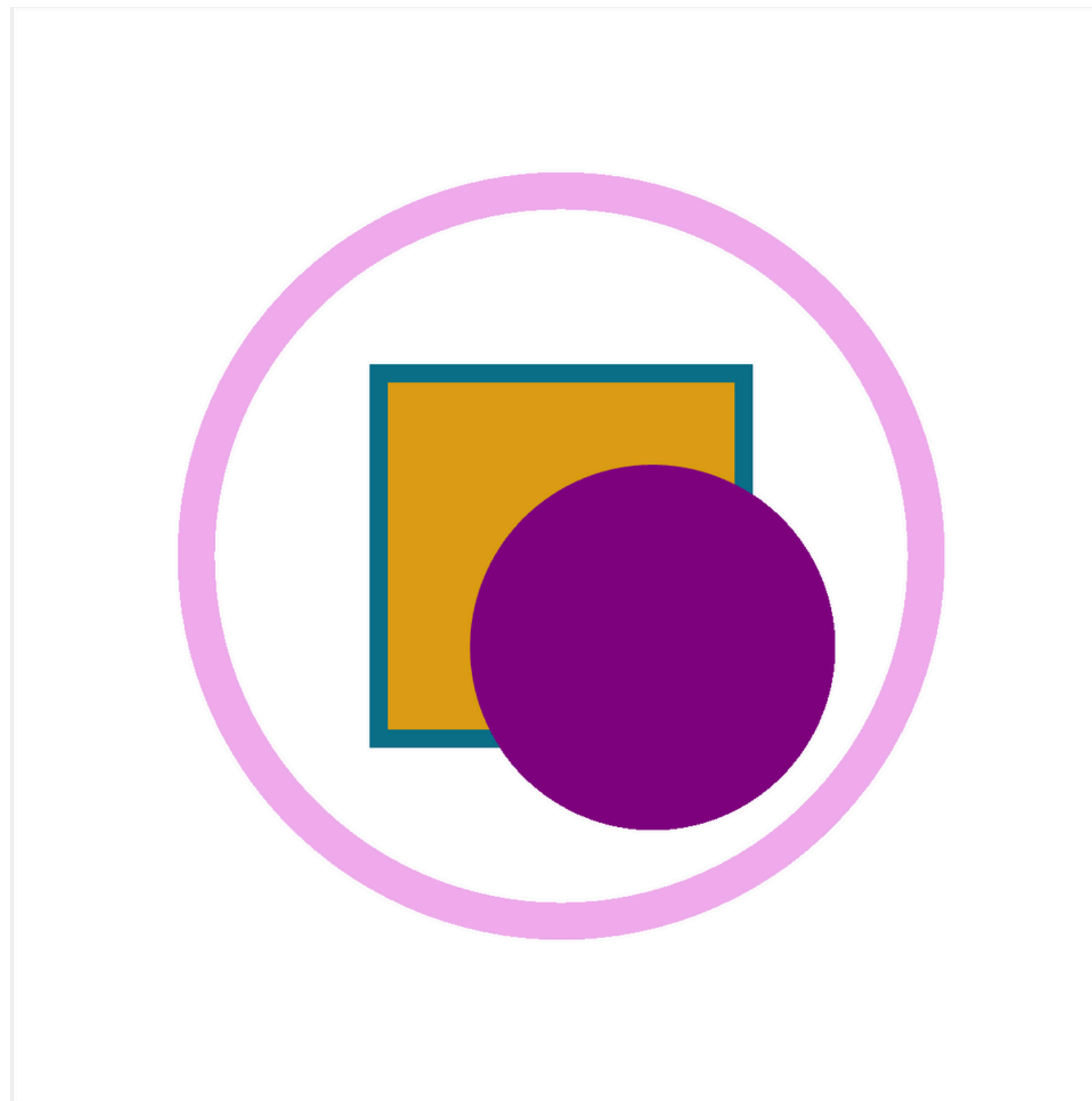


Quickguide til at tegne statiske objekter i OpenProcessing

Find tutorial online, hvor du kan ændre direkte i koderne, her: oppr.org/s/8YqAXGf8



Opsætning

Open Processing starter altid op med en lille eksempelkode.

