

Kapitel 4: Loopar

I denna aktivitet kommer du att lära dig att arbeta med den mångsidiga Loop...EndLoop-strukturen.

Hur fungerar egentligen Loop...EndLoop?

Loop...EndLoop strukturen ger möjlighet till en mer flexibel loop-struktur. Den upprepar exekvering av satser i loopen. Observera att loopen kommer att köras i all oändlighet, såvida inte en Exit-instruktion exekveras någonstans inuti loopen.

Utän en **Exit**-sats kommer loopen att vara oändlig.

Loop...EndLoop strukturen processas åtminstone en gång eftersom det inte finns något villkor som ska uppfyllas

Exit tvingar programmet att processa satsen precis efter **EndLoop**.

Exempel: Hur många slumpstal från 1 till 6 kan du alstra innan två på varandra följande värden är lika?

Notera i programmet till höger hur **Loop...EndLoop** används med en *villkorlig Exit*-sats. När `randInt(1,6)` alstrar två *konsekutiva* (på varandra följande) identiska värden så avslutas loopen och processar `Disp`-satsen i slutet av programmet. **Exit** hittar du i programeditorn i menyn *Överföringar*.

Samma effekt kan uppnås med en *While*-loop.

Syntax: Man kan ha två satser på samma rad i programmet genom att separera satserna med ett kolon (:).

Övning 3: Loop...Endloop

Syfte:

- Use the **Loop...EndLoop** structure
- Use the **Exit** statement to get out of a loop

```
* loop_endloop 3/3
Define loop_endloop()=
Prgm
Loop
EndLoop
EndPrgm
```

```
loop_endloop 2/13
Define loop_endloop()=
Prgm
Local t,c,n
t:=randInt(1,6)
c:=1
Loop
n:=randInt(1,6)
If t=n Then
Exit
Else
t:=n
c:=c+1
EndIf
EndLoop
Disp c
EndPrgm
```

Program: Gissa mitt tal

1. För att demonstrera användningen av Loop... EndLoop, ska vi nu konstruera ett spel för två spelare där det gäller att gissa ett slumpstal mellan 1 och 10. När en spelare gissar rätt tal så avslutas programmet med ett meddelande om vilken spelare som vunnit. Till höger visar vi av resultatet från en programkörning. Vi ska nu komma igång med ett sådant program som vi döper till **gissa**.
2. Vi börjar med att identifiera tre variabler: "spelare nr" (*spelare*), slumptalet (*tal*) and spelarens gissning (*gissning*).

Datorn alstrar ett heltaligt slumpstal mellan 1 och 10. Vi börjar med spelare nr 0. Detta gör det enkelt att ändra spelarens nummer, som vi ska visa senare.

3. Be spelaren att mata in en gissning. Den färdiga satsen blir:

Request "spelare " & string(*spelare*) & " gissa?", *gissning*

& används för *sammanfogning* av strängar. Inmatningsdelen av en **Request**-sats måste vara en sträng så variabeln *spelare* omvandlas till en sträng med **string()**-funktionen.

4. Därefter konstruerar man ett **Exit**-villkor. När en spelare gissar rätt tal så kommer loopen av avslutas. Här använder vi den enkla If-sats:

If gissning=tal

Exit

Tänk på att **If** utan **Then** processar nästa sats när villkoret är sant. I annat fall hoppas nästa sats över.

5. För att växla spelare kan man använda följande listiga sats:

spelare:= 1 - spelare

Denna sats byter värdet hos variabeln *spelare* från 0 till 1 och tvärtom från 1 till 0.

```
gissa()
spelare 0 gissa tal? 1
spelare 1 gissa tal? 2
spelare 0 gissa tal? 3
spelare 1 gissa tal? 4
spelare 1 vinner!!
Klar
```

```
* gissa 4/4
Define gissa()=
Prgm
  Local spelare,tal,gissning
  tal:=randIn(1,10)
  spelare:=0
  {}
EndPrgm
```

```
* gissa 7/7
Define gissa()=
Prgm
  Local spelare,tal,gissning
  tal:=randIn(1,10)
  spelare:=0
  Loop
    Request "spelare "&string(spelare)&" gissa tal?",gissning
  EndLoop
EndPrgm
```

```
* gissa 10/10
Define gissa()=
Prgm
  Local spelare,tal,gissning
  tal:=randIn(1,10)
  spelare:=0
  Loop
    Request "spelare "&string(spelare)&" gissa tal?",gissning
  EndLoop
  If gissning=tal
  Exit
  {}
EndLoop
EndPrgm
```

```
* gissa 9/10
Define gissa()=
Prgm
  Local spelare,tal,gissning
  tal:=randIn(1,10)
  spelare:=0
  Loop
    Request "spelare "&string(spelare)&" gissa tal?",gissning
  EndLoop
  If gissning=tal
  Exit
  spelare:=1-spelare
EndLoop
EndPrgm
```

10 Minutes of Code

TI-Nspire-teknologi

KAPITEL 4: ÖVNING 3

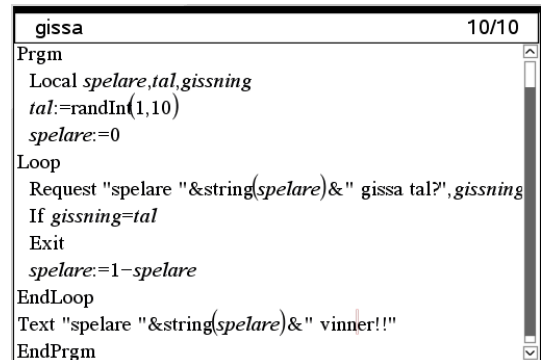
ELEVAKTIVITET

6. Till slut så lägger vi efter loopen till en sats som meddelar vem som vunnit.

Text "spelare" & string(spelare) & "vinner!!"

Tips: Om du inte vill att spelarna ska benämnas "spelare 0 och spelare 1 så kan man lägga till 1 till variabeln *spelare* i satsen här och i **Request**-satsen. Användaren ser 1 eller 2 även om datorn använder 0 och 1 för spelarna.

- 7 Innan du kör programmet se till att spara dokumentet. Du kan t.ex. fastna i en oändlig loop! Så fort programmet har startats så måste du fortsätta tills du har en vinnare.



```
gissa 10/10
Prgm
Local spelare,tal,gissning
tal:=randIn(1,10)
spelare:=0
Loop
Request "spelare "&string(spelare)&" gissa tal?",gissning
If gissning=tal
Exit
spelare:=1-spelare
EndLoop
Text "spelare "&string(spelare)&" vinner!!"
EndPrgm
```