

Måns Andersson  
Sannarpsgymnasiet  
Frennarpsvägen  
Halmstad  
[mans.andersson@halmstad.se](mailto:mans.andersson@halmstad.se)  
0733288233

## **Didaktisk programmeringsuppgift**

Höstterminen 2018

Handledare: Veronica Gaspes

## Bakgrund

I de nya kursplanerna för matematikämnet ställs det tydliga krav på att de digitala färdigheterna ska utvecklas. I de ämnen jag undervisar måste eleverna också kunna angripa matematiska problem med hjälp av programmering. Då det i nuläget är så att eleverna kommer till gymnasieskolan med blandade erfarenheter är detta en utmaning.

## Syfte

Syftet med arbetet är att ta fram ett undervisningsmaterial som underlättar för eleverna att komma i gång med programmering. Efter genomgångna exempel och träningsuppgifter ska de självständigt lösa ett matematiskt problem med hjälp av programmering.

## Målgrupp

Undervisningsmaterialet riktar sig till de elever som studerar på det naturvetenskapliga programmet och kurserna Ma 1c och Ma 2c. Materialet bidrar till att stärka elevernas förståelse för funktionsbegreppet. Där krävs det vissa matematiska förkunskaper. Lämpligt är då att genomföra uppgiften i början på kursen Ma 2c. Inga förkunskaper om programmering krävs.

## Didaktiska överväganden

Läraren måste börja med att visa plattformen *Wescheme* för att köra programspråket online. Därefter kan eleverna arbeta självständigt eller parvis. Om det finns möjlighet kan det vara lämpligt att läraren ger en kort introduktion i grupper om 10 elever. Det är viktigt att alla förstår de absoluta grunderna.

Att färdigställa hela materialet kan beräknas ta 3-5 lektioner. Tanken är att läraren intensivkör hela undervisningsmaterialet på en normal arbetsvecka. Jag har valt att dela upp arbetet i moduler för att eleverna ska få ett naturligt avslut. Det ger också läraren frihet att arbeta i etapper om så önskas.