Størrelse på lærred:

Start med at angive ønsket bredde og højde for dit lærred i parenteser i instruktionen:

```
createCanvas(windowWidth, windowHeight);
```

Farve på lærred:

Skriv farve på engelsk i **citationstegn** i parenteser i instruktionen::

```
background(100);
```

Slet

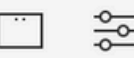
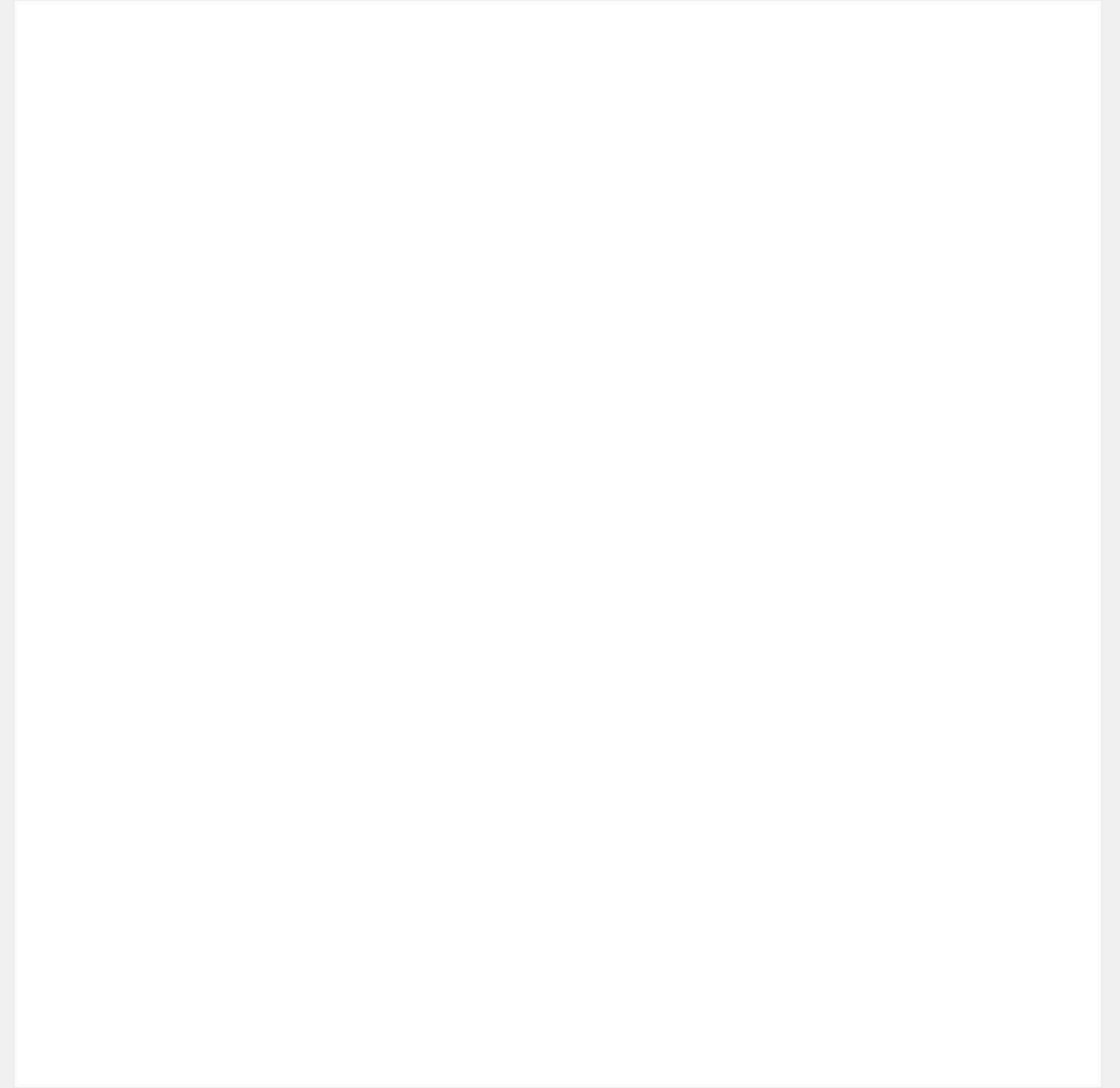
Slet kodeinstruktionen, der vises som default, i tegnefunktionen. **Slet IKKE de to tuborgklammer, der omkranser instruktionen**

```
circle(mouseX, mouseY, 20);
```

I eksemplet er lærredet gjort hvidt og fået størrelsen 600, 600. Nu er du klar til at lave din egen kode. Gå til næste side.

mySketch

```
1 function setup() {  
2   createCanvas(600,600);  
3   background("□white");  
4 }  
5  
6 function draw() {  
7  
8 }
```



Linjer

En linje skal angives som en streg mellem 2 punkter.
Hvert punkt udgøres af et X og et Y.

