

## Textový súbor

Premenné doteraz používaných údajových typov ukladajú svoje hodnoty len do operačnej pamäte a preto, po ukončení behu pythonovského skriptu, sa údaje nenávratne strácajú. Údaje uložené na vonkajšom pamäťovom médiu (najčastejšie na pevnom disku) sú však kedykoľvek k dispozícii, t.j. môžeme ich po spustení nášho programu načítať z disku a spracovať alebo, najneskôr pred ukončením behu programu, uložiť na disk pre neskoršie využitie. Python umožňuje čítať údaje zo súboru na disku aj ich na disk zapisovať. Výmena dát medzi súborom na disku a programom sa môže realizovať vo viacerých formátoch (napr. binárnom, textovom). Výhodou použitia textového formátu, t.j. textových súborov, je jednoduchá editovateľnosť textových súborov užívateľom. Na vytvorenie, zobrazenie obsahu alebo editovanie stačí jednoduchý textový editor, napríklad v systéme Microsoft Windows aplikácia Poznámkový blok. Pre textový súbor je typické členenie na riadky, t.j. súbor obsahuje znak `\n`. Či už chceme čítať z textového súboru alebo do neho zapisovať, musíme najprv vytvoriť prepojenie medzi konkrétnym súborom na disku a programom; hovoríme, že súbor musíme otvoriť na čítanie alebo zápis, prípadne na zápis na koniec súboru. Vo všetkých situáciách používame funkciu `open()`, avšak s rôznymi parametrami.

## Čítanie zo súboru

začínáme príkazom `premenná = open(názov_súboru, režim)`

kde

<code>premenná</code>	sa najčastejšie označuje <code>f</code> (zo slova file),
<code>názov_súboru</code>	je reťazec - názov súboru na disku, môže obsahovať aj cestu k súboru; po stutení programu je súbor hľadaný v priečinku, kde je aj Python skript, pokiaľ tam súbor umiestníme, cestu nemusíme zadať; ak súbor umiestníme do iného priečinka a nevedieme k nemu cestu, nastane výnimka,
<code>režim</code>	znamená použitie znaku <code>"r"</code> (zo slova read), čím sme zvolili režim čítania zo súboru (je predvoleným režimom); pre zápis do súboru sa používa <code>"w"</code> (write) a pre zápis na koniec súboru <code>"a"</code> (append); pre čítanie aj zápis <code>"r+"</code> .

Funkcia `open` môže mať ďalšie parametre, napríklad v prípade problémov s kódovaním môžeme použiť parameter `encoding`.

Po otvorení súboru na čítanie v textovom režime môžeme pomocou funkcií:

<code>read()</code>	načítať celý súbor do jedného reťazca,
<code>readline()</code>	načítať ďalší riadok do reťazca,
<code>readlines()</code>	načítať všetky riadky do zoznamu reťazcov,

Všetko si to ozrejímame na konkrétnych úlohách.

Po ukončení práce so súborom ho treba zavrieť, t.j. zrušiť prepojenie medzi programom a súborom na disku. Zavretie súboru sa realizuje funkciou `close()`.

## Súbor 1

Čítanie zo súboru si začneme ozrejmovvať na spracovaní súboru `mena.txt`, ktorý sme vytvorili v Poznámkovom bloku (obrázok vpravo; príponu pridá aplikácia sama!) a umiestnili do priečinka so zdrojovým skriptom.

### Úloha F.1

Vytvorte program, ktorý zobrazí obsah textového súboru `mena.txt`.