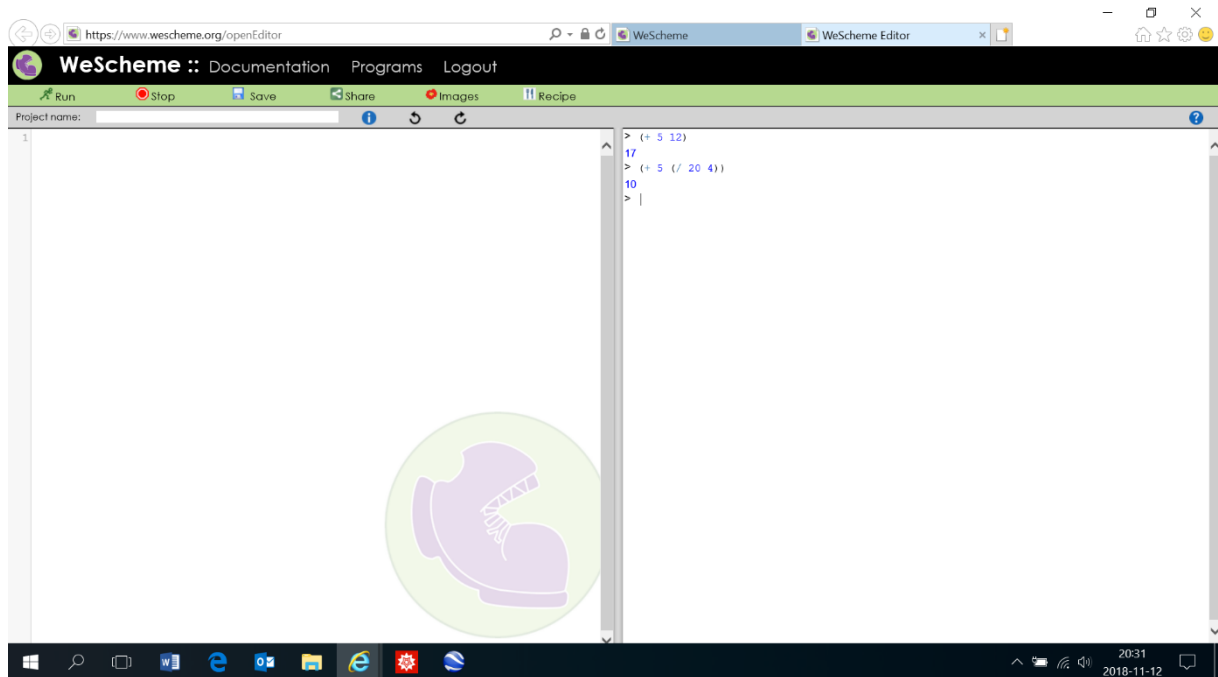
## Innehåll i elevmaterial

- Modul 1 Fönstret, enkla beräkningar.
- Modul 2 Arbeta med geometriska figurer.
- Modul 3 Arbeta i definitionsfönstret.
- Modul 4 Arbeta med bilder.
- Modul 5 Testa dina kunskaper
- Modul 6 Programmeringsuppgift

## Programmering i Wescheme – En introduktion

## Modul 1: Fönstret, enkla beräkningar

När du startar Wecheme ser du ett tvådelat fönster. På vänster sida skriver du in din programkod när du kör programmet visas resultatet i det högra fönstret. Det vänstra fönstret kallas definitionsfönstret och det högra kallas för interaktionsfönstret.



Det första du ska börja med är att genomföra enkla beräkningar med de fyra räknesätten.

Ex1  $5+12$  Skrivs med kod `(+ 5 12)` När du trycker enter visas resultatet.

Ex 2  $5*4$  Skrivs med kod `(* 5 4)`

Ex 3  $5+20/4$  Skrivs med kod `(+ 5 (/ 20 4))`

Parenteserna är således viktiga. Testkör ovanstående i interaktionsfönstret

### Beräkningscirklar

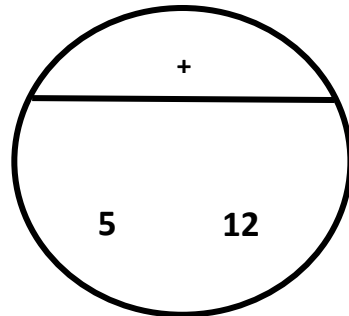
När du tränar på att skriva beräkningar på kodspråk kan du använda dig av beräkningsscirklar som ett mellansteg. Fyll i det som saknas i cirklarna samt kod. Testkör i interaktionsfönstret.

Matematik

Cirkel

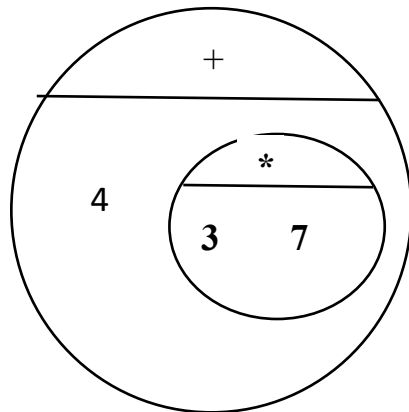
Kod

$5+12$



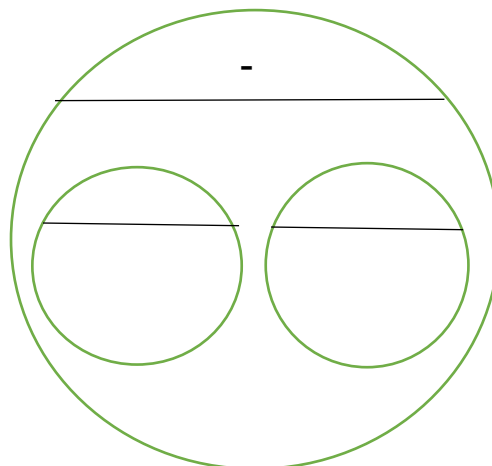
\_\_\_\_\_

$4+3*7$



\_\_\_\_\_

$10/2-8*5$

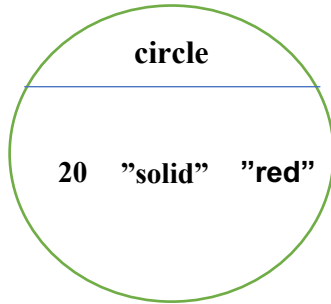


\_\_\_\_\_

**Modul 2: Arbeta med geometriska figurer**

I matematiken har du sedan tidig ålder ritat geometriska figurer. Du vet att cirkelns radie påverkar storleken. I wescheme finns ett helt paket med geometriska figurer. Du kan välja radie, färg och om cirkeln ska bli ifylld.

