

Kapitel 3: Starta programmering på riktigt

I den andra övningen i kapitel 3 kommer du att upptäcka hur du upprepar en uppsättning instruktioner genom att använda den avgränsade loopen FOR.

Slumpmässigt urval

En ogenomskinlig bag innehåller sex röda och fyra gröna bollar. Vi drar slumpmässigt en boll från påsen, noterar dess färg och lägger sedan tillbaka den i påsen.

Programmera en **funktion som simulerar** experimentet med bollarna med en slumpmässig variabel.

- Vilka är de möjliga värden variabeln kan ha?
- Vi vill skriva ett skript som kan skilja de röda bollarna från de gröna med hjälp av ett test.

Starta ett nytt skript och döp det till "DRABOLL".

När du arbetar med slumpantal krävs inläsning av ett bibliotek för arbete med slumpantal. För att göra detta, tryck på f1 (Fns...) och välj i menyn Modul och sedan 2: Random....

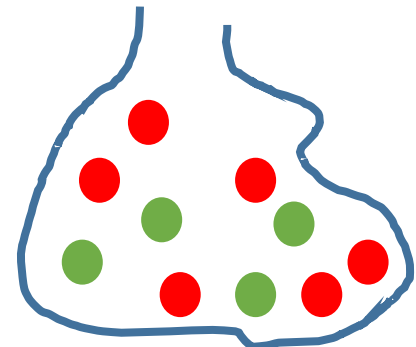
Skriv in skriptet här till höger i editorn och se till att ha de indrag som visas. Det är oerhört viktigt i Python.

Till höger ser vi konsolen några körningar av det korta skriptet. Om slumpvariabeln X får värdet 1 till och med 6 visas "grön". Och om slumpvariabeln X får värdet 7 till och med 10 fås svaret "röd".

Lärarkommentar: Detta skript kan modifiera för vilket antal bollar som helst. man kan t.ex. definiera **färg(n,r)** där n är antalet bollar och r är de röda bollarna.

Övning 1: Funktioner och loopar
Syfte:

- Använda en funktion
- Upptäcka och implementera den avgränsade loopen For
- Använda for-loopen enkla exempel



```

EDITOR: DRABOLL
PROGRAM LINE 0001
# dra bollar
from random import *
def färg():
    X=randint(1,10)
    if X<=6:
        f="röd"
    else:
        f="grön"
    return f
    
```

```

PYTHON SHELL
>>> färg()
'röd'
>>> färg()
'grön'
>>> färg()
'röd'
>>> färg()
'grön'
>>> färg()
'grön'
>>> |
    
```

Vi ska nu tillämpa våra kunskaper genom ett exempel på stickprov och beslutsfattande utifrån stickprovet.

En undersökning genomfördes vid lanseringen av en ny show som föreslagits av en fransk konstnär. Denna undersökning som genomförts i en stor stad visade att två tredjedelar av de som såg showen gillade den. Konstnärens agent ansåg att hela den franska befolkningen hade samma uppfattning. Han beställde en undersökning från ett opinionsinstitut för att kontrollera detta.



Befolkningsstudie:

Vid den statistiska undersökningen måste undersökningsinstitutet skapa en funktion som simulerar svaret på situationen.

Vi ska nu skapa denna funktion i enlighet med följande specifikationer:

- Showen uppskattades med en sannolikhet $p = 2/3$.
- Showen uppskattas inte med sannolikheten $p = 1/3$

Starta ett nytt skript och namnge det STPROV (förkortning av stickprov) Skriv den här funktionen i editorn och testa den flera gånger genom att trycka på f4 (Run) och tryck sedan på `vars` och väljs sedan `fråga()` genom att bekräfta med att klicka Ok. Tryck sedan på `enter`. Du får svaret 1 eller 0.

```

EDITOR: STPROV
PROGRAM LINE 0001
from random import *
def fråga():
    s=randint(0,2)
    if s==0 or s==1:
        svar=1
    else:
        svar=0
    return svar

```

Upprepa nu detta ett antal gånger. Det går fortast om du trycker på `▲` och trycker på `enter`. Om du gör detta många gånger bör du få ettor ungefär två tredjedelar av gångerna.

```

PYTHON SHELL
>>> fråga()
1
>>> fråga()
0
>>> fråga()
1
>>> fråga()
1
>>> fråga()
1
>>> |

```

Göra en stickprovsundersökning med storleken n

Opinionsinstitutet vill nu göra en simulering där man gör undersökningar med olika värden på storleken n . För att göra detta så måste vi skapa en stickprovsfunktion i skriptet som frågar efter stickprovsstorleken n .

