



Rad s datotekama

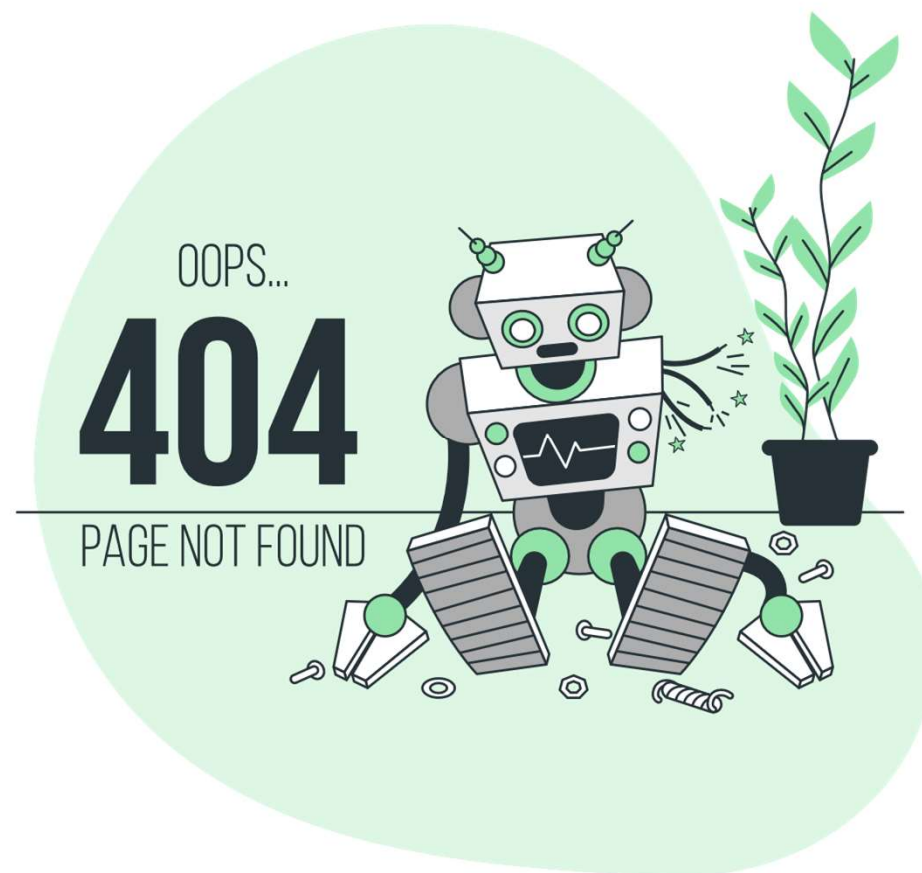
Datoteke

- Python, bez potrebe za instalacijom modula, podržava rad s najvažnijim formatima datoteka
- Za napredni rad sa specifičnim tipovima datoteka (fotografije, video ...), potrebno je instalirati specijalizirane, besplatne module.
- Prvo ćemo se upoznati s uporabom tekstualnih datoteka (.txt, .json, .log, .ini, ...)
- Kada to svladamo, krenut ćemo na foto datoteke i njihovu obradu te načine kako prepoznati lice jedne ili više osoba na fotografiji.



Dostupnost datoteke, baze, web ... ?

- Dok pišemo kôd koji koristi "vanjske" resurse (resurse koji nisu izravno u našem kôdu), NE smijemo smatrati da ćemo uvijek imati pristup tim resursima
- Možda na računalu na kojem se izvršava naš kôd nemamo prava pristupa željenoj datoteci, možda te datoteke nema ili možda nemamo prava kreiranja datoteka
- Možda računalu na kojem se izvršava naš kôd nema sljedeće:
 - mrežni pristup do servera na kojem je datoteka koju koristimo ili baza podataka
 - pristup Internetu kako bi pristupili resursima u Cloudu



try: - except: - ~~else:~~ - finally:

