

Sinus och cosinus 4: Slutuppgift

Nu avslutar vi spelet:

- Se till att kulan inte är synlig i början (tips: ställ in `y_kula` på minus hundra)
- Om du trycker på blanksteg, ställ in `vinkel_kula` lika med `vinkel_kanon`

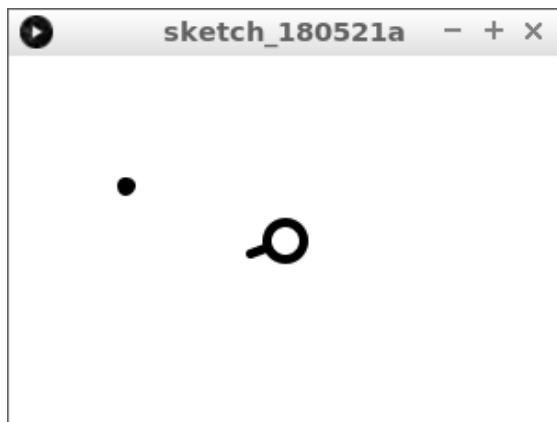


Figure 33: Sinus och cosinus 4: Slutuppgift

Processing

Bok 7



Figure 1: Bok 7: sinus en cosinus

#	Beskrivning
25	Sinus och cosinus 1: graf
26	Sinus och cosinus 2: ritar
27	Sinus och cosinus 3: planteter
28	Sinus och cosinus 4: skjuta

Contents

Förord	1
Sinus och cosinus 1: graf	2
Sinus och cosinus 2: rita	14
Sinus och cosinus 3: solsystem	23
Sinus och cosinus 4: skjutning	33

Förord

Detta är en bok om Processing för ungdomar. Processing är ett programmeringsspråk. Denna bok lär dig det programmeringsspråket.

Om den här boken

Denna bok är licensierad av CC-BY-SA.



Figure 1: Licensen för denna bok

(C) Richèl Bilderbeek och alla lärare och alla elever

Med det här häftet kan du göra vad du vill, så länge du hänvisar till originalversionen på denna webbplats: https://github.com/richelbilderbeek/processing_foer_ungdomar. Detta häfte kommer alltid att förbli gratis, fritt och öppet.

Det är fortfarande en lite slarvig bok. Det finns stafvel och *layouten är inte alltid vacker*. Eftersom den här boken finns på en webbplats kan alla som tycker att den här boken är för slarvig göra den mindre slarvig.

Sinus och cosinus 4: lösning 6

```
float vinkel_kanon = 0;
float x_kula = 0;
float y_kula = 0;
float vinkel_kula = 0;

void setup()
{
    size(300, 200);
    strokeWeight(5);
}

void draw()
{
    background(255, 255, 255);
    final float x_mitten = width / 2;
    final float y_mitten = height / 2;
    final float x_kanon = x_mitten + (cos(vinkel_kanon) * 20);
    final float y_kanon = y_mitten - (sin(vinkel_kanon) * 20);
    x_kula += cos(vinkel_kula);
    y_kula -= sin(vinkel_kula);
    line(x_mitten, y_mitten, x_kanon, y_kanon);
    ellipse(x_mitten, y_mitten, 20, 20);
    ellipse(x_kula, y_kula, 5, 5);
    vinkel_kanon += 0.1;
    if (keyPressed)
    {
        if (key == ' ')
        {
            x_kula = x_kanon;
            y_kula = y_kanon;
        }
    }
}
```

Sinus och cosinus 4: uppgift 6

- Skapa en ny variabel ovanför `setup`-funktionen med namnet `vinkel_kula`
- Ge `vinkel_kula` startvärdet noll
- Låt `x_kula` bli mer och mer med cosinus av `vinkel_kula`
- Låt `y_kula` bli *mindre* med sinus för `vinkel_kula`

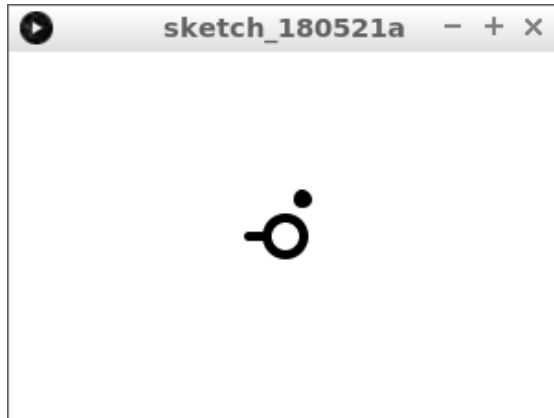


Figure 32: Sinus och cosinus 4: uppgift 6

Sinus och cosinus 1: graf

Under den här lektionen ska vi rita en sinus och en cosinus.



Figure 2: Elite

Sinus och Cosinus 1: uppgift 1

Skriv denna kod över:

```
float x = 0;

void setup()
{
  size(314, 200);
}

void draw()
{
  final float y = sin(x);
  point(x, y);
  x = x + 1;
}
```

Vad ser du?



