Kapitel 7 : micro:bit med Python — Tärningskast med två tärningar

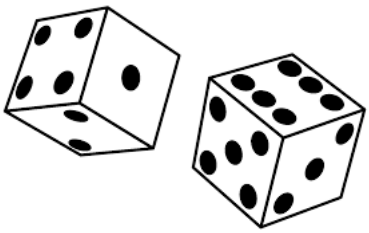
I denna applikation kan du använda ett program för att samla in data med hjälp av micro:bit.och sedan köra programmet i en delad skärm så att du kan se grafiken när du samlar in data.

Doelen :

- Att skriva ett program.
- Att skapa ett dynamiskt diagram med sidan Data och Statistik.

---

1. I denna applikation använder du teknikerna från de tre lektionerna i denna enhet:
   Samla in data med micro:bit med gester eller knappar och använda listor för att lagra och utbyta data med TI-Nspire.
   Skapa sedan en TI-Nspire-sida där:
   - å ena sidan körs Python -programmet
   - å andra sidan visas grafen (punktdiagram eller histogram) medan programmet körs.



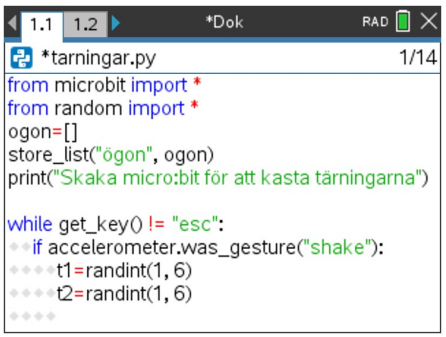
> **Lärartips**: Eleverna bör ha tillräckligt med kunskaper från övningarna 1, 2 och 3 för att utveckla detta program. Om de kämpar med menyer med mera, uppmuntra dem att titta på de tidigare övningarna.

2. Skapa ett nytt program och i det börja med att importera micro:bit -modulen och även den slumpmodulen.
   Definiera sedan en tom lista (“ogon”) och spara den direkt som en TI-Nspire-lista så att den här listan också är tom när programmet startas.
   Låt sedan programmet skriva ut instruktionen för användaren. I det här fallet ska vi använda alternativet att skaka micro:bit-enheten för att “kasta tärningarna”.

3. Programmet bör göra följande varje gång ett "kast" görs:
   - skapa två slumpmässiga heltal (från 1 till 6)
   - addera dem
   - lägg till resultatet i listan "ogon"
   - skriv ut antalet kast, de två utfallen och deras summa på skärmen
   - visa också de två utfallen på micro:bit-displayen, en efter en
   - lagra ogon till en TI-Nspire-lista.

   Tips: Antalet kast är lika med listans längd: ***len(ogon)***

4. Lägg till den sedvanliga while-loopen.



5. Lägg till en if-sats som testar om enheten har skakats. Om så är fallet skapa de två slumptalen.

   Lägg ihop resultaten och lägg denna summa i listan ”ogon”.

6. Visa de två utfallen på micro:bit-displayen, en efter en med en liten paus emellan.

   *Metoden **display.clear()** används för att rensa föregående resultat från displayen så att det är klart att ett nytt värde visas igen.*

> **Lärartips**: Det kan hjälpa att lägga till ytterligare ett sleep()-kommando i loopen för att ge micro:bit-enheten en extra chans att reagera på en gest.

7. Lägg till kod för att skriva ut informationen på TI-Nspire-skärmen och för att uppdatera TI-Nspire-listan.

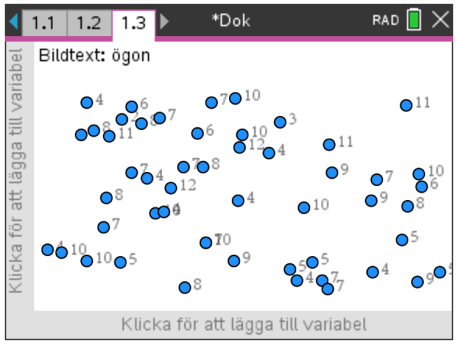
Lärartips: Observera avsaknaden av en räknar-variabel i print-kommandot. Det behövs inte eftersom vi här använder använder len (ogon) för att ta reda på antalet kast.

Vi skulle kunna låta antal vara en räknarvariabel som vi sätter till 0 före loopen. Sedan kan vi öka denna variabel med ett inne i loopen.

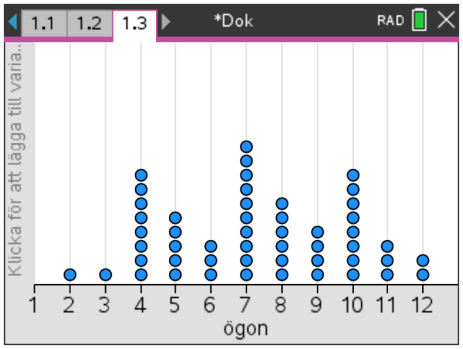
8. Kör programmet och kasta tärningarna ungefär 50 gånger. Tryck på **[esc]** för att stoppa programmet.
   Om Python-programmet går att köra utan fel så ska vi fortsätt med länka inmatad data till TI-Nspires statistikplottning.

   Se till att du är i konsolen (sidan med >>> markören) och tryck sedan på **[ctrl]-[doc]** eller **[ctrl]-[i]** för att lägga till en sida och välj Data & Statistik.

