

Óbudai Egyetem  
Kandó Kálmán Villamosmérnöki Kar  
Python  
fájlkezelés

Dr. Schuster György

2017. november 13.

# Fájlkezelés megnyitás

Ha egy fájlhoz hozzá akarunk férni, azt meg kell nyitni.

# Fájlkezelés megnyitás

Ha egy fájlhoz hozzá akarunk férni, azt meg kell nyitni.

A fájl megnyitására az `open` függvény szolgál.

```
fo=open(path[,mode][,buffering])
```

# Fájlkezelés megnyitás

Ha egy fájlhoz hozzá akarunk férni, azt meg kell nyitni.

A fájl elérési útja



```
fo=open(path[, mode] [, buffering])
```

A fájl megnyitására az open  
függvény szolgál.  
A fájl elérési útja.

# Fájlkezelés megnyitás

Ha egy fájlhoz hozzá akarunk férni, azt meg kell nyitni.

A fájl elérési útja  
↑  
`fo=open(path[, mode] [, buffering])`  
↓  
A fájl leíró objektum

A fájl megnyitására az `open` függvény szolgál.  
A fájl elérési útja.  
A fájl leíró objektum.

# Fájlkezelés megnyitás

Ha egy fájlhoz hozzá akarunk férni, azt meg kell nyitni.

A fájl elérési útja

↑

```
fo=open(path[, mode] [, buffering])
```

↓

A fájl leíró objektum

Megnyitási mód

↓

A fájl megnyitására az open függvény szolgál.

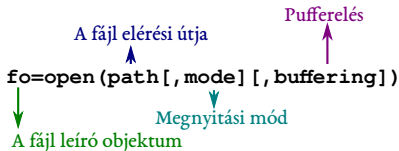
A fájl elérési útja.

A fájl leíró objektum.

Megnyitási mód.

# Fájlkezelés megnyitás

Ha egy fájlhoz hozzá akarunk férni, azt meg kell nyitni.

`fo=open(path[, mode] [, buffering])`  


A fájl megnyitására az `open`  
 függvény szolgál.  
 A fájl elérési útja.  
 A fájl leíró objektum.  
 Megnyitási mód.  
 Pufferelés.

# Fájlkezelés megnyitás

Ha egy fájlhoz hozzá akarunk férni, azt meg kell nyitni.

```
fo=open(path[, mode] [, buffering])
```

A fájl elérési útja

Pufferelés

Megnyitási mód

A fájl leíró objektum

A fájl megnyitására az `open` függvény szolgál.

A fájl elérési útja.

A fájl leíró objektum.

Megnyitási mód.

Pufferelés.

A leíró objektum egy önálló metódusokkal rendelkező entitás. (Erről beszélünk később.)



# Fájlkezelés megnyitás

Ha egy fájlhoz hozzá akarunk férni, azt meg kell nyitni.

A fájl elérési útja
   
`fo=open(path[, mode][, buffering])`
  
 A fájl leíró objektum
   
 Megnyitási mód
   
 Pufferelés

A fájl megnyitására az `open` függvény szolgál.

A fájl elérési útja.

A fájl leíró objektum.

Megnyitási mód.

Pufferelés.

A leíró objektum egy önálló metódusokkal rendelkező entitás. (Erről beszélünk később.)

A megnyitási móddal meghatározhatjuk, hogy a fájlon milyen műveletek hajthatók végre. (Nem kötelező.)

# Fájlkezelés megnyitás

Ha egy fájlhoz hozzá akarunk férni, azt meg kell nyitni.

A fájl elérési útja
   
`fo=open(path[, mode][, buffering])`
  
 A fájl leíró objektum
   
 Megnyitási mód
   
 Pufferelés

A fájl megnyitására az `open` függvény szolgál.

A fájl elérési útja.

A fájl leíró objektum.

Megnyitási mód.

Pufferelés.

A leíró objektum egy önálló metódusokkal rendelkező entitás. (Erről beszélünk később.)

A megnyitási móddal meghatározhatjuk, hogy a fájlon milyen műveletek hajthatók végre. (Nem kötelező.)

A pufferelés a fájl "átmeneti" tár kezelését határozza meg. (Nem kötelező.)

# Fájlkezelés megnyitás mode

A **mode** paraméterek a megnyitás módját határozzák meg.

# Fájlkezelés megnyitás mode

A **mode** paraméterek a megnyitás módját határozzák meg.

Paraméter	Jelentés
-----------	----------

# Fájlkezelés megnyitás mode

A **mode** paraméterek a megnyitás módját határozzák meg.

Paraméter	Jelentés
<b>r</b>	Olvasásra nyitás (ez az alapértelmezett).

# Fájlkezelés megnyitás mode

A **mode** paraméterek a megnyitás módját határozzák meg.

Paraméter	Jelentés
<b>r</b>	Olvasásra nyitás (ez az alapértelmezett).
<b>r+</b>	Olvasására és írásra nyitás.

# Fájlkezelés megnyitás mode

A **mode** paraméterek a megnyitás módját határozzák meg.

Paraméter	Jelentés
<b>r</b>	Olvasásra nyitás (ez az alapértelmezett).
<b>r+</b>	Olvasására és írásra nyitás.
<b>w</b>	Írásra nyitás, a fájlt nyitáskor törli, ha nincs fájl létrehozza.

# Fájlkezelés megnyitás mode

A **mode** paraméterek a megnyitás módját határozzák meg.

Paraméter	Jelentés
<b>r</b>	Olvasásra nyitás (ez az alapértelmezett).
<b>r+</b>	Olvasására és írásra nyitás.
<b>w</b>	Írásra nyitás, a fájlt nyitáskor törli, ha nincs fájl létrehozza.
<b>w+</b>	Írásra olvasásra nyitás, a fájlt nyitáskor törli, ha nincs fájl létrehozza.



# Fájlkezelés megnyitás mode

A **mode** paraméterek a megnyitás módját határozzák meg.

Paraméter	Jelentés
<b>r</b>	Olvasásra nyitás (ez az alapértelmezett).
<b>r+</b>	Olvasására és írásra nyitás.
<b>w</b>	Írásra nyitás, a fájlt nyitáskor törli, ha nincs fájl létrehozza.
<b>w+</b>	Írásra olvasásra nyitás, a fájlt nyitáskor törli, ha nincs fájl létrehozza.
<b>a</b>	Hozzáírásra nyitás.

# Fájlkezelés megnyitás mode

A **mode** paraméterek a megnyitás módját határozzák meg.

Paraméter	Jelentés
<b>r</b>	Olvasásra nyitás (ez az alapértelmezett).
<b>r+</b>	Olvasására és írásra nyitás.
<b>w</b>	Írásra nyitás, a fájlt nyitáskor törli, ha nincs fájl létrehozza.
<b>w+</b>	Írásra olvasásra nyitás, a fájlt nyitáskor törli, ha nincs fájl létrehozza.
<b>a</b>	Hozzáírásra nyitás.
<b>a+</b>	Hozzáírásra és olvasásra nyitás.

