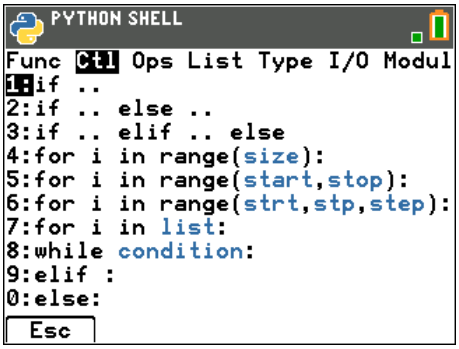
Kapitel 2: Starta programmering på riktigt — Övning 2: Upprepade beräkningar

I den andra övningen i enhet 2 får du lära dig hur du upprepar en process eller en uppsättning instruktioner med hjälp av en förbestämd for-loop.

Syfte:

- Utforska och implementera for-loopar
- Använda for-loopar i enkla exempel

Det är ibland användbart i ett program att upprepa en eller flera instruktioner ett antal gånger. Om antalet repetitioner av processen är känt i förväg används en förbestämd for-loop. Syntaxen för en for-loop är följande:



Naturligt språk
För variabler som sträcker sig från lägsta till högsta instruktioner.

Python-språk
for variabel i intervall ():
  Instruktioner

För funktionen range() kan du ange antalet steg i den förbestämda loopen. Det kan kallas på flera sätt:

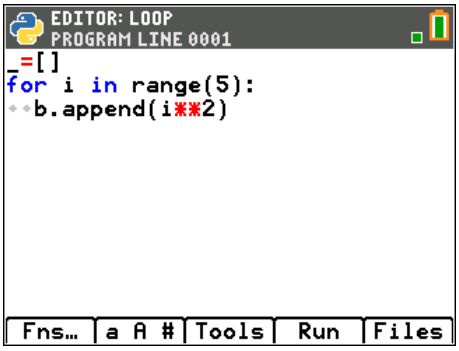
for i in range(size): tar heltalsvärden från 0 till range-1. range(4)==[0, 1, 2, 3]

for i in range:(start,stop): tar heltalvärden från start till stop-1

for i in range:(strt,stp,step): tar heltalsvärden från start till stop-1 med steglängden step.

for i in list: Variabeln använd listvärdena från det första värdet till det sista.

Det finns ingen end-i en loop-sats. Det är *indenteringen*, dvs indraget till höger som gör det möjligt att markera slutet på loopen.



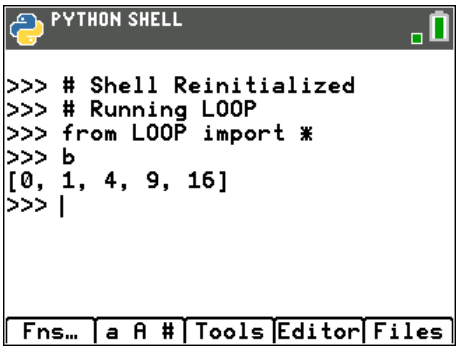
Genomförande
Implementering: Du ska skapa ett skript för att förstå vad en loop är och en iterationsprocess.

- Starta ett nytt skript och namnge det "LOOP"
- Starta en tom lista med satsen b=[ ]. I Python-språket placeras elementen i en lista mellan [ ], åtskilda av kommatecken.
- Tryck på f1 (Fns...) och välj från Ctl-menyn, alternativ 4: *for i in range (size)*
- append-satsen gör att du kan slutföra en lista. b.append (i**2) stegar b-listan med i^2. För varje värde på i så placeras värdet på i^2 i slutet av i^2-listan.
- Variabeln varierar från 0 till och med 4 vilket motsvarar 5 värden.

Lärarkommentar: Tänk på att, loopräknare alltid startas från 0. Instruktionen gäller för listorna. För att nå den, så trycker man på F1 (Fns ...),och väljer sedan listmenyn och slutligen 6: .append (x).

Tryck nu f4 för att köra skriptet.

I konsolen (shell) ber du sedan om visning av b.



Lärarkommentar: I en loop eller instruktion med en indentering är alla indenterade inmatningar ”för hand” ett kommando som är en del av loopen. Slutet av loopen markeras med avslutningen på indenteringen.

Låt oss tillämpa vår kunskap:

Här kommer nu en liten berättelse.

Inuti ett slott finns en kista full av guldmynt. För att komma ut från slottet måste du ta dig igenom fem dörrar, var och en skyddad av en vakt. De låter dig passera så länge du uppfyller deras krav. Här är vakternas krav:

> **”För att kunna gå ut genom dörren måste du ge mig hälften av dina mynt plus ett halvt mynt”**

Vad är det minsta antalet mynt du borde ta med dig från kistan, så att du kan behålla minst ett för dig själv när du kommer ut från slottet?

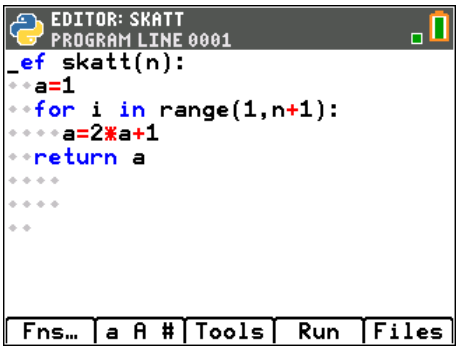
När man har gått igenom 5 dörrar ska man alltså ha 1 mynt kvar. Det måste ju betyda att man ska ha 3 mynt innan man passerar den sista dörren. Man ska ju ge bort hälften av 3 mynt + ett halvt mynt och det blir ju 2 mynt och då har man ett mynt kvar.

Så här kan man fortsätta och räkna ut antalet mynt man ska ha innan man passerar dörr 4, dörr 3 osv. Vi räknar här *baklänges*!

Här är ett algebraiskt uttryck som beskriver relationen mellan antalet mynt a_n man ska ha vid en dörr och den föregående dörren. Vi kallar antalet mynt för *a*.

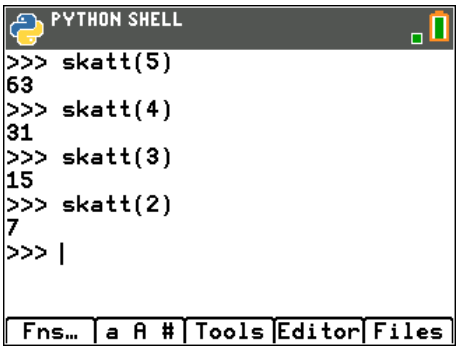
$$a_n = 2 \cdot a_{n-1} + 1$$

Så här då vårt skript ut. Se till att du har indrag enligt skärmen



Ska du ha ett mynt kvar efter 5 dörrar så måste du ha 63 mynt från början. Vi visar också körningar när man passerat 1, 2, och 3 dörrar.

Ändra nu i skriptet och gör nu körningar när antalet kvarvarande mynt efter passage av 5 dörrar ska vara 2 respektive 0! *Vad får du för resultat?*



Lärarkommentarer: Svaret på den sista frågan ovan är 95 dörrar resp. 31 dörrar. Nämn för eleverna att vid körning kan variabeln skatt() kopieras in på skärmen genom att trycka på tangenten ½.

Den här uppgiften finns som en särskild aktivitet för TI-Nspire™. Här finns också ett antal andra bra tillämpningar med upprepade beräkningar.

Gå till https://ti-resurser.com/materialdatabas och sök på *upprepade beräkningar*. De flesta beräkningar (utom de rent algebraiska) kan du utföra med TI-84 Plus CE-räknaren också.