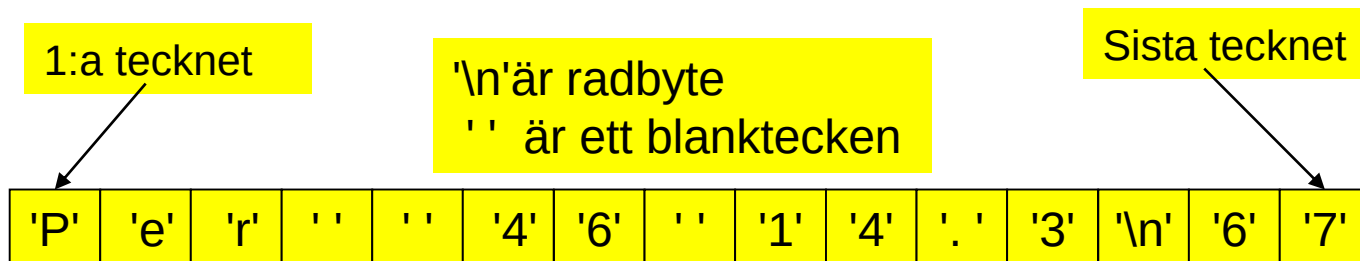


Textfil = sekvens av tecken

Så här kan en textfil se ut om vi öppnat filen i Notepad. Filen ser ut att bestå av tecken som bildar ord och rader.

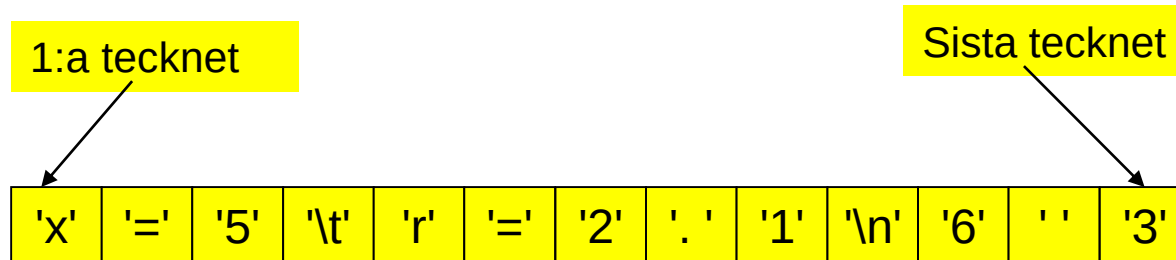
```
Per 46 14.3
67
```

Filen är egentligen en sekvens av tecken, där vissa tecken syns. Andra tecken är specialtecken som TAB och radbyte som t.ex. Notepad använder för att formatera texten, dvs så det ser bra ut på skärmen. Sekvensen av tecken är lagrad permanent i minnet (t.ex. hårddisken) som så här:



Denna sekvens av tecken kan betraktas som en **ström** av tecken.

Om textfilen består av följande **ström** av 13 st tecken:



'\t' är TAB
'\n' är radbyte
' ' är ett blanktecken

Om vi öppnar textfilen i Notepad så ser den ut så här

```
x=5      r=2 . 1
6 3
```

Notepad har formatterat filen.

Öppna och stänga filer

Vi skall nu se hur kan skapa egna strömmar och koppla dem till filer och att man läsa och skriva till dessa strömmar. Öppna en fil för läsning:

```
f = open('data.txt', 'r')
```

