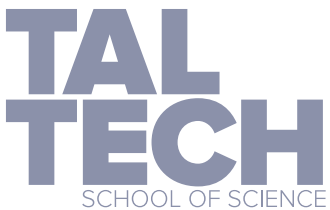
Loeng №1: *Kursusest praktiliselt*, Python, sissejuhatus programmeerimisse Pythonis, Pythoni installeerimine, konsool, Jupyter, minu esimene programm, lihtne aritmeetika, sõne

Dmitri Kartofelev, PhD[1] Aleksandra Zahharova, MSc[2]

[1]Tallinna Tehnikaülikool, loodusteaduskond, küberneetika instituut, tahkisemehhaanika labor
[2]Tallinna Tehnikaülikool, loodusteaduskond, keemia ja biotehnoloogia instituut



Loengu sisu

- *Kursusest praktiliselt*

- Sissejuhatus
  - Arvuti
  - Programmeerimisest üldiselt
  - Programmeerimise ajaloost
- Python
  - Pythonist ja tema olemusest
  - Miks just Python?
  - Pythoni installeerimine ja ülesseadmine
  - Pythoni interpretaatoriga suhtlemine, esimene kokkupuude Pythoniga
  - Lihtne aritmeetika
  - Sõne (string), `print` funktsioon
- Praktikum

Kursusest

Sissejuhatus programmeerimisse Pythoni baasil, YFX0500

Python: Kõrgetasemeline üldotstarbeline programmeerimiskeel, loodud aastatel 1989–1991 hollandlase Guido van Rossum'i poolt, kes oli suur **Monty Python's Flying Circus** fän.

Kursusest: kontakt õppejõuga ja kasulikke linke

Üldinfo: `https://www.tud.ttu.ee/web/dmitri.kartofelev/`

E-mail: `dmitri.kartofelev@taltech.ee`

Elektroonsed abivahendid:

- **TalTech Moodle** (kursuse foorum ja kontakt õppejõuga)
  - Kursuse kood: YFX0500
  - Salasõna: pythonon****
- **Kursuse koduleht:** `https://www.tud.ttu.ee/web/dmitri.kartofelev/python.html`
- Python Docs: `https://www.python.org/`

Kursuse ainekava:

- ÕIS: `http://ois.ttu.ee/aine/YFX0500`

Kursusest: ainekava (täpsustatud)

Programmeerimise olemus. Programmeerimiskeele Python ja selle töövahendite ülevaade. Pythoni süntaks ja programmi struktuur. Standardsed andmetüübid (`int`, `float`, `complex`, `str`, `bool`, `list`, `tuple`, `dict`, `set`, `object`, `type`), objektide defineerimine ja kasutamine. Muutujate defineerimine ja kasutamine. Avaldised (tehted erinevat andmetüüpi objektidega: `+`, `-`, `*`, `/`, `//`, `**`, `%`, `not`, `and`, `or`, `|`, `&`, `^`, `[]`, `()`, `in`, `is`, `<`, `>`, `==`, `!=`, tehete prioriteetsus) ja lausendid (`del`, `for`, `while`, `if`/`elif`/`else`, `try`/`except`/`with`, `break`/`continue`/`pass`). Standardsed sisseehitatud funktsioonid (`print`, `range`, tüübi teisendused, jne), funktsioonide defineerimine (`def` ja `return` lausendid, `lambda` avaldis) ja kasutamine, generaatorid (`yield` lausend), dekoraatorid (`@` operaator). Objektorienteeritud programmeerimise alused (`class` lausend). Standardsed moodulid, moodulite loomine ja kasutamine (`import` ja `from` lausendid), moodulite installeerimine (Anaconda, `conda`, `pip`). Andmefailide loomine ja kasutamine. Teadusarvutuse paketid (numpy, scipy, matplotlib) ja töökeskkonnad (PyCharm, Spyder, Jupyter, IPython).

Kursusest: kursuse ülesehitus

- Sissejuhatus
  - Arvuti
  - Programmeerimine ja programmeerimiskeeled
  - Pythoni programmeerimiskeel
- Pythoni alused
  - Tööriitad, interpretaator, IDE: iPython, Jupyter, Spyder jt.
  - sõne, sõne vorming
  - Pythoni andmetüübid, tehted ja avaldised
  - funktsioon, anonüümne funktsioon, rekursioon, sisseehitatud funktsioonid, generaator, klassi meetodid, protseduur, peaprotseduur
  - Määratud ja määramata tsüklid
  - Silumine ja erisuste haldus
  - Failihaldus ja serialiseerimine, modulaarsus ja teekide import, distributsiooni haldamine
  - Objektorienteeritud programmeerimine (OOP)

Kursusest: kursuse ülesehitus

- Teadusarvutused ja muud rakendused
  - Algoritm
  - NumPy — Andmete esitus massiividena
  - SciPy — Teadusarvutuste algoritmid
  - matplotlib — Andmete analüüs ja visualiseerimine
  - jm.