314 är ungefär hundra gånger talet pi

Sinus och cosinus 4: lösning 5

```
float vinkel_kanon = 0;
float x_kula = 0;
float y_kula = 0;

void setup()
{
  size(300, 200);
  strokeWeight(5);
}

void draw()
{
  background(255, 255, 255);
  final float x_mitten = width / 2;
  final float y_mitten = height / 2;
  final float x_kanon = x_mitten + (cos(vinkel_kanon) * 20);
  final float y_kanon = y_mitten - (sin(vinkel_kanon) * 20);
  line(x_mitten, y_mitten, x_kanon, y_kanon);
  ellipse(x_mitten, y_mitten, 20, 20);
  ellipse(x_kula, y_kula, 5, 5);
  vinkel_kanon += 0.1;
  if (keyPressed)
  {
    if (key == ' ')
    {
      x_kula = x_kanon;
      y_kula = y_kanon;
    }
  }
}
```

Sinus och cosinus 4: uppgift 5

- Om du trycker på mellanslagstangenten så ...
 - får `x_kula` värdet av `x_kanon`
 - får `y_kula` värdet för `y_kanon`
 - kanonen bara fortsätter snurra

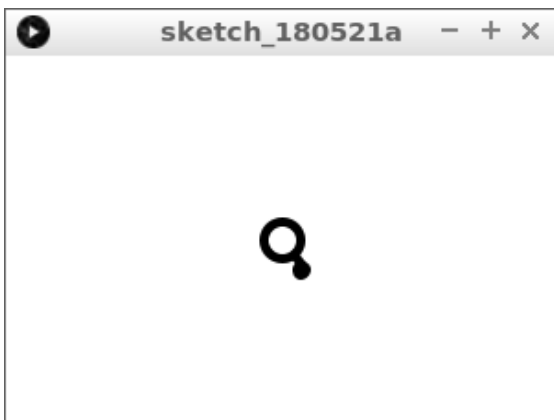


Figure 31: Sinus och cosinus 4: uppgift 5

Sinus och cosinus 1: lösning 1

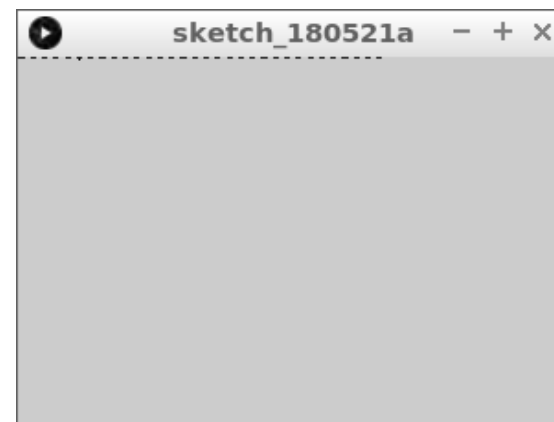


Figure 3: Sinus och cosinus 1: lösning 1

Du kommer att se en sorts prickad linje längst upp.

Sinus och Cosinus 1: uppgift 2

- Flytta den prickade linjen nedåt. Gör detta genom att lägga till halva höjden med y



Tips: det är smart att sätta parenteser runt `height / 2`

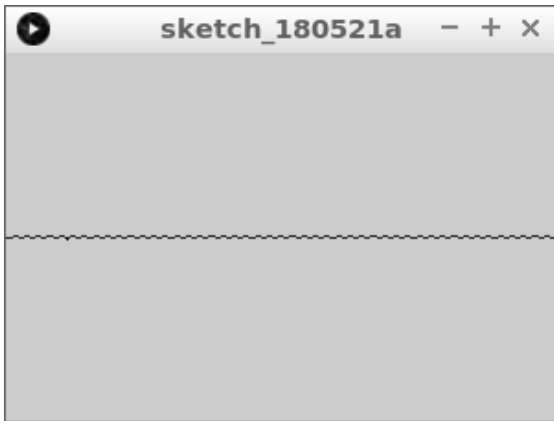


Figure 4: Sinus och cosinus 1: uppgift 2

Sinus och cosinus 4: lösning 4

```
float vinkel_kanon = 0;
float x_kula = 0;
float y_kula = 0;

void setup()
{
  size(300, 200);
  strokeWeight(5);
}

void draw()
{
  background(255, 255, 255);
  final float x_mitten = width / 2;
  final float y_mitten = height / 2;
  final float x_kanon = x_mitten + (cos(vinkel_kanon) * 20);
  final float y_kanon = y_mitten - (sin(vinkel_kanon) * 20);
  line(x_mitten, y_mitten, x_kanon, y_kanon);
  ellipse(x_mitten, y_mitten, 20, 20);
  ellipse(x_kula, y_kula, 5, 5);
  vinkel_kanon += 0.1;
  if (keyPressed)
  {
    if (key == ' ')
    {
      vinkel_kanon = 0;
    }
  }
}
```

Sinus och cosinus 4: uppgift 4

- Gör variablerna `x_kula` och `y_kula` högst upp
- Ställ in de initiala värdena för `x_kula` och `y_kula` till noll
- Rita en kula med mitten (`x_kula`, `y_kula`) med en cirkel som är 5 pixlar bred och hög

