



Kapitel 2 : Programmera i Python

Övning 2: For-loopar

I detta avsnitt skall vi titta på hur man upprepar block av kod. Vi skall här använda så kallade for-loopar.

**Mål:**

- Att använda for-loopar.
- Att kunna syntaxen för for-loopar i Python.

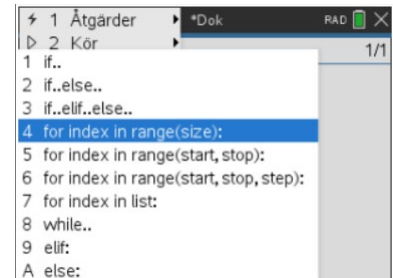
En av styrkorna med datorer är dess förmåga att outtröttligt utföra i stort samma sak om och om igen. Ett kommando för att utföra detta är en for-loop.

Vi skall skriva ett program som skall skriva ut kvadraten och kuben för de första åtta naturliga talen, 0 till och med 7.

Öppna ett nytt Python-program och ge det ett namn.

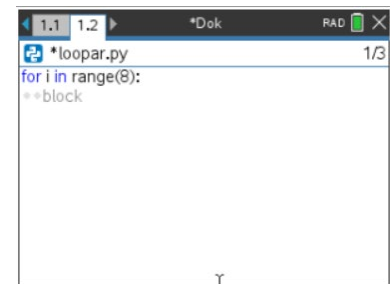
I menyn under Inbyggda under Kontroller hittar du alternativen för en for-loop.

Vi väljer det första alternativet



Ändra koden så att det blir som i exemplet till höger. Variabeln i är nu en så kallad räknare. Om inget initialvärde anges börjar loopen alltid med att i är 0. Så utför blocket av kod varefter i ökar med 1. Loopen körs åtta varv, så det sista värdet i får är 7.

Koden som vi kommer att skriva in i blocket kommer alltså att utföras åtta gånger då vi kör programmet, och i kommer då successiva ha värdena 0, 1, 2, 3, 4, 5, 6 och 7.



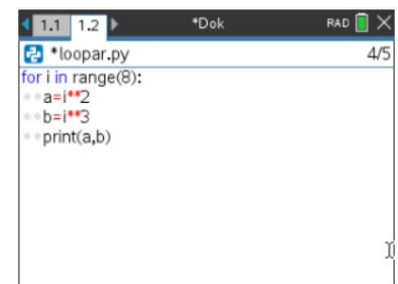
Observera kolontecknet i slutet av raden innan blocket börjar.

Vi vill skriva ut kvadraterna och kuberna så vi kommer att beräkna dem i loopen.

I Python skriver du upphöjt till med två asterisker. Så  $2^5$  är skrivet i Python som  $2^{**}5$ .

I for-slingan måste vi beräkna och skriva ut  $i^{**}2$  och  $i^{**}3$  för varje nytt värde på i.

Lägg till dessa rader och visa resultatet med Ctrl + R.



Vi kan också spara resultat i en lista. En lista i Python skrivs inom [ och ]. Låt oss skapa en lista av kvadrattal.

Vi börjar med en med en tom lista (En tom lista är en lista utan element).

Varje gång loopen körs så skall vi lägga till ett nytt resultat i listan med hjälp av funktion av lite speciellt slag, en så kallad metod. Metoden heter **.append**. och den kan används för att lägga till ett element i en lista.

Du hittar append under Meny > Inbyggda > Listor.



```
loopar.py 7/7
c=[]
for i in range(8):
  a=i**2
  c.append(a)
print(c)
```

Ändra programmet så att den skapar en lista med element och så att listan skrivs ut efter loopen.

Print-kommandot ligger nu utanför for-loopen. Prova att flytta in den in i loopen. Det räcker att lägga till två mellanslag framför print-kommandot. Provkör.

---

Inom matematiken är faktultet-funktionen en mycket viktig funktion. Den är definierad så att man multiplicerar alla naturliga tal upp till och med ett givet naturligt tal med varandra. Vi har till exempel att  $5! = 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1$ .

Vi ska skriva ett program där vi definierar faktultet-funktionen.

Öppna en ny Python-fil (Meny 1: Åtgärder > 1 Nytt) och ge filen namnet faktultet.

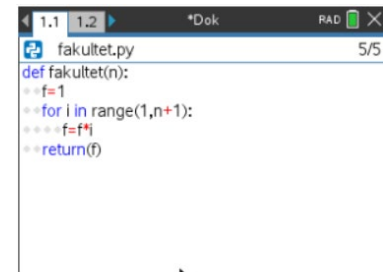
Skapa en funktion med namnet faktultet och en parameter n.

Skapa en variabel f (för faktultet) och sätt den till 1.

Skapa en for-loop som går från n till n+1. Du skall här använda det andra for-loops alternativet med ett start- och ett stopp-värde. Vi slutar med n+1 eftersom for-loopen avslutas då räknaren är lika med slutvärdet och att koden i loopen därmed inte körs för det värdet.

Inne i loopen skall du multiplicera f med i och ge svaret till f. På så vis multiplicerar vi successivt med 1, 2, 3 och så vidare upp till n.

Avsluta med att returnera värdet på a. Tänk på vilken nivå return är i.



```
fakultet.py 5/5
def fakultet(n):
  f=1
  for i in range(1,n+1):
    f=f*i
  return(f)
```

Kontrollera att du kan beräkna faktultet(4) blir 24. Testa gärna med något stor tal, till exempel 1000.

(När du kör programmet kommer du in i konsolen där funktionen nu är definierad. Med var-tangenten eller med Ctrl + L kan du hämta funktionen.)

---

**Tips:** Skillnaden mellan return och print är att ett returvärde ges från en funktion och att returvärdet kan användas vidare i beräkningar. Man kan till exempel skriva  $2 * \text{fakultet}(5)$  för att beräkna 2 gånger  $5!$  Print-kommandot skriver till konsolen, men man kan inte på samma sätt direkt använda det som skrivits ut i vidare beräkningar. Däremot ger det oss möjlighet att skriva ut flera saker istället för att bara returnera ett värde.