Loengute ja arutatud teemade loetelu: `https://www.tud.ttu.ee/web/dmitri.kartofelev/YFX0500/loengud.pdf`

Kursusest: loengud, praktikumid ja kursuse töö

Kontaktõppe kestus iga nädal on **kolm astronoomilist tundi**.

- Esimene tund:
  - teooria, näited, seletused, soovitused, süntaksi reeglid
- Ülejäänud kaks tundi:
  - praktikum, iseseisev töö, diskusioon, seminar

Kursuse töö: Enda kirjutatud programm koos arlgoritmi seletuse või voodiagrammiga. **Tähtaeg:** enne sinu eksamiaega.

Eksamist

Eksamist:

1. Algteadmiste kontroll (teooria)
   - Valikvastustega küsimused/ülesanded (Moodles, ajapiirang)
2. Rakenduslike ülesannete lahendamine (praktika)
   - Kodeerimine ilma arvutita (paberil ilma abivahenditeta)
     - Pythoni süntaksi peast tundmine
   - Etteantud lähtekoodi silumine või erisuste haldamine
     - Koodi lugemine ja mõistmine
     - Koodi silumine (IDE, konsool)
     - Erisuste haldamine
   - Kodeerimine
     - Sisendandmete küsimine kasutajalt
     - Algoritmi kodeerimine Pythonis
     - Tulemuste väljastamine (konsooli, graafikule, faili)

Iseseisvaks harjutamiseks:

Teooria: `https://www.sanfoundry.com/1000-python-questions-answers/`

Praktika: `http://www.practicepython.org/`

Eksamist: praktiline osa, kohapeal

- Eksamisessioon:
  - I konsultatsioon: *Lepi aeg kokku*
  - I eksam: *vt. Moodle kalender*
  - II konsultatsioon: *Lepi aeg kokku*
  - II eksam: *vt. Moodle kalender*
  - III eksam: *vt. Moodle kalender*
- Eksamile registreerimine ÕISi kaudu
  - Registreerida saab eksamieelse tööpäeva kella 16:00-ni. Jälgige ÕISis toodud tingimusi!
  - Maksimaalselt 2 eksamisooritust, viimane hinne jääb kehtima
- Konsultatsiooni kestvus: 1.5 h
- Eksami kestvus: 1.0–2.0 h
- Mitteilmunud üliõpilase eksamitulemuseks on “MI”
- Kõrvalist abi kasutanud üliõpilane eemaldatakse eksamilt ja eksamihindeks märgitakse “0”
- Tagasiside eksamist ühe nädala jooksul

Eksamist: praktiline osa, kohapeal

- Lubatud abivahendid (praktiline osa ül. 2 ja 3):
  - kirjutusvahend ja konspekt
  - arvutiklassi või enda arvuti
  - Python, (PyCharm, JuPyter, Spyder, IPython)
  - Pythoni juhendid avatud veebilehitseja aknas
- Kõik muu on keelatud, sh.
  - nutitelefon
  - kõikvõimalikud sotsiaalmeedia kanalid
  - suhtlemine kaaslastega
  - tehisintellekt (nt. *chatGPT*)
- Keelatud tegevusele järgneb hoiatus, ja edaspidi eksamilt eemaldamine
  - Oma lahenduse jagamisel langeb hinne 1 palli võrra kõigil osapooltel

Eksamist kokkuvõtvalt: eksamijuhend

KÜSIMUSTELE VASTAMINE
Moodle keskkonnas piiratud ajaga valikvastused/ülesanded. Loe küsimusi hoolikalt.

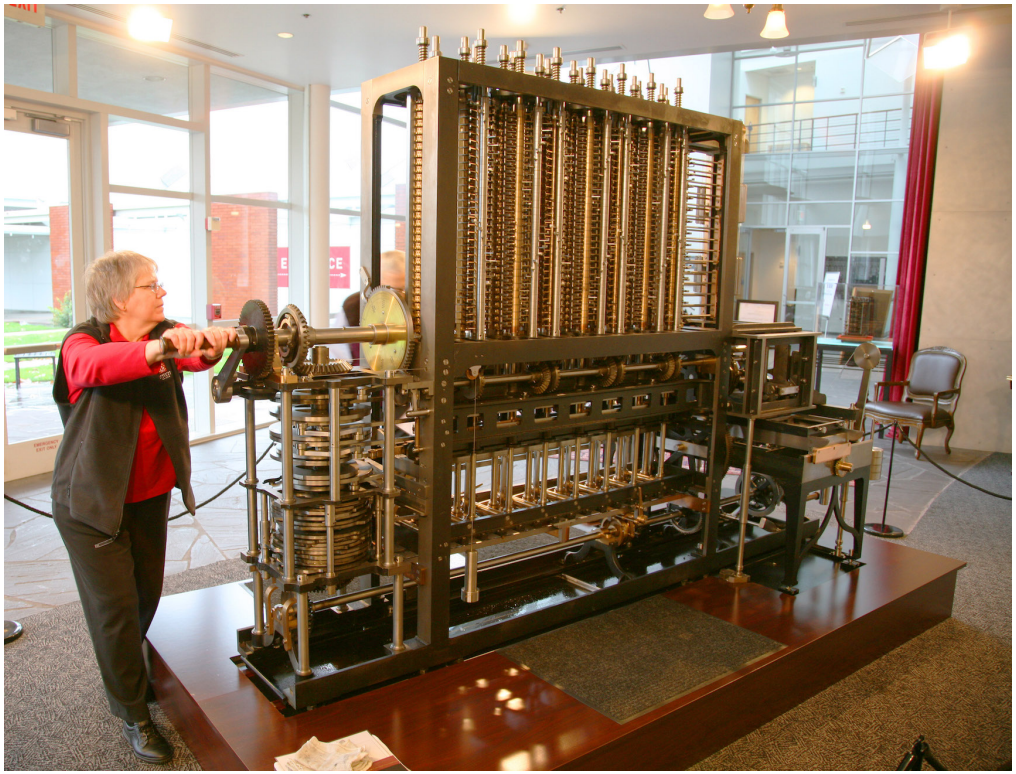
AJA PLANEERIMINE
Algteadmisi kontrolliva osa (Moodle) täitmiseks võiks kuluda 10–30 min. Praktilise osa peale u 60 min.