Se beräkningscirkeln nedan:



Vid kodning skriver du följande i interaktionsfönstret (**circle 20 "solid" "red"**). Tryck enter. Du får fram en helfärgad cirkel med radien 20. Kopiera gärna ovanstående kod och klistra in. (ctrlc, ctrlv)

"solid" och "red" är exempel på en ny datatyp som kallas strängar (strings)

Namnet på funktionen är circle och den använder en siffra och två strängar för att producera en bild

```
;circle, number, string, string->image
```

Ovanstående kan kategoriseras i ett så kallat Contract

```
; circle ; number, string, string -> image  
name domain range
```

Testa att allt fungerar genom att konstruera egna cirklar med valfri radie och färg.

## Övning

Konstruera nu en rektangel med bredden 50 och höjden 20 som helfärgad (solid) och blå  
Finns i Wescheme och koden inleds enligt nedan

(**rectangle** .....

Fyll i nedanstående Contract

; \_\_\_\_\_ ; \_\_\_\_\_ -> \_\_\_\_\_  
name domain range

**Modul 3: Arbeta i definitionsfönstret**

I föregående modul konstruerade vi en cirkel med en given radie. Om vi vill slippa skriva om hela koden kan vi använda funktionsbegreppet.

Funktioner är mycket viktiga inom matematik såväl som programmering. ( Repetera vid behov enkla funktionsberäkningar

I Wescheme definieras funktionen allmänt:

```
(define (funktionsnamnet variabel/variabler) ( funktionskroppen))
```

Funktionskroppen kan vara ett matematiskt uttryck eller liknande men kan också vara geometriska verktyg( t.ex cirkeln som du kodat ).

Vi vill kunna skriva ett enkelt program som skapar en ny cirkel om vi ändrar r. Kopiera koden nedan och klistra in i definitionsfönstret

```
( define (dot r) (circle r "solid" "blue"))
```

dot är namnet på funktionen.

r är variabeln som kan ändras och ger cirkelns radie

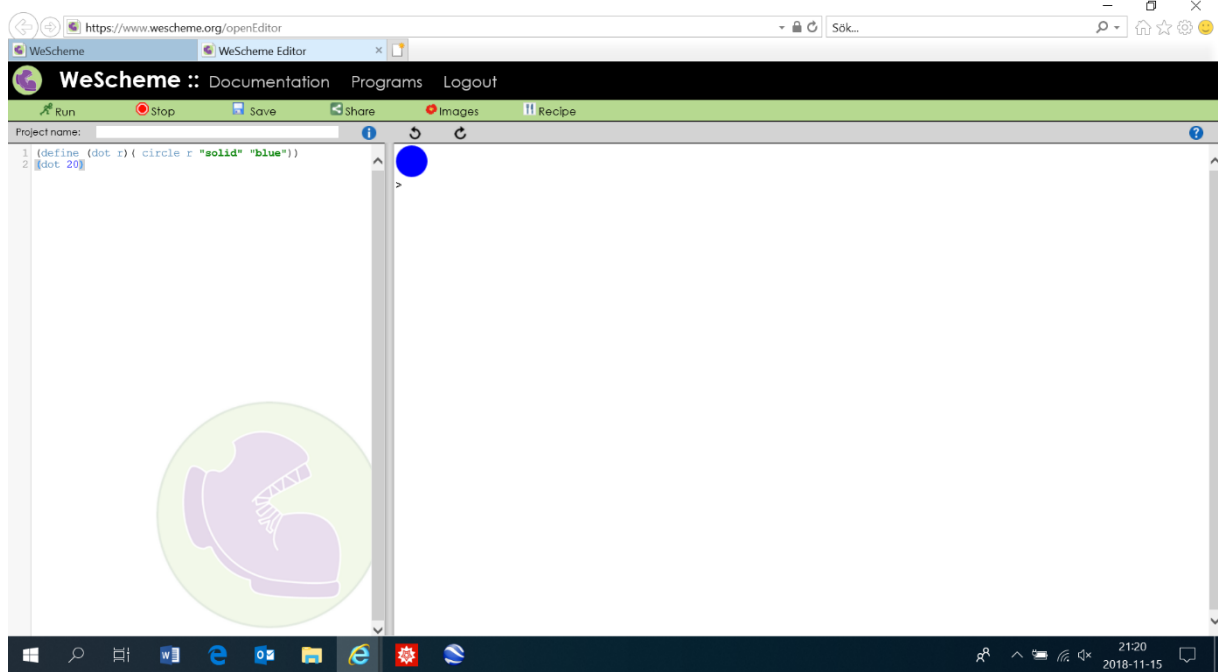
(circle r "solid" "blue")) är funktionskroppen och ritat en helfärgad cirkeln med den radie som du angett

