

Lektion 6

LT1023
Programmeringsteknik

Exempel

```
# split.py
```

```
mening = 'Liten tuva välter ofta stort lass!'
orden = mening.split(' ')
for ord in orden:
    print (ord)
```

Innehåll

- Strängmetoden *split()*
- Felhantering
- Filhantering

Felhantering

- Används för att undvika att ett program kraschar i olika situationer
- Det finns många fördefinierade "Error" i Python t ex
 - *ValueError* – t e x om en sträng inte kan tolkas som ett tal vid konvertering
 - *NameError* – då ej deklarerad variabel används
- När ett fel uppstår kan man välja mellan att "kasta det" eller att hantera det.
- Kasta
 - Som standard kastas felet till anropande metod om inget skrivs.
 - Om man har flera anropsnivåer sker detta hela vägen upp till huvudprogrammet, om man inte hanterat felet där heller kraschar programmet.
- Hantera
 - Med *try*: inkapslar man den "känsliga" koden.
 - Med *except*: "fångar" man det fel som skett samt vidtar åtgärd, p s s slipper man exekveringsfel.

split()

- Metoden *split()* i klassen *str* används för att dela upp en sträng i delar (tokens), t e x ta fram orden ur en mening.
- Tecknet som ska dela strängen skickas med som parameter
- Returtypen är en lista med beståndsdelarna, t e x orden

Exempel

```
# exception.py
fel = True
while fel:
    fel = False
    ålderText = input('Hur gammal är du? ')
    try:
        ålder = int(ålderText)
    except ValueError:
        print ('Det här är inget helta!!')
        fel = True
```

Filer

- Vill man spara data mellan programkörningar måste man lagra data på fil. Läsning från fil har många likheter med att läsa från tangentbordet och skrivning till fil har många likheter med att skriva till skärmen.
- Man öppnar en fil för läsning / skrivning med funktionen `open()`.
- När man arbetar klart med filen stänger man den för vidare inläsning / utskrift med metoden `close()`

priser.txt

```
fiskpinnar/26
yoghurt/12
mjölk/8
ägg/22
kakor/16
bröd/24
```

Exempel

```
# Filen 'infil.txt' förutsätts finnas i samma mapp som programmet
# Filen 'utfil.txt' skapas om den ej existerar, skrivs annars över
```

```
infil = open('infil.txt','r') # parametern 'r' står för read
utfil = open('utfil.txt','w') # parametern 'w' står för write
```

```
rad = infil.readline() # läser första raden från filen
print (rad)
rad = infil.readline() # läser andra raden från filen
print (rad)
infil.close()
```

```
namn = input('Vad heter du? ')
utfil.write(namn)
utfil.close()
```

Exempel

```
kundvagn = {'mjölk':1,'ägg':1,'bröd':2}
```

```
infil = open('priser.txt', 'r', encoding='latin-1') #utf-8
rad = infil.readline()
rad = rad.rstrip('\n')
prislista = {}
while rad != "":
    raddelar = rad.split('/')
    prislista[raddelar[0]] = float(raddelar[1])
    rad = infil.readline().rstrip('\n')
```

```
summa = 0
for nyckel in kundvagn.keys():
    print (nyckel, 'Pris/enhet:', prislista[nyckel], 'Antal:', kundvagn[nyckel])
    summa += prislista[nyckel]*kundvagn[nyckel]
```

```
print ('Summan blir', summa, 'kr')
```