Rakendusülesande lähtekoodid tuleb saata e-kirja aadressile: `dmitri.kartofelev@taltech.ee`. Lisa kirjale oma nimi, üliõpilase kood ja eksami ülesannete lahenduste fail. Subjekti reale märgi, et tegemist on eksamiga. E-kiri tuleb saata eksami aja jooksul. Hiljem saadetud E-kirju ei arvestata.

TAGASISIDE
Tagasiside eksamisoorituse kohta antakse ühe nädala jooksul.

Arvuti

Arvuti[1] on *seade*, mis töötleb andmeid automaatselt, sealhulgas teeb aritmeetika- ja loogikatehteid *programmeeritavate* algoritmide alusel.

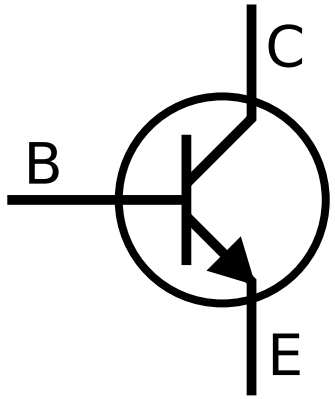


Joonis: Charles Babbage'i "Analytical Engine".

[1] Wikipedia.org

Moodne arvuti

Moodsa arvuti (*elektroonika seade*) protsessori töö põhineb **transistorite** ekspluatatsioonil.
Vt. iseseisvalt: loogikavärav[2] (logic gate) ehk loogikaelement.



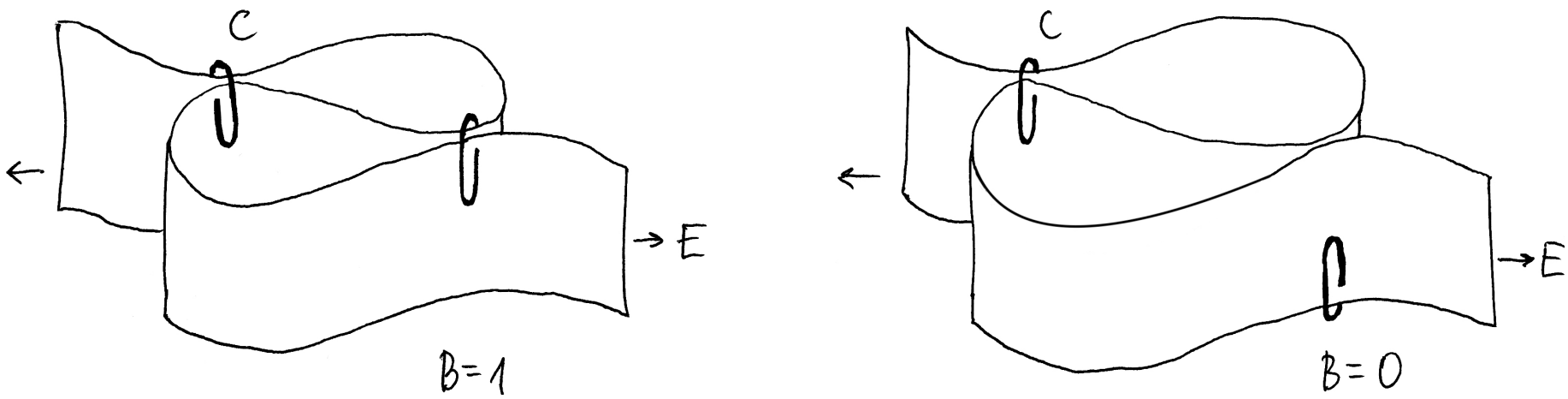
Joonis: Transistori sümbol.

Transistor on kolme väljaviiguga pooljuhtseadis ehk triood mida kasutatakse elektriahelate *avamiseks/sulgemiseks* ja elektrisignaalide *võimendamiseks*.