Koden er altså:

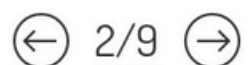
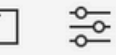
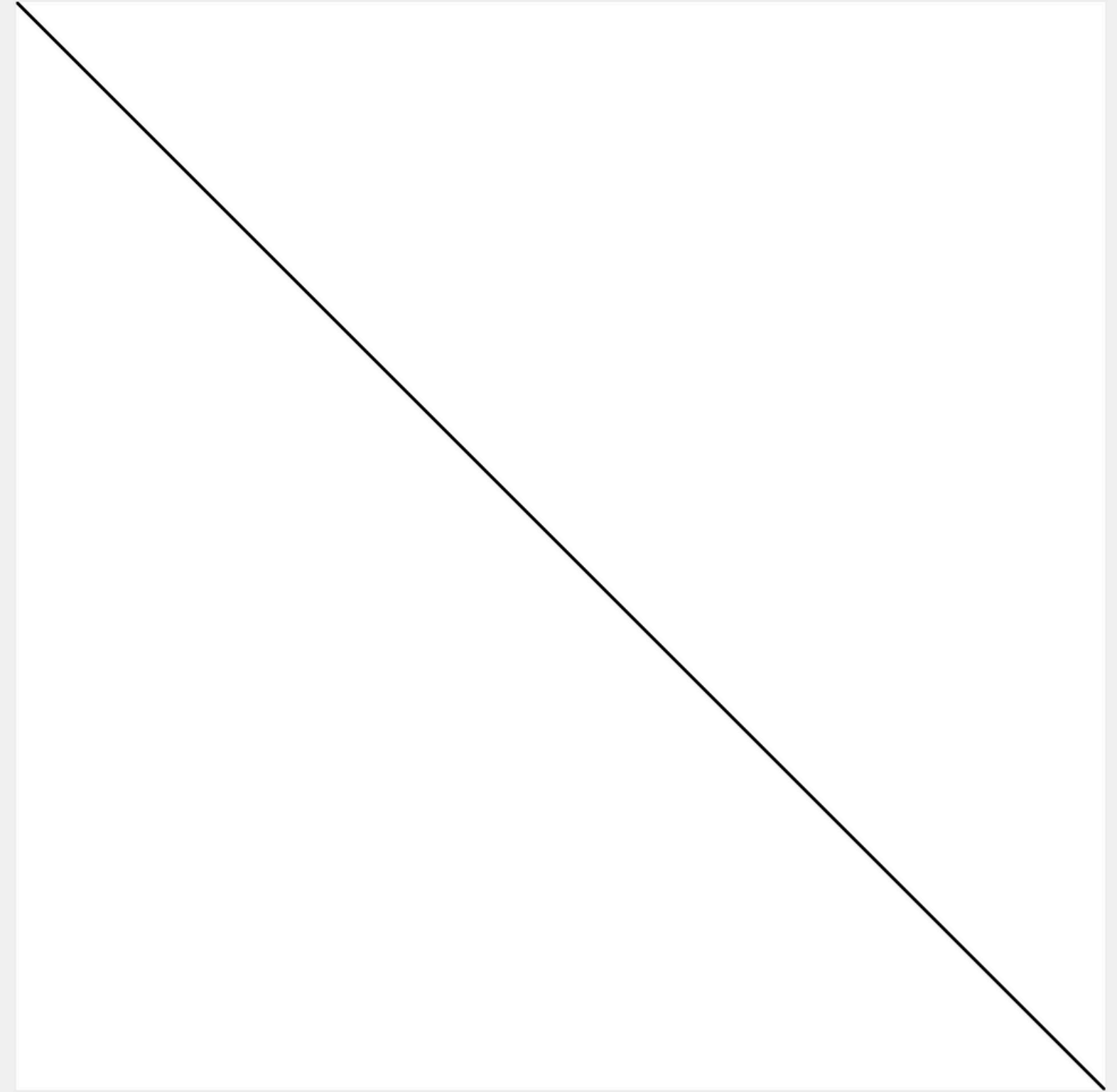
```
line (X1, Y1, X2, Y2);
```

Prøv selv at lave linjer andre steder på lærredet.