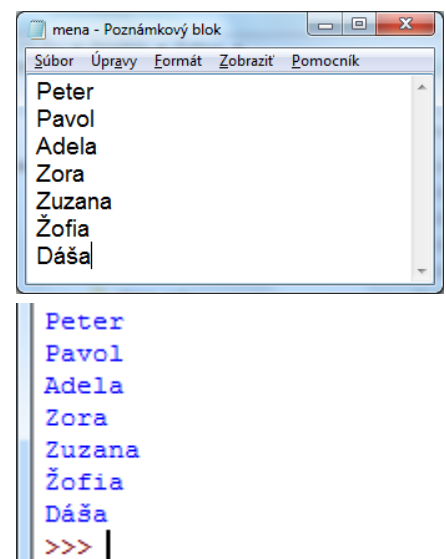
#### Riešenie 1.1

```
f = open("mena.txt", "r")
obsahSuboru = f.read()
f.close()
print(obsahSuboru)
```

Príkaz v prvom riadku otvorí súbor `mena.txt` na čítanie.

`obsahSuboru = f.read()` uloží celý obsah súboru do premennej `obsahSuboru` typu `str` vrátane znakov `\n`.

`f.close()` uzavrie súbor a `print(obsahSuboru)` ho zobrazí, ako to je na obrázku vpravo.



V prípade, že ste súbor `mena.txt` zabudli vytvoriť alebo neumiestnili k zdrojovému skriptu, program pri pokuse otvoriť súbor `mena.txt` havaruje (nastane neošetrená výnimka) a zobrazí sa chybové hlásenie

... `FileNotFoundError`: Žiadny takýto súbor alebo adresár: "mena.txt". Ošetrenie výnimky konštrukciou `try-except` je pomerne jednoduché. Chceme, aby sa beh programu, ak nastane výnimka, ukončil. Použijeme príkaz `exit` z knižnice `sys` (preto `import sys`):

```
import sys
try:
    f = open("mena.txt", "r")
except FileNotFoundError:
    print("Súbor nenájdený, ukončil som program!")
    sys.exit()
obsahSuboru = f.read()
f.close()
print(obsahSuboru)
```

V ďalších riešeniach predpokladáme, že pri otváraaní súboru nedôjde k výnimke a ošetrenie kvôli prehľadnosti riešenia vynecháme.

### Riešenie 1.2

Použijeme funkciu `readlines`. Keďže funkcia `readlines` automaticky načíta všetky riadky otvoreného súboru do zoznamu reťazcov, na vytvorenie zoznamu stačí jediný príkaz `zoznamRiadkov = f.readlines()`.

```
f = open("mena.txt") # otvorenie súboru na čítanie je predvoleným režimom - vynecháme "r"
zoznamRiadkov = f.readlines()
f.close()
print(zoznamRiadkov)
```

```
>>>
['Peter\n', 'Pavol\n', 'Adela\n', 'Žofia\n', 'Dáša\n', 'Zuzana\n', 'Zora']
>>>
```

Analýza tohto výstupu je veľmi dôležitá. Možno z neho vyčítať, že na konci každého riadka je zapísaný aj znak `\n` - prikazujúci prejsť na nový riadok. Tento znak nie je za posledným menom v súbore, pretože pri písaní súboru `mena.txt` sme za posledným riadkom nestlačili kláves `Enter`. Dosiahnúť „krajší“ výpis prvkov zoznamu možno použitím príkazu `for riadok in zoznamRiadkov: print(riadok)` - pozri Výpis 1. Zrejme nám budú vadit prázdne riadky, ktorých sa môžeme zbaviť viacerými spôsobmi. Buď ovplyvníme len výpis použitím parametra `end=""` v príkaze `print`: `for riadok in zoznamRiadkov: print(riadok, end="")` alebo pred vypísaním z konca každého riadka dáme odstrániť znak `\n` (pozri funkciu `strip`): `for riadok in zoznamRiadkov: print(riadok.rstrip())`. Oba prípady vedú k Výpisu 2.