[2] `https://en.wikipedia.org/wiki/Logic_gate`

Transistori analoog

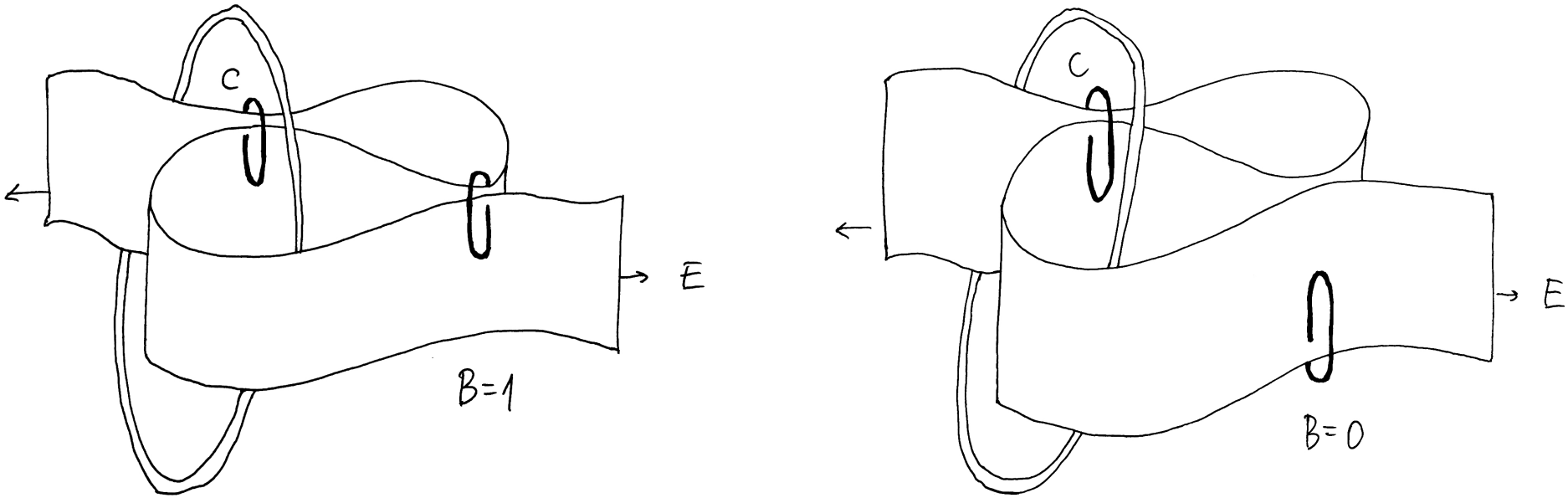
Lihtsustatud *ebatäpne* analoog: üksnes lüliti funktsioon.



Joonis: Kirjaklamber – elektron/id, vool.

Transistori analoog

Täpsem analoog: lüliti ja võimendi.



Joonis: Kirjaklamber – elektron/id, vool; kummipael – palju elektrone, rohkem voolu.

Doomino klotsidest arvuti

Vt. iseseisvalt:

Loogika elemendid ja muu vajalik (arvude liitmine):
`https://www.youtube.com/watch?v=w6E7aQnA4Ws`

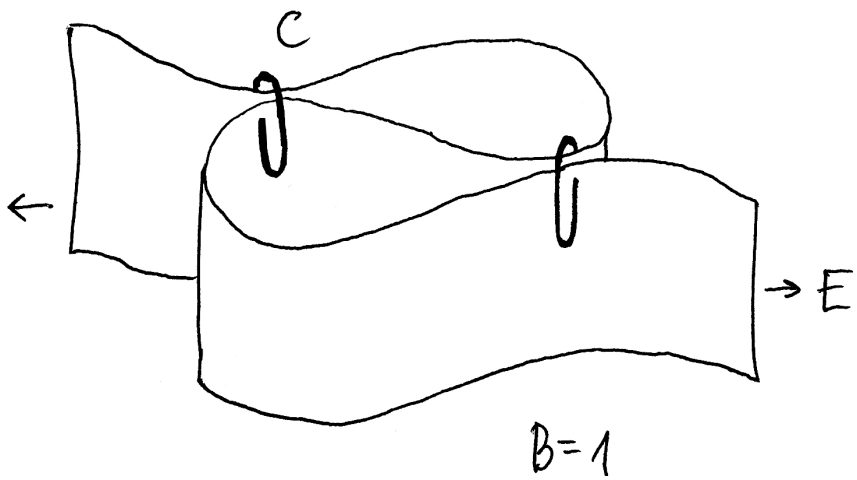
10 000e Doomino kalkulaator:
`https://www.youtube.com/watch?v=OpLU__bhu2w`

Kirjaklamberkalkulaator

Allolev pole seotud transistori analoogiaga.

Huvitav, kas eelmainitud mudel suudaks ühendatud kirjaklambreid *lahutada* – teostada algebrat?

