

## 1. nedēļa — Pašrefleksijas jautājumi

Atbildi uz šiem jautājumiem saviem vārdiem. Mērķis nav “pareiza atbilde” — mērķis ir pārliecināties, ka tu saproti, ko esi iemācījies. Ja kādā punktā jūties nedrošs — tas ir signāls, ka jāpiestrādā.

**Iesniedz šo dokumentu kopā ar pārējo mājasdarbu.**

---

### Koncepti

1. Kas ir programma? Kā tu to izskaidrotu cilvēkam, kurš nekad nav programmējis?  
*Vieta, kur ir janorada precīzas darbības ko darīt, piem., ka receptes.*
2. Kas ir algoritms? Kā tas atšķiras no vienkāršas instrukcijas?  
*Instrukcija – viens solis. Algoritms – vairaki soļi.*
3. Kādas trīs īpašības padara algoritmu derīgu (valid)? Pastāsti katru saviem vārdiem un dod piemēru, kas šo īpašību pārkāpj.  
*Precīzs – katram solim ir jābūt skaidram.  
Tam ir jābeidzas – tas nevar būt bezgalīgs.  
Izpildāms – tehniski izpildāms.*
4. Ko nozīmē “secīga izpilde” (sequential execution)? Kāpēc datoram ir svarīgi, kādā secībā soļi tiek izpildīti?  
*Jo tas izpilda komandas secībā kādā es tās esmu norādījusi. Ja sajauc secību, mainas rezultāts.*
5. Izskaidro modeli “ievade → apstrāde → izvade” ar vienu konkrētu piemēru no ikdienas vai no sava mājasdarba.  
*Ievade – ievadu datus.  
Apstrāde – apstrādāju, ko e vēlos ar tiem datiem izveidot.  
Izvade – izprintēju rezultātu.*
6. Kas ir interpretators (interpreter) un ko tas dara ar tavu Python kodu?  
*Nolasa Python kodu un izpilda to rindu pa rindai. Python ir interpretējama valoda — kods netiek iepriekš pilnībā pārvērsts mašīnkodā.*
  - Pārbauda sintaksi.
  - Pārvērš kodu darbībā.
  - Uzreiz izpilda.
7. Kāda ir procesora (CPU) un atmiņas (memory) loma programmas izpildē? Izskaidro vienkāršoti, saviem vārdiem.  
*CPU – “datora smadzenes” – veic aprēķinus, izpilda komandas, pieņem lēmumus (if). (pavārs)  
RAM (atmiņa) – glabājas mainīgie, tiek turēti dati programmas darbības laikā. (darba galds ar sastāvdaļām)*

8. Kas notiek, ja tavā kodā ir sintakses kļūda? Kāpēc programma netiek izpildīta pat daļēji?  
[Python apstājas. Programma netiek izpildīta.](#)  
[Interpretators pārbauda vai kods ir uzrakstīts pareizi, j atrod kļūdu, tas neuzdrosinās turpināt.](#)
- 

## Praktiskie darbi

9. Vai Python ir veiksmīgi uzstādīts tavā datorā? Kādu Python versiju parāda `python -version`?  
[Python 3.14.2](#)
10. Vai tu spēj palaist kodu no termināļa vai IDE? Apraksti, kā tu to dari — kādus soļus veic.  
[Izmantoju VSCode, Colab, Anaconda - JupiterLab](#)
11. Vai tu spēj izveidot .py failu un to izpildīt? Kas notiek, ja mēģini palaist failu, kura nav?  
[Maku izveidot failu.](#)
12. Vai tu uzrakstīji un palaidi savu "Hello, world!" programmu? Kas tieši parādījās ekrānā?
- ```
print ("Hello Worl")
```
- 
- ```
Hello Worl
```
- Jā.
13. Vai tu veiksmīgi nomainīji izvades tekstu? Ko tu mainīji un kāds bija rezultāts?  
[Mēģināju ievadīt sviesmaizes pagatavošanas komandas.](#)
- 

## Refleksija

14. Kāpēc programmētājam ir jābūt precīzam katrā instrukcijā? Ko dators nevar "uzminēt"?  
[Dators nedomā, nesaprot kontekstu, neuzmin nozīmi.](#)
15. Kāpēc precizitāte ir svarīga programmās? Apraksti vienu situāciju, kur neliela neprecizitāte varētu radīt lielu problēmu.  
[Programma izpilda precīzas instrukcijas.](#)  
[Nepareiza datu ievade, piem., uzņēmējdarbībā](#)