Skriv de nye instruktioner inde i tegnefunktionen
(mellem de to tuborgklammer)

mySketch

```
1 function setup() {  
2   createCanvas(600,600);  
3   background("white");  
4 }  
5  
6 function draw() {  
7   line(0,0, 600, 600)  
8   //skriv nye instruktioner her  
9  
10  
11 }
```



Trekanten

En trekant tegnes ved at trække streger mellem tre punkter. Hvert punkt udgøres af et X og et Y.

Koden er altså:

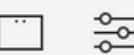
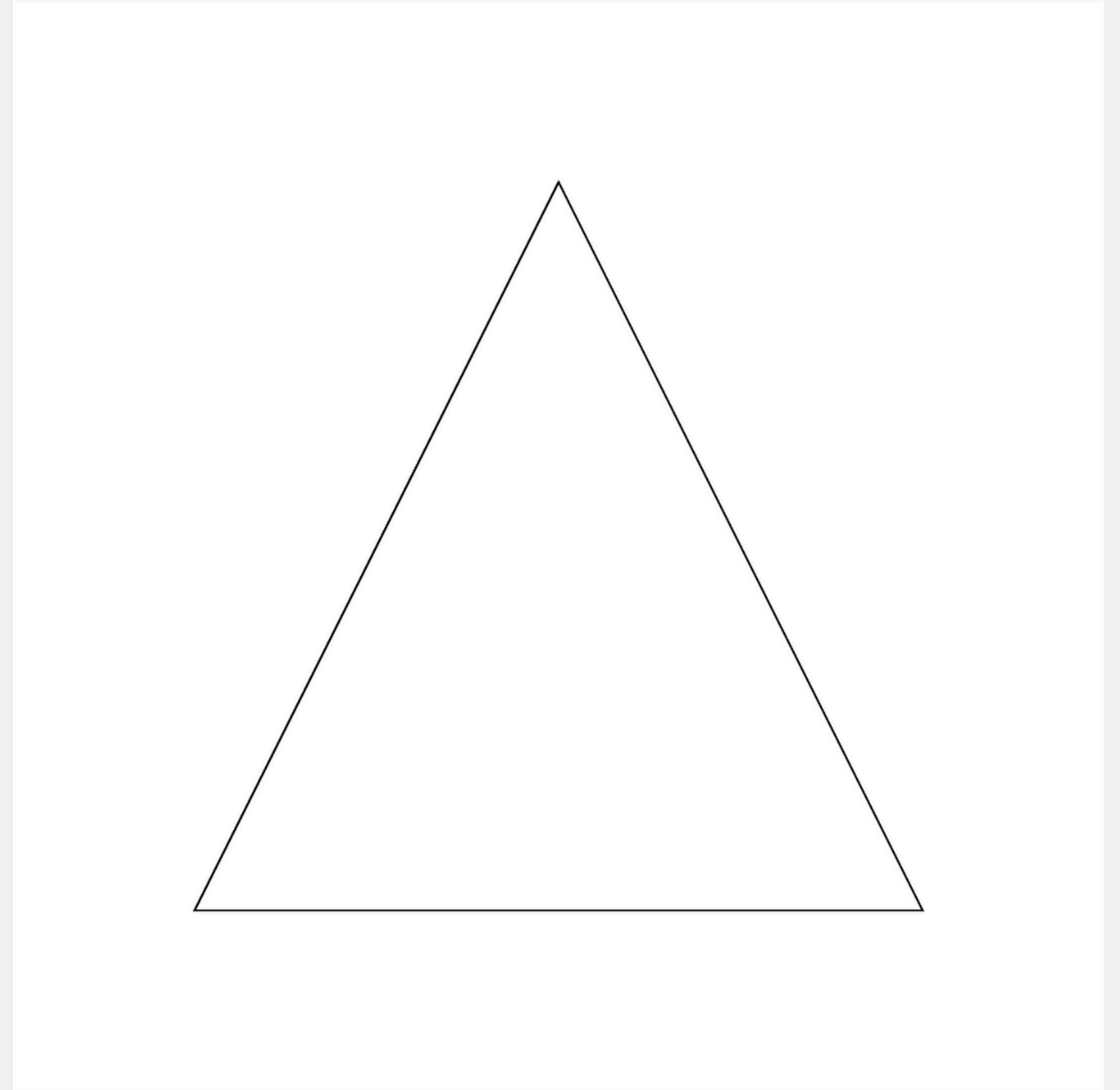
```
triangle (X1, Y1, X2, Y2, X3, Y3);
```

Prøv selv at lave trekanten andre steder på lærredet.