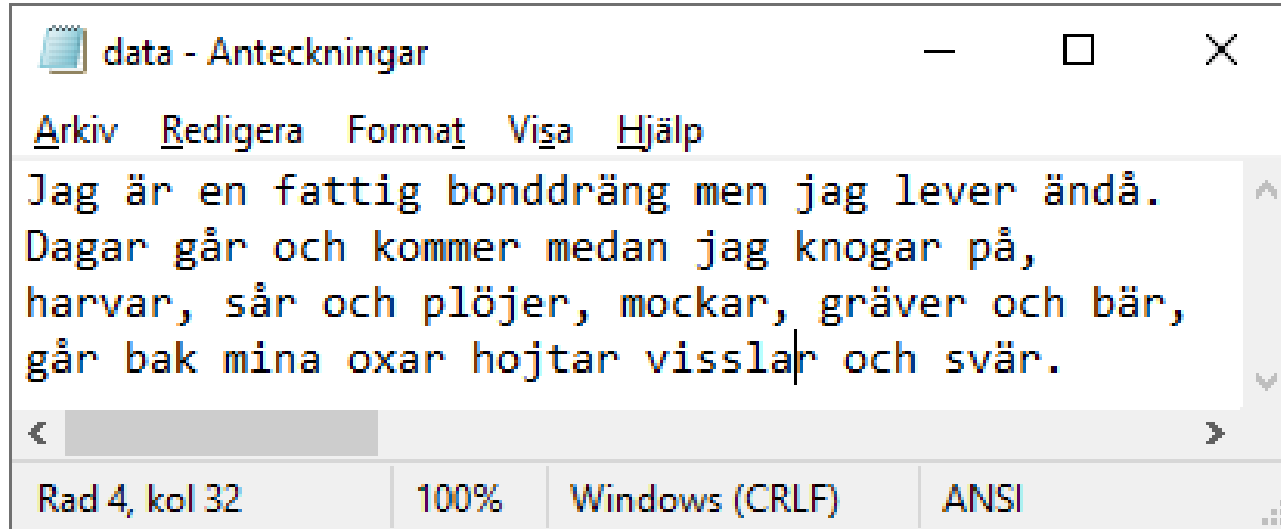
- Funktionen `open` skapar (returnerar) en referens (`f`) till ett filobjekt och `f` används för att läsa filen `data.txt`
- `'r'` betyder vi öppnar filen för läsning

När vi kör programmet fås en felutskrift:

```
No such file or directory: 'data.txt'
```

Det beror på att filen `data.txt` ej finns i den mapp där pythonfilen (programmet) finns.

Vi skapar en fil i Notepad, som ser ut så här:



Spela upp

Döper filen till [data.txt](#) och spar den i samma mapp som python-filen

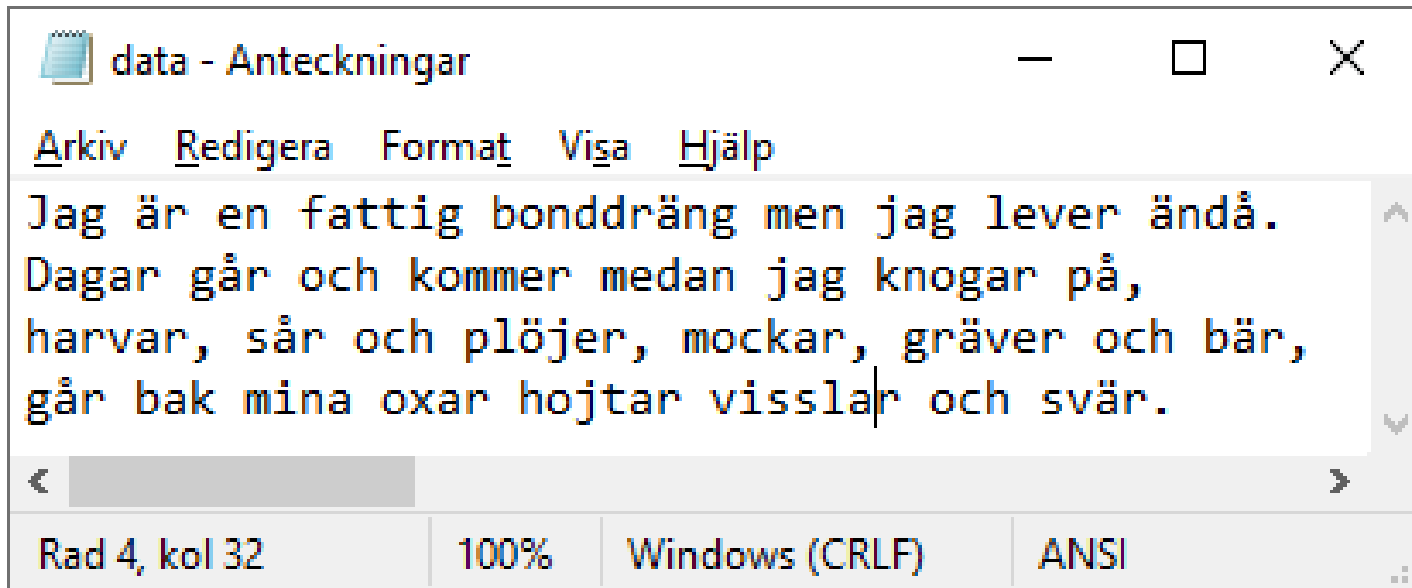
Nu går det att köra programmet utan det tidigare felet.