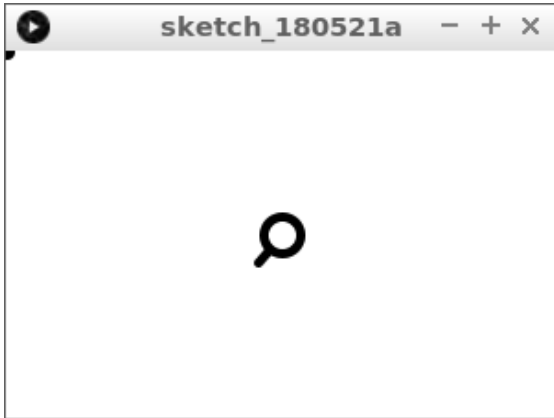


Figure 30: Sinus och cosinus 4: uppgift 4

Sinus och cosinus 1: lösning 2

```
float x = 0;

void setup()
{
  size(314, 200);
}

void draw()
{
  final float y = sin(x) + (height / 2);
  point(x, y);
  x = x + 1;
}
```



Fästena hjälper datorn att veta ordningen på en beräkning



En sinusvåg är formad som en våg som svänger mellan -1 och 1

Sinus och cosinus 1: uppgift 3

- Ersätt $\sin(x)$ med $(\sin(x) * 50)$

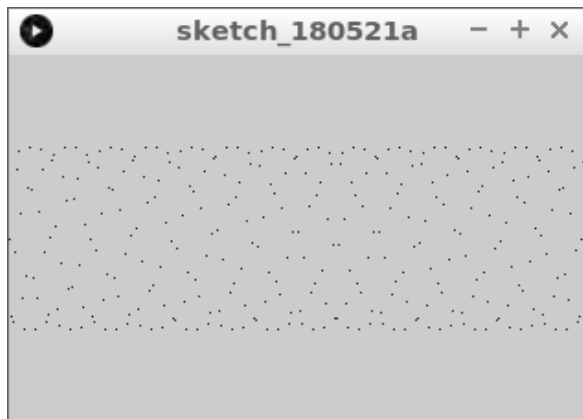


Figure 5: Sinus och cosinus 1: uppgift 3



Parentes är viktigt!

```
}
```

Sinus och cosinus 4: lösning 3

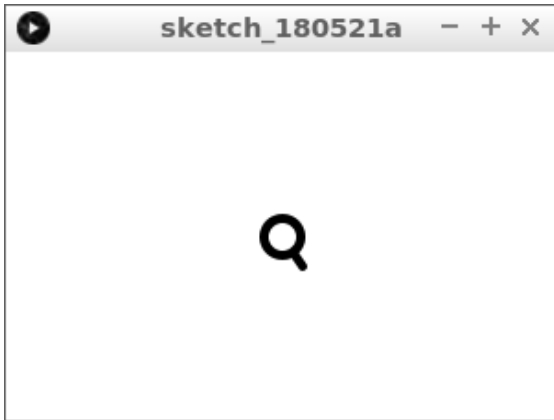


Figure 29: Sinus och cosinus 4: uppgift 3

```
float vinkel_kanon = 0;

void setup()
{
  size(300, 200);
  strokeWeight(5);
}

void draw()
{
  background(255, 255, 255);
  final float x_mitten = width / 2;
  final float y_mitten = height / 2;
  final float x_kanon = x_mitten + (cos(vinkel_kanon) * 20);
  final float y_kanon = y_mitten - (sin(vinkel_kanon) * 20);
  line(x_mitten, y_mitten, x_kanon, y_kanon);
  ellipse(x_mitten, y_mitten, 20, 20);
  vinkel_kanon += 0.1;
  if (keyPressed)
  {
    if (key == ' ')
    {
      vinkel_kanon = 0;
    }
  }
}
```

Sinus och cosinus 1: lösning 3

Du kommer nu att se ett mönster.

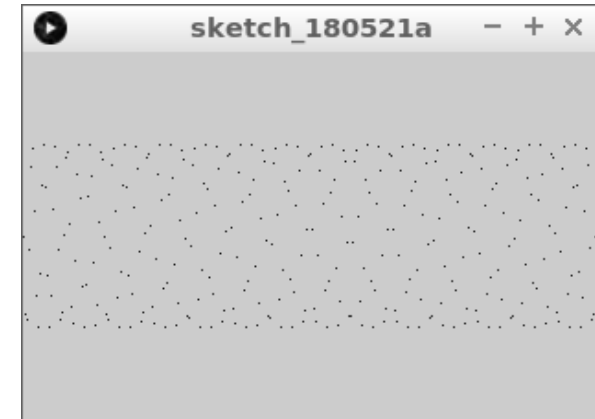


Figure 6: Sinus och cosinus 3: uppgift 2

```
float x = 0;

void setup()
{
  size(314, 200);
}

void draw()
{
  final float y = (sin(x) * 50) + (height / 2);
  point(x, y);
  x = x + 1;
}
```