Výpis 1	Výpis 2	Výpis 3	Výpis 4
<pre>&gt;&gt;&gt; Peter Pavol Adela Žofia Dáša Zuzana Zora &gt;&gt;&gt;</pre>	<pre>&gt;&gt;&gt; Peter Pavol Adela Žofia Dáša Zuzana Zora &gt;&gt;&gt;</pre>	<pre>&gt;&gt;&gt; Peter Pavol Adela Žofia Dáša Zuzana Zor &gt;&gt;&gt;</pre>	<pre>&gt;&gt;&gt; Adela Dáša Pavol Peter Zora Zuzana Žofia &gt;&gt;&gt;</pre>

Požiadavka zbaviť sa znaku `\n` môže viesť aj k myšlienke odrezat' znak `\n` na konci každého riadka, t.j. použit' napríklad príkaz `for riadok in zoznamRiadkov: print(riadok[:-1])`, čo ale spôsobí aj nežiaduce odrezanie posledného znaku v poslednom mene - pozri Výpis 3. **Vhodnejšie aj univerzálnejšie je použiť funkciu `rstrip`, ktorá vyrieši problém aj s prípadnými bielymi znakmi vloženými za posledný platný údaj, najčastejšie stlačenie klávesu `Enter` - vložený prázdny riadok na konci súboru.**

Použitie funkcie `readlines()` má aj tú výhodu, že môžeme ľahko doceliť utriedenie reťazcov v zozname použitím funkcie `sort()` (príkaz `zoznamRiadkov.sort()`), ako to vidieť vo Výpise 4.

**Riešenie 1.3a**

Ďalšou alternatívou je použiť funkciu `readline`, teda čítať súbor po riadkoch. Riešenie 3a využíva príkaz `for`:

```
f = open("mena.txt", "r")
for riadok in f:
    print(riadok.rstrip())
f.close()
```

Toto riešenie nám dá Výpis 2.

**Riešenie 1.3b**

Riešenie 3b vychádza z úvahy, že koniec súboru možno interpretovať ako neexistenciu ďalšieho riadka, t.j. „načíta sa“ prázdny riadok resp. vráti hodnota `None` (vyhodnotená ako `False`). Riešenie 3b využíva príkaz `while`:

```
f = open("mena.txt", "r")
riadok = f.readline()
while riadok != "":
    print(riadok.rstrip())
    riadok = f.readline()
f.close()
# stačí aj while riadok:
```

Všetky riešenia otestujte aj na prázdny súbor, súbor obsahujúci len jeden riadok, súbor obsahujúci aj prázdny riadok a súbor, v ktorom je po poslednom neprázdnom riadku ešte stlačený kláves `Enter` prípadne vložených viac bielych znakov.

Postúpme vo výklade na vyššiu úroveň.

**Súbor 2**

Textový súbor `mena.txt` doplňte o riadky so študijným priemerom pod každým menom a uložte pod názvom `priemery.txt` (obrázok vpravo).

Pokiaľ môžeme rozhodnúť o tom, ako budú údaje v textovom súbore organizované, je veľmi praktické, keď údaj, ktorý reprezentuje nejaký text (meno, adresu,...) je na samostatnom riadku. Číselných údajov môže byť aj viacej v jednom riadku, dajú sa ľahko vyselektovať (väčšinou bývajú jednotlivé čísla oddelené medzerou) najčastejšie funkciou `split`. Jednotlivým prípadom sa teraz budeme venovať.

**Úloha F.2**

Vytvorte program, ktorý zobrazí obsah textového súboru `priemery.txt`.

**Riešenie 2.1**

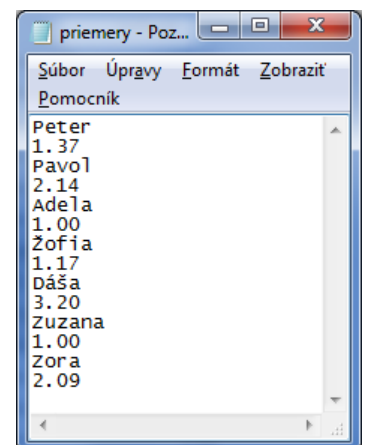
Riešenie je opakovaním riešenia 1.1 z predchádzajúcej úlohy, takže bez komentára:

```
import sys
try:
    f = open("priemery.txt", "r")
except FileNotFoundError:
    print("Súbor nenájdený, ukončil som program!")
    sys.exit()
obsahSuboru = f.read()
f.close()
print(obsahSuboru)
```

Úlohu si sťažíme a budeme požadovať, aby vo výpise bolo v jednom riadku meno aj k nemu prislúchajúci priemer (obrázok vpravo).