Men behöver lägga till satser för att läsa filen →

Vi skall läsa filen och skriva ut innehållet på skärmen:

```
# fileEx1.py
f = open('data.txt', 'r') # öppna filen
for rad in f: # för varje rad i filen
    print(rad, end=' ')
f.close()
```

- Börjar med att öppna filen för läsning. En "filpekare" är nu högst upp i filen och "pekar" på det första tecknet i den första raden i filen.
- Loopen itererar så många ggr som det finns rader i filen, dvs tills filen är slut.
- Variabeln `rad`, tilldelas den sträng av text som är aktuell rad i filen. Första varvet i loopen består strängen av:
Jag är en fattig bonddräng men jag lever ändå.\n
- `end=' '` för att slippa dubbla radbyten
- Funktionen `close` stänger filen efter loopen.



data - Anteckningar

Arkiv Redigera Format Visa Hjälp

Jag är en fattig bonddräng men jag lever ändå.
Dagar går och kommer medan jag knogar på,
harvar, sår och plöjer, mockar, gräver och bär,
går bak mina oxar hojtar visslar och svär.

Rad 4, kol 32 100% Windows (CRLF) ANSI

Kör programmet igen, vilket ger följande utskrift:

Jag är en fattig bonddräng men jag lever ändå.
Dagar går och kommer medan jag knogar på,
harvar, sår och plöjer, mockar, gräver och bär,
går bak mina oxar hojtar visslar och svär.

```
# fileEx2.py
f = open('data.txt', 'r')
rader = 0
tecken = 0
for rad in f:
    rader += 1                # Radräknare
    tecken += len(rad)        # Teckenräknare
f.close()
print('#rader, #tecken = ', rader, tecken)
```

- För varje rad i filen läser programmet raden från filen och placerar den inlästa raden (texten, dvs strängen) i variabeln `rad`, som därmed en vanlig strängvariabel.
- `for`-satsen innebär en iteration för varje rad i filen.

Programkörning:

```
#rader, #tecken = 4 179
```

Läsa teckenvis med read

```
f = open('data.txt', 'r')  
# Läs filens 5 första tecken  
s = f.read(5) # s är en sträng med 5 tecken  
print(s)
```

Jag ä

Läsa radvis med readline

```
f = open('data.txt', 'r')  
f.readline() # Läs förbi första raden  
rad = f.readline() # Läs och spara nästa rad  
print(rad) # rad inkluderar '\n')
```

Dagar går och kommer medan jag knogar på,

Läsa heltal från en fil, heltal.txt, lagra dem i en lista

Antag att man vet att filen innehåller ett tal per rad

```
# filEx3.py
f = open('heltal.txt', 'r')
# Spara talen i en lista
talList = []
for rad in f: # För varje rad i filen
    talList.append(int(rad)) #Omv tecken->heltal
f.close()
print(talList)
```

Ger:

```
[413, 410, 405, 43, 393, 394, 390, 417, 39, 404]
```

with, ett alternativ till open.

Exvis

```
with open('data.txt', 'r') as f:  
    for rad in f:
```

...

Filen stängs per auto

Allmänt: **f = open(filnamn, mode)**

mode anger hur filen skall användas:

r	läsning
w	Skrivas, om den finns skrivs den över, annars skapas ny
x	Skrivas, får ej finnas tidigare
a	som w, men lägger till från slutet om filen redan finns

Sammanfattning

```
f.read(n)
```

Läser `n` st tecken från filen `f`. Om `n` är negativ, läses hela filen.
Returnerar `None` vid filslut.

```
f.readline()
```

Läser en rad i filen. Returnerar `None` vid filslut.

```
for rad in f:
```

Läser en rad i taget till rad. Som upprepade `readline`

```
f.readlines()
```

Läser alla raderna i filen. Ger en lista med alla raderna, dvs elementen är resp rad.

```
f.write(s)
```

Skriver ut `s` till filen `f`. Returnerar antal skrivna tecken.

```
print(..., file=f)
```

Som vanliga `print` på skärmen, men skriver till filen `f`.