# Fájlkezelés megnyitás mode

A **mode** paraméterek a megnyitás módját határozzák meg.

Paraméter	Jelentés
<b>r</b>	Olvasásra nyitás (ez az alapértelmezett).
<b>r+</b>	Olvasására és írásra nyitás.
<b>rb</b>	Olvasására nyitás binárisan.
<b>rb+</b>	Olvasására és írásra nyitás binárisan.
<b>w</b>	Írásra nyitás, a fájlt nyitáskor törli, ha nincs fájl létrehozza.
<b>w+</b>	Írásra olvasásra nyitás, a fájlt nyitáskor törli, ha nincs fájl létrehozza.
<b>wb</b>	Írásra nyitás, a fájlt nyitáskor törli, ha nincs fájl létrehozza binárisan.
<b>wb+</b>	Írásra olvasásra nyitás, a fájlt nyitáskor törli, ha nincs fájl létrehozza binárisan.
<b>a</b>	Hozzáadásra nyitás.
<b>a+</b>	Hozzáadásra és olvasásra nyitás.
<b>ab</b>	Hozzáadásra nyitás binárisan.
<b>ab+</b>	Hozzáadásra és olvasásra nyitás binárisan.

# Fájlkezelés megnyitás mode

A **mode** paraméterek a megnyitás módját határozzák meg.

Paraméter	Jelentés
<b>r</b>	Olvasásra nyitás (ez az alapértelmezett).
<b>r+</b>	Olvasására és írásra nyitás.
<b>rb</b>	Olvasásra nyitás binárisan.
<b>rb+</b>	Olvasására és írásra nyitás binárisan.
<b>w</b>	Írásra nyitás, a fájl nem létezik, akkor létrehozza.
<b>w+</b>	Írásra olvasásra nyitás, a fájl nem létezik, akkor létrehozza.
<b>wb</b>	Írásra nyitás, a fájl nem létezik, akkor létrehozza binárisan.
<b>wb+</b>	Írásra olvasásra nyitás, a fájl nem létezik, akkor létrehozza binárisan.
<b>a</b>	Hozzáírásra nyitás, a fájl nem létezik, akkor létrehozza.
<b>a+</b>	Hozzáírásra olvasásra nyitás, a fájl nem létezik, akkor létrehozza.
<b>ab</b>	Hozzáírásra nyitás, a fájl nem létezik, akkor létrehozza binárisan.
<b>ab+</b>	Hozzáírásra olvasásra nyitás, a fájl nem létezik, akkor létrehozza binárisan.

## Bináris fájlkezelés

A bináris fájlkezelésnek Windows-nál van szerepe és akkor, ha a fájlban tetszőleges irányban szeretnénk mozogni. Windows sor vég kezelés:

Fájl	Memória
...CR LF	⇒ ...LF
...CR LF	⇐ ...LF

# Fájlkezelés megnyitás `buffering`

A `buffering` paraméter a puffelés módját határozza meg.

# Fájlkezelés megnyitás `buffering`

A `buffering` paraméter a puffereles módját határozza meg.

A puffereles az jelenti, hogy a fájl és a fájl kezelő alkalmazás között hogyan történik az adatmozgatás.

Ha nincs puffereles, akkor a fájl és az alkalmazás között az adatmozgás azonnal megtörténik.

Puffereles esetén, először az adatok a memóriába kerülnek és - vagy egy idő után (30 - 60s) után, vagy a puffer méretének betelése után kerülnek az adatok kiírásra a fájlba.

# Fájlkezelés megnyitás `buffering`

A `buffering` paraméter a puffelés módját határozza meg.

Paraméter	Jelentés
-----------	----------

# Fájlkezelés megnyitás `buffering`

A `buffering` paraméter a pufferelés módját határozza meg.

Paraméter	Jelentés
0	Nincs pufferelés.



# Fájkezelés megnyitás `buffering`

A `buffering` paraméter a puffereles módját határozza meg.

Paraméter	Jelentés
0	Nincs puffereles.
1	Van puffereles, a puffer mérete az alapértelmezett.

# Fájkezelés megnyitás `buffering`

A `buffering` paraméter a puffereles módját határozza meg.

Paraméter	Jelentés
0	Nincs puffereles.
1	Van puffereles, a puffer mérete az alapértelmezett.
$n > 1$	Van puffereles, $n$ a puffer mérete.

# Fájlkezelés megnyitás `buffering`

A `buffering` paraméter a puffereles módját határozza meg.

Paraméter	Jelentés
0	Nincs puffereles.
1	Van puffereles, a puffer mérete az alapértelmezett.
<code>n&gt;1</code>	Van puffereles, <code>n</code> a puffer mérete.
<code>n&lt;0</code>	Van puffereles, a puffer mérete az alapértelmezett.

# Fájl objektum

Az open függvény egy fájl objektummal tér vissza.

# Fájl objektum

Az open függvény egy fájl objektummal tér vissza.

Ez egy OOP fogalom, még lesz róla szó.

# Fájl objektum

Az open függvény egy fájl objektummal tér vissza.

Legfontosabb metódusai:

# Fájl objektum

Az `open` függvény egy fájl objektummal tér vissza.

Legfontosabb metódusai:

`close`            lezárja a fájlt.

# Fájl objektum

Az open függvény egy fájl objektummal tér vissza.

Legfontosabb metódusai:

`close`

lezárja a fájlt.

`read`

olvas a fájlból.



# Fájl objektum

Az open függvény egy fájl objektummal tér vissza.

Legfontosabb metódusai:

<code>close</code>	lezárja a fájlt.
<code>read</code>	olvas a fájlból.
<code>readline</code>	egy teljes sor beolvasása.

# Fájl objektum

Az open függvény egy fájl objektummal tér vissza.

Legfontosabb metódusai:

<code>close</code>	lezárja a fájlt.
<code>read</code>	olvas a fájlból.
<code>readline</code>	egy teljes sor beolvasása.
<code>write</code>	ír a fájlba.

# Fájl objektum

Az open függvény egy fájl objektummal tér vissza.

Legfontosabb metódusai:

<code>close</code>	lezárja a fájlt.
<code>read</code>	olvas a fájlból.
<code>readline</code>	egy teljes sor beolvasása.
<code>write</code>	ír a fájlba.
<code>tell</code>	a fájl pozíció mutató értékéte adja meg.

# Fájl objektum

Az open függvény egy fájl objektummal tér vissza.

Legfontosabb metódusai:

<code>close</code>	lezárja a fájlt.
<code>read</code>	olvas a fájlból.
<code>readline</code>	egy teljes sor beolvasása.
<code>write</code>	ír a fájlba.
<code>tell</code>	a fájl pozíció mutató értékéte adja meg.