Det är en röra, eftersom vi blandar ungefär 100 bihålor



*50 gör att sinus förstoras femtio gånger

Sinus och cosinus 1: uppgift 4

- Ersätt $\sin(x)$ med $\sin(x / 100)$

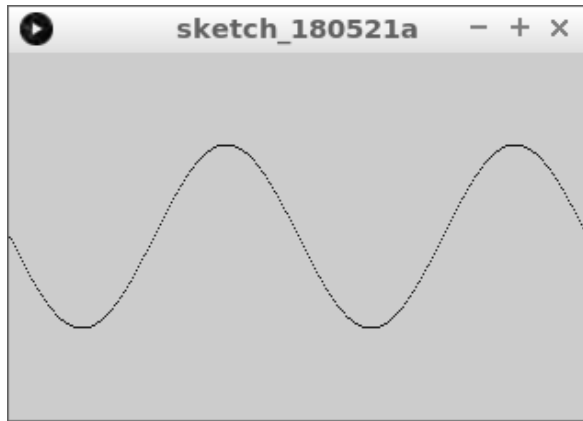


Figure 7: Sinus och cosinus 1: uppgift 4

Sinus och cosinus 4: uppgift 3

- Se till att `vinkel_kanon` blir noll när du trycker på mellanslagstangenten

Dricks:

```
if (keyPressed)
{
  if (key == ' ')
  {
    //Sett vinkel_kanon pa nol
  }
}
```

Sinus och cosinus 4: lösning 2

```
float vinkel_kanon = 0;

void setup()
{
  size(300, 200);
  strokeWeight(5);
}

void draw()
{
  background(255, 255, 255);
  final float x_mitten = width / 2;
  final float y_mitten = height / 2;
  final float x_kanon = x_mitten + (cos(vinkel_kanon) * 20);
  final float y_kanon = y_mitten - (sin(vinkel_kanon) * 20);
  line(x_mitten, y_mitten, x_kanon, y_kanon);
  ellipse(x_mitten, y_mitten, 20, 20);
  vinkel_kanon += 0.1;
}
```

Sinus och cosinus 1: lösning 4

Du ser nu en våg: en sinus!

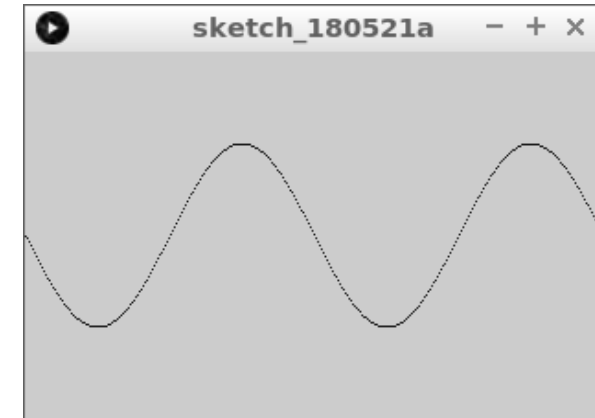


Figure 8: Sinus och cosinus 1: uppgift 4

```
float x = 0;

void setup()
{
  size(314, 200);
}

void draw()
{
  final float y = (sin(x / 25) * 50) + (height / 2);
  point(x, y);
  x = x + 1;
}
```

Sinus och Cosinus 1: uppgift 5

- Ersätt `sin(x)` med `cos(x)`

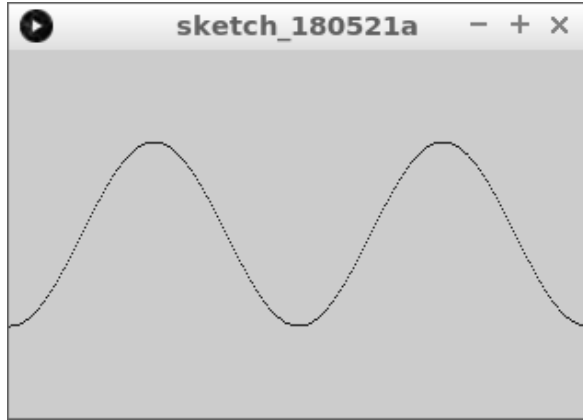


Figure 9: Sinus och cosinus 1: uppgift 5

Sinus och cosinus 1: lösning 5

Du ser nu en våg: en sinus!

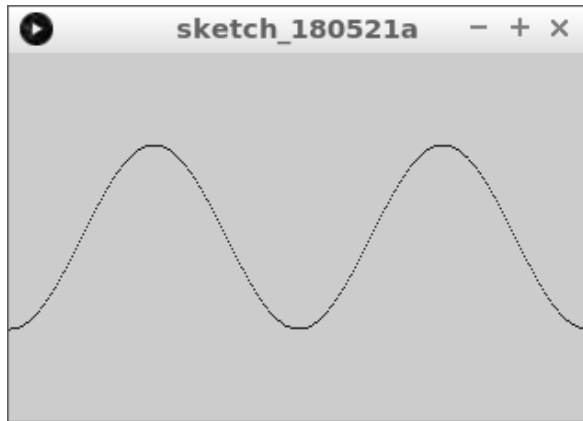


Figure 10: Sinus och cosinus 1: uppgift 5

```
float x = 0;

void setup()
{
  size(314, 200);
}
```

