilima ...
- ▶ sećate se zbirke Demidović?
- ▶ sve ovo može da se automatizuje i automatizuje se odavno, Macsyma početa 1968, ...
- ▶ computer algebra systems (CAS)? http://en.wikipedia.org/wiki/Comparison_of_computer_algebra_systems

SageMath, uvod 2: CAS proprietary alternative

- ▶ **Mathematica**, \$2,495 (Professional), \$1095 (Education), \$140 (Student), \$69.95 (Student annual license), \$295 (Personal), **free on Raspberry Pi hardware**
- ▶ **Maple**, \$2,390 (Commercial), \$2,265 (Government), \$995 (Academic), \$239 (Personal Edition), \$99 (Student), \$79 (Student, 12-Month term)
- ▶ navijačke strasti, Maple vs. Mathematica
- ▶ Symbolic Math Toolbox (MATLAB), \$3,150 (Commercial), \$99 (Student Suite), \$700 (Academic), \$194 (Home) including required Matlab
- ▶ bio popularan **Derive**, [http://en.wikipedia.org/wiki/Derive_\(computer_algebra_system\)](http://en.wikipedia.org/wiki/Derive_(computer_algebra_system)), discontinued 2007

SageMath, uvod 3: CAS slobodne alternative

- ▶ Maxima, a computer algebra system, GPL
- ▶ [http://en.wikipedia.org/wiki/Maxima_\(software\)](http://en.wikipedia.org/wiki/Maxima_(software))
- ▶ <http://maxima.sourceforge.net/>
- ▶ zasnovana na MIT Macsyma,
<http://en.wikipedia.org/wiki/Macsyma>
- ▶ William Frederick Schelter, GPL, DOE Macsyma 1982
http://en.wikipedia.org/wiki/Bill_Schelter
- ▶ star program, ali aktivno se razvija
- ▶ uključen u **Scilab** i **Euler Math Toolbox**
- ▶ uključen u **SageMath**, [http://en.wikipedia.org/wiki/Sage_\(mathematics_software\)](http://en.wikipedia.org/wiki/Sage_(mathematics_software))
- ▶ napredno: **SageMath**, <http://www.sagemath.org/>

SageMath, uvod 4: CAS slobodne alternative

- ▶ SymPy paket za simboličko računanje koristeći Python
- ▶ cilj: isto što i wxMaxima
- ▶ ideja: Python sintaksa, poznata
- ▶ moduli, funkcije za simboličko računanje
- ▶ <http://sympy.org/en/index.html>
- ▶ <https://github.com/sympy/sympy/releases>
- ▶ <http://live.sympy.org/>
- ▶ uputstvo 1.7.1, 2592 strane, 11.12.2020; čitate stranu po stranu?
- ▶ usput, uputstvo 1.3, 2044 strane, 14.09.2018; napreduje?
- ▶ Ubuntu: Software Center ili Synaptic
- ▶ volite Mathematica sintaksu?
- ▶ **odličan pregled i tutorial**
- ▶ za one koji vole matematiku: <http://www.sagemath.org/>

SageMath, uvod 5: SageMath, konačno

- ▶ <http://www.sagemath.org/>
- ▶ od 24.02.2005, GPL
- ▶ Python sintaksa
- ▶ koristi Maxima i SymPy i mnoge druge free programe, kombinuje ih
- ▶ zato je napuštena politika dva alata za jedan posao
- ▶ **Literatura:**
 1. Mathematical Computation with Sage
 2. Sage for Undergraduates
- ▶ za Ubuntu 20.04 uključen u repository
- ▶ koristi pun kapacitet free software ideologije, kombinuje puno različitih alata u novu funkcionalnost, izbegava “*reinventing the wheel*”
- ▶ autor (pokretač) programa: William A. Stein

SageMath: početak sa komandne linije

- ▶ za moj uzrast i vid na manje od 1 m bilo je zgodno prometiti color scheme terminala na light
- ▶ sage
- ▶ pogledajte ime terminala, IPython
- ▶ whos
- ▶ 3 + 2
- ▶ factorial(100)
- ▶ x = factorial(100)
- ▶ type(x)
- ▶ whos
- ▶ del x
- ▶ whos

SageMath: faktorizacija celih brojeva, \wedge i $**$

- ▶ `factor(123456789)`
- ▶ `_`
- ▶ `type(_)`
- ▶ `_`
- ▶ `factor(123456789)`
- ▶ `expand(_)`
- ▶ `type(_)`
- ▶ zaista dynamic typing!
- ▶ `3^2`
- ▶ `3**2`