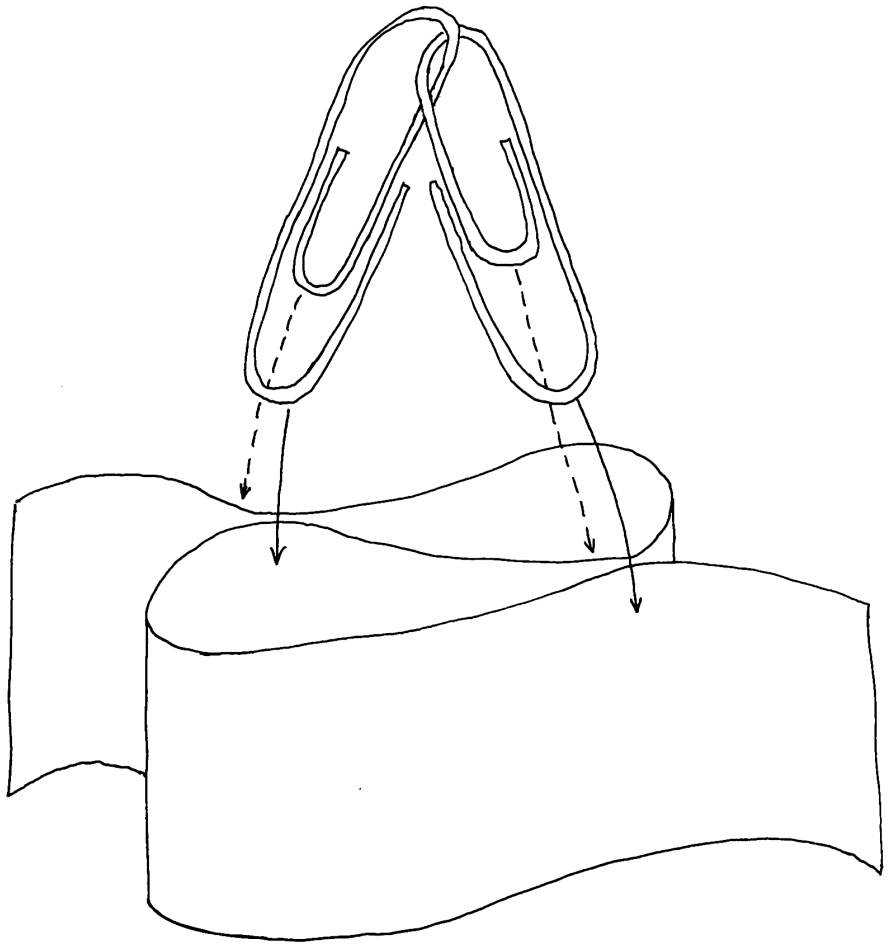
Me teame, et me saame neid liita (ühendada).



Joonis: Kirjaklamber – liidetav.

Kirjaklamberkalkulaator

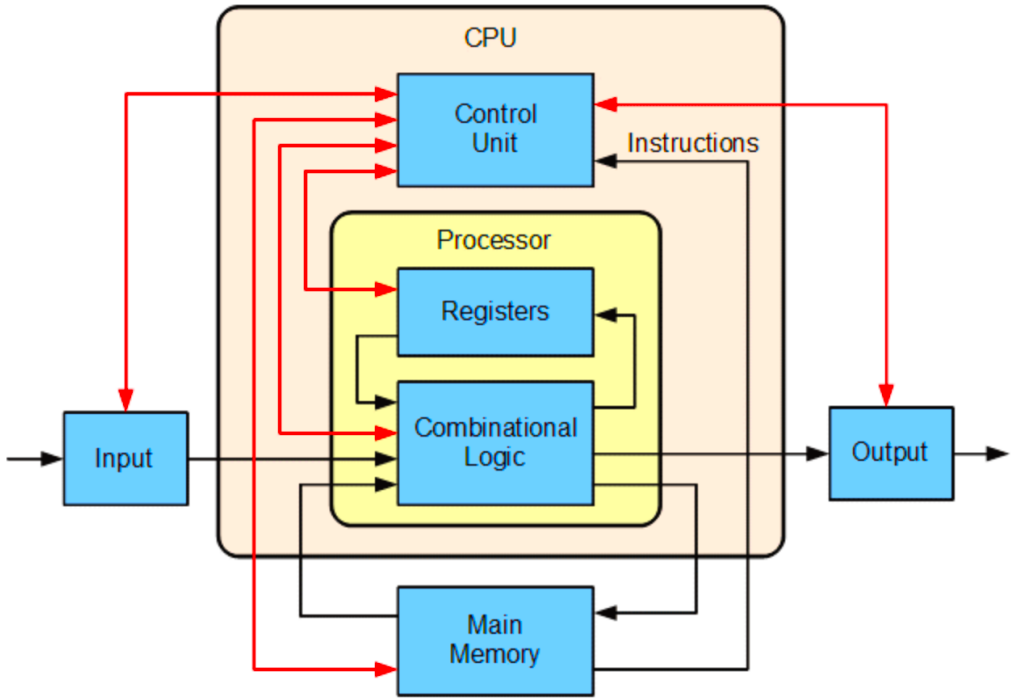
Huvitav, kas eelmainitud mudel suudaks ühendatud kirjaklambreid *lahutada*?