Skriv de nye instruktioner inde i tegnefunktionen (mellem de to tuborgklammer).

mySketch

```
1 function setup() {  
2   createCanvas(600,600);  
3   background("white");  
4 }  
5  
6 function draw() {  
7   triangle(300, 100, 500, 500, 100, 500);  
8   //Skriv nye instruktioner her  
9  
10 }
```



Cirkler

Cirkler tegnes ved at angive X og Y for cirkelens centrum + diameter.

Koden er altså:

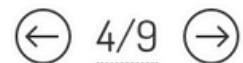
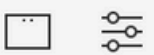
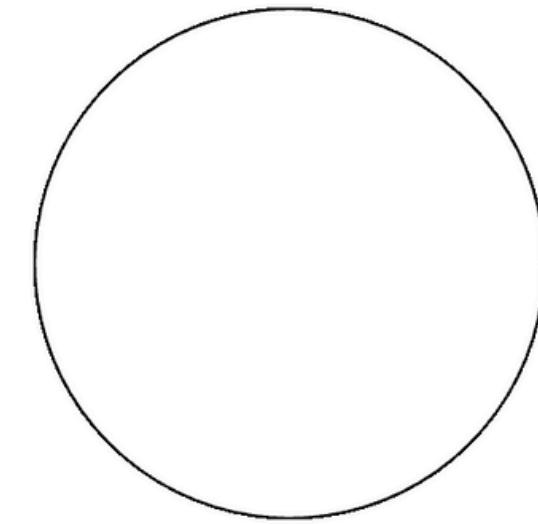
```
circle(X, Y, Diameter);
```

Prøv selv at lave cirkler andre steder på lærredet.

Skriv de nye instruktioner inde i tegnefunktionen (mellem de to tuborgklammer).

mySketch

```
1 function setup() {  
2   createCanvas(600,600);  
3   background("white");  
4 }  
5  
6 function draw() {  
7   circle(300, 300, 200);  
8   //Skriv nye instruktioner her  
9  
10 }
```



Ellipser

Ellipser er næsten som cirkler og her skal du også starte med at angive X og Y for ellipsens centrum, men i stedet for diameter, skal du skrive ellipsens bredde og højde

Koden er altså:

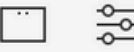
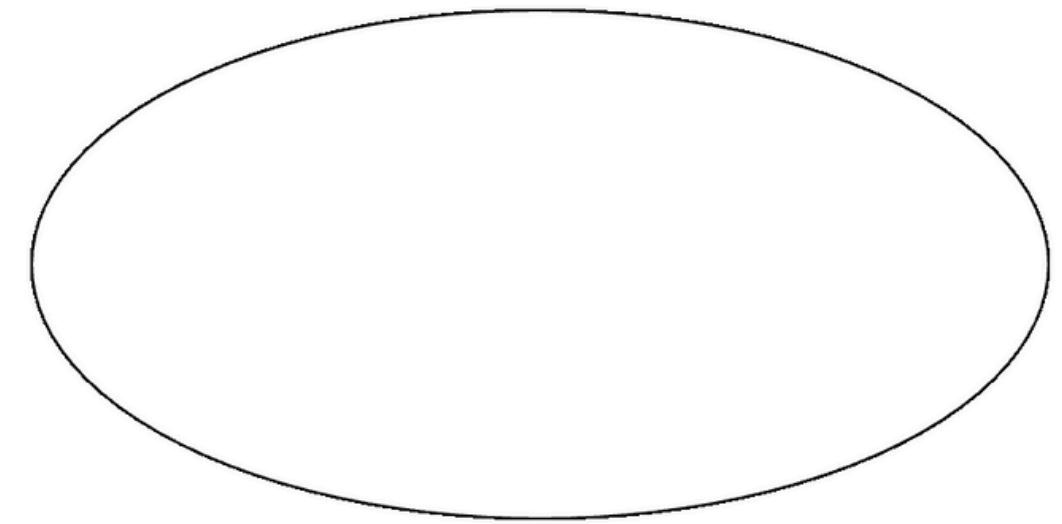
```
ellipse(X, Y, Bredde, Højde)
```

Prøv selv at lave ellipser andre steder på lærredet.

