

## Kapitel 2: Tilldela värden till variabler

I denna första aktivitet för kapitel 2 ska du använda variabler i ett program och upptäcka deras påverkan.

## Lagra ett värde i en variabel

Ibland behöver man tilldela ett värde till en variabel inom ett program. I exemplet använder vi Herons formel för att utifrån de tre sidornas längder i en triangel beräkna arean. Vi använder här en tvåstegsformel:

Först bestämmer vi halva omkretsen:  $s = \frac{1}{2} \cdot (a + b + c)$

och sedan arean:  $A = \sqrt{s \cdot (s - a) \cdot (s - b) \cdot (s - c)}$

Till höger så ser du programlistningen. Observera att vi använder `:=` för att *tilldela* värdet från en beräkning till en variabel. Alltså inget likhetstecken, =.

`:=` betyder "får värdet av ..." eller enklare, "får", så satsen  $s := 1/2 \cdot (a + b + c)$  läses "s får 1/2 gånger omkretsen av triangeln"

Använda operatoren för  $\rightarrow$  för att lagra

Du kan använda lagraoperatoren  $\rightarrow$  istället för tilldela-symbolen `:=`.

Se programlistningen till höger. Observera den motsatta ordningen i satsen där beräkningen med formeln kommer först, sedan lagraoperatoren  $\rightarrow$  och sedan variabeln. Det är den ordning i vilken satsen exekveras (från vänster till höger).

Både `:=` och  $\rightarrow$  kan användas i tilldelningssatserna. Kom bara ihåg ordningen som vi beskrev i förra stycket.

Båda metoderna tilldelar ett värde (oftast ett resultat av en beräkning) till en variabel så dessa kallas tilldelningssatser.

Skriv nu färdigt programmet så som det visas i skärmbilden till höger.

När hela koden är inskriven så kan du kontrollera programmet genom att "Kontrollera syntax och lagra". Den åtgärden finns i verktygsmenyn till vänster. Du kan också trycka *Ctrl B*, som är snabbkommandot för samma sak.

Testa nu programmet med värdena 3, 4, 5. Du ska då få värdet 6. Varför då?

## Övning 1: Globala variabler

## Syfte:

- Använda en variabel i ett program.
- Förstå den inverkan på resten av programmet som en variabel har.
- Utforska betydelsen av *argument* i ett program, som t.ex. Define program(a,b,c)

```

heron(3,4,5)
      arean= 6
      Klar

heron(5,5,5)
      arean= 25·√3
             4
      Klar

```

```

heron
3/3
Define heron(a,b,c)=
Prgm
s:=1/2·(a+b+c)
a:=√s·(s-a)·(s-b)·(s-c)
Disp "arean=",a
EndPrgm

```

```

heron(3,4,5)
      arean= 6
      Klar

heron(5,5,5)
      arean= 25·√3
             4
      Klar

```

```

heron
3/3
Define heron(a,b,c)=
Prgm
1/2·(a+b+c)→s
√s·(s-a)·(s-b)·(s-c)→a
Disp "arean=",a
EndPrgm

```

## 10 Minutes of Code

### TI-Nspire-teknologi

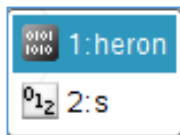
## KAPITEL 2: ÖVNING 1

### ELEVAKTIVITET

Om du klickar på knappen märkt **var**



så får du upp ett fönster med variablerna **heron** och **s**. heron är programnamnet och s skapades av programmet. Men a, b och c användes ju också i programmet. Vart har de tagit vägen?



Svaret är att a, b och c är *argument* till programmet och de kan bara existera inom programmet och de är inte skapade i dokumentet.

Dessutom vill vi inte att variabeln s ska vara med i dokumentet heller. I nästa aktivitet visar vi hur man fixar till det.

Kom ihåg att spara dokumentet så att också programmet sparas. Vi ska använda programmet heron i nästa aktivitet.