**Riešenie 2.2a**

Zrejme musíme čítať súbor po riadkoch. Treba si tiež uvedomiť usporiadanie dát v súbore a pri čítaní dát zo súboru to využiť. V našom prípade sa striedajú riadky s menom a priemerom. Je viacej možností, ako zobrazí načítané dáta. Nám sa zdá najpraktickejšie použiť `while`-cyklus a hneď po načítaní reťazca priemer využiť konverziu na `float` (zbavíme sa tým problému so znakom `\n`).



```
>>>
Peter      1.37
Pavol      2.14
Adela      1.00
Zofia      1.17
Dáša       3.20
Zuzana     1.00
Zora       2.09
>>>
```

```
import sys
try:
    f = open("priemery.txt", "r")
except FileNotFoundError:
    print("Súbor nenájdený, ukončil som program!")
    sys.exit()

meno = f.readline()
while meno != "":
    # stačí aj while meno:
    meno = meno.strip()
    priemer = float(f.readline())
    print("{meno:10} {priemer:.2f}".format(**locals()))
    meno = f.readline()
f.close()
```

### Riešenie 2.2b

„Násilné“ použitie for-cyklu. Keďže každým prechodom for-cyklom sa načíta nový riadok, či ide o nepárny obsahujúci meno alebo párny obsahujúci priemer, musíme „ustrážiť“ my. Použijeme premenú `neparny_riadok`, ktorá bude striedať hodnoty `True` (pravda) a `False` (nepravda). Počiatočná hodnota je `True` (riadok č.1 je nepárny) a po načítaní riadka sa má hodnota zmeniť na opačnú (z `True` na `False` alebo z `False` na `True`), čo zabezpečí negácia aktuálnej hodnoty, teda príkaz `neparny_riadok = not(neparny_riadok)`. Použiť by sa mohla aj číselná premenná.

```
try:
    f = open("priemery.txt", "r")
except FileNotFoundError:
    print("Súbor nenájdený, ukončil som program!")
    sys.exit()

neparny_riadok = True
for riadok in f:
    if neparny_riadok:
        meno = riadok.strip()
    else:
        priemer = float(riadok)
        print("{meno:10} {priemer:.2f}".format(**locals()))
        neparny_riadok = not(neparny_riadok)
f.close()
```

### Súbor 3

Textový súbor `mena.txt` doplňte o riadky so známami jednotlivých študentov a uložte pod názvom `znamky.txt` (v súbore sa striedajú riadky s menami a známami, ako to vidieť na obrázku vpravo).

#### Úloha F.3.1

Vytvorte program, ktorý zobrazí obsah textového súboru `znamky.txt`.

#### Riešenie 3.1a

je analogické s riešeniami 2.1 a 1.1, iný je len názov súboru, ktorý treba otvoriť

```
f = open("znamky.txt", "r").
```

Úlohu si opäť sťažíme a budeme požadovať, aby vo výpise bolo v jednom riadku meno aj k nemu prislúchajúce známky (obrázok vpravo nižšie).

