

Textové súbory

- „zásobáreň“ dát – životnosť dát
- Viditeľné a neviditeľné znaky
- 3 módy práce so súborom:
 1. Otvorenie
 2. Spracovanie
 3. Zatvorenie – dôležité!!!

1. Otvorenie súboru

```
premenná = open('meno_súboru', 'r')
```

absolútnej a relatívnej ceste, parameter – r, r+, w, a

```
subor = open('pribeh.txt', 'r')
kniha = open('c:/dokumenty/python.txt', 'r')
subor_s_cislami = open('../texty/cisla.txt', 'r')
```

2. Zatvorenie súboru

Ked' skončíme prácu so súborom, **uzavrieme** otvorené spojenie volaním metódy:

```
subor.close()
```

3. Práca so súborom – čítanie

*funkcia **read(line(s))**
read() načítať celý súbor do jedného reťazca,
readline() načítať ďalší riadok do reťazca,
readlines() načítať všetky riadky do listu reťazcov,*

```
riadok = subor.readline()
```

Funkcia vráti znakový reťazec - prečítaný riadok aj s koncovým znakom '**\n**'. Súborová premenná si zároveň zapamätá, kde v súbore sa práve nachádza toto čítanie, aby každé ďalšie zavolanie **readline()** čítaло ďalšie a ďalšie riadky.

Funkcia vráti prázdný reťazec '**'**', ak sa už prečítali všetky riadky a teda pozícia čítania v súbore je na konci súboru.

```
subor = open('subor.txt', 'r')
print(subor.readline())
print('dlzka =', len(subor.readline()))
print(subor.readline()[:-1])
```

Predpokladajme, že máme pripravený nejaký textový súbor, napríklad 'subor.txt':

```
Od učenia este  
nikto nezomrel,  
ale naco riskovat.
```

Albert Einstein

Tento súbor má 5 riadkov (štvrty je prázdny) a preto ho môžeme celý prečítať a vypísať takto:

```
t = open('subor.txt', 'r')

for i in range(5):
    riadok = t.readline()
    print(riadok)

t.close()
```

program vypíše:

```
Od učenia este  
nikto nezomrel,  
ale naco riskovat.
```

Albert Einstein

Kedže metóda `readline()` prečíta zo súboru celý riadok aj s koncovým '`\n`', príkaz `print()` k tomu pridáva ešte jeden svoj '`\n`' a preto je za každým vypísaným riadkom ešte jeden prázdny. Budú príkazu `print()` povieme, aby na koniec riadka nevkladal prechod na nový riadok (napríklad `print(riadok, end='')`), alebo pred samotným výpisom z reťazca `riadok` vyhodíme posledný znak, napríklad:

```
t = open('subor.txt', 'r')    alebo      t = open('subor.txt', 'r')
for i in range(5):           for riadok in t:
    riadok = t.readline()   print(riadok[:-1])
    print(riadok[:-1])       t.close()
```

Takéto vyhadzovanie posledného znaku z reťazca môže nefungovať celkom správne pre posledný riadok súboru, ktorý nemusí byť ukončený znakom '`\n`'. Výhodné je použiť funkciu `strip`.

Zistenie konca súboru

Najväčším nedostatkom predchádzajúceho programu je to, že predpokladá veľkosť vstupného súboru presne 5 riadkov. Ak by sme tento počet dopredu nepoznali, musíme použiť nejaký iný spôsob. Kedže metóda `readline()` vráti na konci súboru **prázdny reťazec** `''` (pozor, nie jednoznakový reťazec `'\n'`), môžeme práve túto podmienku využiť na testovanie konca súboru:

```
t = open('subor.txt', 'r')
riadok = t.readline()
while riadok != '':
    print(riadok, end='')
    riadok = t.readline()
t.close()
```

Tento program už správne vypíše všetky riadky súboru, hoci nevidíme, či je štvrtý riadok prázdný alebo obsahuje aj nejaké medzery:

Od uenia este
nikto nezomrel,
ale naco riskovat.