Tryck Enter och på nästa rad skriver du (dot 20) . Klicka run. Vad händer?

Testa (dot 50) och ( dot 100)

### Övning:

Skriv en kod där variabeln istället styr vilken färg som önskas. Om du vill så kan du experimentera med två variabler



Om vi vill beräkna arean för cirkel kan vi definiera en funktion enligt nedan

**(define (area r) (\* pi r r) alternativt (\* pi (sqr r))**

(area 20) Kör programmet med run

### Övning:

Skriv ett program som ritar upp en given rektangel. Där du väljer bredd och längd-

Skriv ett program som räknar ut arean för vilken rektangel som helst.



## Modul 4: Arbeta med bilder

Nu ska vi titta på hur du kan placera geometriska figurer ovanpå varandra något som görs lättas med kommandot `place-image`

Kopiera koden till definitionsfönstret och kör programmet

```
(define (bild) (place-image (rectangle 60 40 "solid" "blue") 100 50 (rectangle 200 100 "solid" "green")))
```

Nu kommer rektangeln med bredden 20 och längden 40 att placeras ovanpå rektangeln med bredden 200 och längden 100.

Observera att `(bild)` är funktionens namn och hade istället kunnat heta t.ex `figur`.

I koden så finns siffrorna `100 50`. Undersök vad de styr genom att ändra siffrorna i koden.

### Skriv ut text

Om du vill skriva en text på ovanstående bild skriver du

```
(place-image (text "hej" 20 "black") 50 50 (bild))
```

”hej” Det du ska skriva ut

20 textstorlek

”black” färg på texten

50 50 textens placering i x- o y-led

### Skriv ut siffra som text

Som ovan men du måste göra om siffran till en sträng `number->string`

```
(place-image (text (number->string 48) 20 "black") 50 50 (bild))
```

Således skriv siffran 48 ut på samma ställe där det tidigare stod hej.

Testa

## Modul 5: Testa dina kunskaper

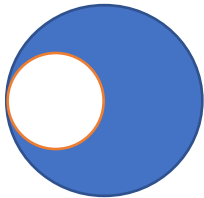
Nu har du uppnått en viss säkerhet och kan skriva enkla program. Försök nu självständigt lösa nedanstående uppgift

Undersök förhållandet mellan två cirklar. Den stora cirkeln har dubbelt så stor radie som den lilla.

Konstruera en bild där man ser att den lilla cirkeln ligger inuti den stora. När du ändrar radien så kommer bildens storlek att förändras.

Börja med att definiera de funktioner du behöver och beräkna aktuella areor.

Avslutningsvis ska förhållandet skrivas ut på den aktuella bilden.



## **Modul 6: Programmeringsuppgift**

Du har 600 m stängsel som ska användas för att bygga en hästhage. Vilka mått ska hagen ha för att du ska få så stor area som möjligt.

Med hjälp av programmering ska du visualisera hur hagen ser ut för olika mått på bredd och längd. Du ska också kunna beräkna den aktuella arean så att rätt slutsats kan dras.

### **Fördjupning**

Få dina figurer att förändras i realtid med det som kallas big-bang.

Du kan även göra så att din bild ändras när du t.ex trycker på piltangenten.

Din lärare tillhandahåller ramverket för den kod som behövs för big-bang