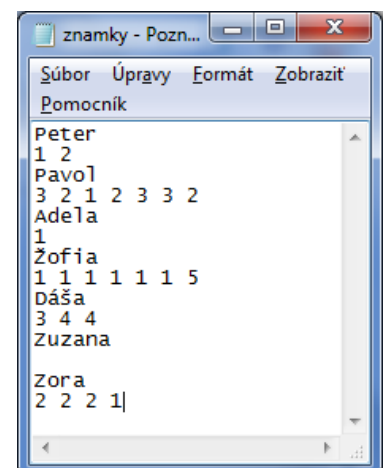
#### Riešenie 3.1b

Použijeme `while`-cyklus. Keďže známky netreba spracovávať, stačí s celým riadkom so známami pracovať ako s reťazcom.

Výstup riešenia je na obrázku vpravo.

```
import sys

try:
    f = open("znamky.txt", "r")
```



```
Peter      1 2
Pavol     3 2 1 2 3 3 2
Adela      1
Žofia     1 1 1 1 1 5
Dáša      3 4 4
Zuzana
Zora      2 2 2 1
```

```
except FileNotFoundError:
    print("Súbor nenájdený, ukončil som program!")
    sys.exit()

meno = f.readline()
while meno != "":
    meno = meno.strip()
    znamky = f.readline().strip()
    print("{meno:10} {znamky}".format(**locals()))
    meno = f.readline()
f.close()
```

### Úloha F.3.2

Vytvorte program, ktorý zo známok zadaných v súbore znamky.txt vypočíta a zobrazí priemery prislúchajúce k jednotlivým menám.

V súbore znamky.txt sme spravili jednu zmenu, Zuzane sme nedali ani jednu známku (riadok so známkami má prázdny). Program má pri chýbajúcich známkach vypísať priemer 0.00. Pozri obrázok nižšie.

### Riešenie 3.2

Výpočet priemeru sme umiestnili do samostatnej funkcie `vratPriemerZnamok` s parametrom `znamky` – zoznam známok typu `str` získaný funkciou `split`.

```
import sys

def vratPriemerZnamok(znamky):
    if len(znamky) == 0:
        return 0.0
    else:
        sucet = 0
        for znamka in znamky:
            sucet += int(znamka)
        return sucet/len(znamky)

# ===== HLAVNÝ PROGRAM =====
try:
    f = open("znamky.txt", "r")
except FileNotFoundError:
    print("Súbor nenájdený, ukončil som program!")
    sys.exit()

meno = f.readline().strip()
while meno != "":
    znamky = f.readline().split()
    priemer = vratPriemerZnamok(znamky)
    print("{meno:10} {priemer:.2f}".format(**locals()))
    meno = f.readline().strip()
f.close()
```

```
>>>
Peter      1.50
Pavol      2.29
Adela      1.00
Žofia      1.57
Dáša       3.67
Zuzana     0.00
Zora       1.75
>>>
```

Program otestujte aj na prázdny súbor.

### Úloha F.4 (rovnaký problém ako F.3.1 a F.3.2)

Vytvorte program, ktorý spracuje dáta uložené v textovom súbore v tvare meno žiaka a na novom riadku jeho známky oddelené medzerami, napríklad:

```
Fero
1 1 4
Katka
1 1
Karol

Dušan
5 4 4 2
```

Program nech umožňuje zadať meno textového súboru, ktorého dáta sa majú spracovať; vo funkciách vypísať súbor; vypísať po riadkoch formátovane meno a známky žiaka a vypísať po riadkoch formátovane meno a priemer jeho známok. Ak študent nemá ani jednu známku, jeho priemer je 0,00.

**Ukážka výstupu - vypísať súbor:**

```
Spracuj triedu (data) zo súboru: tretiaci
Fero
1 1 4
Katka
1 1
Karol

Dušan
5 4 4 2
```

**Ukážka výstupu - vypísať po riadkoch formátovane meno a známky žiaka:**

```
Meno      Znamky
Fero      1 1 4
Katka     1 1
Karol
Dušan     5 4 4 2
```

**Ukážka výstupu - vypísať po riadkoch formátovane meno žiaka a priemer jeho známok:**

```
Meno      Priemer
Fero      2.00
Katka     1.00
Karol     0.00
Dušan     3.75
```