SageMath: polinomi

```
▶ type(x)
▶ x = var('x')
▶ type(x)
▶ p = x^2 + 2 * x + 1
▶ p
▶ show(p)
▶ print(p)
▶ pretty_print(p)
▶ latex(p)
▶ factor(p)
▶ p.factor()
▶ p.roots()
```

SageMath: jupyter

- ▶ izđete iz sage, ^d (ctrl/d)
- ▶ na komandnoj liniji:
- ▶ **sage -n jupyter**
- ▶ sačekate da se otvori browser, New, SageMath 9.0
- ▶ alternativa, na komandnoj liniji
- ▶ **jupyter-notebook**
- ▶ dalje isto ...
- ▶ modernija alternativa, na komandnoj liniji
- ▶ **jupyter lab**
- ▶ dalje slično, drugačiji interfejs
- ▶ server se gasi sa ctrl/c (^c) na komandnoj liniji ili sa Quit u osnovnom jupyter prozoru
- ▶ shortcut keys, da se pojave h
- ▶ ponavljamo dosadašnje ...

SageMath: razlomci, konstante i numerika 1

1/2

type(_)

10/26

$z = 1/2 - 1/3; z$

$zz = 1./2 - 1/3; \text{print}(zz); \text{print}(\text{type}(zz))$

show(z)

type(z)

$(z).n(\text{digits} = 10)$

$(z).n(\text{digits} = 100)$

SageMath: razlomci, konstante i numerika 2

i^2

I^2

pi

show(_)

pi.n()

e^(i * pi) + 1

e.n(digits = 100)

pi.n(digits = 500)

SageMath: razlomci, konstante i numerika 3

```
r = e^(pi * sqrt(163))
```

```
r.n(digits = 27)
```

```
r.n(digits = 100)
```

```
n(sqrt(2))
```

```
n(sqrt(2), digits = 50)
```

```
N(sqrt(2))
```

```
(sqrt(2)).numerical_approx()
```

```
numerical_approx(sqrt(2))
```

SageMath: razlomci, konstante i numerika 4

infinity

Infinity

oo

1 / oo

2 + oo

SageMath: algebra 1

```
show((x + 1)^2).expand())
```

```
show((x + 1) * (x - 1)).expand())
```

```
y = (x-3)^7.expand(); show(y)
```

```
show(factor(y))
```

```
eq = (x - 4) * (x - 5) * (x - 6) * 7).expand()  
show(eq)
```

```
seq = solve(eq, x); show(seq)
```

```
seq[0].rhs(), seq[1].rhs(), seq[2].rhs()
```

```
show(factor(eq))
```

SageMath: algebra 2

```
show(factor(4 * x^5 - 4 * x^4 -  
13 * x^3 + x^2 - 17 * x + 5))  
  
s = var('s')  
ex = (s^3 + 4 * s^2 + 6 * s + 4) / \  
(s^3 + 3 * s^2 + 3 * s + 1)  
show(ex)  
  
show(factor(ex))  
  
pfex = ex.partial_fraction(); show(pfex)  
  
show(pfex.expand())  
  
show(pfex.simplify_rational())
```

SageMath: algebra 3

```
expression = sin(x / (x^2 + x)) == \
exp((log(x) + 1)^2 - log(x)^2); show(expression)

show(expression.simplify())

show(expression.expand())

show(expression.simplify_rational())
```

SageMath: algebra 4

```
a = var('a')
expr = (x^(a / 2) + 1)^2 * (x^(a / 2) - 1)^2 \
/ (x^a - 1)
show(expr)

show(expr.expand())

show(expr.simplify())

show(expr.rational_simplify())

show(expr.simplify_full())
```

SageMath: trigonometrija 1

```
cos(pi / 3)
```

```
show(sin(pi / 3))
```

```
sin(pi / 3).n()
```

```
sin(pi / 3.)
```

```
sin(1)
```

```
sin(1.)
```

SageMath: trigonometrija 2

```
show(csc(45 * pi / 180))
```

```
tan(pi / 4)
```

```
show(tan(pi / 8))
```

```
show(tan(pi / 16))
```

```
show(tan(pi / 32))
```

```
show(acos(1/2))
```

```
180 / pi * asin(sqrt(3) / 2)
```

```
show(acsc(1))
```

