

Kapitel 4: Loopar

I denna aktivitet kommer du att lära dig hur begreppet looping fungerar i programmering och närmare undersöka For-loopen.

Lärarkommentar: Det finns tre grundläggande loopar i TI-Nspire TI-Basic: For, While, och Loop. En loopstruktur ger ett program förmågan att processa en uppsättning av satser om och om igen, antingen upprepning över en sekvens av värden (precis som i For-loopen) eller tills ett speciellt villkor är uppfyllt (eller inte) som i While och Loop. Aktiviteterna i kapitel 4 introducerar var och en av dessa strukturer.

Program kan bli komplicerade eftersom det ofta är nödvändigt att blanda in alla kontrollstrukturer (If-satser och loopar) i ett program för att kunna arbeta med mer komplexa algoritmer. Det är ju detta som gör programmering så spännande och intressant ... och roligt!

Om Loopar

Programmeringsspråket TI-Basic har förmågan att bearbeta en uppsättning programsatser om och om igen. Denna upprepning av uttalanden kallas **looping**.

De tre loop-strukturer du lär dig i detta kapitel är nås genom att välja kontrollmenyn i programeditorn. Se skärmbild till höger. While... och Loop... strukturerna kommer att utforskas i senare aktiviteter i detta kapitel.

For...EndFor

For... loopen används för att processa en aritmetisk sekvens av värden. Det kallas för *iteration*.

Genom att välja For...EndFor-satsen från kontrollmenyn får du tillgång till de nödvändiga komponenterna som behövs för att bygga resten av strukturen:

For , , ,

EndFor

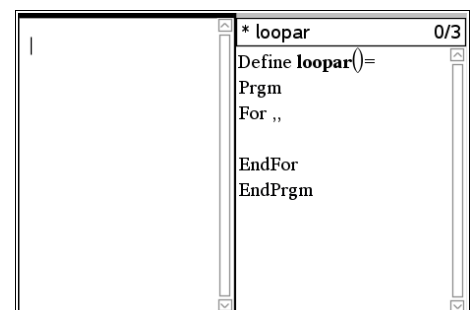
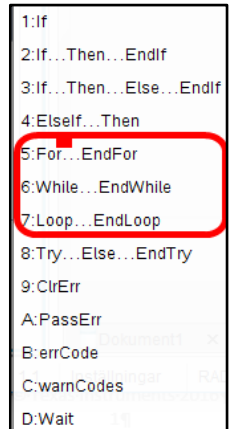
Kommatecknen efter ordet **For** indikerar att du behöver lägga till **fyra** poster:

For *i*, 1, *n*, 1

Övning 1: For-loopar

Syfte:

- Beskriva hur looping fungerar i programmering
- Konstruera program där man använder For...EndFor



10 Minutes of Code

TI-Nspire-teknologi

KAPITEL 4: ÖVNING 1

LÄRARKOMMENTAR

1. i är kontrollvariabeln för **loopen**. Den första posten måste vara en variabel.
2. 1 är **startvärdet**. Varje gång loopen processas ökar värdet, eller räkneverket om man så vill, från det specificerade **startvärdet** (1).
3. n är **slutvärdet**. Varje gång loopen processas ökar värdet, eller räkneverket om man så vill, till det specificerade **slutvärdet** (n).
4. 1 är **stegvärdet**. Varje gång loopen processas ökar värdet, eller räkneverket om man så vill, med ett bestämt stegvärde (1).

Startvärde, slutvärde och stegvärde kan alla vara tal eller variabler.

```
* loopar 4/4
Define loopar(n)=
Prgm
local i
For i,1,n,1
  Disp i,i2
EndFor
EndPrgm
```

Lärarkommentar:

Om steglängden är 1 så kan den fjärde posten utelämnas:

For $i, 1, n$ använder ett standardvärde på 1 för steglängden

Värdet på steglängden kan vara vilket tal som helst, även negativa tal:

For $i, 5, -5, -1$ startar vid 5 och går ner till -5

Vid stegningen är det inte nödvändigt att precis "träffa" slutvärdet. När värdet på i överstiger slutvärdet så avslutas loopen:

For $i, 1, 10, 6$ kommer att processa värdena 1 and 7 och sedan sluta

Tilldela inte loopkontroll-variabeln några värden inne i loopen. Det kommer att störa loopprocessen och orsaka oönskade resultat. Vanligtvis är kontrollvariabeln en lokal variabel.

Köra programmet

Till höger ser du resultatet av en programkörning där värdet på n är 5 (loopar(5)) Observera TI-Nspires funktion med delade fönster. Till höger har vi listningen av programmet och till vänster kör vi programmet i applikationen Räkna.

- Värdet på n är ett argument för programmet.
- Kontrollvariabeln är en lokal variabel och den påverkar inte resten av problemet där programmet finns. (Alla TI-Nspire-dokument består av ett eller flera problem).
- Loopen börjar med $i=1$ och visar värden på i och i^2 .
- Efter Disp-satsen skickar EndFor-satsen kontrollen tillbaka till For-satsen, där stegvärdet 1 adderas till i .
- Om värdet är mindre än eller lika med slutvärdet processas loopen igen med det uppstegade värdet på i .
- Denna process upprepas tills slutvärdet nås eller överträffas.

```
loopar(5) 4/4
1 1
2 4
3 9
4 16
5 25
Klar

loopar 4/4
Define loopar(n)=
Prgm
Local i
For i,1,n,1
  Disp i,i2
EndFor
EndPrgm
```

10 Minutes of Code

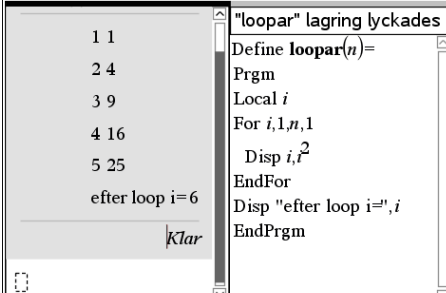
TI-Nspire-teknologi

KAPITEL 4: ÖVNING 1

LÄRARKOMMENTAR

När loopen avslutats så kommer i att vara större än slutvärdet!
Låter skumt! Du kan kontrollera detta genom att lägga till en ännu en Disp-sats efter EndFor-satsen och kontrollera själv.

Lärarkommentar: När loopen har avslutats så är värdet på loopvariabeln större än slutvärdet:



The screenshot shows a TI-Nspire calculator interface. On the left, the output of a program is displayed: 1 1, 2 4, 3 9, 4 16, 5 25, and after the loop i=6. The word 'Klar' is visible below the output. On the right, the program code is shown: 'loopar' lagring lyckades, Define loopar(n)=, Prgm, Local i, For i,1,n,1, Disp i,i^2, EndFor, Disp "efter loop i=",i, EndPrgm.

```
1 1
2 4
3 9
4 16
5 25
  efter loop i=6
  Klar

"loopar" lagring lyckades
Define loopar(n)=
Prgm
Local i
For i,1,n,1
  Disp i,i^2
EndFor
Disp "efter loop i=",i
EndPrgm
```