Skolan för Datavetenskap och kommunikation

DD1311 Programmeringsteknik med PBL

Föreläsning 13

Bra att ha till P-uppgiften:

- Idag:
  - Metodnamn
  - Sortering
  - Kopiera en lista

Nyheter

- CL & DL:
  - labbar torsdagar 10-12
  - gruppmöten torsdagar 15-16
  - Strukturering på gruppmötet
- S:
  - Oregano i D35
  - Persilja i D41
  - Specredovisning på gruppmötet
  - Ny enkät på webbsidan!

Metodnamn

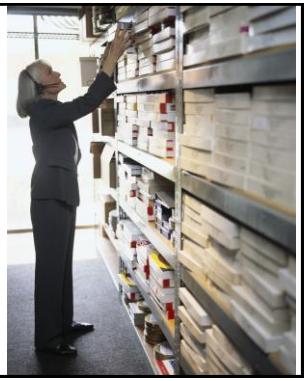
- En åtkomstmetod returnerar värdet av ett attribut.
- Kan den ha samma namn som attributet?
- Nej, då blir det **fel** (t ex AttributeError eller TypeError)

Fel	Rätt
`def namn(self):` `  return self.namn`	`def ge_namn(self):` `  return self.namn`

Sortering

Man kan sortera böcker efter t ex

- Författare
- Titel
- Ämne



Sortering

- Python-listor har ju en inbyggd sort-metod:
  `lista.sort()`
- En lista med tal sorteras i stigande ordning, en lista med strängar sorteras i bokstavsordning. ..
- Men om man har en lista av objekt - vilket attribut sorterar `sort()` på då?
- Ta t ex Vitaminklassen...

Vitaminsortering

- Skapar tre objekt
- Lägger dom i en lista
- Skriver ut listan
- Sorterar efter D-vitamin-halt
- Skriver ut den sorterade listan

```
class Vitamin(object):
    """Representerar ett livsmedel"""
    def __init__(self, namn, a, d):
        self.a = a
        self.d = d
        self.namn = namn

    def __str__(self):
        return str(self.a)+str(self.d)

    def aHalt(self):
        return self.a

    def dHalt(self):
        return self.d
```

Parametrar till sort

- Nu kan `sort` anropas med parametern *key* satt till åtkomstmetoden för det vi vill sortera på!
- Parametern blir *Klass.metodnamn*

  `listan.sort(key=`**`Vitamin.aHalt`**`)`
- För att vända på ordningen kan vi använda parametern reverse.

  `listan.sort(key=Vitamin.aHalt,`
  `reverse=True``)`

```
#Huvudprogram
v1 = Vitamin("Morötter",1600,0)
v2 = Vitamin("Kantareller",1300,13)
v3 = Vitamin("Mjölk",26,0.38)
lista = [v1,v2,v3]

for v in lista:
    print(v)

lista.sort(key = Vitamin.dHalt)
for v in lista:
    print(v)
```

Urvalssortering

Ibland vill man hellre göra en egen sorteringsfunktion, och då är urvalssortering enklast:

- Sök igenom listan efter minsta värdet
- Byt plats på minsta och första värdet
- Nu är första elementet klart
- Gör om samma sak med resten av listan

```
def urvalssortera(lista):
   n = len(lista)
   for i in range(n):
      minst = i
      for j in range(i+1,n):
         if lista[j] < lista[minst]:
            minst = j
      lista[minst],lista[i] =
      lista[i], lista[minst]
    return lista
```

Moduler - dokumentation

- Allt står inte i boken...
- Hur vet man vilka moduler som finns?
- Titta på sidan:
  http://docs.python.org/release/x.y/
  där x.y är versionsnummer, 3.1 eller 2.7
- Under Global Module Index finns en lista med länkar till dokumentation för alla moduler.

Exempel: modulen **copy**

This module provides generic (shallow and deep) copying operations.

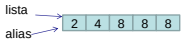
Interface summary:

```
import copy

x = copy.copy(y) #make a shallow copy of y
x = copy.deepcopy(y) #make a deep copy ofy
```

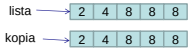
Kopiera en lista

```
alias = lista
```



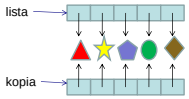
Vill man ha en *kopia* av hela listan kan man använda `copy` i modulen `copy`:

```
import copy
kopia = copy.copy(lista)
```



Kopiera en lista av objekt

Om det är objekt i listan kopieras *referenserna* till varje objekt!

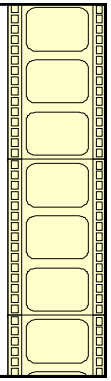


Använd deepcopy för att istället göra kopior av objekten:

```
import copy
kopia = copy.deepcopy(lista)
```

Större exempel

- Programmet filmer.py:
  - läser in namn, kategori, och längd (i minuter) för en YouTube-film
  - man får ge betyg på filmerna
  - och välja på vilket sätt dom ska sorteras



Föreläsningen på torsdag?

- Formatterad utskrift (snygga tabeller)
- Avlusning
- Testning