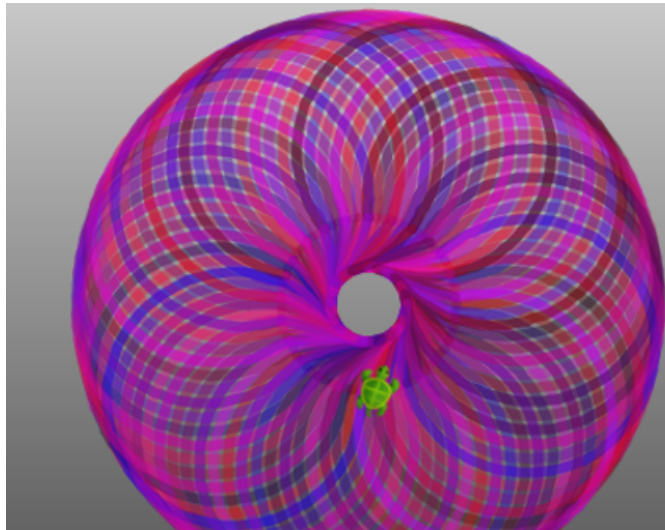


Uppdrag med Kojo

Redaktör: Björn Regnell
www.lth.se/programmera

Senast ändrad: 11 februari 2015





License: Creative Commons *Attribution-NonCommercial-ShareAlike 4.0 International* CC BY-NC-SA 4.0

Editor: Björn Regnell

Contributors: Björn Regnell, Lalit Pant, Sandra Nilsson, Maja Johansson, ...

© Björn Regnell, Lund University, 2015

<http://lth.se/programmera>

Innehåll

		Blanda dina egna färger	16
		Prova färgväljaren	17
		Rita slumpcirkclar	18
		Rita en blomma	19
		Rita många blommor	20
Om Kojo	1	Hur snabb är din dator?	21
Ditt första program	2	Byt kostym på paddan	22
Rita en kvadrat	3	Gör en timer	23
Rita en trappa	4	Träna multiplikation	24
Gör en loop	5	Spara saker i en vektor	25
Rita en gubbe	6	Simulera ett trafikljus	26
Gör din egen funktion	7		
Stapla kvadrater	8		
Gör en stapelfunktion	9		
Gör ett rutnät	10		
Kvadrat med parameter	11		
Rita en kvadratgubbe	12		
Rita en polygon	13		
Rita många polygoner	14		
Slumptal	15		

Om Kojo

Vad är Kojo?

Kojo är en app som hjälper dig att lära dig att programmera. Med Kojo kan du koda i det moderna och kraftfulla programspråket **Scala**. Kojo är gratis och finns på Svenska. Kojo fungerar med Linux, Windows och Mac OSX.

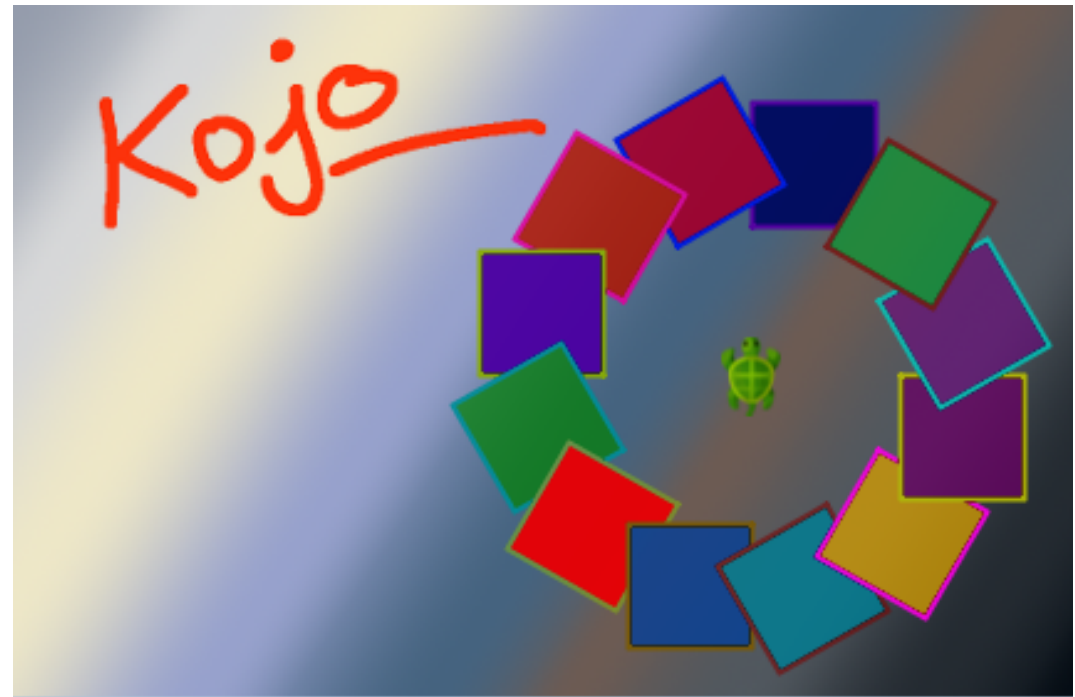
Var hittar jag Kojo?

