


DD1310/DD1314/DA3009 PROGRAMMERINGSTEKNIK

FÖRELÄSNING 4

DATASTRUKTURER

Kap 4-5 i Dawson

- Strängar 'N' 'O' 'U' 'G' 'A' 'T'
- Listor ['blå', 'grön', 'röd', 'gul']
- Tupler ('blå', 'grön', 'röd', 'gul') 
- Dictionary {'blå': 'grön', 'röd': 'gul', 'blå': 'vert', 'rouge': 'jaune'}

STRÄNGMETODER

Metod	Exempel
<code>upper()</code>	"kanin" -> "KANIN"
<code>lower()</code>	"KANIN" -> "kanin"
<code>swapcase()</code>	"KaniN" -> "kANin"
<code>capitalize()</code>	"kanin" -> "Kanin"
<code>title()</code>	"liten blå kanin" -> "Liten Blå Kanin"
<code>strip()</code>	" kanin " -> "kanin"
<code>replace(x, y)</code>	"kanin" -> "kinin" (om x="a",y="i")

PYTHONS DOKUMENTATION

Hur kan man veta vilka funktioner och metoder som finns i python?

- På <http://docs.python.org/3/> finns dokumentation
- Se även "Pythonlänkar" på kurshemsidan

METODANROP

- Så här anropas en metod:

```
mening=input("Skriv en föreläsning: ")  
print("Så här ser den ut med versaler:")  
print(mening.upper())
```

metod = method
anropa = invoke

STRÄNGMETODEN SPLIT()

- Det är enkelt att dela upp en mening i ord:
`mening.split()`
- Metoden split delar vid mellanslag
- Varje del läggs i en numererad låda
- Numret kallas *index*



FOR-SLINGAN I REPRIS

Exempel	Vad blir <i>i</i> ?
<pre>for i in range(5): print(i)</pre>	Heltal
<pre>for i in "abcde": print(i)</pre>	Bokstav
<pre>mening = "en dag som idag" for i in mening.split(): print(i)</pre>	Ord

STRÄNGAR - INDEX

- Varje tecken i en sträng har också index.
- Exempel:

mat="pizza"

p	i	z	z	a
0	1	2	3	4

mat[0]	mat[1]	mat[2]	mat[3]	mat[4]
"p"	"i"	"z"	"z"	"a"

STRÄNGAR: SKIVNING

mat[0] → "p"	Titta går bra...
mat[0] = "m"	...men inte ändra värdet.
mat[1:3] → "iz"	Delsträng (3:e ingår inte).
mat[4:] → "pizz"	Från början till 4.
mat[2:] → "zza"	Från 2 till slutet.

strängskivning = string slicing

UPPGIFT: VAD BLIR S[3:]?

```
s = "aktie"  
print(s[3:])
```

akt

ie

t

tie

ALGORITM FÖR ATT SKAPA ANAGRAM

1. Börja med ett tomt anagram
2. Slumpa ett index i ursprungsordet
3. Kopiera bokstaven från den platsen och lägg till den till anagrammet
4. Plocka bort den från ursprungsordet
5. Upprepa punkt 2-4 så länge som det finns bokstäver kvar i ursprungsordet

EXEMPEL

slumpad plats	namn = "LISA"	anagram = ""
1	"LSA"	"I"
2	"LS"	"IA"
1	"L"	"IAS"
0	""	"IASL"

ANAGRAM

```
import random
namn = input("Vad heter du? ")

anagram = ""
while namn != "":
    antal = len(namn)
    index = random.randrange(antal)
    anagram = anagram + namn[index]
    namn = namn[:index] + namn[(index+1):]
    print(index, namn, anagram)

print("Ett anagram på ditt namn är ", anagram)
```

LISTAN

- En följd av data, omgiven av hakparenteser
tom = []
primtal = [2, 3, 5, 7, 11, 13, 17, 19]
dagar = ["tisdag", "onsdag", "fredag"]
- Kan innehålla data av valfri typ
- Har index precis som strängar
- Och en massa metoder

NÅGRA LISTMETODER

Metod	Beskrivning
append(x)	Lägger till x sist i listan.
sort()	Sorterar i stigande ordning.
reverse()	Vänder listan.
insert(i,x)	Stoppar in x på plats i.
pop(i)	Plockar ut elementet på plats i.
remove(x)	Tar bort första förekomsten av x.

REFERENSER

- En listvariabel är en *referens* till listan. Tilldelning ger inte en kopia.

```
dagar = ["tisdag", "onsdag", "fredag"]
vecka = dagar
```

```
vecka → tisdag onsdag fredag
dagar → tisdag onsdag fredag
```

UPPGIFT: HUR SER DAGAR UT SEN?

```
dagar = ["tisdag", "onsdag", "fredag"]
vecka = dagar
vecka[0] = "måndag"
```

■ ["tisdag", "onsdag", "fredag"]

■ ["måndag", "onsdag", "fredag"]

Läsa in till en lista

```
# Frågar efter ord som rimmar och läser
# in i en lista, som sedan skrivs ut sorterad.
lista = []
ord = input("Vad rimmar på hus? ")
while ord != "":
    lista.append(ord)
    ord = input("Kan du ett till? ")
print("Du hittade", len(lista), "ord!")
lista.sort()
print(lista)
```

DICTIONARY

- Skapas med klamrar {}:
`engelska = {}`
- Lagrar datapar:
`engelska['trollstav'] = 'wand'`
- Varje datapar har en *nyckel* och ett *värde*.
- Nycklarna kan t ex vara strängar eller tal
- Värdena kan vara av vilken typ som helst.
- En dictionary har inte ordning!

nyckel = uppslagslista
värde = value

DICTIONARY - METODER

Metod	Beskrivning
<code>if n in d:</code>	Kollar om nyckeln n finns i d
<code>d[n] = x</code>	Stoppar in x med nyckel n
<code>print (d[n])</code>	Skriver ut värdet med nyckel n
<code>keys ()</code>	Ger alla nycklar
<code>values ()</code>	Ger alla värden
<code>pop (n)</code>	Plockar ut det värde som har nyckel n

UPPGIFT: GIVET D = {} VILKA SATSER ÄR GILTIGA?

- `d[5] = "fem"`
- `d["fem"] = 5`
- `d["5"] = 5`
- `d[fem] = "5"`

LÄXA 2

Se "Labbar, föreläsningar och övningar" på kurswebbsidan.

- I labb 3 ska ni skriva ett glosförhörprogram.
- Besök <http://pythontutor.com/visualize.html#py=3>, klistra in det givna programmet och kör steg för steg så att du förstår vad som händer.
- Bygg på programmet med en funktion som genomför ett glosförhör (det är OK att dela upp i flera funktioner).
- Läs även Kap 7 i kursboken (om filer) till nästa vecka.