

## Programmeringsteknik



Föreläsning 5

## Idag



- ☞ Kap 7 i Dawson
- ☞ Filer
- ☞ Felhantering med try...except

## Filer



- ☞ Nästan alla program använder filer.
- ☞ Exempel:
  - ☞ Mailprogram
  - ☞ Webbbläsare
  - ☞ IDLE
- ☞ Enda sättet att spara data mellan programkörningar!
- ☞ Filer kan innehålla olika typer av data men vi kommer mest att titta på *textfiler*.

## Filvariabler



- ☞ En filvariabel lagrar
  - ☞ Sökvägen till filen
  - ☞ Vilken mod som ska användas (infil/utfil)
  - ☞ Aktuell position i filen

## Läsa från fil



- ☞ Att hämta data från en fil kallas läsning..
- ☞ Det här behöver man kunna:
  - ☞ Öppna filen för läsning
 

```
infil = open('C:\bok.txt', "r")
```
  - ☞ Läsa in en rad från en fil
 

```
rad = infil.readline()
```
  - ☞ Läsa in alla rader
 

```
lista = infil.readlines()
```
  - ☞ Stänga filen
 

```
infil.close()
```

"r" för read = läsning

## Uppgift: Rätta felen!



```
namn = "s.txt"      ■ namn = s.txt
fil = open(namn, r) ■ fil = open(namn, "r")
rad = readline(fil) ■ rad = fil.readline()
fil.close()         ■ close(fil)
```

## Skriva ut på fil



- ☞ Att spara data på fil kallas skrivning.
- ☞ Det här behöver man kunna:
  - ☞ Öppna filen för läsning eller skrivning:
 

```
utfil = open("C:\nytt.txt", "w")
```
  - ☞ Skriva ut en rad på en fil
 

```
utfil.write("Natten var mörk.")
```
  - ☞ Skriva ut en lista på en fil
 

```
utfil.writelines(lista)
```
  - ☞ Stänga filen
 

```
utfil.close()
```

☞ för write = skrivning

## Läsa med for-slinga



- ☞ I Python kan man läsa igenom hela filen med en for-slinga. Exempel:
 

```
fil = open("universitet.txt")
antalOrd = 0
for rad in fil:
    orden = rad.split()
    antalOrd += len(orden)
print(antalOrd, "ord i filen.")
```

## Exception



- ☞ Fel att öppna en fil för läsning om den inte finns.

- ☞ Felet heter IOError:

```
>>> fil = open("kudde.txt", "r")
Traceback (most recent call last):
File "<pyshell#2>", line 1, in ?
fil = open("kudde.txt", "r")
IOError: [Errno 2] No such file:
'kudde.txt'
```

## Olika feltyper

Exception	När uppkommer det?
IOError	Om man försöker öppna en fil som inte finns.
SyntaxError	Om man skrivit programkod som Python inte kan tolka.
KeyError	Om man försöker använda en nyckel som inte finns i en ordlista.
NameError	Om man använder en variabel utan att ha gett den ett värde .

## try-except



- ☞ Man kan ta hand om fel genom att införa try-except-satser. Exempel:

```
def öppna(namn, mod = "r"):
    """Öppna en fil säkert för
    läsning eller utskrift."""
    try:
        fil = open(namn, mod)
        return fil
    except (IOError):
        print("Kan inte öppna filen", namn)
```

## Uppgift: Varför?



- Utan try...except går det inte att läsa från fil.
- Programmet går snabbare
- Man slipper stänga filen med close
- Man kan få felutskrift på svenska

## sortering.py



- ☞ På textfilen universitet.txt finns namnen på ett antal universitet lagrade:  
*University of Cape Town*  
*St. Petersburg State University*  
*Massachusetts Institute of Technology*  
 ...
- ☞ Programmet ska läsa in alla rader från filen, konvertera till gemena, sortera dom i bokstavsordning, och skriva ut på en annan fil

## Uppdelning i funktioner



- ☞ öppna(filnamn, mod)
  - ☞ Öppnar en fil.
  - ☞ Indata: filnamnet (sträng) Utdata: en filvariabel
- ☞ läsFil(filvariabel)
  - ☞ Läser alla rader från filen.
  - ☞ Indata: filvariabel Utdata: lista med gemena rader
- ☞ skrivFil(filvariabel, lista)
  - ☞ Skriver ut listan på filen
  - ☞ Indata: filvariabel, lista med strängar Utdata: -

```
def läsFil(infil):
    """Läs in alla rader från infil"""
    lista = []
    for rad in infil:
        lista.append(rad.strip().lower())
    infil.close()
    return lista
```

```
def skrivFil(lista, utfil):
    """Skriver ut lista på utfil"""
    for rad in lista:
        utfil.write(rad + "\n")
    utfil.close()
```

```
def main():
    infil = öppna("universitet.txt", "r")
    lista = läsFil(infil)
    lista.sort()
    utfil = öppna("sorterat.txt", "w")
    skrivFil(lista, utfil)

#Anropar main
main()
```