## Fájl pozíció mutató

az a pozíció érték a fájlban, ahol a következő művelet történik.

# Fájl objektum

Az open függvény egy fájl objektummal tér vissza.

Legfontosabb metódusai:

<code>close</code>	lezárja a fájlt.
<code>read</code>	olvas a fájlból.
<code>readline</code>	egy teljes sor beolvasása.
<code>write</code>	ír a fájlba.
<code>tell</code>	a fájl pozíció mutató értékéte adja meg.
<code>seek</code>	beállítja a fájl pozíció mutatót.

Használat:

# Fájl objektum

Az open függvény egy fájl objektummal tér vissza.

Legfontosabb metódusai:

<code>close</code>	lezárja a fájlt.
<code>read</code>	olvas a fájlból.
<code>readline</code>	egy teljes sor beolvasása.
<code>write</code>	ír a fájlba.
<code>tell</code>	a fájl pozíció mutató értékéte adja meg.
<code>seek</code>	beállítja a fájl pozíció mutatót.

Használat: `FileObject.method(...)`

# Fájl objektum

Legfontosabb változói:

# Fájl objektum

Legfontosabb változói:

**closed**

igaz, ha a fájl le van zárva.



# Fájl objektum

Legfontosabb változói:

**closed**  
**mode**

igaz, ha a fájl le van zárva.  
a megnyitás módjával tér vissza.

# Fájl objektum

Legfontosabb változói:

<code>closed</code>	igaz, ha a fájl le van zárva.
<code>mode</code>	a megnyitás módjával tér vissza.
<code>name</code>	a fájl nevével tér vissza.

# Fájl objektum

Legfontosabb változói:

<code>closed</code>	igaz, ha a fájl le van zárva.
<code>mode</code>	a megnyitás módjával tér vissza.
<code>name</code>	a fájl nevével tér vissza.
<code>softspace</code>	???

# A fájl objektum metódusai

A fájlt a használat után célszerű lezárni.

# A fájl objektum metódusai

A fájlt a használat után célszerű lezárni.

```
fo=open(...)
```

A fájl megnyitása.

# A fájl objektum metódusai

A fájlt a használat után célszerű lezárni.

```
fo=open(...)  
    ⋮
```

A fájl megnyitása.  
A fájl használata.

# A fájl objektum metódusai

A fájlt a használat után célszerű lezárni.

```
fo=open(...)  
    ⋮  
fo.close()
```

A fájl megnyitása.

A fájl használata.

A fájl lezárása.

# A fájl objektum metódusai

A fájlt a használat után célszerű lezárni.

```
fo=open(...)  
    :  
fo.close()
```

A fájl megnyitása.

A fájl használata.

A fájl lezárása.

Ha egy már megnyitott fájl objektumra egy újabb megnyitás történik, akkor a Python az előzőleg megnyitott fájlt lezárja.



# Fájl objektum metódusai

Olvásás a fájlból:

# Fájl objektum metódusai

Olvasás a fájlból:

```
values=
```

```
.
```

A változó, amibe olvasunk.

# Fájl objektum metódusai

Olvadás a fájlból:

```
values=FileObject.
```

A változó, amibe olvasunk.  
A fájl objektum.

# Fájl objektum metódusai

Olvadás a fájlból:

```
values=FileObject.read( )
```

A változó, amibe olvasunk.

A fájl objektum.

A `read` függvény.

# Fájl objektum metódusai

Olvadás a fájlból:

```
values=FileObject.read(n)
```

A változó, amibe olvasunk.

A fájl objektum.

A **read** függvény.

A bolvasandó bájtok száma.

# Fájl objektum metódusai

Olvadás a fájlból:

```
values=FileObject.read(n)
```

```
values=FileObject.read()
```

A változó, amibe olvasunk.

A fájl objektum.

A **read** függvény.

A olvasandó bájtok száma.

Ha a számot nem adjuk meg, az egész fájl beolvassa.

# Fájl objektum metódusai

Olvadás a fájlból:

```
values=FileObject.read(n)
```

```
values=FileObject.read()
```

```
values=FileObject.readline()
```

A változó, amibe olvasunk.

A fájl objektum.

A **read** függvény.

A bolvasandó bájtok száma.

Ha a számot nem adjuk meg, az egész fájl beolvassa.

Egy sort olvas be a **value** változóba, a sort lezáró LF karakterrel együtt.

# Fájl objektum metódusai

Olvásás a fájlból:

```
values=FileObject.read(n)
```

```
values=FileObject.read()
```

```
values=FileObject.readline()
```

Írás a fájlba:

A változó, amibe olvasunk.

A fájl objektum.

A **read** függvény.

A olvasandó bájtok száma.

Ha a számot nem adjuk meg, az egész fájl beolvassa.

Egy sort olvas be a **value** változóba, a sort lezáró LF karakterrel együtt.



# Fájl objektum metódusai

Olvadás a fájlból:

```
values=FileObject.read(n)
```

```
values=FileObject.read()
```

```
values=FileObject.readline()
```

Írás a fájlba:

```
FileObject.write(value)
```

A változó, amibe olvasunk.

A fájl objektum.

A **read** függvény.

A olvasandó bájtok száma.

Ha a számot nem adjuk meg, az egész fájlt beolvassa.

Egy sort olvas be a **value** változóba, a sort lezáró LF karakterrel együtt.

A **value** változót kírja fájlba.

# Fájl objektum metódusai

Fájl pozíció olvasása:

# Fájl objektum metódusai

Fájl pozíció olvasása:

A **value** változóba kerül a fájl pozíció mutató értéke

**values=** .

# Fájl objektum metódusai

Fájl pozíció olvasása:

```
values=FileObject.
```

A **value** változóba kerül a fájl pozíció mutató értéke

A fájl objektum.

# Fájl objektum metódusai

Fájl pozíció olvasása:

```
values=FileObject.tell()
```

A **value** változóba kerül a fájl pozíció mutató értéke

A fájl objektum.

A **tell** függvény.

# Fájl objektum metódusai

Fájl pozíció olvasása:

```
values=FileObject.tell()
```

A **value** változóba kerül a fájl pozíció mutató értéke

A fájl objektum.

A **tell** függvény.

Fájl pozíció beállítása:

# Fájl objektum metódusai

Fájl pozíció olvasása:

```
values=FileObject.tell()
```

A **value** változóba kerül a fájl pozíció mutató értéke

A fájl objektum.

A **tell** függvény.

Fájl pozíció beállítása:

Fájl objektum.

FileObject.

# Fájl objektum metódusai

Fájl pozíció olvasása:

```
values=FileObject.tell()
```

A **value** változóba kerül a fájl pozíció mutató értéke

A fájl objektum.

A **tell** függvény.

