

Kapitel 6: Koordinater

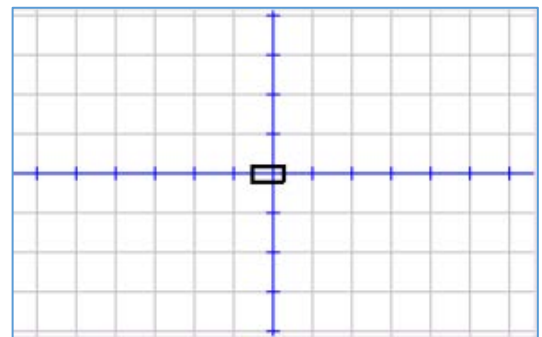
Övning 1: Introduktion till koordinater

I denna första övning för kapitel 6 får du lära dig mer om Rovers koordinatsystem och förflyttningar till koordinater.

Syfte:

- Förstå Rover's koordinatsystem och dess startposition och riktning
- Få Rover att förflytta sig till en speciell punkt i koordinatplanet
- Använda matematik för att bestämma avstånd.

Rover har ett inbyggt koordinatsystem precis som grafiksystemet på en räknare. När du ger instruktionen "Send "CONNECT RV" så sätts Rover's position i koordinatsystemet till (0, 0) och dess riktning är 0 grader vilket betyder att den pekar mot den positiva x-axeln.



Vårt program kommer att tala om för Rover att flytta till en punkt i sitt koordinatsystem. Vi använder en **Request**-sats för att ange värden för X och Y och sedan få Rover att köra till punkten (X, Y) och sedan köra tillbaka till origo.

1. Starta ditt program med de vanliga instruktionerna.
2. Lägg till **Request**-sats för x och y
Du minns väl att **Request** visar meddelandet innanför citattecknen (promten).
3. **Text**-satsen ger dig tid att placera Rover i origo och rikta den i riktning "höger".
4. Lägg till Rover-kommandot att köra "TO XY". Kommandot hittar du i > **Hub** > **Rover (RV)** > **Drive RV**-menyn, och visas i ditt program som en ofullständig sats: **Send "RV TO XY"**.

```
* rover61 0/9
Define rover61()=
Prgm
Send "CONNECT RV"
Request "x-koordinat ",x
Request "y-koordinat ",y
"Tryck enter för att starta. "
```

```
* rover61 5/5
Define rover61()=
Prgm
Send "CONNECT RV"
Request "x-koordinat ",x
Request "y-koordinat ",y
"Tryck enter för att starta. "
Send "RV TO XY "
EndPrgm
```

10 Minutes of Code

TI-Nspire-teknologi med TI-Innovator™ Rover

KAPITEL 6: ÖVNING 1

ELEVAKTIVITET

5. x - och y -koordinaterna måste läggas till och lagras i variablerna x och y . För att TI-Innovator™ Hub ska kunna använda dessa värden så måste du använda **eval()**-funktionen två gånger.
6. Lägg till **eval(x) eval(y)** till kommandot:
Send "RV TO XY eval(x) eval(y)"
7. Testa nu ditt program. Rover rör sig nu till den position som du anger när du matar in koordinatvärden.
8. Lägg till ett **Wait**-kommando för att låta Rover röra sig och ge sedan Rover order om att återvända till origo. Använd talen 0 och 0 separerade med ett mellanslag. Vi lägger också till en sats för att få Rover att rikta in sig i sin ursprungliga riktning (**TO ANGLE 0**).
9. Testa nu ditt program igen. Den här gången ska Rover flytta sig till din inmatade punkt och sedan återvända och ställa in sig i sin ursprungliga riktning.

```
rover61 2/5
Define rover61()=
Prgm
Send "CONNECT RV"
Request "x-koordinat",x
Request "y-koordinat",y
"Tryck enter för att starta. "
Send "RV TO XY eval(x) eval(y)"
EndPrgm
```

```
* rover61 0/8
Define rover61()=
Prgm
Send "CONNECT RV"
Request "x-koordinat",x
Request "y-koordinat",y
"Tryck enter för att starta. "
Send "RV TO XY eval(x) eval(y)"
Wait 5
Send "RV TO XY 0 0"
Send "RV TO ANGLE 0"
EndPrgm
```