- Vi börjar med att skapa en tom lista `L`.
- Vi fyller denna lista genom att använda funktionen `fråga()`.

```

EDITOR: STPROV
PROGRAM LINE 0013
if s==0 or s==1:
    svar=1
else:
    svar=0
return svar
def stickprov(n):
    L=[]
    L=[fråga() for i in range(n)]
    return L

```

Lärarkommentar: Med Python-språket kan du använda en funktion för att fylla en lista som ökas i steg av en For-loop, som argument i listan. Det här exemplet kan uppnås utan användning av listor. Skriptet måste då ändras något genom att listinstruktionerna ersätts med avgränsade loopar som stegvis ökar en variabel.

Vi ska nu prova stickprovfunktionen genom att göra ett test med storleken 30.

- Tryck på Run och tryck sedan på tangenten $\frac{1}{2}$ och välj sedan stickprov(). Bekräfta med Ok.
- Fyll i stickprovstorleken och tryck på Í. Nu får vi en lista med nollor och ettor på skärmen.

I exemplet på skärmen får vi 17 ettor. Kör nu några gånger och räkna antalet ettor.

```

PYTHON SHELL
>>> # Shell Reinitialized
>>> # Running STPROV
>>> from STPROV import *
>>> stickprov(30)
[1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1,
, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1,
0, 0, 1, 0, 1, 0, 1, 1, 1]
>>> |
    
```

Vad händer om vi kör skriptet 1000 gånger?

Vi får hela skärmen full av nollor och ettor. Vi måste hitta ett sätt att titta på andelen ettor. Istället för att returnera listan med satsen return L ska vi nu istället returnera ett värde. Värdet vi ska ha utskrivet vid körning är andelen ettor i listan.

```

PYTHON SHELL
0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1
, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0,
0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1,
1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1
, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0,
1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1
, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1,
1, 0, 1, 1, 0, 0, 0, 1]
>>> |
    
```

Tryck f1 (Fns...) och sedan List. Där finns nu ett antal funktioner för listor. Det uttryck vi nu ska skriva är

return sum(L)/len(L)

och instruktionen **sum** ger den aritmetiska summan, dvs vi adderar alla ettor. **len** ger antalet element i listan.

Körning av programmet

Tryck på Run och sedan på **vars** för att välja stickprov(). Titta både på små och stora stickprov och kör programmet ca 10 gånger. Det går snabbt om du efter varje körning trycker på **↵** för att göra en ny.

```

EDITOR: STPROV
PROGRAM LINE 0004
*if _==0 or s==1:
*   svar=1
*else:
*   svar=0
*return svar
def stickprov(n):
*   L=[]
*   L=[fråga() for i in range(n)
   ]
*   return sum(L)/len(L)
*
    
```

Vad kan du säga om resultaten? Vilka slutsatser kan du dra?

```

PYTHON SHELL
>>> # Shell Reinitialized
>>> # Running STPROV
>>> from STPROV import *
>>> stickprov(1000)
0.666
>>> stickprov(1000)
0.6800000000000001
>>> stickprov(1000)
0.6959999999999999
>>> |
    
```

Lärarkommentar: Vi ska nu titta lite närmare på osäkerheten bakom stickprov på andelar. Man stöter ofta på detta i samband med undersökningar om partisympatier.

Vi antar att stickprovsstorleken är 1000 personer, ungefär vad man brukar ha i opinionsundersökningar. Frågan är vad man kan förvänta sig om hur nära andelen i stickprovet kommer andelen i hela befolkningen. Vi kallar den för p .

Vi börjar här med teorin och genomför sedan några undersökningar med slumpen. Man kan visa att i 19 fall av 20 (95 % konfidensintervall) så hamnar stickprovets andel i intervallet

$$p \pm 1,96 \cdot \sqrt{\frac{p(1-p)}{n}} \quad \text{där } n \text{ är stickprovsstorleken.}$$

Vi antar den *sanna* proportionen i populationen är 65 % och stickprovsstorleken 1000. Då får vi

$$0,65 \pm 1,96 \cdot \sqrt{\frac{0,65 \cdot (1-0,65)}{1000}} \approx 0,65 \pm 0,030$$

Vi får alltså då en felmarginal på 3 %. Av uttrycket ser vi att felmarginalen beror både på andelen och stickprovsstorleken.

Vi har gjort en plottning för $p = 0,65$ i marginalen. Man ser hur osäkerheten ökar kraftigt när stickprovsstorleken minskar.

På <https://ti-resurser.com/materialdatabas> finns en aktivitet, Opinionsundersökning, för TI-Nspire som ger en ordentlig av teorin bakom stickprovsundersökningar.