Joonis: Kirjaklambrite lahutamine.

Arvuti meie kursuses kontekstis

Arvuti ehk **kompuuter** ehk **raal** on *elektroonika seade*, mis koosneb kolmest põhikomponendist: protsessorist, mälust ja sisend-väljund seadmetest.



Joonis: von Neumann'i arhitektuuriga personaalarvuti.

Programmeerimisest üldiselt: mõisteid

Programmeerimine on inimese tegevus eesmärgiga panna arvuti tegema seda, mida inimene tahab. Programmeerimise tulemuseks on programm või tarkvara, mis kontrollib arvuti tegevust. Arvuti tegevus koosneb informatsiooni töötlusest, milleks on nt. andmete hankimine, teisendamine ja edastamine.

Programm on juhiste jada, mida arvuti/protsessor hakkab programmi käivitamisel täitma. Protsessori tasemel täidetavad käsklused on suhteliselt lihtsad.

Programmeerimiskeel kompilaatorile või interpretaatorile mõistetav keel (süntaksi- ja semantikareeglite kogum) milles programmid on kirjutatud.

Kompilaator/interpretaator on programm mis tõlgib inimloetava programmi koodi arvutile mõistetavateks juhisteks (0-d ja 1-d).

Programmeerimisest üldiselt: ajalugu

IT ja programmeerimiskeelte arengulugu

- **Assemblerkeel:** raudvarast/arhitektuurist sõltuv (iga uus arvuti nõuab olemasolevate programmide ümberkirjutamist) madala taseme keel (sisuliselt kirjeldad tegelikku arvutis toimuvat).
  - *probleem: toimub pidev arvutite areng (efektiivsuse tõstmine)*
- **Kõrgema taseme (KT) keeled:** loodi **kompilaatorid** (igal arhitektuuril oma) mis tõlgib KT keele arvutile mõistetavaks. Tulemus: keele konstruktsioon ei sõltu (nii palju) arhitektuurist.
  - *probleem: programmid kasvavad keerukuses. Kompileerimine võtab liiga palju aega. Programme levitatakse kompileeritud kujul. Programm töötab vaid vastava arhitektuuriga arvutis.*
- **Veel kõrgema tasemega keeled:** millega kirjutatud programme on võimalik koheselt arvutis käivitada (ilma programmi kompileerimata) eeldades, et arvutisse on installeeritud vastava keele **interpretaator**.

Programmeerimisest üldiselt: ajalugu

Interpretaator koosneb sisuliselt ettekompileeritud programmijuppidest, mis vastavad programmeerimiskeele konstruktsioonidele. Seega kui interpretaator loeb programmi lähtekoodist keele lausendi, siis käivitatakse koheselt sellele lausendile vastav programmeerimiskeele konstruktsioon antud arvutis.

Selliseid programmeerimiskeeli, mille lausenditega juhitakse programmitööd, nimetatakse ka **skriptimiskeelteks**.