Sinus och cosinus 4: uppgift 2

- Få kanonen att rotera genom att göra `vinkel_kanon` mer och mer 0.1

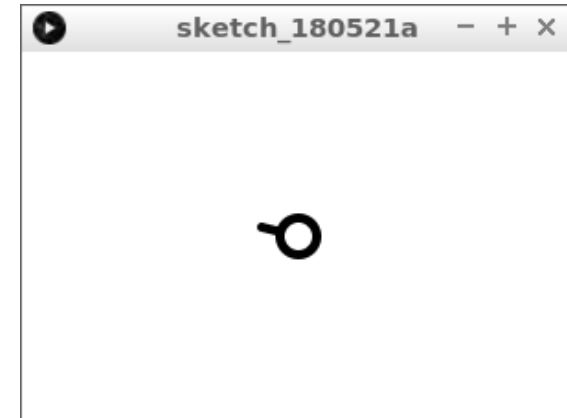


Figure 28: Sinus och cosinus 4: uppgift 2

Sinus och cosinus 4: lösning 1

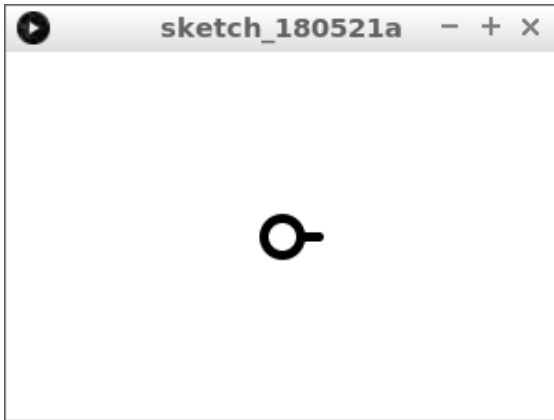


Figure 27: Sinus och cosinus 4: lösning 1

Du ser en cirkel i mitten, med en kanon

```
void draw()
{
  final float y = (sin(x / 25) * 50) + (height / 2);
  point(x, y);
  x = x + 1;
}
```



$x / 25$ ser till att vi går 25 gånger lugnare genom
sinus

Sinus och cosinus 1: Slutuppgift

- Rita både sinus och cosinus
- Få varje våg att gå upp och ner en gång
- Rita cosinus röd, sinus blå

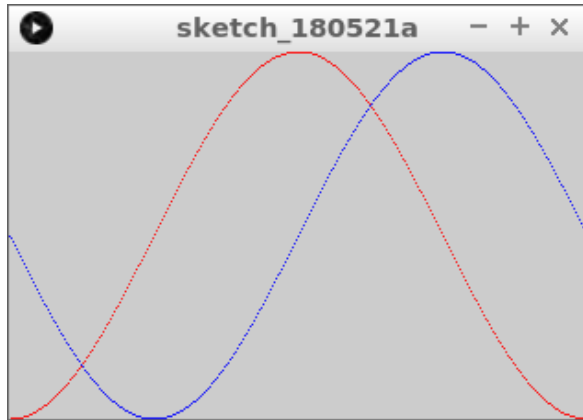


Figure 11: Sinus och cosinus 1: Slutuppgift

Sinus och cosinus 4: uppgift 1

Skriv denna kod över:

```
float vinkel_kanon = 0;

void setup()
{
  size(300, 200);
  strokeWeight(5);
}

void draw()
{
  background(255, 255, 255);
  final float x_mitten = width / 2;
  final float y_mitten = height / 2;
  final float x_kanon = x_mitten + (cos(vinkel_kanon) * 20);
  final float y_kanon = y_mitten - (sin(vinkel_kanon) * 20);
  line(x_mitten, y_mitten, x_kanon, y_kanon);
  ellipse(x_mitten, y_mitten, 20, 20);
}
```

Vad ser du?

Sinus och cosinus 4: skjutning

Under den här lektionen ska vi använda en sinus och cosinus för att skjuta en kula



Figure 26: Xybots

Sinus och cosinus 2: rita

Under den här lektionen ska vi använda en sinus och cosinus för att få en måne att kretsa runt en planet.