Ladda ner Kojo här:

www.kogics.net/kojo-download

Läs mer här:

lth.se/programmera



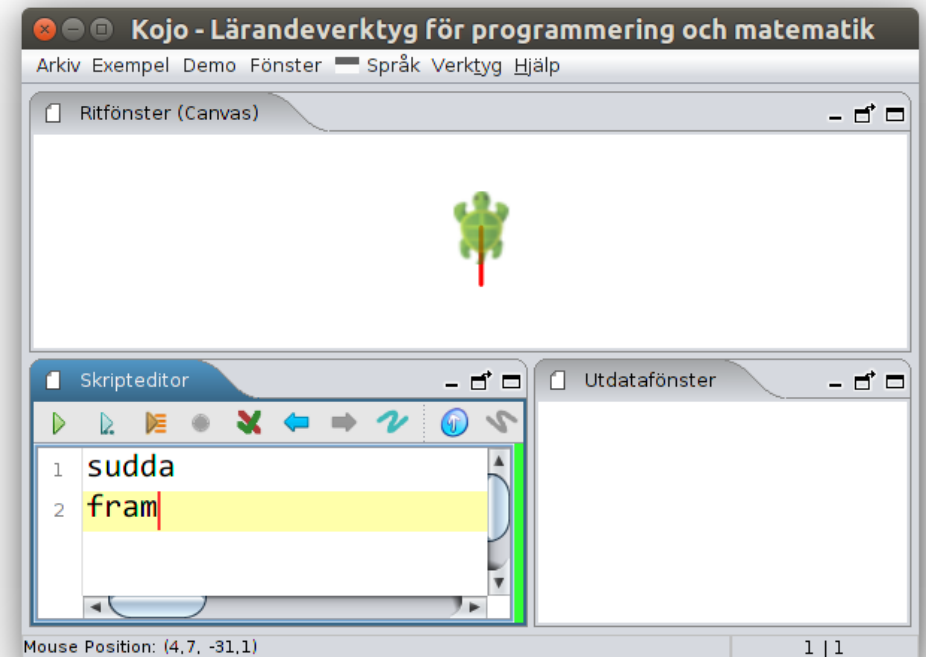
Ditt första program

Uppdrag:

Skriv så här i Kojos skripteditor-fönster:

```
sudda  
fram
```

Tryck på den gröna play-knappen
för att köra igång ditt program.



Rita en kvadrat

sudda
fram
höger

Om du skriver vänster eller höger så
vrider sig paddan.

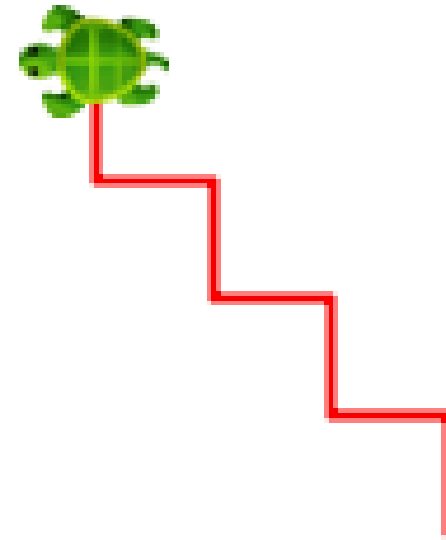
Uppdrag:

Utöka programmet så att det blir en
kvadrat.



Rita en trappa

```
sudda  
fram; vänster  
fram; höger
```



Med semikolon ; kan du ha flera satser på samma rad.

Uppdrag:

Utöka programmet så att det blir en trappa.

Gör en loop

sudda

```
upprepa(4){ fram; höger }
```



Uppdrag:

- Vad händer om du ändrar 4 till 100?
- Rita en trappa med 100 trappsteg.