Python

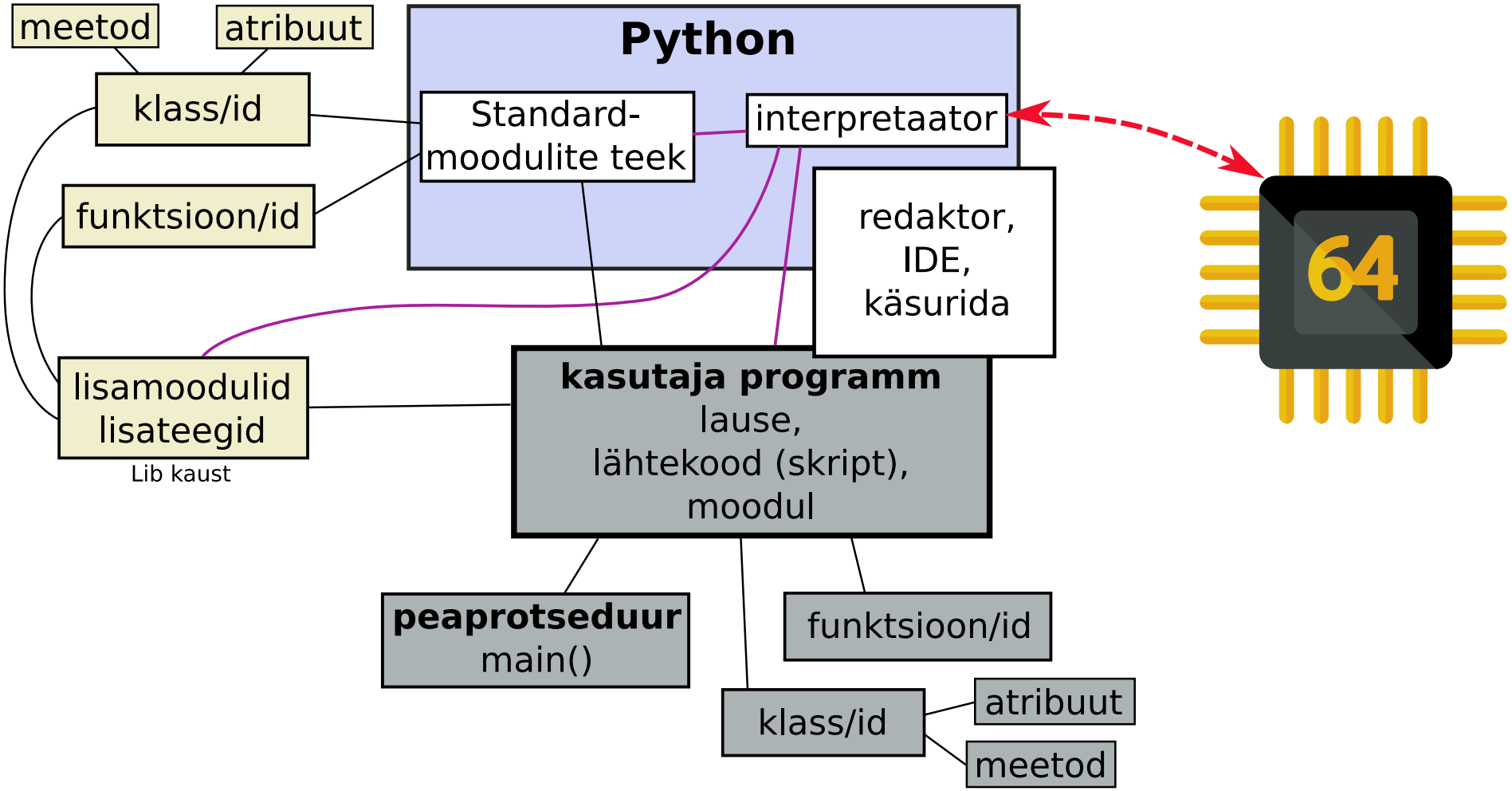
Python on *väga kõrge tasemega üldotstarbeline dünaamiline objektorienteeritud skriptimiskeel*. Python võimaldavad kasutada erinevaid programmeerimise paradigmasid nagu funktsionaalseid, protseduraalseid, ja objektorienteeritud programmeerimise võtteid.

Funktsionaalne programmeerimine: funktsiooni rakendamisel saadakse väärtus.

Protseduraalne programmeerimine: programmi poolt tehtavad arvutused on koondatud nn. protseduuri blokkidesse, mida saab taaskasutada.

Objektorienteeritud programmeerimine: programmi töös manipuleeritakse objektidega, mis on spetsiaalsed andmestruktuurid objekti andmetest (parameetrid, muutujad, atribuudid) ja objektiga seotud tegevustest (meetodid). Objekti võib käsitleda kui iseseisvat üksust, mis oskab andmeid vastu võtta, neid töödelda, salvestada ja väljastada.

Python



Joonis: Programmeerimine Pythonis. Kasutaja programm ja selle osad.

Miks just Python?

Maailmas on aktiivses kasutuses üle 500 programmeerimiskeele. Miks me valisime just Pythoni?[3]

- Üldkasutatav keel
- Inimloetav lähtekood
- Suhteliselt kergesti õpitav
- Palju arendajaid, palju mooduleid/teeke importimiseks
- Objektorientreeritus on võimas tööriist
- Laialt kasutuses teaduses

[3] Sobiva programmeerimiskeele valik on mittetriviaalne probleem.

Python, kus kasutatakse

Walt Disney Feature Animation – uses Python as a scripting language for animation. All the magic that happens in Disneyland has a bit of Python behind it.
Yahoo! Maps – uses Python in many of its mapping lookup services & addresses.
Instagram – moved to Python 3 is just great example of a gigantic tech company using python in combination with Django. Instagram has about 400M of daily active users who share more than 95M photos & videos. Instagram choose Python because of its simplicity & popularity.
Spotify – trust Python & applies it in its backend service as well as Data Analysis purposes. The backend of a Spotify consist of a Plethora of separate services, connected by means of the messaging protocol developed in house. 80% of the services are based on Python while remaining 20% on languages like Java and C or C++.
Amazon – uses Python because of its popular, scalable & appropriate for dealing with Big Data that's the big plus for the kind of solutions Amazon strived to create.
Survey Monkey – use Python were its simplicity, tons of libraries allowing to build Web Apps faster, as well as facilitating working with deployment, Unit Testing, etc.
Facebook – decided to use Python as the core language for the backend of their applications connected with image processing.
Google – is one of the most popular search engine in the world, has been built using Python. It allows google to switch the traffic & figure out the requirements of search.
YouTube - Python has been the driving force behind YouTube, used by millions for downloading & uploading videos of all hues and sizes. YouTube has been coded in a way which makes it easier & extremely interactive for the user.
Quora – is a portal where you get your answers. Quora's language programming has been developed using Python's framework.
Dropbox – Many of our choices to store our data are going online. We create a document, we save it & we share it. It is the ideal way to preserve your documents online. This file hosting has been created by using Python.
Reddit – It is a place where you can find a lot of information & entertainment across thousands of categories. Popularly called internet's front page has been developed by using Python.
Bitly – The popular link management platform created by Peter Stern in 2008 shortens close to 600 million links annually. This website also owes greatly to Python as it came into existence because of Python.
IBM – uses Python for its factory tool control applications.
Nokia – makes uses of Python for S60 and also Python for Maemo for its S60 & Maemo software platforms.
NASA – uses Workflow Automation System, an application written & developed in Python. It also uses Python for Astronomy Picture of the Day, API, PyMDP Toolbox, Everest, PyTransit.