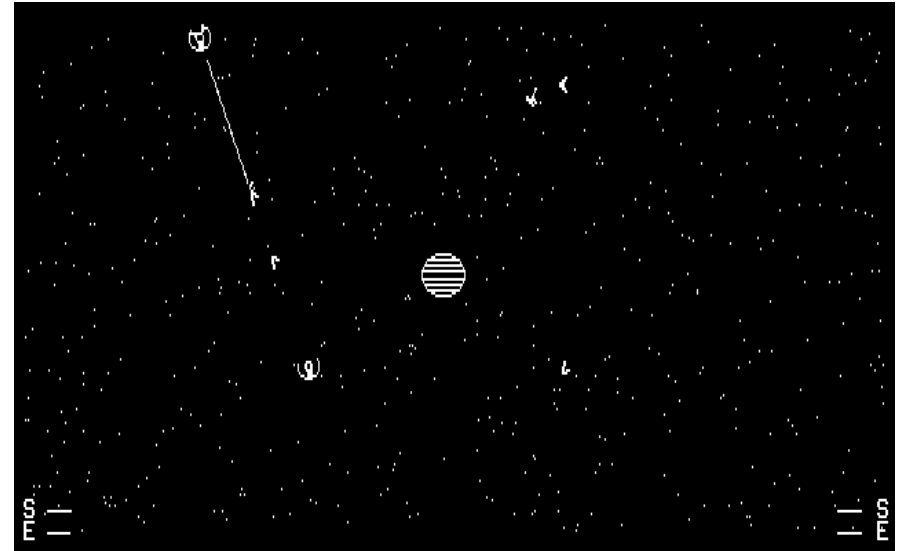


Figure 12: Spacewar

Sinus och cosinus 2: uppgift 1

Skriv denna kod över:

```
float vinkel = 0;

void setup()
{
  size(300, 200);
}

void draw()
{
  final float x = cos(vinkel) * 100;
  final float y = sin(vinkel) * 100;
  point(x, y);
  vinkel = vinkel + 0.01;
}
```

Vad ser du?

Sinus och cosinus 3: Slutuppgift

Låt månen kretsa runt jorden!

- Skapa en ny variabel `vinkel_man`, lägg den högst upp
- Gör `vinkel_man` mer i slutet av `draw`-funktionen. Använd ett värde som 0.1.
- Gör `x_man` nu `x_jord` plus tio gånger sinus för `vinkel_jord`
- Samma för `y_man`.

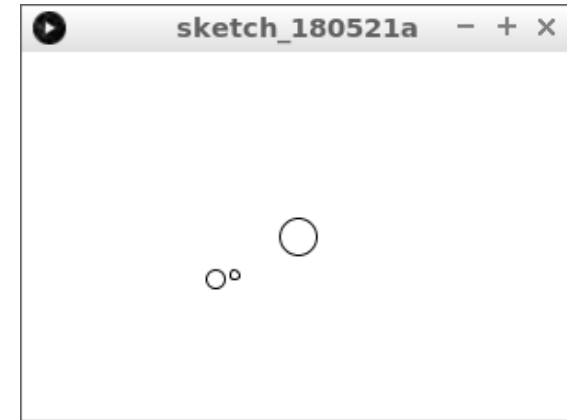


Figure 25: Sinus och cosinus 3: Slutuppgift

Sinus och Cosinus 3: uppgift 5

- Skapa två nya variabler för månen: `x_man` och `y_man` (inom `draw`-funktionen)
- Värdet på `x_man` är värdet på `x_jord` plus 10
- Värdet på `y_man` är värdet på `y_jord` plus 10
- Rita månen som en cirkel med en bredd och höjd på 5 pixlar

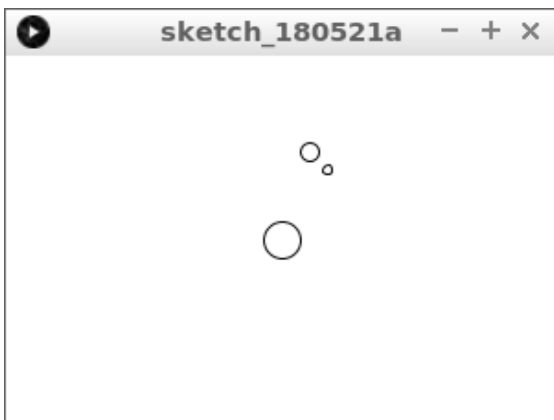


Figure 24: Sinus och cosinus 3: uppgift 5

Sinus och cosinus 2: lösning 1

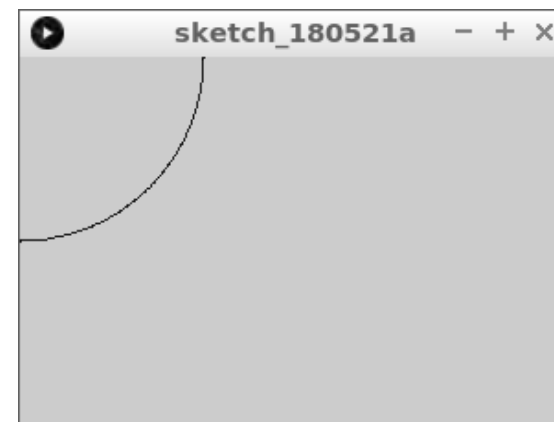


Figure 13: Sinus och cosinus 2: lösning 1

Du kommer att se en kvartscirkel i det övre vänstra hörnet.

Sinus och Cosinus 2: uppgift 2

- Flytta cirkeln till mitten
- Gör detta genom att lägga till halva höjden med y
- Gör detta genom att lägga till halva bredden med x .