Skriv de nye instruktioner inde i tegnefunktionen (mellem de to tuborgklammer).

mySketch

```
1 function setup() {  
2   createCanvas(600,600);  
3   background("white");  
4 }  
5  
6 function draw() {  
7   ellipse(300, 300, 400, 200)  
8   //Skriv nye instruktioner her  
9  
10 }
```



Rektangler

Rektangler tegnes ved at angive X og Y for øverste venstre hjørne. Derefter angives bredde og højde (nedad).

Koden er altså:

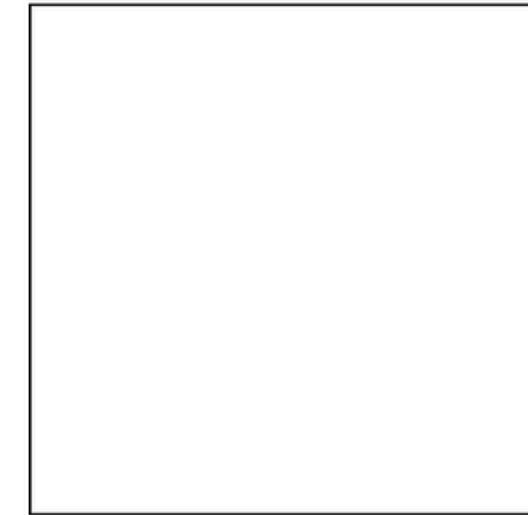
```
rect(X, Y, Bredde, Højde);
```

Prøv selv at lave rektangler andre steder på lærredet.

Skriv de nye instruktioner inde i tegnefunktionen (mellem de to tuborgklammer).

mySketch

```
1 function setup() {  
2   createCanvas(600,600);  
3   background("white");  
4 }  
5  
6 function draw() {  
7   rect(200, 200, 200, 200);  
8   //Skriv nye instruktioner her  
9  
10 }
```



Farver

Farver angives ved at bruge følgende instruktioner:

Udfyldning:

```
fill("farve på engelsk");
```

For konturlinje og ved linjer, som ikke kan farvelægges med "fill":

```
stroke("farve på engelsk");
```