Pythoni installatsioon

Klassiruumis kasutame Anaconda distributsiooni:
`https://www.anaconda.com`

Pythoni ja Pythoni teekide haldamise (`pip` ja `conda`) seletame lahti kursuse jooksul.

Programmeerimise õpetamisest ja õppimisest

"People confuse programming with coding. Coding is to programming what typing is to writing. Writing is something that involves mental effort, you're thinking about what you're going to say the words have some importance but in some sense even they are secondary to the ideas...
If people are trying to learn programming by being taught to code well, they're being taught writing by being taught how to type and that doesn't make much sense." — Leslie Lamport[4]

Iga programmi taga on mingi **idee**, inspiratsioon, loominguline otsus, probleemi lahendus, algoritm, jne.

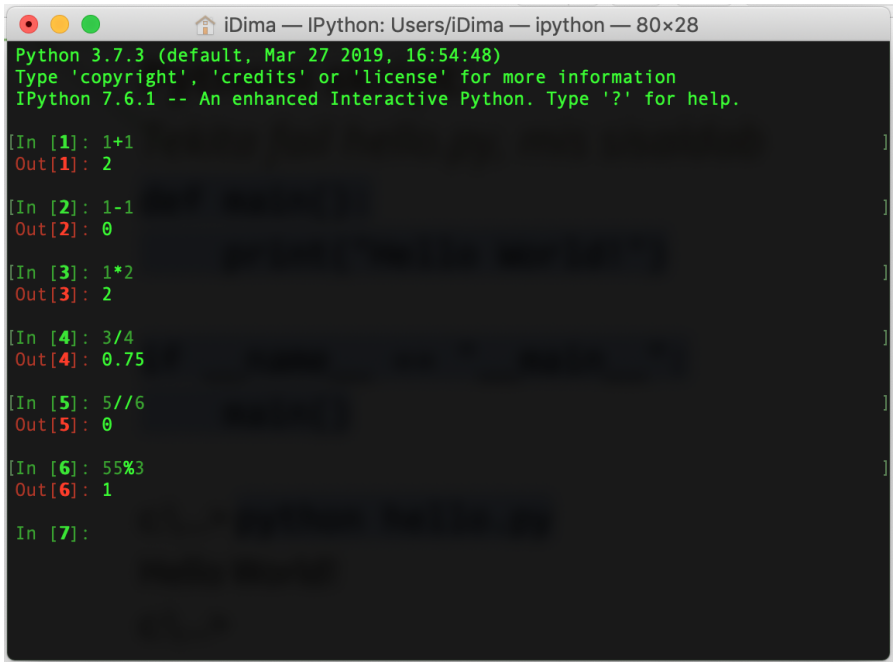
Programmid on ehitatud ideedele!

[4] Allikas: `https://www.youtube.com/watch?v=rkZzg7Vowao`

Pythoni interpretaatorist

Kuskilt peab alustama: IDLE, konsool, Jupyter.

Sisestame konsooli (Windows: *Anaconda Prompt*, MacOS, UNIX, Linux: *Terminal*) `python` või `ipython`.



"Arvuti" viitab mõistele "arvutama". Vaatame kas Python oskab arvutada. Sisesta:

$$2+2, \quad 2-2, \quad 2/2, \quad \text{jne.}$$

Pythoni interpretaatorist

Kursuse koduleht või TalTech Moodle keskkond:

- Avame `Naited_L1.html` faili (neti brauser/veebilehitseja)
- Avame `Naited_L1.ipynb` faili (ava Jupyter ja lae fail)
- Avame Jupyteri ja loome uue `.ipynb` (Jupyteri tööleht) faili ning hakkame kodeerima....

`.ipynb` fail on Jupyteri töölehe fail. Jyputer on IDE tarkvara, sisuliselt on tegemist konsooliga millele on lisatud rohkem funktsionaalsust.

Praktikum

Kursuse koduleht või TalTech Moodle keskkond:

- Avame Jupyteri ja lae fail `Praktikum_L1.ipynb`

Fail sisaldab ülesandeid.

Kokkuvõte

- *Kursusest praktiliselt*

- Sissejuhatus
  - Arvuti
  - Programmeerimisest üldiselt
  - Programmeerimise ajaloost
- Python
  - Pythonist ja tema olemusest
  - Miks just Python?
  - Pythoni installeerimine ja ülesseadmine
  - Pythoni interpretaatoriga suhtlemine, esimene kokkupuude Pythoniga
  - Lihtne aritmeetika
  - Sõne (string), `print` funktsioon
- Praktikum