Fájl pozíció beállítása:

```
FileObject.seek( , )
```

Fájl objektum.

**seek** függvény.



# Fájl objektum metódusai

Fájl pozíció olvasása:

```
values=FileObject.tell()
```

A **value** változóba kerül a fájl pozíció mutató értéke

A fájl objektum.

A **tell** függvény.

Fájl pozíció beállítása:

```
FileObject.seek(offset, )
```

Fájl objektum.

**seek** függvény.

**offset** érték, ezzel módosítjuk a fájl pozíció mutatót.

# Fájl objektum metódusai

Fájl pozíció olvasása:

```
values=FileObject.tell()
```

A **value** változóba kerül a fájl pozíció mutató értéke

A fájl objektum.

A **tell** függvény.

Fájl pozíció beállítása:

```
FileObject.seek(offset,[whence])
```

Fájl objektum.

**seek** függvény.

**offset** érték, ezzel módosítjuk a fájl pozíció mutatót.

**whence** a módosítás honnan történik (nem kötelező):

# Fájl objektum metódusai

Fájl pozíció olvasása:

```
values=FileObject.tell()
```

A **value** változóba kerül a fájl pozíció mutató értéke

A fájl objektum.

A **tell** függvény.

Fájl pozíció beállítása:

```
FileObject.seek(offset,[whence])
```

Fájl objektum.

**seek** függvény.

**offset** érték, ezzel módosítjuk a fájl pozíció mutatót.

**whence** a módosítás honnan történik (nem kötelező):

0 fájl elejéről (alapértelmezett),

# Fájl objektum metódusai

Fájl pozíció olvasása:

```
values=FileObject.tell()
```

A **value** változóba kerül a fájl pozíció mutató értéke

A fájl objektum.

A **tell** függvény.

Fájl pozíció beállítása:

```
FileObject.seek(offset,[whence])
```

Fájl objektum.

**seek** függvény.

**offset** érték, ezzel módosítjuk a fájl pozíció mutatót.

**whence** a módosítás honnan történik (nem kötelező):

0 fájl elejéről (alapértelmezett),

1 aktuális pozíciótól,

# Fájl objektum metódusai

Fájl pozíció olvasása:

```
values=FileObject.tell()
```

A **value** változóba kerül a fájl pozíció mutató értéke

A fájl objektum.

A **tell** függvény.

Fájl pozíció beállítása:

```
FileObject.seek(offset,[whence])
```

Fájl objektum.

**seek** függvény.

**offset** érték, ezzel módosítjuk a fájl pozíció mutatót.

**whence** a módosítás honnan történik (nem kötelező):

0 fájl elejétől (alapértelmezett),

1 aktuális pozíciótól,

2 végétől.

# Példa: fájl listázás karakterenként

Listázzuk ki karakterenként a következő fájlt!

# Példa: fájl listázás karakterenként

Listázzuk ki karakterenként a következő fájlt!

```
file.txt
```

```
Humpty Dumpty sat on the wall  
Humpty Dumpty had a great fall  
All the king's horses  
All the king's men  
Couldn't put Humpty together again.
```

# Példa: fájl listázás karakterenként

```
#!/usr/bin/python3
fo=open("file.txt","r")
while True:
    c=fo.read(1)
    if c == '':
        break
    else:
        print(c,end='')
fo.close()
```

A futtató program.



# Példa: fájl listázás karakterenként

```
#!/usr/bin/python3
fo=open("file.txt","r")
while True:
    c=fo.read(1)
    if c == '':
        break
    else:
        print(c,end='')
fo.close()
```

A futtató program.

A fájl megnyitása olvasásra.

# Példa: fájl listázás karakterenként

```
#!/usr/bin/python3
fo=open("file.txt","r")
while True:
    c=fo.read(1)
    if c == '':
        break
    else:
        print(c,end='')
fo.close()
```

A futtató program.

A fájl megnyitása olvasásra.

Az olvasó ciklus feje.

# Példa: fájl listázás karakterenként

```
#!/usr/bin/python3
fo=open("file.txt","r")
while True:
    c=fo.read(1)
    if c == '':
        break
    else:
        print(c,end='')
fo.close()
```

A futtató program.

A fájl megnyitása olvasásra.

Az olvasó ciklus feje.

Olvasás a `c` változóba.

# Példa: fájl listázás karakterenként

```
#!/usr/bin/python3
fo=open("file.txt","r")
while True:
    c=fo.read(1)
    if c == '':
        break
    else:
        print(c,end='')
fo.close()
```

A futtató program.

A fájl megnyitása olvasásra.

Az olvasó ciklus feje.

Olvasás a `c` változóba.

Vége a fájlnek?

# Példa: fájl listázás karakterenként

A fájl végét az '' üres karakter jelzi.

```
#!/usr/bin/python3
fo=open("file.txt","r")
while True:
    c=fo.read(1)
    if c == '':
        break
    else:
        print(c,end='')
fo.close()
```

A futtató program.

A fájl megnyitása olvasásra.

Az olvasó ciklus feje.

Olvasás a `c` változóba.

Vége a fájlnek?

# Példa: fájl listázás karakterenként

```
#!/usr/bin/python3
fo=open("file.txt","r")
while True:
    c=fo.read(1)
    if c == '':
        break
    else:
        print(c,end='')
fo.close()
```

A futtató program.

A fájl megnyitása olvasásra.

Az olvasó ciklus feje.

Olvasás a `c` változóba.

Vége a fájlnek?

Igen, kiugrás a ciklusból.

# Példa: fájl listázás karakterenként

```
#!/usr/bin/python3
fo=open("file.txt","r")
while True:
    c=fo.read(1)
    if c == '':
        break
    else:
        print(c,end='')
fo.close()
```

A futtató program.

A fájl megnyitása olvasásra.

Az olvasó ciklus feje.

Olvasás a `c` változóba.

Vége a fájlnek?

Igen, kiugrás a ciklusból.

Ha nem, akkor kiiratás.

# Példa: fájl listázás karakterenként

```
#!/usr/bin/python3
fo=open("file.txt","r")
while True:
    c=fo.read(1)
    if c == '':
        break
    else:
        print(c,end='')
fo.close()
```

A futtató program.

A fájl megnyitása olvasásra.

Az olvasó ciklus feje.

Olvasás a `c` változóba.

Vége a fájlnek?

Igen, kiugrás a ciklusból.

Ha nem, akkor kiiratás.

Ha kész, akkor a fájl lezárása.



# Példa: fájl listázás karakterenként

```
#!/usr/bin/python3
fo=open("file.txt","r")
while True:
    c=fo.read(1)
    if c == '':
        break
    else:
        print(c,end='')
fo.close()
```

A futtató program.

A fájl megnyitása olvasásra.

Az olvasó ciklus feje.

Olvasás a `c` változóba.

Vége a fájlnek?

Igen, kiugrás a ciklusból.