Tips: det är smart att sätta parenteser runt `height / 2`

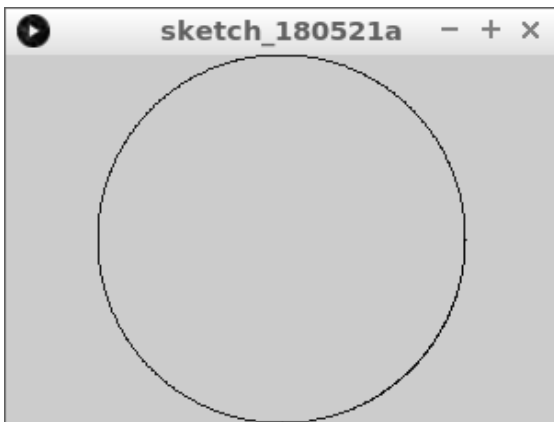


Figure 14: Sinus och cosinus 2: uppgift 2

Sinus och cosinus 3: uppgift 4

- Se till att jorden går runt solen nu
- Gör nu `y_jord` `y_sol` plus femtio gånger cosinus för `vinkel_jord`

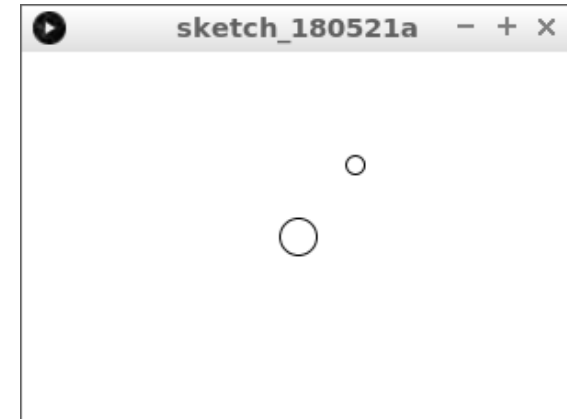


Figure 23: Sinus och cosinus 3: uppgift 4

Sinus och cosinus 3: lösning 3

Du ser nu jorden gå fram och tillbaka

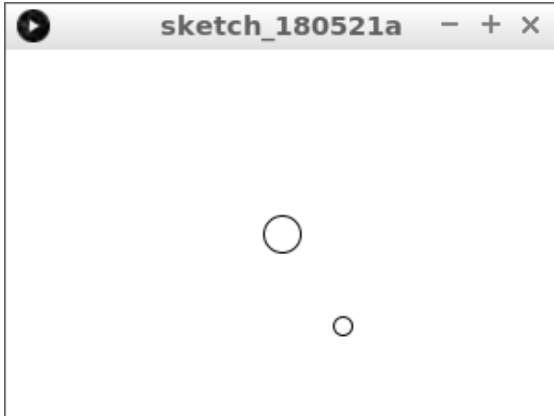


Figure 22: Sinus och cosinus 3: uppgift 2

```
float vinkel_jord = 0;

void setup()
{
  size(300, 200);
}

void draw()
{
  background(255, 255, 255);
  final float x_sol = width / 2;
  final float y_sol = height / 2;
  ellipse(x_sol, y_sol, 20, 20);
  final float x_jord = x_sol + (50 * sin(vinkel_jord));
  final float y_jord = y_sol + 50;
  ellipse(x_jord, y_jord, 10, 10);
  vinkel_jord += 0.01;
}
```

Sinus och cosinus 2: lösning 2

```
float vinkel = 0;

void setup()
{
  size(300, 200);
}

void draw()
{
  final float x = cos(vinkel) * 100 + (width / 2);
  final float y = sin(vinkel) * 100 + (height / 2);
  point(x, y);
  vinkel = vinkel + 0.01;
}
```



Fästena hjälper datorn att veta ordningen på en
beräkning



En sinus och cosinus tillsammans kan rita en cirkel

Sinus och cosinus 2: uppgift 3

- Rita en cirkel istället för en punkt. Cirkeln ska vara 20 pixlar bred och 20 pixlar hög

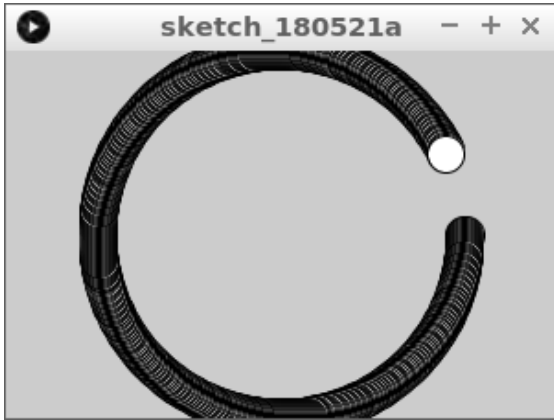


Figure 15: Sinus och cosinus 2: uppgift 3

Sinus och Cosinus 3: uppgift 3

- Skapa en ny variabel `vinkel_jord`, lägg den högst upp
- Gör `vinkel_jord` mer i slutet av `draw`-funktionen. Använd ett värde som `0.01`.
- Gör `x_jord` nu `x_sol` plus femtio gånger sinus för `vinkel_jord`

Vad ser du?

Sinus och cosinus 3: lösning 2

```
void setup()
{
  size(300, 200);
}

void draw()
{
  background(255, 255, 255);
  final float x_sol = width / 2;
  final float y_sol = height / 2;
  ellipse(x_sol, y_sol, 20, 20);
  final float x_jord = x_sol + 50;
  final float y_jord = y_sol + 50;
  ellipse(x_jord, y_jord, 10, 10);
}
```

Sinus och cosinus 2: lösning 3

Du kommer nu att se ett mönster.