Rita en gubbe

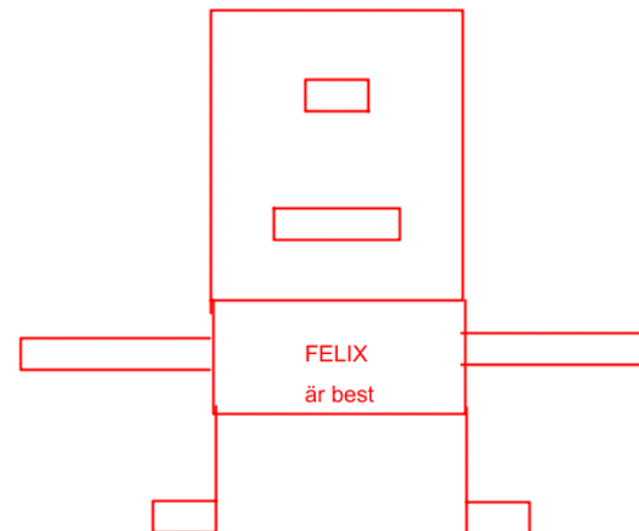
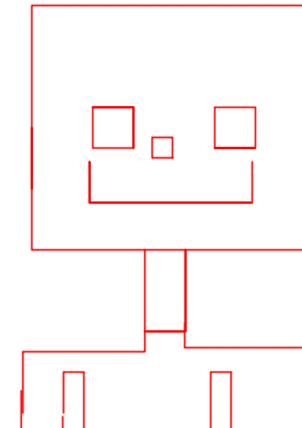
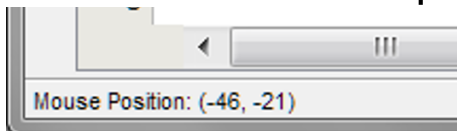
Uppdrag:

Rita en gubbe som du själv vill.

Tips:

```
hoppa  
vänster(180)  
fram(300)  
hoppa(100)  
hoppaTill(25,-28)  
skriv("FELIX är bäst")
```

Du kan se paddans läge nere till vänster medan du rör muspekaren i Ritfönstret:



Gör din egen funktion

Med **def** kan du göra egna *funktioner* som du själv väljer namn på.

```
def kvadrat = upprepa(4){ fram; höger }
```

sudda

```
kvadrat    //använd din kvadrat-funktion
```

hoppa

```
kvadrat
```

Uppdrag:

- Byt färg på kvadraterna.
- Gör fler kvadrater.

Tips:

```
fyll(grön); färg(lila)
```

Stapla kvadrater

Uppdrag:

Gör en stapel med 10 kvadrater.

Tips:

```
def kvadrat = upprepa(4){ fram; höger }
```

```
sudda; sakta(100)  
upprepa(10){ ??? }
```



Gör en stapelfunktion

Uppdrag:

Gör en funktion som heter `stapel`, som ritar en stapel med 10 kvadrater.

Tips:

```
def kvadrat = upprepa(4){ fram; höger }
```

```
def stapel = ???
```

```
sudda; sakta(100)
```

```
stapel
```



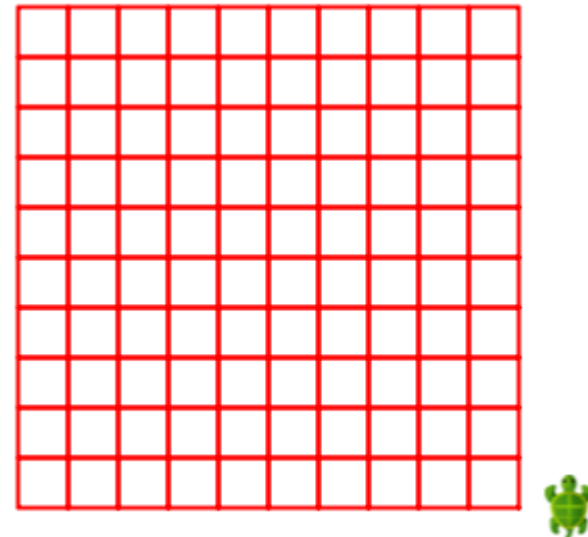
Gör ett rutnät

Uppdrag:

Gör ett rutnät med 10*10 kvadrater.

Tips:

- Använd din stapelfunktion från tidigare.
- Du kan hoppa baklänges en hel stapelhöjd med `hoppa(-10*25)`
- Du kan sedan hoppa till rätt plats med `höger; hoppa; vänster`



Kvadrat med parameter

Uppdrag:

Rita olika stora kvadrater.

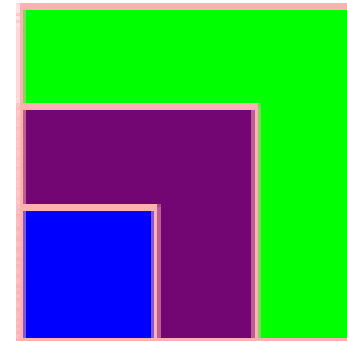
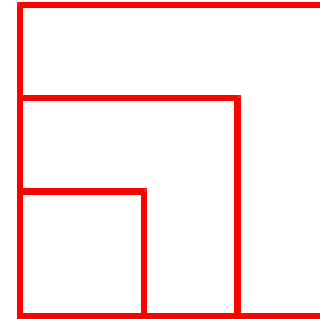
Tips:

Ge din kvadrat-funktion en *parameter*, med namnet *sidlängd* och typen *Heltal*:

```
def kvadrat(sidlängd : Heltal) =  
  upprepa(4){ fram(sidlängd); höger }
```

```
sudda; sakta(100); osynlig  
kvadrat(100)  
kvadrat(70)  
kvadrat(40)
```

Du kan byta färg med:
fyll(blå); färg(rosa)

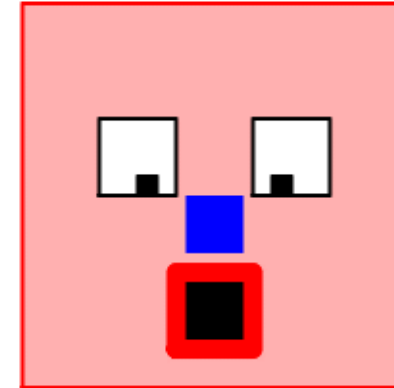


Rita en kvadratgubbe

Uppdrag:

Rita en gubbe med hjälp av olika stora kvadrater.

Tips:



```
def kvadrat(x: Heltal, y: Heltal, sidlängd: Heltal) = {  
  hoppaTill(x, y)  
  upprepa(4) { fram(sidlängd); höger }  
}  
def huvud(x: Heltal, y: Heltal) = { fyll(rosa); färg(röd); kvadrat(x, y, 200) }  
def öga(x: Heltal, y: Heltal) = { fyll(vit); färg(svart); kvadrat(x, y, 40) }  
def pupill(x: Heltal, y: Heltal) = { fyll(svart); färg(svart); kvadrat(x, y, 10) }  
def näsa(x: Heltal, y: Heltal) = { fyll(blå); färg(genomskinlig); kvadrat(x, y, 30) }  
def mun(x: Heltal, y: Heltal) = { bredd(10); fyll(svart); färg(röd); kvadrat(x, y, 40) }
```

```
sudda; sakta(20); osynlig  
huvud(0, 0)  
öga(40, 100); pupill(60, 100)  
???
```

Rita en polygon

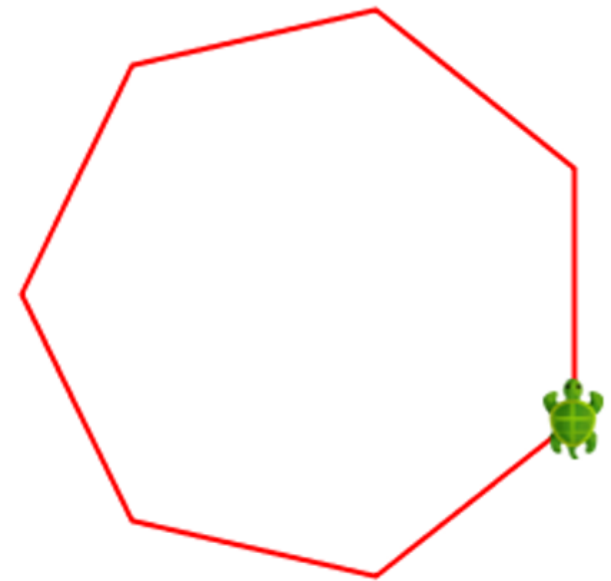
Uppdrag:

- Prova koden nedan. Rita olika slags polygoner.
- Lägg till en parameter sidlängd och rita olika stora polygoner.
- Hur stort behöver n vara för att det ska se ut som en cirkel?

Tips:

```
def polygon(n:Heltal) = upprepa(n){  
  fram(100)  
  vänster(360.0/n)  
}
```