- **try: – except: – ~~else:~~ – finally:**
- Desno je prikazan puni blok koji koristimo kada želimo pristupiti "vanjskom" resursu iz našeg kôda.
- **Try:** predstavlja blok kôda koji želimo izvršiti
- **Except:** predstavlja blok kôda koji će se izvršiti ukoliko se dogodi neka iznimka kao npr. nemamo pravo pristup, nema konekcije na mrežu. Može biti više *except* blokova.
- **Else:** ima gotovo istu namjenu kao i Try pa se zato najčešće izostavlja
- **Finally:** je dio kôda koji će se gotovo uvijek izvršiti bez obzira je li se dogodila greška ili je blok uspješno izvršen. Najčešće se dodaje kôd, koji će otpustiti zauzete "vanjske" resurse.
- **try:**
 - Blok naredbi
- **except Exception1 as ex1:**
 - Blok naredbi
- **except Exception2 as ex2:**
 - Blok naredbi
- ...
- **else:**
 - Blok naredbi
- **finally:**
 - Blok naredbi

Tekstualne datoteke – čitanje i pisanje

Tri jednostavna koraka:

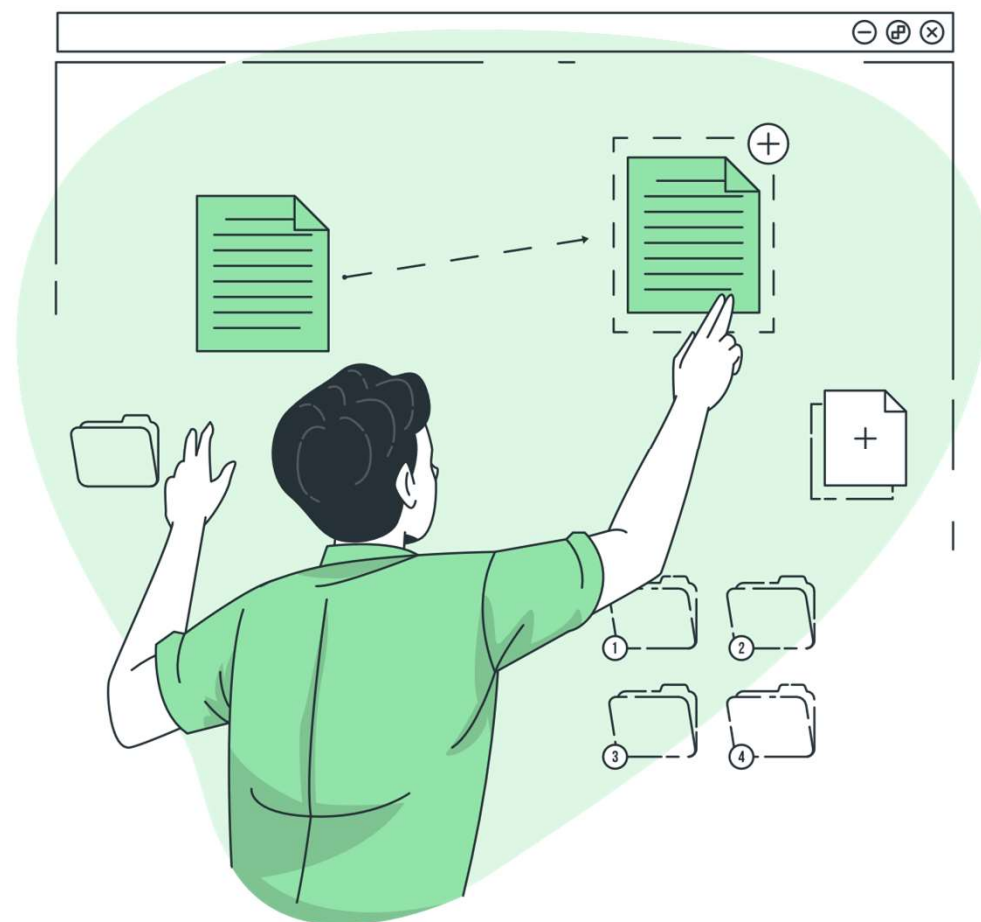
1. Otvori datoteku za čitanje / pisanje
 - Ovaj korak "zaključava", odnosno onemogućava drugima pristup datoteci
 - Kao da smo otvorili konekciju pomoću koje smo se povezali na datoteku
 2. Pročitaj sadržaj / zapiši tekst
 3. Zatvori datoteku – NE ZABORAVITI!
 - Ovaj korak oslobađa drugim aplikacijama pristup datoteci
 - Kao da smo zatvorili konekciju / vezu prema datoteci
- SAVJET: Otvoriti datoteku što "kasnije" (neposredno prije korištenja) i zatvoriti najranije moguće (odmah nakon što više nema potrebe za pristup datoteci).