Albert Einstein

šablóna čítania s while-cykлом

```
subor = open('meno súboru', 'r')

riadok = subor.readline()
while riadok != '':
    # ... spracuj riadok
    riadok = subor.readline()

subor.close()
```

šablóna čítania s for-cykлом

```
subor = open('meno súboru', 'r')

for riadok in subor:
    ... # spracuj riadok

subor.close()
```

4. Práca so súborom – zápis

```
subor = open('meno_súboru', 'w')
```

Možnosti, ako zapisovať riadky do súboru je viac. Zápis pomocou základnej metódy pre zápis `write()` a pomocou nám známej štandardnej funkcie `print()`.

```
subor.write(retezec)
```

```
subor = open('subor1.txt', 'w')
subor.write('zoznam prvocisel:\n')
for ix in 2, 3, 5, 7, 11, 13:
    subor.write(f'cislo {ix} je prvocislo\n')
subor.close()
```

Program najprv do súboru zapísal jeden riadok 'zoznam prvocisel:' a za ním ďalších 6 riadkov:

```
zoznam prvocisel:
cislo 2 je prvocislo
cislo 3 je prvocislo
cislo 5 je prvocislo
cislo 7 je prvocislo
cislo 11 je prvocislo
cislo 13 je prvocislo
```

```
import random

subor = open('nahodne_cisla.txt', 'w')
for i in range(100):
    print(random.randint(1, 100), file=subor)
subor.close()
```

Všimnite si nový parameter pri volaní funkcie `print()`, pomocou ktorého presmerujeme výstup do nášho súboru (tu musíme uviesť súborovú premennú už otvoreného súboru na zápis).

Ak by sme chceli, aby boli čísla v súbore nie v jednom stĺpci ale v jednom riadku oddelené medzerou, zapísali by sme:

```
import random

subor = open('nahodne_cisla.txt', 'w')
for i in range(100):
    print(random.randint(1, 100), end=' ', file=subor)
print(file=subor)
subor.close()
```

Konštrukcia with

Všeobecný tvar príkazu je:

```
with open(...) as premenna:  
    prikaz  
    prikaz  
    ...
```

Tento príkazovou konštrukciou sa otvorí požadovaný súbor a referencia na súbor sa priradí do premennej uvedenej za `as`. Ďalej sa vykonajú všetky príkazy v bloku a po ich skončení sa súbor **automaticky** zatvorí. Urobí sa skoro to isté, ako

```
premenna = open(...)  
prikaz  
prikaz  
...  
premenna.close()
```

5. Práca so súborom – pridanie riadkov

Zoznámime sa s ešte jednou voľbou, pri ktorej sa súbor otvorí na zápis, ale nezruší sa jeho pôvodný obsah. Namiesto toho sa nové riadky budú pridávať na koniec súboru. Napríklad, ak máme súbor `'subor3.txt'` s tromi riadkami:

```
first line  
second line  
end of file
```

môžeme do neho pripísať ďalšie riadky, napríklad takto: namiesto `'r'` a `'w'` pri otváraní súboru použijeme `'a'`, ktoré označuje anglické slovo **append**:

```
t = open('subor3.txt', 'a')  
t.write('pridany riadok na koniec\nna este jeden\n')  
t.close()
```

v súbore je teraz:

```
first line  
second line  
end of file  
pridany riadok na koniec  
a este jeden
```

Príklad:

Súbor (*cisla.txt*) obsahuje jeden riadok:

1 2 3 4 5 6 7 8 9 10

Úloha:

1. načítať čísla zo súboru
 2. premeniť ich na celé čísla
 3. vypočítať dvojnásobok každého
 4. vypísať na obrazovku
-

Riešenie v Pythone

```
with open("cisla.txt", "r") as f:  
    riadok = f.readline()      # načítame jeden riadok  
    hodnoty = riadok.split()   # rozdelíme podľa medzery  
    cisla = [int(x) for x in hodnoty] # prevedieme na int  
  
for c in cisla:  
    print(c * 2)
```