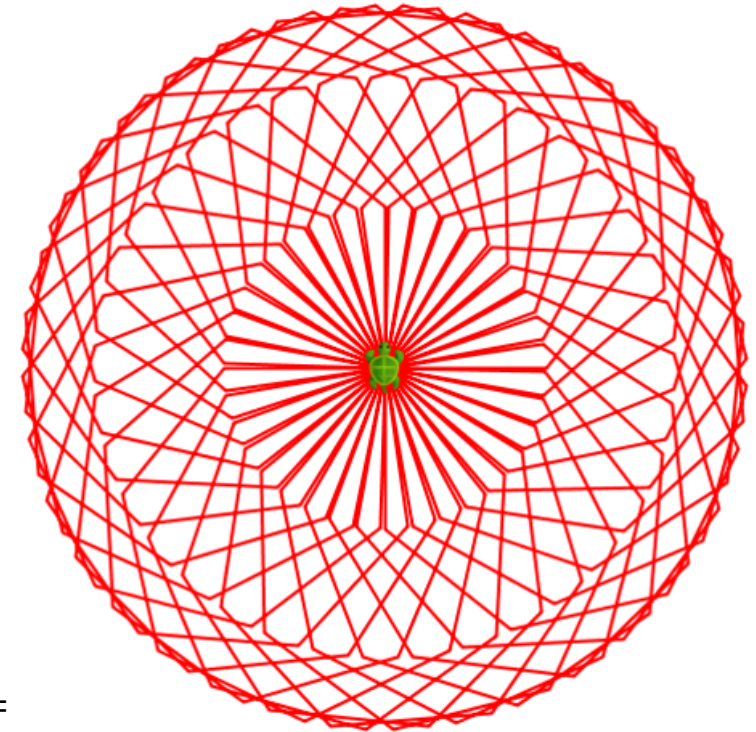
```
sudda; sakta(100)  
polygon(7)
```



Rita många polygoner

Uppdrag:

- Prova programmet nedan.
- Prova ändra antalet sidor och vinkel.
- Fyll polygonerna med färg.



```
def polygon(n: Heltal, sidlängd: Heltal) = upprepa(n){  
  fram(sidlängd)  
  vänster(360.0/n)  
}  
def snurra(n: Heltal, vinkel: Heltal, sidlängd: Heltal) =  
  upprepa(360/vinkel){ polygon(n, sidlängd); vänster(vinkel) }  
  
sudda; sakta(5)  
snurra(7, 10, 100)
```

Slumptal

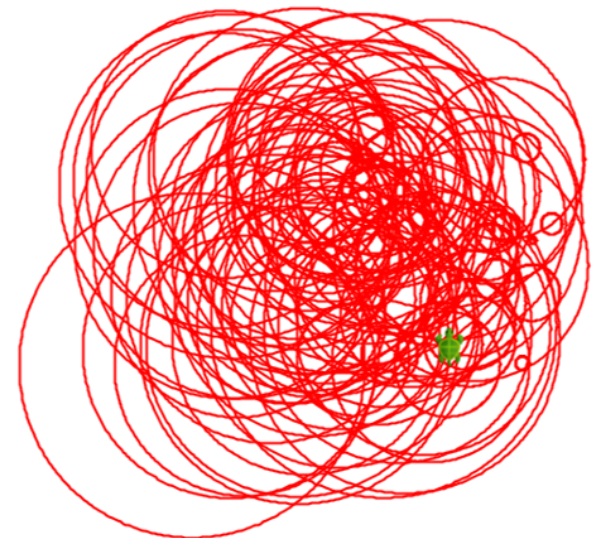
Uppdrag:

- Kör programmet nedan många gånger. Vad händer?
- Vilket är det minsta och största möjliga värdet på radien r ?
- Ändra så att r blir ett slumptal mellan 3 och 200.
- Rita 100 cirklar med slumpmässig radie på slumpmässig plats, som bilden visar.

```
//värdet r blir ett slumptal mellan 10 och 89:
```

```
val r = slumptal(90) + 10
```

```
sudda; sakta(10); osynlig  
skriv("Radie = " + r)  
cirkel(r)
```



Blanda dina egna färger

- Med Color kan du blanda egna färger, till exempel `Color(0, 70, 0)`
- De tre parametrarna anger mängden *rött*, *grönt* och *blått*
- Du kan också lägga till en fjärde parameter som anger *genomskinligheten*
- Alla parametrar ska vara mellan 0 och 255

Uppdrag:

Prova programmet nedan. Ändra genomskinligheten.

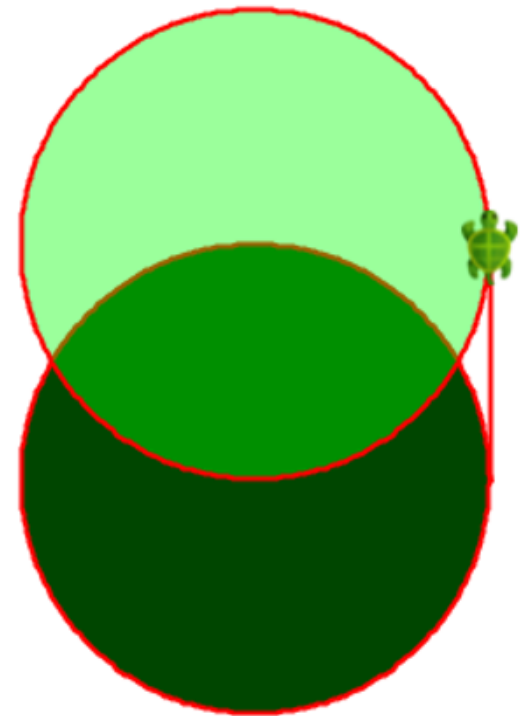
```
sudda; sakta(100)
```

```
val olivgrön = Color(0,70,0)
```

```
val pistageglass = Color(0,255,0,100)
```

```
fyll(olivgrön); cirkel(100)
```