Farver angives altid, **FØR** objekt tegnes

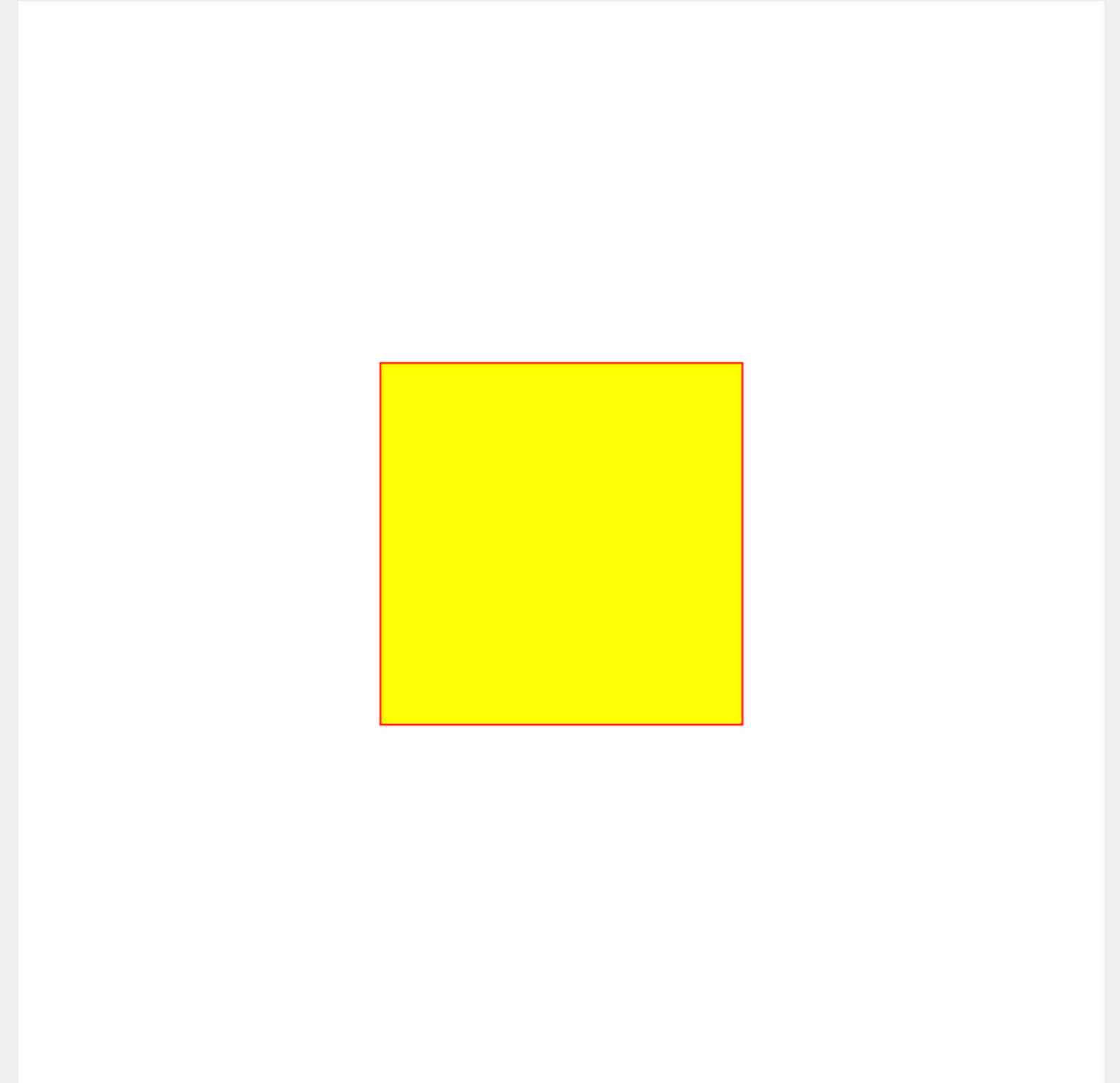
Husk citationstegn omkring farvens navn - de fortæller programmet, at dit input skal læses som tekst.

Klik evt. på den farvede firkant, der vises ud for navnet - den giver adgang til farvevælger.

På www.algoritmisk-kunst.dk kan du desuden se, hvordan du kan angive farver med HEX-kode eller RGB-værdier, hvis du vil have en helt bestemt nuance,

mySketch

```
1 function setup() {  
2   createCanvas(600,600);  
3   background("white");  
4 }  
5  
6 function draw() {  
7   stroke("red");  
8   fill("yellow");  
9   rect(200, 200, 200, 200);  
10 }
```