Ha nem, akkor kiiratás.

Ha kész, akkor a fájl lezárása.

# Példa: fájl listázás soronként

# Példa: fájl listázás soronként

```
#!/usr/bin/python3
fo=open("file.txt","r")
while True:
    c=fo.readline()
    if c == '':
        break
    else:
        print(c,end='')
fo.close()
```

# Példa: fájl listázás soronként

```
#!/usr/bin/python3
fo=open("file.txt","r")
while True:
    c=fo.readline()
    if c == '':
        break
    else:
        print(c,end='')
fo.close()
```

Mint az előző példában.

## Példa: fájl listázás soronként

```
#!/usr/bin/python3
fo=open("file.txt","r")
while True:
    c=fo.readline()
    if c == '':
        break
    else:
        print(c,end='')
fo.close()
```

Mint az előző példában.  
Egy teljes sor beolvasásása.

# Példa: fájl listázás soronként

```
#!/usr/bin/python3
fo=open("file.txt","r")
while True:
    c=fo.readline()
    if c == '':
        break
    else:
        print(c,end='')
fo.close()
```

Mint az előző példában.

Egy teljes sor beolvasásása.

Mint az előző példában. Csak itt egy teljes sort ír ki

# Példa: fájl listázás soronként

```
#!/usr/bin/python3
fo=open("file.txt","r")
while True:
    c=fo.readline()
    if c == '':
        break
    else:
        print(c,end='')
fo.close()
```

Mint az előző példában.

Egy teljes sor beolvasásása.

Mint az előző példában. Csak itt egy teljes sort ír ki

# Példa: fájl listázás soronként és egyben

```
#!/usr/bin/python3
fo=open("file.txt","r")
while True:
    c=fo.readline()
    if c == '':
        break
    else:
        print(c,end='')
fo.close()
```

Mint az előző példában.

Egy teljes sor beolvasásása.

Mint az előző példában. Csak itt egy teljes sort ír ki



# Példa: fájl listázás soronként és egyben

```
#!/usr/bin/python3
fo=open("file.txt","r")
while True:
    c=fo.readline()
    if c == '':
        break
    else:
        print(c,end='')
fo.close()
```

```
#!/usr/bin/python3
fo=open("file.txt","r")
c=fo.read()
fo.close()
```

Mint az előző példában.

Egy teljes sor beolvasásása.

Mint az előző példában. Csak itt egy teljes sort ír ki

# Példa: fájl listázás soronként és egyben

```
#!/usr/bin/python3
fo=open("file.txt","r")
while True:
    c=fo.readline()
    if c == '':
        break
    else:
        print(c,end='')
fo.close()
```

Mint az előző példában.

Egy teljes sor beolvasásása.

Mint az előző példában. Csak itt egy teljes sort ír ki

```
#!/usr/bin/python3
fo=open("file.txt","r")
c=fo.read()
fo.close()
```

Semmi új.

# Példa: fájl listázás soronként és egyben

```
#!/usr/bin/python3
fo=open("file.txt","r")
while True:
    c=fo.readline()
    if c == '':
        break
    else:
        print(c,end='')
fo.close()
```

Mint az előző példában.

Egy teljes sor beolvasásása.

Mint az előző példában. Csak itt egy teljes sort ír ki

```
#!/usr/bin/python3
fo=open("file.txt","r")
c=fo.read()
fo.close()
```

Semmi új.

Az egész fájl beolvassa.

# Példa: fájl listázás soronként és egyben

```
#!/usr/bin/python3
fo=open("file.txt","r")
while True:
    c=fo.readline()
    if c == '':
        break
    else:
        print(c,end='')
fo.close()
```

Mint az előző példában.

Egy teljes sor beolvasásása.

Mint az előző példában. Csak itt egy teljes sort ír ki

```
#!/usr/bin/python3
fo=open("file.txt","r")
c=fo.read()
fo.close()
```

Semmi új.

Az egész fájl beolvassa.

Illik lezárni.

# Példa: fájl listázás soronként és egyben

```
#!/usr/bin/python3
fo=open("file.txt","r")
while True:
    c=fo.readline()
    if c == '':
        break
    else:
        print(c,end='')
fo.close()
```

Mint az előző példában.

Egy teljes sor beolvasásása.

Mint az előző példában. Csak itt egy teljes sort ír ki

```
#!/usr/bin/python3
fo=open("file.txt","r")
c=fo.read()
fo.close()
```

Semmi új.

Az egész fájl beolvassa.

Illik lezárni.

Példa: fájl másolás egy másik fájlba fordított sorrendben

Példa: fájl másolás egy másik fájlba fordított sorrendben

A futtató program.

```
#!/usr/bin/python3
ff=open("file.txt","rb")
fc=open("filer.txt","wb")
ff.seek(-1,2)
while True:
    c=ff.read(1)
    fc.write(c)
    p=ff.tell()
    if p==1:
        break
    else:
        ff.seek(-2,1)
ff.close()
fc.close()
```

Példa: fájl másolás egy másik fájlba fordított sorrendben

```
#!/usr/bin/python3
ff=open("file.txt","rb")
fc=open("filer.txt","wb")
ff.seek(-1,2)
while True:
    c=ff.read(1)
    fc.write(c)
    p=ff.tell()
    if p==1:
        break
    else:
        ff.seek(-2,1)
ff.close()
fc.close()
```

A futtató program.

A forrás fájl bináris megnyitása.



Példa: fájl másolás egy másik fájlba fordított sorrendben

A futtató program.

A forrás fájl bináris megnyitása.

```
#!/usr/bin/python3
ff=open("file.txt", "rb")
fc=open("file")
ff.seek(-1,2)
while True:
    c=ff.read(1)
    fc.write(c)
    p=ff.tell()
    if p==1:
        break
    else:
        ff.seek(-2,1)
ff.close()
fc.close()
```

A forrás fájl binárisan kell megnyitni,  
különböztetve a `seek` függvényt nem hajlandó a fájl  
eleje felé mozogni a fájl pozíció mutatót.

Példa: fájl másolás egy másik fájlba fordított sorrendben

```
#!/usr/bin/python3
ff=open("file.txt","rb")
fc=open("filer.txt","wb")
ff.seek(-1,2)
while True:
    c=ff.read(1)
    fc.write(c)
    p=ff.tell()
    if p==1:
        break
    else:
        ff.seek(-2,1)
ff.close()
fc.close()
```

A futtató program.

A forrás fájl bináris megnyitása.

A cél fájl megnyitása létrehozásra és írásra.

Példa: fájl másolás egy másik fájlba fordított sorrendben

```
#!/usr/bin/python3
ff=open("file.txt", "rb")
fc=open("filer.txt", "wb")
ff.seek(-1, 2)
while True:
    c=ff.read(1)
    fc.write(c)
    p=ff.tell()
    if p==1:
        break
    else:
        ff.seek(-2, 1)
ff.close()
fc.close()
```

A futtató program.