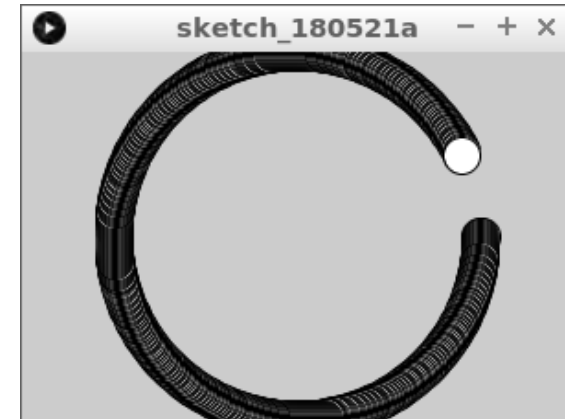


Figure 16: Sinus och cosinus 3: uppgift 2

```
float vinkel = 0;

void setup()
{
  size(300, 200);
}

void draw()
{
  final float x = cos(vinkel) * 100 + (width / 2);
  final float y = sin(vinkel) * 100 + (height / 2);
  ellipse(x, y, 20, 20);
  vinkel = vinkel + 0.01;
}
```

Sinus och cosinus 2: uppgift 4

- Se till att bollen inte lämnar en linje längre
- Använd en vit bakgrund för detta



Figure 17: Sinus och cosinus 2: uppgift 4



Tips: använd `background`

Sinus och Cosinus 3: uppgift 2

- Skapa två nya variabler: `x_jord` och `y_jord` (inom `draw`-funktionen)
- Värdet på `x_jord` är värdet på `x_sol` plus 50
- Värdet på `y_jord` är värdet på `y_sol` plus 50
- Rita jorden som en cirkel med en bredd och höjd på 10 pixlar

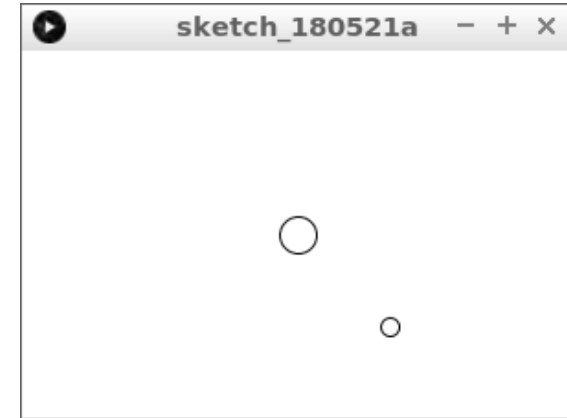


Figure 21: Sinus och cosinus 3: uppgift 2

Sinus och cosinus 3: lösning 1

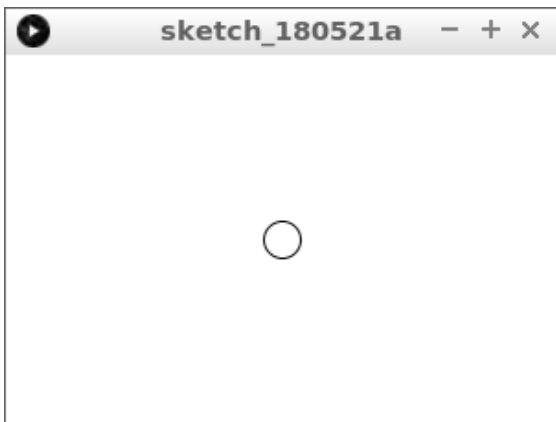


Figure 20: Sinus och cosinus 3: lösning 1

Du ser en cirkel i mitten: solen!

Sinus och cosinus 2: Slutuppgift

- Rita också en cirkel i mitten, med en bredd och höjd på 50 pixlar
- Gör den rörliga cirkeln moturs
- Se till att den rörliga cirkeln inte längre går halvvägs ut från skärmen. Gör det här, genom att göra avståndet från centrum mindre

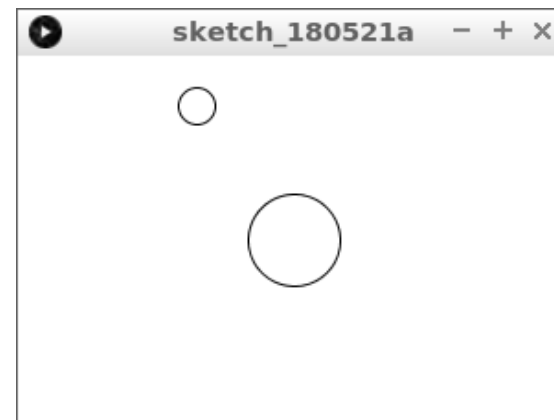


Figure 18: Sinus och cosinus 2: Slutuppgift

Sinus och cosinus 3: solsystem

I den här lektionen ska vi använda en sinus och cosinus för att skapa ett solsystem



Figure 19: Chaos Engine

Sinus och Cosinus 3: uppgift 1

Skriv denna kod över:

```
void setup()
{
  size(300, 200);
}

void draw()
{
  background(255, 255, 255);
  final float x_sol = width / 2;
  final float y_sol = height / 2;
  ellipse(x_sol, y_sol, 20, 20);
}
```

Vad ser du?