with open('naziv_datoteke') as naziv

- Sintaksa **with open('naziv_datoteke', '[r, w, a]') as naziv**: se koristi kako bi se konekcija prema datoteci nakon kraja korištenja, automatski zatvorila. Ovo je preporučeni način pristupa datotekama.
 - Blok **'[r, w, a]'** označava najčešće korištene tipove pristupa datoteci.
 - **r** – read – čitanje
 - **w** – write – pisanje
 - **a** – append – dodavanje na kraj
- Metoda **rstrip()**, koju ima svaki tekstualni tip podatka, se koristi kako bi se uklonio znak **'\n'** na kraju teksta.

FileManager Paket

- Python Paket koji nudi mogućnost:
 - Provjere postoji li datoteka ili neka putanja do datoteke
 - Kreiranje/otvaranje konekcije prema datoteci
 - Zapisa podataka u datoteku
 - Dodavanja podataka u već postojeću datoteku
 - Čitanje podataka iz datoteke
 - Zatvaranje konekcije prema datoteci



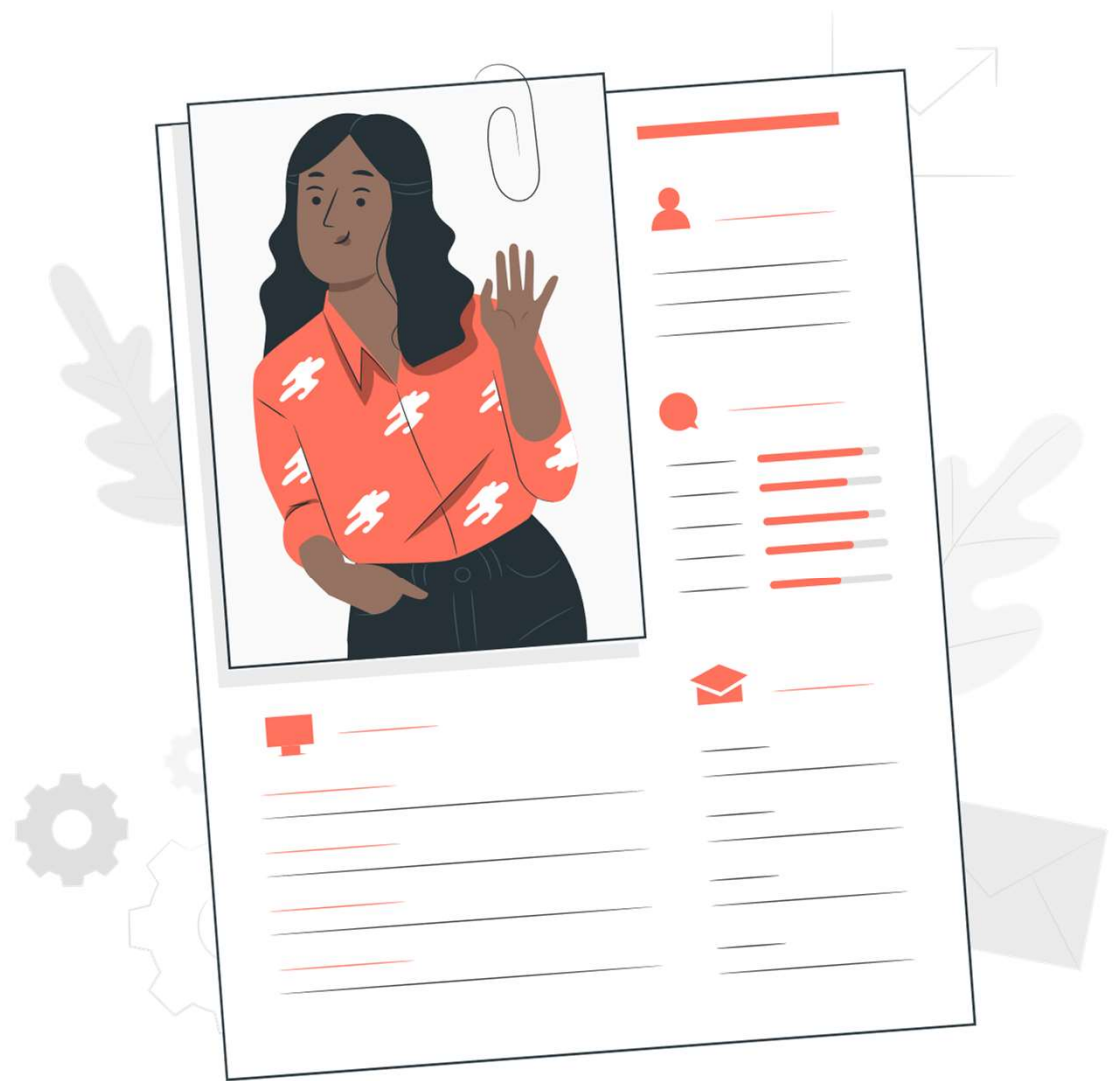
Contact Manager

Kreirajte novi folder ContactManager i otvorite ga unutar VS Code kao novi projekt.

Napravite jednostavnu aplikaciju **Contact Manager** koja vodi evidenciju o podacima **Tvrtke (Organization)**, svim **Kupcima (Customers)** te svim **Djelatnicima (Employees)**.

Aplikaciju napravite tako da svaka komponenta ima svoj Paket sa svim pripadajućim modulima.

Podaci su pohranjeni u tekstualne datoteke. Za rad s datotekama koristite zaseban Paket **FileManager**.



JSON – JavaScript Object Notation

- Standardni format razmjene podataka na Internetu
- Tekstualni tip datoteke
- Nekada se koristio kompliciraniji XML kojeg je JSON zamijenio
- Format jednako čitljiv ljudima kao i računalima.
- Python ima podršku za rad s datotekama u json formatu
<https://docs.python.org/3/library/json.html>
- Format zapisa sličan kao u rječniku (Dictionary - key:value)

```
{  
  "ime" : "Josip",  