SageMath: trigonometrija 3

```
eq = sin(x)^2 + cos(x)^2; print(eq)
show(eq)
latex(eq)

eq.simplify()

eq.trig_simplify()

a, b = var('a b')
e1 = sin(a + b)
show(e1)
```

SageMath: trigonometrija 4

```
e2 = e1.trig_simplify(); show(e2)
```

```
e2.trig_reduce()
```

```
e3 = sin(x)^2; show(e3)
```

```
e4 = e3.trig_reduce(); show(e4)
```

```
show(e4.simplify_full())
```

SageMath: jednačine 1

```
a, b, c = var('a b c')
eq = a * x^2 + b * x + c
s = solve(eq, x)
show(s)

x1 = s[0].rhs(); show(x1)

x2 = s[1].rhs(); show(x2)

eq.subs(x = x1)

_.expand()

eq.subs(x = x2).expand()
```

SageMath: jednačine 2

```
s2 = solve(x^3 + 4 * x^2 - 3 * x + 1, x); show(s2)

latex(s2)

n(s2[0].rhs()), n(s2[1].rhs()), n(s2[2].rhs())

show(n(s2[1].rhs()))

y = var('y')
solve([x + y == 3, x - y == 1], [x, y])

solve([x + y == 2, 2 * x + 2 * y == 4], [x, y])

solve([x + y == 2, 2 * x + 2 * y == 5], [x, y])
```

SageMath: algebarske jednačine, sistemi

```
x, y = var('x y')
e1 = x^2 + y^2 == 41
e2 = y == x + 1
show(e1)
show(e2)

s = solve([e1, e2], [x, y])
show(s)

s1 = solve(e1, y); show(s1)

y1 = s1[0].rhs(); show(y1)

y2 = s1[1].rhs(); show(y2)

set_verbose(-1)
plot([y1, y2, e2.rhs()], (x, -10, 10), aspect_ratio = 1)
```

SageMath: limesi 1

```
limit((1 + 1 / x)^(3 * x), x = infinity)
```

```
_ .n()
```

```
show(_)
```

```
f = (x - 2) / (x^2 - 4)
```

```
#f(x = 2)
```

```
limit(f, x = 2)
```

```
show(limit(x^3, x = oo))
```

```
show(limit(x^3, x = - oo))
```

SageMath: limesi 2

```
show(limit(atan(x), x = oo))

show(limit(atan(x), x = - oo))

show(limit(sin(17 * x) / x, x = 0))

show(limit(1 / x, x = 0))

show(limit(1 / x, x = 0, dir = '+'))

show(limit(1 / x, x = 0, dir = '-'))

f(x) = atan(x)
h = var('h')
show(limit((f(x + h) - f(x)) / h, h = 0))
```

SageMath: izvodi 1

```
diff(x^2, x)
```

```
(x^2).diff()
```

```
diff(x^2, x, 2)
```

```
(x^2).diff(x, 2)
```

```
(x^2).diff(2)
```

```
(x^2).diff(x)
```

```
diff(sin(x))
```

```
diff(sin(x), 2)
```

SageMath: izvodi 2

```
diff(sin(x), x, 3)
```

```
(sin(x)).diff(4)
```

```
show((atan(x)).diff(x))
```

```
diff((x^2 + 2 * x + 1) * e^(3 * x), x)
```

```
_.factor()
```

```
show(_)
```

```
y = var('y')
```

```
diff(sin(x * y), x)
```

```
diff(sin(x * y), y)
```

SageMath: integrali 1

```
integrate(sin(x), x)
```

```
integrate(sin(x), (x, 0, pi))
```

```
integrate(1/(1 + x^2), x)
```

```
integrate(1/(1 + x^2), (x, 0, 1))
```

```
show(_)
```

SageMath: integrali 2

```
show(integrate((3 * x + 5) / (x^2 + x + 1), x))
```

```
integrate(x^-(1/5), (x, 4, 5))
```

```
show(_)
```

```
(__).n()
```

```
integrate((log(x) / x)^2, (x, 1, oo))
```

SageMath: kompleksni brojevi 1

```
z = 3 + 4j; show(z); show(type(z))

Z = 3 + 4 * i; show(Z); show(type(Z))

abs(z)

show(abs(Z)); show(arg(Z))

x, y = var('x, y')

z = x + i * y; show(z)

print(z.real()); show(z.real());
print(z.imag()); show(z.imag())
```