A forrás fájl bináris megnyitása.

A cél fájl megnyitása létrehozásra és írásra.

A forrásban a fájl pozíció mutató beállítása az utolsó karakterre.

Példa: fájl másolás egy másik fájlba fordított sorrendben

```
#!/usr/bin/python3
ff=open("file.txt", "rb")
fc=open("filer.txt", "wb")
ff.seek(-1, 2)
while True:
    c=ff.read(1)
    fc.write(c)
    p=ff.tell()
    if p==1:
        break
    else:
        ff.seek(-2, 1)
ff.close()
fc.close()
```

A futtató program.

A forrás fájl bináris megnyitása.

A cél fájl megnyitása létrehozásra és írásra.

A forrásban a fájl pozíció mutató beállítása az utolsó karakterre.

( Az utolsó karakterre kell állni, mert azt már be kell olvasni.

Példa: fájl másolás egy másik fájlba fordított sorrendben

```
#!/usr/bin/python3
ff=open("file.txt", "rb")
fc=open("filer.txt", "wb")
ff.seek(-1, 2)
while True:
    c=ff.read(1)
    fc.write(c)
    p=ff.tell()
    if p==1:
        break
    else:
        ff.seek(-2, 1)
ff.close()
fc.close()
```

A futtató program.

A forrás fájl bináris megnyitása.

A cél fájl megnyitása létrehozásra és írásra.

A forrásban a fájl pozíció mutató beállítása az utolsó karakterre.

A végtelen ciklus feje.

Példa: fájl másolás egy másik fájlba fordított sorrendben

```
#!/usr/bin/python3
ff=open("file.txt", "rb")
fc=open("filer.txt", "wb")
ff.seek(-1, 2)
while True:
    c=ff.read(1)
    fc.write(c)
    p=ff.tell()
    if p==1:
        break
    else:
        ff.seek(-2, 1)
ff.close()
fc.close()
```

A futtató program.

A forrás fájl bináris megnyitása.

A cél fájl megnyitása létrehozásra és írásra.

A forrásban a fájl pozíció mutató beállítása az utolsó karakterre.

A végtelen ciklus feje.

Egy karakter olvasása a forrásból.

Példa: fájl másolás egy másik fájlba fordított sorrendben

```
#!/usr/bin/python3
ff=open("file.txt", "rb")
fc=open("filer.txt", "wb")
ff.seek(-1,2)
while True:
    c=ff.read(1)
    fc.write(c)
    p=ff.tell()
    if p==1:
        break
    else:
        ff.seek(-2,1)
ff.close()
fc.close()
```

A futtató program.

A forrás fájl bináris megnyitása.

A cél fájl megnyitása létrehozásra és írásra.

A forrásban a fájl pozíció mutató beállítása az utolsó karakterre.

A végtelen ciklus feje.

Egy karakter olvasása a forrásból.

A kiolvasott karakter kiírása a cél fájlba.

Példa: fájl másolás egy másik fájlba fordított sorrendben

```
#!/usr/bin/python3
ff=open("file.txt", "rb")
fc=open("filer.txt", "wb")
ff.seek(-1,2)
while True:
    c=ff.read(1)
    fc.write(c)
    p=ff.tell()
    if p==1:
        break
    else:
        ff.seek(-2,1)
ff.close()
fc.close()
```

A futtató program.

A forrás fájl bináris megnyitása.

A cél fájl megnyitása létrehozásra és írásra.

A forrásban a fájl pozíció mutató beállítása az utolsó karakterre.

A végtelen ciklus feje.

Egy karakter olvasása a forrásból.

A kiolvasott karakter kiírása a cél fájlba.

A forrás fájl pozíció mutatójának olvasása.



Példa: fájl másolás egy másik fájlba fordított sorrendben

```
#!/usr/bin/python3
ff=open("file.txt", "rb")
fc=open("filer.txt", "wb")
ff.seek(-1, 2)
while True:
    c=ff.read(1)
    fc.write(c)
    p=ff.tell()
    if p==1:
        break
    else:
        ff.seek(-2, 1)
ff.close()
fc.close()
```

A futtató program.

A forrás fájl bináris megnyitása.

A cél fájl megnyitása létrehozásra és írásra.

A forrásban a fájl pozíció mutató beállítása az utolsó karakterre.

A végtelen ciklus feje.

Egy karakter olvasása a forrásból.

A kiolvasott karakter kiírása a cél fájlba.

A forrás fájl pozíció mutatójának olvasása.

Ha a `p` értéke egy, akkor beolvasta a 0. karaktert.

Példa: fájl másolás egy másik fájlba fordított sorrendben

```
#!/usr/bin/python3
ff=open("file.txt", "rb")
fc=open("filer.txt", "wb")
ff.seek(-1,2)
while True:
    c=ff.read(1)
    fc.write(c)
    p=ff.tell()
    if p==1:
        break
    else:
        ff.seek
ff.close()
fc.close()
```

A karakterek beolvasása után a fájl pozíció mutató értéke egyel növekszik, ezért vizsgáljuk az 1 értéket.

A futtató program.

A forrás fájl bináris megnyitása.

A cél fájl megnyitása létrehozásra és írásra.

A forrásban a fájl pozíció mutató beállítása az utolsó karakterre.

A végtelen ciklus feje.

Egy karakter olvasása a forrásból.

A kiolvasott karakter kiírása a cél fájlba.

A forrás fájl pozíció mutatójának olvasása.

Ha a `p` értéke egy, akkor beolvasta a 0.

Példa: fájl másolás egy másik fájlba fordított sorrendben

```
#!/usr/bin/python3
ff=open("file.txt", "rb")
fc=open("filer.txt", "wb")
ff.seek(-1, 2)
while True:
    c=ff.read(1)
    fc.write(c)
    p=ff.tell()
    if p==1:
        break
    else:
        ff.seek(-2, 1)
ff.close()
fc.close()
```

A futtató program.

A forrás fájl bináris megnyitása.

A cél fájl megnyitása létrehozásra és írásra.

A forrásban a fájl pozíció mutató beállítása az utolsó karakterre.

A végtelen ciklus feje.

Egy karakter olvasása a forrásból.

A kiolvasott karakter kiírása a cél fájlba.

A forrás fájl pozíció mutatójának olvasása.

Ha a `p` értéke egy, akkor beolvasta a 0. karaktert.

Ha nem a forrás fájl pozíció mutatóját kettővel előre módosítja.

Példa: fájl másolás egy másik fájlba fordított sorrendben

```
#!/usr/bin/python3
ff=open("file.txt", "rb")
fc=open("filer.txt", "wb")
ff.seek(-1,2)
while True:
    c=ff.read(1)
    fc.write(c)
    p=ff.tell()
    if p==1:
        break
    else:
        ff.seek(-2,1)
ff.close()
fc.close()
```

A futtató program.