Stregtykkelse

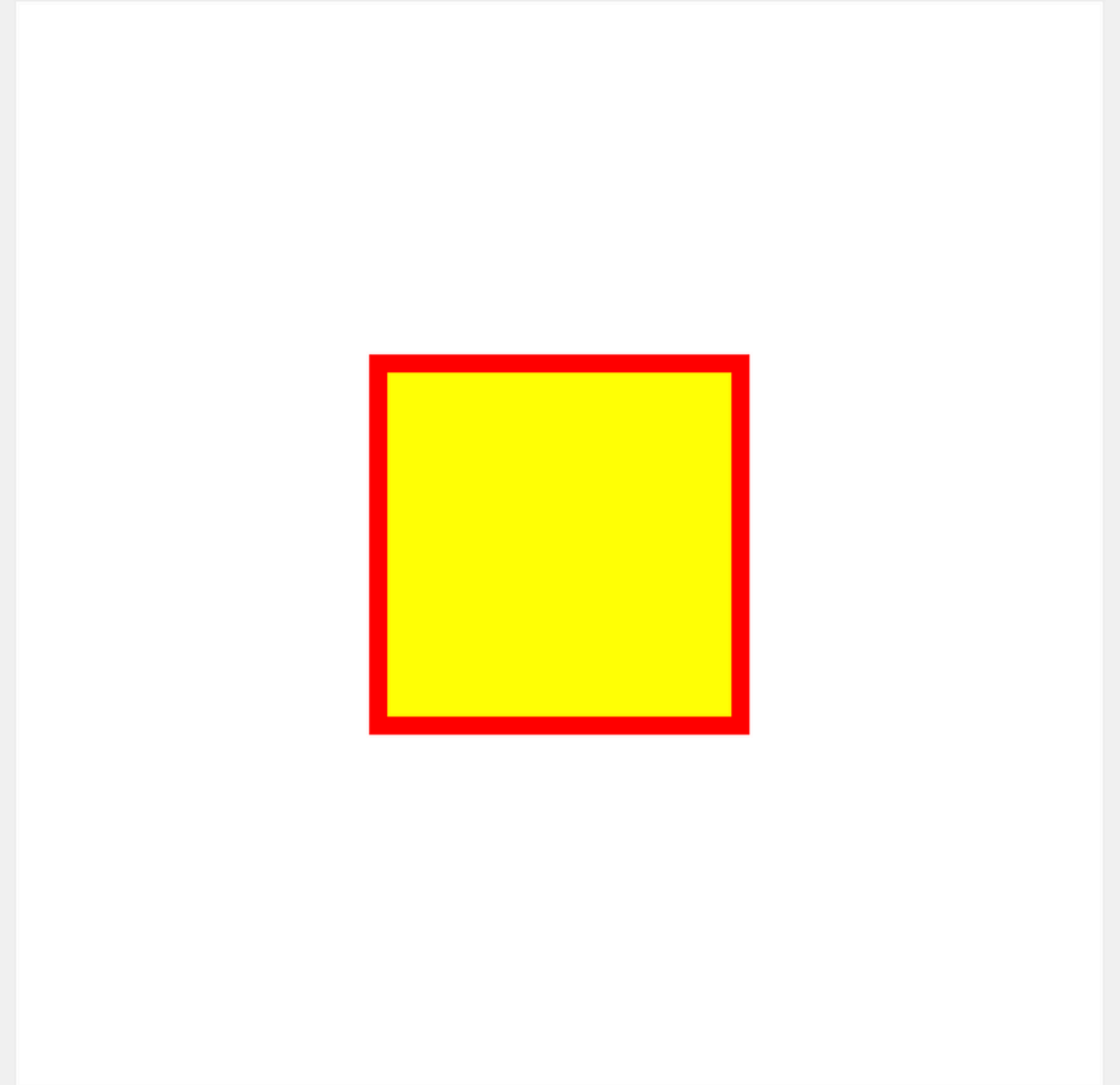
Du kan også nemt ændre tykkelsen på konturlinjen.
Det gør du med instruktionen:

```
strokeWeight(skriv tal her);
```

Stregtykkelse skal angives, **FØR** objekt tegnes.

mySketch

```
1 function setup() {  
2   createCanvas(600,600);  
3   background("white");  
4 }  
5  
6 function draw() {  
7   strokeWeight(10);  
8   stroke("red");  
9   fill("yellow");  
10  rect(200, 200, 200, 200);  
11 }
```



Ophæv udfyldning og fjern konturlinje

Nogle gange vil du måske tegne flere objekter med forskellige indstillinger. Du kan bruge instruktionerne:

Ophævning af konturlinje:

```
noStroke();
```

Ophævning af udfyldning

```
noFill();
```

Programmet læser oppefra og ned og husker din seneste indstilling, så definér hver gang, du vil ændre noget. Da programmet tegner 60 gange i sekundet, ville konturtykkelsen i eksemplet her derfor være 20 på kvadratet efter første gennemløb af koden, hvis ikke den havde sin egen instruktion, der angiver den til 10.

Tænk også over, at rækkefølgen på dine objekter bestemmer, hvordan objekterne lægger sig, hvis de overlapper hinanden.

mySketch

```
1 function setup() {
2   createCanvas(600,600);
3   background("white");
4 }
5
6 function draw() {
7   strokeWeight(10);
8   stroke("rgb(11,110,136)");
9   fill("rgb(220,158,19)");
10  rect(200, 200, 200, 200);
11
12  noStroke();
13  fill("purple");
14  circle(350, 350, 200);
15
16  noFill();
17  strokeWeight (20);
18  stroke("rgb(240,173,235)");
19  circle(300, 300, 400)
20
21
22 }
```