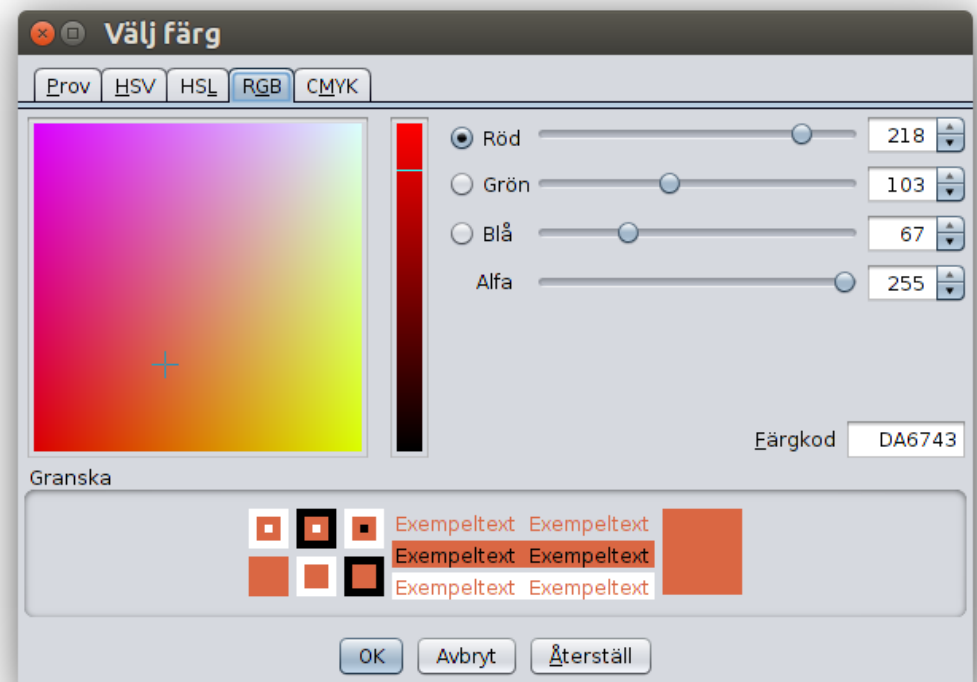
```
fyll(pistageglass); fram(100); cirkel(100)
```



Prova färgväljaren

Uppdrag:

- Högerklicka i editor-fönstret och klicka på "Välj färg".
- Om du väljer fliken **RGB** i färgväljaren kan du blanda nya RGB-färger.
- Tryck OK och titta i Utdatafönstret. Där syns de tre RGB-värdena för rött, grönt och blått.
- Du kan använda dessa värden i ditt program för att rita med din nya färg med `färg(Color(218, 153, 67))`.



Rita slumpcirkclar

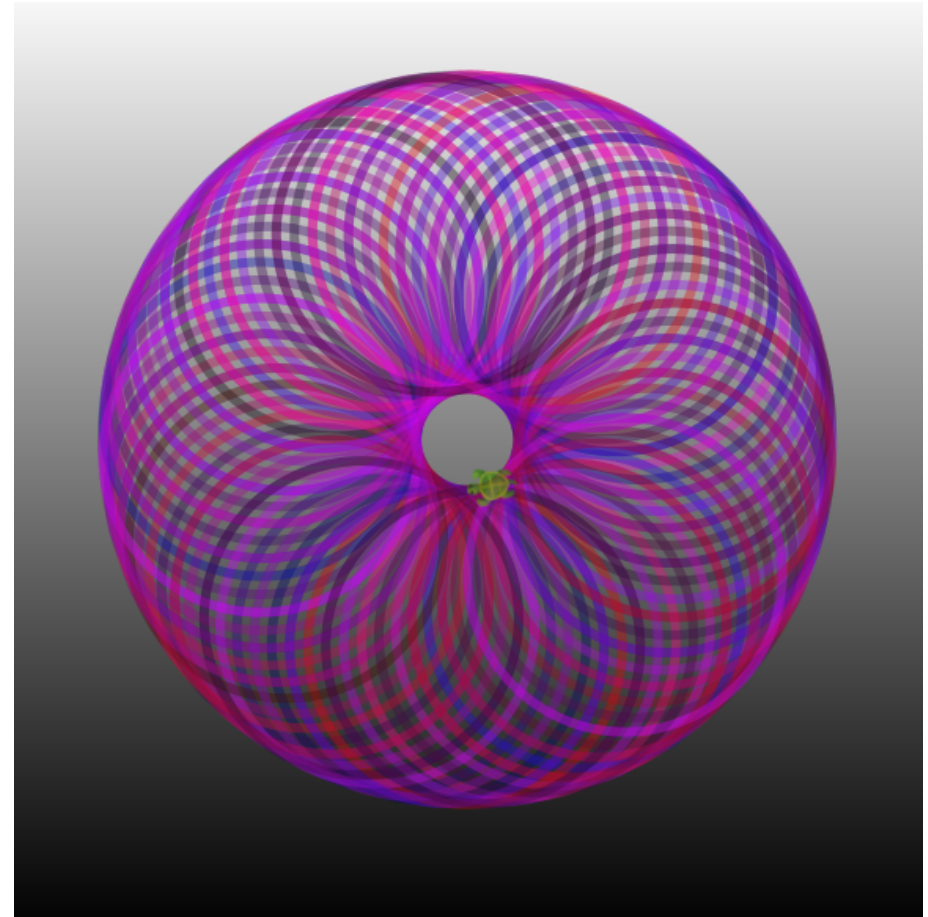
```
def slump = slumptal(256)
def slumpfärg = Color(slump,10,slump,100)

sudda; sakta(5)
bakgrund2(svart,vit)
bredd(6)

upprepa(100) {
  färg(slumpfärg)
  cirkel(100)
  hoppa(20)
  höger(35)
}
```