A forrás fájl bináris megnyitása.

A cél fájl megnyitása létrehozásra és írásra.

A forrásban a fájl pozíció mutató beállítása az utolsó karakterre.

A végtelen ciklus feje.

Egy karakter olvasása a forrásból.

A kiolvasott karakter kiírása a cél fájlba.

A forrás fájl pozíció mutatójának olvasása.

Ha a `p` értéke egy, akkor beolvasta a 0. karaktert.

Ha nem a forrás fájl pozíció mutatóját kettővel előre módosítja.

Egyet előre, kettőt vissza.

Példa: fájl másolás egy másik fájlba fordított sorrendben

```
#!/usr/bin/python3
ff=open("file.txt", "rb")
fc=open("filer.txt", "wb")
ff.seek(-1,2)
while True:
    c=ff.read(1)
    fc.write(c)
    p=ff.tell()
    if p==1:
        break
    else:
        ff.seek(-2,1)
ff.close()
fc.close()
```

A futtató program.

A forrás fájl bináris megnyitása.

A cél fájl megnyitása létrehozásra és írásra.

A forrásban a fájl pozíció mutató beállítása az utolsó karakterre.

A végtelen ciklus feje.

Egy karakter olvasása a forrásból.

A kiolvasott karakter kiírása a cél fájlba.

A forrás fájl pozíció mutatójának olvasása.

Ha a `p` értéke egy, akkor beolvasta a 0. karaktert.

Ha nem a forrás fájl pozíció mutatóját kettővel előre módosítja.

Fájlok lezárása.

Példa: fájl másolás egy másik fájlba fordított sorrendben

```
#!/usr/bin/python3
ff=open("file.txt", "rb")
fc=open("filer.txt", "wb")
ff.seek(-1,2)
while True:
    c=ff.read(1)
    fc.write(c)
    p=ff.tell()
    if p==1:
        break
    else:
        ff.seek(-2,1)
ff.close()
fc.close()
```

A futtató program.

A forrás fájl bináris megnyitása.

A cél fájl megnyitása létrehozásra és írásra.

A forrásban a fájl pozíció mutató beállítása az utolsó karakterre.

A végtelen ciklus feje.

Egy karakter olvasása a forrásból.

A kiolvasott karakter kiírása a cél fájlba.

A forrás fájl pozíció mutatójának olvasása.

Ha a `p` értéke egy, akkor beolvasta a 0. karaktert.

Ha nem a forrás fájl pozíció mutatóját kettővel előre módosítja.

Fájlok lezárása.

Példa: fájl másolás egy másik fájlba fordított sorrendben

```
#!/usr/bin/python3
ff=open("file.txt", "rb")
fc=open("filer.txt", "wb")
ff.seek(-1,2)
while True:
    c=ff.read(1)
    fc.write(c)
    p=ff.tell()
    if p==1:
        break
    else:
        ff.seek(-2,1)
ff.close()
fc.close()
```

A futtató program.

A forrás fájl bináris megnyitása.

A cél fájl megnyitása létrehozásra és írásra.

A forrásban a fájl pozíció mutató beállítása az utolsó karakterre.

A végtelen ciklus feje.

Egy karakter olvasása a forrásból.

A kiolvasott karakter kiírása a cél fájlba.

A forrás fájl pozíció mutatójának olvasása.

Ha a `p` értéke egy, akkor beolvasta a 0. karaktert.

Ha nem a forrás fájl pozíció mutatóját kettővel előre módosítja.

Fájlok lezárása.

Példa: fájl másolás egy másik fájlba fordított sorrendben

```
#!/usr/bin/python3
ff=open("file.txt", "rb")
fc=open("filer.txt", "wb")
ff.seek(-1,2)
while True:
    c=ff.read(1)
    fc.write(c)
    p=ff.tell()
    if p==1:
        break
    else:
        ff.seek(p-1,0)
ff.close()
fc.close()
```

Az eredmény a `filer.txt` fájlban

```
.niaga rehtegot ytpmuH tup
t'ndluoC
nem s'gnik eht l1A
sesroh s'gnik eht l1A
llaf taerg a dah ytpmuD ytpmuH
llaw eht no tas ytpmuD ytpmuH
```

A futtató program.

A forrás fájl bináris megnyitása.

A cél fájl megnyitása létrehozásra és írásra.

A forrásban a fájl pozíció mutató

...rrásból.

...a a cél fájlba.

...jának olvasása.

...eolvasta a 0.

...ő mutatóját

Fájlok lezárása.



# Példa: ugyanez másképp

# Példa: ugyanez másképp

Az előző esetben az volt a probléma, hogy a szöveg fájlként megnyitott állományban nem lehet visszafelé (negatív offsettel) **seek**-elni.

# Példa: ugyanez másképp

Az előző esetben az volt a probléma, hogy szöveg fájlként megnyitott állományban nem lehet visszafelé (negatív offsettel) **seek**-elni.

Akkor ne **seek**-eljünk visszafelé.

# Példa: ugyanez másképp

```
#!/usr/bin/python3
ff=open("file.txt","r")
fc=open("filer.txt","w")
ff.seek(0,2)
e=ff.tell()
for p in range(e,0,-1):
    ff.seek(p-1,0)
    c=ff.read(1)
    fc.write(c)
ff.close()
fc.close()
```

A futtató program.

# Példa: ugyanez másképp

```
#!/usr/bin/python3
ff=open("file.txt","r")
fc=open("filer.txt","w")
ff.seek(0,2)
e=ff.tell()
for p in range(e,0,-1):
    ff.seek(p-1,0)
    c=ff.read(1)
    fc.write(c)
ff.close()
fc.close()
```

A futtató program.  
A fájlok megnyitása.

# Példa: ugyanez másképp

```
#!/usr/bin/python3
ff=open("file.txt","r")
fc=open("file")
ff.seek(0,2)
e=ff.tell()
for p in range(e,0,-1):
    ff.seek(p-1,0)
    c=ff.read(1)
    fc.write(c)
ff.close()
fc.close()
```

A futtató program.

A fájlok megnyitása.

A fájlok sima text fájlként vannak nyitva.

# Példa: ugyanez másképp

```
#!/usr/bin/python3
ff=open("file.txt", "r")
fc=open("filer.txt", "w")
ff.seek(0, 2)
e=ff.tell()
for p in range(e, 0, -1):
    ff.seek(p-1, 0)
    c=ff.read(1)
    fc.write(c)
ff.close()
fc.close()
```

A futtató program.

A fájlok megnyitása.

A forrás fájl pozíció mutatója a fájl legvégére.

