

DD1311 Programmeringsteknik med PBL



Föreläsning 6

Idag



- ☞ Kap 7 i Dawson
- ☞ Filer
- ☞ Särfall (exceptions)
- ☞ Konservering (pickling)

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*.

Filer



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


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


```
rad = infil.readline()
```
 - ☞ Skriva ut på en fil


```
utfil.write("Natten var mörk.")
```
 - ☞ Stänga filen


```
utfil.close()
```

"r" för läsning, "w" för skrivning

Funktioner/metoder för filer

Skapa filvariabel, öppna filen	<code>fil = open(filnamn, mod)</code>
Läs en rad/ lista med rader	<code>rad = fil.readline()</code> <code>lista = fil.readlines()</code>
Skriv en rad/ lista med rader	<code>fil.write(rad)</code> <code>fil.writelines(lista)</code>
Stäng filen	<code>fil.close()</code>

Läsa hela filen



- ☞ 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.")
```

Särfall



☞ 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'
```

`särfall=exception`

Olika särfall

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

try-except



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

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

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, sortera dem i bokstavsordning, och skriva ut på en annan fil

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

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

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

#Anropar main
main()
```

Läsa webbsidor



☞ Man kan läsa från en webbsida på ungefär samma sätt som från en fil:

```
import urllib
url = "http://www.kth.se"
websida = urllib.urlopen(url, "r")
for rad in websida:
    print rad
```

Konservering



☞ En hel datastruktur lagras på fil så här:

```
import cPickle
strumplista = ["gula", "vita", "ylle"]
konserv = open("strumpor.txt", "w")
cPickle.dump(strumplista, konserv)
konserv.close()
```

☞ Och hämtas igen:

```
infil = open("strumpor.txt", "r")
x = cPickle.load(infil)
```

konservera = pickle