Uppdrag:

Prova olika slumpfärger och bakgrunder.

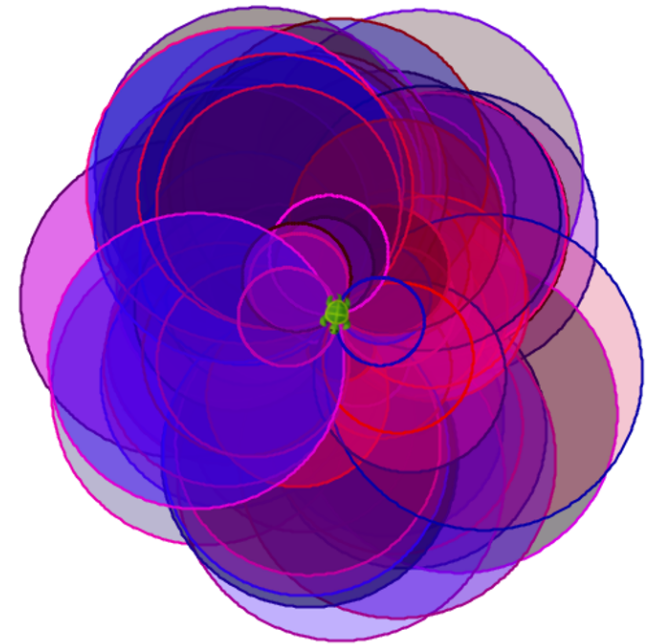


Rita en blomma

Uppdrag:

Programmet nedan ritar 100 slumpfärgade cirklar på slumpmässig plats med slumpmässig radie. Prova att ändra de olika slumptalens gränser och försök förklara vad som händer.

```
sudda(); sakta(5)
bredd(2)
upprepa(100){
  färg(Color(slumptal(256),0,slumptal(256)))
  fyll(Color(slumptal(256),0,slumptal(256),slumptal(100)+50))
  vänster(slumptal(360))
  cirkel(slumptal(30)*4+10)
}
```



Rita många blommor

Uppdrag:

- Gör en funktion som heter blomma, som ritar en krona och en grön stjälk från kronans mitt med ett grönt blad.
- Rita 5 blommor bredvid varandra.

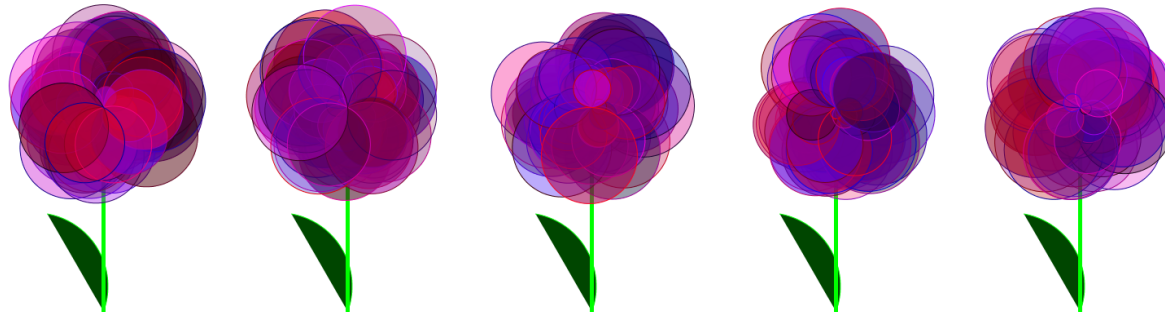
Tips:

Du kan rita blad med `båge(radie, vinkel)`.