**Jednotlivé funkcie:**

```
def vypisSubor():
    subor = open(nazov_triedy, "r")
    print(subor.read())
    subor.close()

def vypisMenoZnamky():
    subor = open(nazov_triedy, "r")
    print("\n{:10} {}".format("Meno", "Znamky"))
    meno = subor.readline().strip()
    while meno:
        znamkyStr = subor.readline().rstrip()
        print("{:10} {}".format(meno, znamkyStr))
        meno = subor.readline().strip()
    subor.close()

def vypisMenoPriemer():
    subor = open(nazov_triedy, "r")
    print("\n{:10} {}".format("Meno", "Priemer"))
    meno = subor.readline().strip()
    while meno:
        znamky = subor.readline().split()
        sucet = 0
        for i in range(len(znamky)):
            sucet += int(znamky[i])
        if len(znamky) > 0:
            priemer = sucet/len(znamky)
        else:
            priemer = 0.0
        print("{:10} {:.4.2f}".format(meno, priemer))
        meno = subor.readline().strip()
    subor.close()

def main():
    vypisSubor()
    vypisMenoZnamky()
    vypisMenoPriemer()

# =====
nazov_triedy = input("Spracuj triedu (data) zo súboru: ") + ".txt"
main()
```

**Úloha F.5**

Vytvorte program, ktorý spracuje dáta uložené v textovom súbore v tvare meno pracovníka a na novom riadku počet jeho odpracovaných hodín, napríklad:

```
Novák
187.5
Kováčová
168
Malý
200
Úzky
180
```

Program nech umožňuje zadať meno textového súboru, ktorého dáta sa majú spracovať; vypísať súbor; vypísať po riadkoch formátovane meno a odpracované hodiny zamestnanca a vypísať po riadkoch formátovane meno a plat zamestnanca (plat zamestnanca sa vypočíta ako súčin počtu odpracovaných hodín a zadanej konštanty hodinová mzda). Koniec posledného výstupu nech je doplnený o celkovú vyplatenú sumu, t.j. súčet plátov jednotlivých zamestnancov.

Ukážka výstupu:

```
Spracovať súbor: platy
Novák
187.5
Kováčová
168
Malý
200
Úzky
180
```

Meno	Odpr.hod.
Novák	187.5
Kováčová	168.0
Malý	200.0
Úzky	180.0

Meno	Plat v €
Novák	937.50
Kováčová	840.00
Malý	1000.00
Úzky	900.00
=====	
Súčet	3677.50

Riešenie:

```
def vypisSubor():
    subor = open(nazov_suboru, "r")
    print(subor.read())
    subor.close()

def vypisMenoHodiny():
    print("\n{:10} {}".format("Meno", "Odpr.hod. "))
    subor = open(nazov_suboru)
    meno = subor.readline().rstrip()
    while meno:
        hodinyStr = subor.readline().rstrip()
        print("{:10} {:.1f}".format(meno, float(hodinyStr)))
        meno = subor.readline().rstrip()
    subor.close()

def vypisMenoPlat():
    print("\n{:10} {:>8}".format("Meno", "Plat v €"))
    subor = open(nazov_suboru)
    sucet = 0
    meno = subor.readline().rstrip()
```

```
while meno:
    hodiny = float(subor.readline().rstrip())
    sucet += hodiny*HM
    print("{:10} {:8.2f}".format(meno, hodiny*HM))          #
    meno= subor.readline().rstrip()
subor.close()
print("="*19)
print("{:10} {:8.2f}".format("Súčet", sucet))

def main():
    vypisSubor()
    vypisMenoPlat()

# =====
HM = 5 # hodinová mzda v eurách
nazov_suboru = input("Spracovať súbor: ") + ".txt"
main()

Čo sa zmení, ak riadok označený # nahradíme riadkom print("{:10} {:7.1f}0".format(meno, hodiny*HM)?
```