  "prezime" : "Jelačić Bužimski",  
  "godinaRodenja" : 1801,  
  "jeziciKojeGovori" :  
  [  
    "hrvatski",  
    "njemački",  
    "francuski",  
    "mađarski",  
    "latinski"  
  ]  
}
```

JSON u Pythonu

- Importiranje modula json
 - `import json`
- Učitavanje sadržaja datoteke u json formatu
 - `varijabla1 = json.load(fileReader)` ili
 - `varijabla2 = json.load(varijabla1)`
- Kreiranje tekst varijable s json formatiranim sadržajem
 - `varijabla = json.dump(data_dictionary, fileWriter, indent=4)`
 - `varijabla = json.dumps(data_dictionary, indent=4)`
- Zapisivanje u datoteku – jednako kao i u prethodnim primjerima:
 - Kreiranje i formatiranje sadržaja pomoću `json.dumps()` metode
 - Pomoću `with open('naziv_datoteke', 'w') as fileWriter:` te `json.dump()` metode zapisati u datoteku

Address Book

Stranica

<https://jsonplaceholder.typicode.com>

nudi besplatni pristup do web resursa koje ćemo koristiti u nekoliko primjera.

<https://jsonplaceholder.typicode.com/users>

je link s kojeg možete preuzeti popis korisnika prikazanih u json formatu.

Potrebno je izraditi aplikaciju koja će koristiti FileManager paket za rad s datotekama te JsonManager za rad s JSON formatiranim tekstom.



Contact Manager

Prethodno kreiranu aplikaciju **Contact Manager** prepravite tako da se podaci čuvaju u JSON formatu.

Osim podataka o **Tvrтки (Organization)**, svim **Kupcima (Customers)**, svim **Djelatnicima (Employees)**

Proširite aplikaciju tako što ćete omogućiti čuvanje podataka i o **Proizvodima (Products)** tvrtke.

Strukturu aplikacije zadržite kao što je i bila: svaka komponenta ima svoj Paket sa svim pripadajućim modulima.

I dalje za rad s datotekama koristite Paket **FileManager**, a za rad s json formatom koristite novi **JsonManager** Paket.