Låt funktionen `blomma` ha två parametrar `x` och `y` och använd `hoppaTill(x,y)`

Du kan loopa 5 gånger och räkna ut platsen så här:

```
var i = 0
upprepa(5){
  blomma(600*i,0)
  i = i + 1
}
```



Hur snabb är din dator?

I Kojo finns en funktion `räknaTill` som mäter hur snabbt datorn kan räkna. När jag kör `räknaTill(5000)` på min snabba dator skrivs detta i utdata-fönstret:

```
*** Räknar från 1 till ... 5000 *** KLAR!  
Det tog 0.32 millisekunder.
```

Uppdrag:

- Kör `räknaTill(5000)` och kolla om din dator är snabbare än min.
- Hur lång tid tar det för din dator att räkna till en miljon?
- Hur långt hinner din dator räkna till på en sekund?

Byt kostym på paddan

Uppdrag:

Ladda ner mediafiler från Kojos hemsida: www.kogics.net/kojo-download#media
Skapa en mapp Kojo i din hemkatalog om det inte redan finns en sådan.
Packa upp filen scratch-media.zip och lägg mappen Media i mappen Kojo.
Prova att byta kostym på paddan till en häst så här:

```
sudda  
kostym("~/Kojo/Media/Costumes/Animals/horse1-a.png")  
sakta(2000)  
fram(1000)
```



Tips:

Du kan också använda dina egna bilder av typen .png eller .jpg

Gör en timer

Uppdrag:

Prova programmet nedan och mät din reaktionstid. Hur snabb är du?

```
object timer {  
  def nu = System.currentTimeMillis //ger nutid i millisekunder  
  var tid = nu  
  def nollställ = { tid = nu }  
  def mät = nu - tid  
  def slumpvänta(min: Int, max: Int) = //vänta mellan min och max sekunder  
    Thread.sleep((slumptal(max-min)+min)*1000) //Thread.sleep(1000) väntar 1 sekund  
}  
  
utdata("Klicka i utdatafönstret och vänta...")  
timer.slumpvänta(3,6) //vänta mellan 3 och 6 sekunder  
timer.nollställ  
indata("Tryck Enter så snabbt du kan.")  
utdata("Reaktionstid: " + (timer.mät/1000.0) + " sekunder")
```

Träna multiplikation

Uppdrag:

Prova programmet nedan. Ändra så att man bara tränar 8:ans och 9:ans tabell.

```
var antalRätt = 0
val startTid = System.currentTimeMillis / 1000
upprepa(12) {
    val tal1 = slumptal(12)+1
    val tal2 = slumptal(12)+1
    val svar = indata("Vad är " + tal1 + "*" + tal2 + "?")
    if (svar == (tal1 * tal2).toString) {
        utdata("Rätt!")
        antalRätt = antalRätt + 1
    }
    else utdata("Fel. Rätt svar är " + (tal1 * tal2))
}
val stoppTid = System.currentTimeMillis / 1000
val sek = stoppTid - startTid
utdata("Du fick " + antalRätt + " rätt på " + sek + " sekunder.")
```

Spara saker i en vektor

Uppdrag:

Prova programmet nedan. Vad skrivs ut? Lägg till fler djur i vektorn.

```
var djur = Vector("älg", "ko", "kanin", "kvalster")
utdata("Första djuret i listan är: " + djur(0)) //platserna i vektorn räknas från 0
utdata("Andra djuret i listan är: " + djur(1))
utdata("Det finns så här många djur: " + djur.size)
utdata("Sista djuret i listan är: " + djur(djur.size-1))
```

```
val s = slumpTal(djur.size) //dra ett slumpal mellan 0 och antalet djur minus 1
utdata("Ett slumpmässigt djur: " + djur(s))
```

```
djur = djur :+ "Kamel" //lägg till ett djur sist i vektorn
djur = djur.updated(0, "Dromedar") //Ändra djuret på plats 0
utdata("Alla djur i vektorn baklänges:")
djur.foreach{x => utdata(x.reverse)} //för alla x i vektorn: skriv ut x baklänges
```

Simulera ett trafikljus

Uppdrag:

Prova programmet nedan. Ändra så att trafikljuset är rött dubbelt så länge.

```
def släckAlla = draw(penColor(gray) * fillColor(black) -> PicShape.rect(130,40))
def ljus(c: Color, h: Int) = penColor(noColor) * fillColor(c) * trans(20,h) -> PicShape.circle(15)
def rött = draw(ljus(red, 100))
def gult = draw(ljus(yellow, 65))
def grönt = draw(ljus(green, 30))
def vänta(sekunder: Int) = Thread.sleep(sekunder*1000)

clear; invisible
while (true) { //en oändlig loop
  släckAlla
  rött; vänta(3)
  gult; vänta(1)
  släckAlla
  grönt; vänta(3)
  gult; vänta(1)
}
```