SageMath: kompleksni brojevi 2

```
show(z.conjugate())  
  
w = 1 / z; show(w)  
  
show(w.rectform())  
  
show(abs(z)); show(abs(w))  
  
show(z.norm()); show(w.norm());  
show((w.norm()).simplify_full())
```

SageMath: Laplace 1

```
t = var('t')
s = var('s')
a = var('a')
w = var('w')

show(laplace(exp(- a * t), t, s))

show(laplace(sin(w * t), t, s))

show(laplace(cos(w * t), t, s))
```

SageMath: Laplace 2

```
inverse_laplace(s / (s^2 + 1), s, t)
```

```
inverse_laplace(1 / (s^2 + 1), s, t)
```

```
inverse_laplace(1 / s, s, t)
```

```
inverse_laplace(1 / s^2, s, t)
```

```
inverse_laplace(1 / (s + 1), s, t)
```

SageMath: Laplace 3

```
inverse_laplace(1, s, t)

ex = (s^3 + 4 * s^2 + 6 * s + 4) / \
(s^3 + 3 * s^2 + 3 * s + 1)
show(ex)

show(inverse_laplace(ex * 1 / s, s, t))
```

SageMath: diferencijalne jednačine 1

```
t = var('t')
y = function('y')(t)

deq1 = diff(y, t) + y == 0
show(deq1)
desolve(deq1, [y, t])
print(desolve(deq1, [y, t]))
show(desolve(deq1, [y, t]))

deq2 = diff(y, t) - y == 0
show(deq2)
show(desolve(deq2, [y, t]))

deq3 = diff(y, t) + y == 1
show(deq3)
s3 = desolve(deq3, [y, t])
print(s3); show(s3)
```

SageMath: diferencijalne jednačine 2

```
print(expand(s3))
show(expand(s3))

deq4 = diff(y, t) + y == cos(2 * t)
show(deq4)
desolve(deq4, [y, t])

expand(_)

sol = _

show(sol)

deq5 = diff(y, t, 2) + 4 * y == 0
show(deq5)
show(desolve(deq5, [y, t]))
```

SageMath: diferencijalne jednačine 3

```
deq6 = diff(y, t, 2) - 4 * y == 0
show(deq6)
show(desolve(deq6, [y, t]))  
  
deq7 = diff(y, t, 2) + 4 * y == cos(2 * t)
show(deq7)
show(desolve(deq7, [y, t]))  
  
deq8 = diff(y, t, 2) + 4 * y == cos(t)
show(deq8)
show(desolve(deq8, [y, t]))  
  
deq9 = diff(y, t, 2) + 2 * diff(y, t) + 4 * y == cos(t)
show(deq9)
show(desolve(deq9, [y, t]))
```

SageMath: diferencijalne jednačine 3

```
deq10 = diff(y, t, 2) + 4 * diff(y, t) + 4 * y == cos(t)
show(deq10)
show(desolve(deq10, [y, t]))
```

```
deq11 = diff(y, t, 2) + 4 * diff(y, t) + 4 * y == cos(t)
show(deq11)
show(desolve(deq11, [y, t]))
```

so, we're done . . .

šta nismo stigli, a bilo je u planu punom optimizma?

- ▶ SageMath, plotovanje
- ▶ SageMath, linearna algebra (za ovo mi je baš žao)
- ▶ **Julia** (za ovo mi je najviše žao)
- ▶ samo ovo nije kraj, idite dalje, samo sada idete sami . . .
- ▶ nadam se da ste kroz PSAE prošli početne korake i postali slobodni i samostalni
- ▶ happy hacking!

šta nismo ni planirali, a potrebno je, t ≪ (potrebnog)

- ▶ komandna linija
- ▶ regular expressions, Charles Severance, video, 35', 23"
- ▶ detaljnije Code::Blocks ili neki drugi IDE
- ▶ Eclipse <http://www.eclipse.org/>
- ▶ ukratko LibreOffice, više pravila pisanja
- ▶ makar malo vremena za SciLab
- ▶ malo vremena za GIMP, mada nije problem
- ▶ malo vremena za Inkscape, ovo je veći problem
- ▶ Ngspice <http://ngspice.sourceforge.net/>
- ▶ moj izbor: julia
- ▶ ovo je samo početak, mada je za sada ...

— K R A J —