

Programmeringsteknik

Föreläsning 4

Datastrukturer

Kap 4-5 i Dawson

- Strängar
- Listor
- Tuple
- Dictionary

Sträng - exempel

En sträng består av noll eller flera tecken:

```
blodkärl = "artär"
```

```
mening="I can haz Cheezburger?"
```

```
årtal = "2012"
```

```
mellanslag = "      "
```

```
tomSträng = ""
```

Lista - exempel

En lista består av noll eller flera värden:

```
tomLista = []
```

```
blod = ["artär", "ven", "kapillär"]
```

```
 dagar = [31, 29, 31, 30, 31, 30]
```

Tuppel - exempel

Som en lista, men oföränderlig:

```
blod = ("artär", "ven", "kapillär")
```

```
 dagar = (31, 29, 31, 30, 31, 30)
```

Dictionary - exempel

Lagrar noll eller flera datapar:

```
färger = {"artär":"röd",
```

```
         "ven":"blå", "kranskärl":"röd"}
```

```
dagar = {"jan":31, "feb":29}
```

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")

Metodanrop

- Så här anropas en metod:

```
mening=input("Skriv en förolämpning: ")
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 numrerad låda
- Numret kallas *index*



for-slingan i repris

Exempel	Vad blir 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

<code>mat[0]</code>	<code>mat[1]</code>	<code>mat[2]</code>	<code>mat[3]</code>	<code>mat[4]</code>
"p"	"i"	"z"	"z"	"a"

Strängar: skivning

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

strängskivning = string slicing

Uppgift: Vad blir s[3:]?

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

■ akt

■ ie

■ t

■ tie

Listan

- En följd av data, omgiven av hakparenteser
tom = []
printal = [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.

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!

dictionary = uppslagslista

Använda en dictionary

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