

Koordinatsystem i Processing:

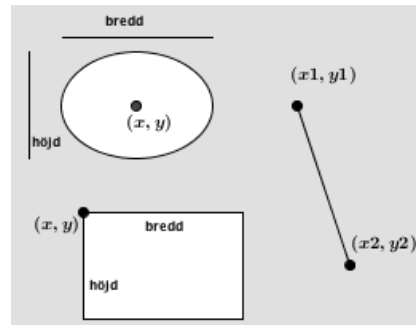
I alla kodexempel skall de kursiverade blå orden bytas mot siffror

Sätt fönstrets storlek

```
size(bredd, höjd);
```

Gör geometriska figurer

```
rect(x, y, bredd, höjd);
ellipse(x, y, bredd, höjd);
line(x1, y1, x2, y2);
```



Ändra färgerna

```
background(röd, grön, blå);
fill(röd, grön, blå);
stroke(röd, grön, blå); //kantens färg
```

röd, grön, blå, skall vara tal mellan 0 och 255

Testa även

```
strokeWeight(pennStorlek);
noFill();
noStroke();
```

Lägg in en bild

Spara din sketch. Välj Sketch -> Add file...

Lägg in en bild som du har sparad på datorn.

Om bilden heter *minBild.png*, skriv in:

```
PImage bild = loadImage("minBild.png");
image(bild, x, y);
```

eller

```
PImage bild = loadImage("minBild.png");
image(bild, x, y, bredd, höjd);
```

Visa text

```
textSize(textStorlek);
text("Hello World!", x, y);
```

Få saker att röra på sig

Gör en ny Processing-sketch och skriv in:

```
void setup() {
  size(600, 400);
  //kod som körs när programmet startar
}

void draw() {
  //kod som körs om och om igen
  //som ett "för alltid"-block i Scratch
}
```

Det som står efter // är så kallade kommentarer. Kommentarer är till för den som läser koden och exekveras inte.

Slumptal

Ett slumptal mellan talen *a* och *b* genereras med koden `random(a, b)`.
(För att vara noga är $a \leq \text{random}(a, b) < b$)

Testa att lägga in denna rad i draw-metoden:

```
rect(random(0, 300), random(0, 200), 10, 10);
```

Man kan också slumpa färgerna röd, grön, blå, eller andra tal som används.

Om man skriver `random(b)`, får man ett slumptal mellan 0 och *b*.

Exempel: `fill(random(256), 0, 0);`

Följ muspekaren!

Muspekarens koordinater ges av `mouseX` och `mouseY`. Testa:
`ellipse(mouseX, mouseY, 20, 20);`

Testa att lägga in denna rad överst i draw-metoden:

```
background(0, 0, 0);
```

Villkorsatser (if-satser)

Motsvarigheterna till Scratchs "om då"- och "om då annars"-block:

```
if (mousePressed) {  
    ellipse(mouseX, mouseY, 20, 20);  
}
```

eller

```
if (mousePressed) {  
    fill(200, 0, 0);  
} else {  
    fill(0, 0, 200);  
}
```

Variabler och datatyper

Om du vill använda en variabel för att lagra värden, måste du ange vilken **datatyp** variabeln skall vara. Tal kan antingen var `float` (decimaltal) eller `int` (heltal)

Jämförelse med Scratch

Scratch-kod	Processing-kod
Skapa variabeln decimaltal	<code>float decimaltal;</code>
Skapa variabeln heltal	<code>int heltal;</code>
sätt (eller tilldela)	<code>=</code>
ändra (addera till)	<code>+=</code>
större än	<code>></code>
mindre än	<code><</code>
lika med	<code>==</code>
och	<code>&&</code>
eller	<code> </code>
avstånd till muspekaren	<code>dist(mouseX, mouseY, x, y);</code>

Förflyttningar

I Scratch finns en sprajts position lagrad i "x-läge" och "y-läge". I Processing måste du själv införa variabler för att ha koll på positioner. I Scratch finns det särskilda block som hanterar kollisioner, exempelvis "studsas, vid kanten", eller "rör muspekaren". I Processing måste du själv hantera all kollisionsdetektering, med hjälp av koordinater. En enkel förflyttning nedåt, visas nedan. Testkör koden!

```
float xPos, yPos;
```

```
void setup() {  
    size(600, 400);  
    xPos = random(600);  
    yPos = 0;  
    textSize(24);  
}
```

```
void draw() {  
    background(0, 0, 0); //kan även skrivas background(0)  
    ellipse(xPos, yPos, 20, 20);  
    if (mousePressed && dist(mouseX, mouseY, xPos, yPos) < 10) {  
        text("träff!", mouseX, mouseY-20);  
    }  
    if (yPos > 400) { //kollision med nedre sidan av fönstret  
        xPos = random(600);  
        yPos = 0;  
        fill(random(256), random(256), random(256));  
    }  
    yPos += 1;  
}
```

Indentering

För att koden skall vara lättläst, drar man in vissa rader. Detta kallas att **indentera**. Exempelvis är alla rader inom en if-sats indragna, så att man lättare kan se vad som tillhör if-satsen.

Om du inte redan har indenterat koden, tryck `ctrl+t` (Windows/Linux) eller `cmd+t` (Mac)!