

Придодавање података у постојећи фајл

Када се користи 'w' мод за отварање излазног фајла а фајл са тим именом већ постоји на диску, постојећи фајл ће бити обрисан и нов празан фајл истог имена ће бити креиран.

Понекад је потребно сачувати стари фајл и придодати нове податке на већ постојећи садржај фајла.

Придодавање података у фајл значи уписивање нових података на крај постојећих података.

У Пајтону се може користити мод 'a' за отварање излазног фајла у моду придодавања (append mode), што значи:

- ако фајл већ постоји, неће бити избрисан; ако фајл не постоји, беће креиран
- када се подаци уписују у фајл, биће уписани на крају постојећег садржаја фајла

Пример: нека у фајлу prijatelji.txt већ постоје уписана имена пријатеља:

Miki

Ana

Darko

Следећи код отвара фајл и придодaje податке у постојећи садржај:

```
moj_fajl = open('prijatelji.txt', 'a')
moj_fajl.write('Zoran\n')
moj_fajl.write('Milena\n')
moj_fajl.write('Jovan\n')
moj_fajl.close()
```

Добија се:

Miki

Ana

Darko

Zoran

Milena

Jovan

Упис и читање нумеричких података

Стрингови могу бити уписани директно у фајл са методом write, али се прво бројеви морају конвертовати у стрингове пре уписа у фајл.

Коришћењем уграђене функције str, врши се конвертовање из бројног приказа у стринг.

Пример:

```
def main():
    izlazni_fajl = open('brojevi.txt', 'w')
    broj1 = int(input('Unesi prvi broj: '))
    broj2 = int(input('Unesi drugi broj: '))
    broj3 = int(input('Unesi treci broj: '))
    izlazni_fajl.write(str(broj1) + '\n')
    izlazni_fajl.write(str(broj2) + '\n')
    izlazni_fajl.write(str(broj3) + '\n')
    izlazni_fajl.close()
    print('Podaci su upisani u brojevi.txt fajl.')
```

main()

Unesi prvi broj: 34

Unesi drugi broj: 2345

Unesi treci broj: 2

Podaci su upisani u brojevi.txt fajl.

brojevi.txt - Notepad

File Edit Format View Help

34

2345

2

Када се читају бројеви из текст фајла, они се увек читају као стрингови.

Ако је потребно прочитати претходно унешене бројеве из фајла:

```
ulazni_fajl = open('brojevi.txt', 'r')
vrednost = ulazni_fajl.readline()
ulazni_fajl.close()
```

У коду се користи метода `readline` за читање линије из фајла.

Промењива `vrednost` ће указивати на прву прочитану линију у фајлу а то је `'34\n'`.

Ово може представљати проблем ако је била намера да радимо са бројчаним вредностима, па се зато мора извршити конверзија у неки од нумеричких типова податка:

```
ulazni_fajl = open('brojevi.txt', 'r')
string_ulaz = ulazni_fajl.readline()
vrednost = int(string_ulaz)
ulazni_fajl.close()
```

Сада промењива `string_ulaz` указује на прву линију у фајлу: `'34\n'`.

Затим у следећој линији се врши конверзија промењиве `string_ulaz` из стринга у целобројну вредност и резултат се додељује промењивој `vrednost`.

После овог исказа, промењива `vrednost` ће указивати на целобројну вредност 34 пошто функције `int` и `float` игноришу приликом конверзије ескејп карактере на крају стринга који им се додели као аргумент.

Бољи начин за читање стринга из фајла и његову конверзију:

```
ulazni_fajl = open('brojevi.txt', 'r')
vrednost = int(ulazni_fajl.readline())
ulazni_fajl.close()
```

У другој линији кода се позива метод `readline` као аргумент функције `int`.

При његовом позиву, метод `readline` враћа стринг, који се предаје функцији `int` која га конвертује у целобројну вредност и резултат конверзије се додељује промењивој `vrednost`.

Пример:

```
def main():
    ulazni_fajl = open('brojevi.txt', 'r')
    broj1 = int(ulazni_fajl.readline())
    broj2 = int(ulazni_fajl.readline())
    broj3 = int(ulazni_fajl.readline())
    ulazni_fajl.close()
    suma = broj1 + broj2 + broj3
    print('Brojevi su:', broj1, broj2, broj3)
    print('Njihov zbir je:', suma)
```

```
main()
```