

DD1311 Programmeringsteknik med PBL

Föreläsning 3

- Slumptal
 - modulen `random`
- En ny *styrstruktur*:
 - for-slingan
- En *datastruktur*:
 - strängar

Slumptal

- Oförutsägbara program är roligare!
- Slumptal får man genom att
 1. Importera `random`-modulen med satsen `import random`
 2. Anropa någon av `random`-funktionerna, till exempel `random.randint(1,100)`

slumptal=random number

Random-funktioner

Funktion	Resultat
<code>randint(1,6)</code>	Slumpar ett heltal: 1,2,3,4,5 eller 6
<code>randrange(1,6,2)</code>	Slumpar ett udda heltal: 1,3 eller 5
<code>random()</code>	Slumpar ett decimaltal mellan 0.0 och 0.999...
<code>choice(("lo","uv"))</code>	Slumpar ett värde bland de givna: "lo" eller "uv"

Talgissning 3.0

```
# Talgissning, version 3.0
import random
tal = random.randint(1,100)
gissning = input("Gissa mitt tal: ")
while gissning != tal:
    if gissning > tal:
        gissning = input("Lägre:")
    elif gissning < tal:
        gissning = input("Högre:")
print "Bravo, du gissade rätt!"
```

for-slingan

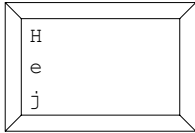
En for-slinga upprepar ett antal satser för varje element i en följd. Exempel:

- För varje telefonnummer på listan:
 - Ring upp personen!
- För varje chokladbit i asken:
 - Ät upp den!
- För varje bokstav i en sträng:
 - Skriv ut bokstaven!

följd=sequence

Exempel 1

```
for tkn in "Hej":  
    print tkn
```



Hur skapar man talföljder?

Funktionen range() ger en lista av heltal.

range(10) ger [0,1,2,3,4,5,6,7,8,9]

Man kan också ange både start och slut:

range(8,15) ger [8,9,10,11,12,13,14]

Och även steg:

range(44,55,3) ger [44,47,50,53]

Uppgift: Vad händer här?

```
summa = 0  
for i in range(5):  
    summa = summa + i  
print summa
```

Strängar - index

- Varje tecken i en sträng är numrerad, med början från noll.
- Numret kallas *index*.
- Exempel:

mat="pizza"

p	i	z	z	a
0	1	2	3	4

- Varje fack i strängen fungerar som en egen variabel, som man når via index inom hakparenteser [].

Strängar: skivning

mat[0] → "p"	Man kan titta vad som finns på platsen...
mat[0] = "m"	men inte ändra värdet.
mat[1:3] → "iz"	En delsträng – från och med 1 upp till (men inte med) 3.
mat[:4] → "pizz"	Utelämnat tal före kolon betyder början av strängen.
mat[2:] → "zza"	Utelämnat tal efter kolon betyder slutet av strängen.

strängskivning = string slicing

Algoritm för att skapa anagram

- Börja med ett tomt anagram
- Så länge som det finns bokstäver kvar i ursprungsordet:
 1. Slumpa en plats i ursprungsordet
 2. Ta bokstaven på den platsen
 3. Lägg till den till anagrammet
 4. Plocka bort den från ursprungsordet

Exempel

slumpad plats	ordet = "citron"	anagram = ""
4	"citrn"	"o"
3	"citn"	"or"
1	"ctn"	"ori"
2	"ct"	"orin"
0	"t"	"orinc"
0	""	"orinct"

```
# Anagramprogram
import random
namn = raw_input("Vad heter du? ")

# Skapa en tom sträng
anagram = ""
# Så länge som ursprungsordet inte är tomt
while ordet:
    # Slumpa en plats i ordet
    plats = random.randrange(len(ordet))
    # Ta bokstaven där och lägg den i trassel
    anagram = anagram + ordet[plats]
    # Plocka bort bokstaven ur ursprungsordet
    ordet = ordet[:plats] + ordet[(plats+1):]

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