

8 Práce se soubory v programování

Soubor v informatice označuje pojmenovanou sadu dat uloženou na nějakém datovém médiu, se kterou lze pracovat nástroji operačního systému jako s jedním celkem. Může obsahovat různá data vytvářející celek (textový dokument, obrázek,...) nebo se může jednat o soubor složené (Kancelářské dokumenty, ISO obrazy disků...)

Pro naše potřeby postačí soubory .txt .csv a .json. Jedná se o samostatné soubory, které slouží k uchování dat.

- .txt -> textová data (poznámky z hodiny, ...)
- .csv -> textová data oddělená znakem (, nebo |...) většinou ve formě tabulek (Využívá se pro ukládání dat ve formátu tabulek, aby byla čitelná pro velké množství programů (textové editory, tabulkové kalkulátory...))
- .json -> JavaScript Object Notation - využívá se pro ukládání dat ve formátu "klíč": "hodnota", často využíván při komunikaci mezi programy(servery) pro svoji jednoduchost a čitelnost (i pro programátora)

Operace se soubory (v pythonu)

- Čtení ('r') -> Slouží k pouhému přečtení dat v souboru (nelze jej upravit)
- Zápis ('w') -> Slouží k zápisu do souboru. Pokud soubor neexistuje, tak jej vytvoří. Pokud v sobě již má nějaká data, tak je přepíše.
- Připsání ('a') -> Slouží k přidání dat do souboru (narozdíl od 'w' jej nepřepíše). Pokud soubor neexistuje, tak jej vytvoří

Postup u .csv a .json souborů je stejný, pouze využíváme knihoven "csv" a "json"

- [csv](#)
- [json](#)

```
# Čtení

# test.txt:
# Hello World

with open("test.txt", "r") as file:
    read_content = file.read()
    print(read_content)

# output:
# Hello World
```

```
# Zápis

# test.txt:
# Hello World

with open("test.txt", "w") as file:
    file.write('Goodbye World')

# test.txt:
# Goodbye World
```

```
# Připsání

# test.txt:
# Hello World

with open("test.txt", "a") as file:
    file.write('\nGoodbye World')

# test.txt:
# Hello World
# Goodbye World
```

Ukázka:

Program na čištění zbytečných informací z logů vygenerovaných ROBOCOPY (využívám při kopírování záloh na jiný disk)

```
import re

cleaned_logs_list = []

# Zjednodušení práce s regexem
def regex(regex_code, text):
    found = re.search(regex_code, text)
    if not found == None:
        return True

# Načtení logu
log_file_obj = open("log.txt", mode='r')
logs_list = log_file_obj.read().split('\n')

# Vytřídění logu
for i in range (len(logs_list)):
    # Pokud neobsahuje text, tak to řádek přidá do listu vyčištěných log
    # zápisů
    if regex('Newer|New File|New Dir|EXTRA Dir|EXTRA File|Started :|Source
:|Dest =|Files :|Options :|Total    Copied    Skipped    Mismatch    FAILED
Extras|Dirs :|Files :|Bytes :|Times :|Ended :|-----|ROBOCOPY      ::
Robust File Copy for Windows',logs_list[i]) == True:
        cleaned_logs_list.append(logs_list[i])

log_file_obj.close()

# Přepsání logu
log_file_obj = open("log.txt", mode='w')

for i in range (len(cleaned_logs_list)):
    log_file_obj.write(cleaned_logs_list[i] + "\n")

log_file_obj.close()
```