# Példa: ugyanez másképp

```
#!/usr/bin/python3
ff=open("file.txt", "r")
fc=open("filer.txt", "w")
ff.seek(0,2)
e=ff.tell()
for p in range(e,0,-1):
    ff.seek(p-1,0)
    c=ff.read(1)
    fc.write(c)
ff.close()
fc.close()
```

A futtató program.

A fájlok megnyitása.

A forrás fájl pozíció mutatója a fájl legvégére.

Nincs negatív offset.



# Példa: ugyanez másképp

```
#!/usr/bin/python3
ff=open("file.txt","r")
fc=open("filer.txt","w")
ff.seek(0,2)
e=ff.tell()
for p in range(e,0,-1):
    ff.seek(p-1,0)
    c=ff.read(1)
    fc.write(c)
ff.close()
fc.close()
```

A futtató program.

A fájlok megnyitása.

A forrás fájl pozíció mutatója a fájl legvégére.

Az **e**-ben a fájl hossza.

# Példa: ugyanez másképp

```
#!/usr/bin/python3
ff=open("file.txt","r")
fc=open("filer.txt","w")
ff.seek(0,2)
e=ff.tell()
for p in range(e,0,-1):
    ff.seek(p-1,0)
    c=ff.read(1)
    fc.write(c)
ff.close()
fc.close()
```

A futtató program.

A fájlok megnyitása.

A forrás fájl pozíció mutatója a fájl legvégére.

Az **e**-ben a fájl hossza.

Számláló ciklus **e**-től 1-ig **p**-be.

# Példa: ugyanez másképp

```
#!/usr/bin/python3
ff=open("file.txt","r")
fc=open("filer.txt","w")
ff.seek(0,2)
e=ff.tell()
for p in range(e,0,-1):
    ff.seek(p-1,0)
    c=ff.read(1)
    fc.write(c)
ff.close()
fc.close()
```

A futtató program.

A fájlok megnyitása.

A forrás fájl pozíció mutatója a fájl legvégére.

Az **e**-ben a fájl hossza.

Számláló ciklus **e**-től 1-ig **p**-be.

Pozícionálás **p-1**-el előlről.

# Példa: ugyanez másképp

```
#!/usr/bin/python3
ff=open("file.txt", "r")
fc=open("filer.txt", "w")
ff.seek(0,2)
e=ff.tell()
for p in range(e,0,-1):
    ff.seek(p-1,0)
    c=ff.read(1)
    fc.write(c)
ff.close()
fc.close()
```

A futtató program.

A fájlok megnyitása.

A forrás fájl pozíció mutatója a fájl legvégére.

Az eredeti fájl hossza

Nincs negatív offset.

>-be.

pozícionais  $p-1$ -ei előiről.

# Példa: ugyanez másképp

```
#!/usr/bin/python3
ff=open("file.txt","r")
fc=open("filer.txt","w")
ff.seek(0,2)
e=ff.tell()
for p in range(e,0,-1):
    ff.seek(p-1,0)
    c=ff.read(1)
    fc.write(c)
ff.close()
fc.close()
```

A futtató program.

A fájlok megnyitása.

A forrás fájl pozíció mutatója a fájl legvégére.

Az **e**-ben a fájl hossza.

Számláló ciklus **e**-től 1-ig **p**-be.

Pozicionálás **p-1**-el előlről.

Az aktuális **c** karakter beolvasásáa.

# Példa: ugyanez másképp

```
#!/usr/bin/python3
ff=open("file.txt","r")
fc=open("filer.txt","w")
ff.seek(0,2)
e=ff.tell()
for p in range(e,0,-1):
    ff.seek(p-1,0)
    c=ff.read(1)
    fc.write(c)
ff.close()
fc.close()
```

A futtató program.

A fájlok megnyitása.

A forrás fájl pozíció mutatója a fájl legvégére.

Az **e**-ben a fájl hossza.

Számláló ciklus **e**-től 1-ig **p**-be.

Pozicionálás **p-1**-el előlről.

Az aktuális **c** karakter beolvasásáa.

A **c** kiírása a cél fájlba.

# Példa: ugyanez másképp

```
#!/usr/bin/python3
ff=open("file.txt","r")
fc=open("filer.txt","w")
ff.seek(0,2)
e=ff.tell()
for p in range(e,0,-1):
    ff.seek(p-1,0)
    c=ff.read(1)
    fc.write(c)
ff.close()
fc.close()
```

A futtató program.

A fájlok megnyitása.

A forrás fájl pozíció mutatója a fájl legvégére.

Az **e**-ben a fájl hossza.

Számláló ciklus **e**-től 1-ig **p**-be.

Pozicionálás **p-1**-el előlről.

Az aktuális **c** karakter beolvasásáa.

A **c** kiírása a cél fájlba.

A fájlok lezárása. .

# Példa: ugyanez másképp

```
#!/usr/bin/python3
ff=open("file.txt","r")
fc=open("filer.txt","w")
ff.seek(0,2)
e=ff.tell()
for p in range(e,0,-1):
    ff.seek(p-1,0)
    c=ff.read(1)
    fc.write(c)
ff.close()
fc.close()
```

A futtató program.

A fájlok megnyitása.

A forrás fájl pozíció mutatója a fájl legvégére.

Az **e**-ben a fájl hossza.

Számláló ciklus **e**-től 1-ig **p**-be.

Pozicionálás **p-1**-el előlről.

Az aktuális **c** karakter beolvasásáa.

A **c** kiírása a cél fájlba.

A fájlok lezárása. .



# Példa: ugyanez másképp

```
#!/usr/bin/python3
ff=open("file.txt","r")
fc=open("filer.txt","w")
ff.seek(0,2)
e=ff.tell()
for p in range(e,0,-1):
    ff.seek(p-1,0)
    c=ff.read(1)
    fc.write(c)
ff.close()
fc.close()
```

A futtató program.

A fájlok megnyitása.

A forrás fájl pozíció mutatója a fájl legvégére.

Az `e`-ben a fájl hossza.

Számláló ciklus `e`-től 1-ig `p`-be.

Pozicionálás `p-1`-el előlről.

Az aktuális `c` karakter beolvasásáa.

A `c` kiírása a cél fájlba.

A fájlok lezárása. .

# Példa: ugyanez másképp

```
#!/usr/bin/python3
ff=open("file.txt","r")
fc=open("file.txt","w")
ff.seek(0,2)
e=ff.tell()
for p in range(0,e):
    ff.seek(p)
    c=ff.read(1)
    fc.write(c)
ff.close()
fc.close()
```

Az eredmény a `filer.txt` fájlban

```
.niaga rehtegot ytpmuH tup
t'ndluoC
nem s'gnik eht l1A
sesroh s'gnik eht l1A
llaf taerg a dah ytpmuD ytpmuH
llaw eht no tas ytpmuD ytpmuH
```

A futtató program.

A fájlok megnyitása.

ja a fájl

>-be.

vasásáa.

**Egyelőre ennyi, de még lesz!**