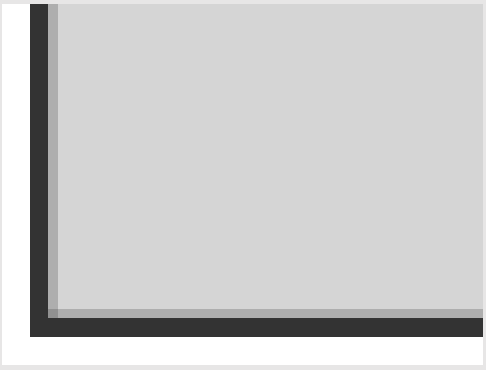
   Förhoppningsvis ser du nu en skärm som den till höger.

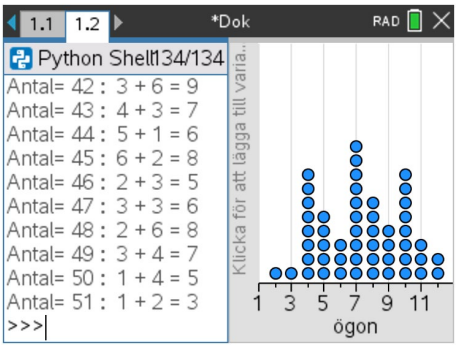


9. Klicka på ”Klicka för att lägga till variabel” längst ned på skärmen och välj ögon. Du kommer då att se en punktdiagram över de data vi samlat in.



Lärartips: Det är möjligt att ändra diagrammet till ett histogram: tryck på **[meny]> Diagramtyp> Histogram**. Du bör också justera staplarnas orientering så att de startar på varje halvvärde (orientering = 0.5) så att staplarna är centrerade över sina *x*-axelvärden.
Tryck på [**meny]> Diagramegenskaper> Histogramegenskaper> Inställningar stapelvärden>Lika fackbredd** och ställ in Orientering: 0.5.





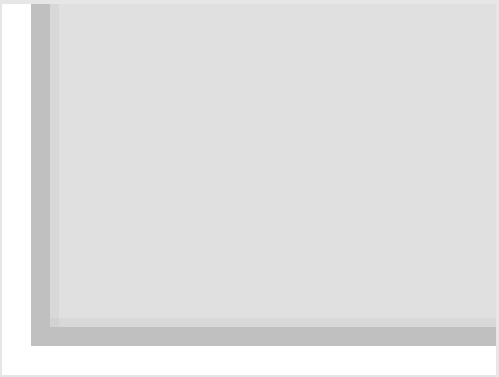
10. Gå tillbaka en sida till konsolen med **[ctrl]-[vänsterpil]** och tryck på **[ctrl]-[4]** för att slå ihop de två sidorna till en delad sida.
    Till vänster i fönstret visas konsolen och till höger applikationen Data & Statistik.

11. Gå tillbaka till Python-redigeraren och starta om programmet med **[ctrl]-[R]**. Variablerna nollställs nu.
    Du kommer sedan tillbaka till den delade skärmen och om du nu skakar micro:bit-enheten fram och tillbaka ser du en ny graf dyka upp.

    Du kan också starta programmet från Python-konsolen (den vänstra delen av skärmen) genom att trycka **[ctrl]-[R]** när konsolen är vald.

Lärartips: (Lite repetition och lite nytt) TI-Nspires delade skärmlayout kan vara antingen vertikal eller horisontell. När du trycker på **[ctrl]-[4]** för att slå ihop två fönster till en sida, är standardvalet vertikalt så som det visas i elevlektionen. För att växla till den horisontella layouten på handenheten trycker du på **[doc]>Sidlayout>Välj layout>Layout 3**

Du kan också justera separatorlinjen för att göra python Shell -appen mindre:



Du kan återgå till separata fönster för de bägge delfönstrena genom att trycka **[doc]>Sidlayout>Välj layout>Dela grupp**

För att få det till ett histogram, se föregående tips.

Exempel på möjlig kod:

Lite extra: Att visa tärningens ögon på micro:bit-displayen.
Det är enkelt att designa anpassade bilder som ska visas på micro:bit-displayen. Följande kod skapar de sex sidorna på tärningen. Notera att Image börjar med ett stort I.
Vi börjar med att importera nödvändiga moduler:

```
############################################################
from microbit import *
from random import *
```

För att skapa bilden för ett, kan vi skriva följande kod:

```
ett= Image(
"00000:"
"00000:"
"00900:"
"00000:"
"00000")
```

Python tolkar två strängar skrivna på separata rader utan avgränsare (utan, till exempel, ett komma) som en enda sträng, så
"Aaa"
“Bbb”
är det samma som
“Aaabbb”.

I följande kod visas de resterande bilderna för 2 till 6. Värdet på varje siffra i bilden kan vara från 0 till 9 för att styra ljusstyrkan på motsvarande lysdiod.
Vi namnger bilderna för att matcha utfallen.

```
tva=Image(  # Kan inte använda å.
"00000:"
"09000:"
"00000:"
"00090:"
"00000")

tre=Image(
"00000:"
"09000:"
"00900:"
"00090:"
"00000")
```

```
fyra=Image(
"00000:"
"09090:"
"00000:"
"09090:"
"00000")

fem =Image(
"00000:"
"09090:"
"00900:"
"09090:"
"00000")

sex=Image(
"00000:"
"09090:"
"09090:"
"09090:"
"00000")
```

Vi kan sedan placera dessa element i en array för att sedan kunna visa dem:

```
tarnings_bilder=[None, ett, tva, tre, fyra, fem, sex]
# index:                0      1    2    3      4       5     6
print("Startat")
while get_key() != "esc":
  if button_a.is_pressed():
     ogon=randint(1,6)
     display.show(tarnings_bilder[ogon])  # Visa ett utfall som en tärningsbild.
```