



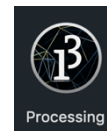
Tutorial

Zeichenprogramm in Java mit Processing

In diesem Tutorial lernst Du, wie man ein einfaches **Zeichenprogramm** in **Java** mit Hilfe von **Processing** programmiert. Folge dabei den Anweisungen Schritt für Schritt.

Schritt 1

Öffne **Processing** durch Doppelklick auf das Processing-Symbol am Desktop:



Schritt 2

Gib nun folgenden Text ein:



```
sketch_190120a
1 ellipse(30, 30, 30, 30);
2
```

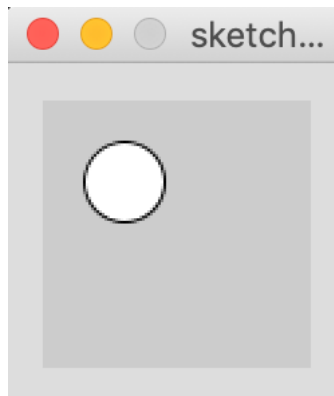
Schritt 3

Starte Dein Programm durch Klick auf den **Play**-Knopf:



```
sketch_190120a
1 ellipse(30, 30, 30, 30);
2
```

Nun solltest Du folgendes Ergebnis sehen:



Wir haben also einen Kreis gezeichnet, dessen Mittelpunkt sich 30 Pixel links und 30 Pixel unterhalb der linken oberen Ecke befindet. Der Kreis hat den Radius 30 Pixel.

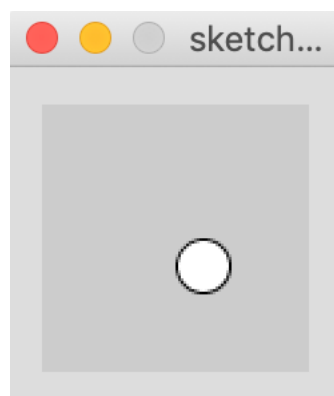
Schritt 4

Verschiebe nun den Kreis an die Stelle 60/60 und ändere den Radius auf 20 Pixel. Öffne dazu wieder den Texteditor und gib dort folgendes ein:



Schritt 5

Starte Dein Programm erneut durch Klick auf den **Play**-Knopf. Dein Ergebnis sollte nun so aussehen:



Schritt 6

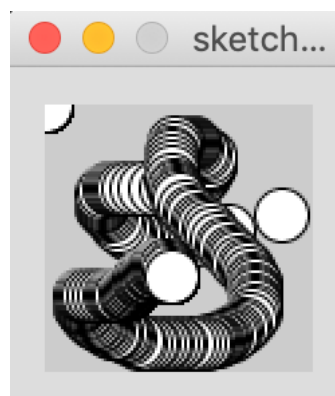
Nun ändern wir Dein Programm so, dass der Kreis nicht mehr an einer fixen, sondern bei der Position des **Mauszeigers** gezeichnet wird. Ersetze dazu den Text wie folgt:

The image shows a screenshot of the Processing IDE window titled "sketch_190120a | Processing 3...". At the top left, there are two circular buttons: a play button (green triangle) and a stop button (black square). Below these is a dropdown menu showing "sketch_190120a". The main area is a code editor with a dark background. The code is as follows:

```
1 void draw()  
2 {  
3   ellipse(mouseX, mouseY, 20, 20);  
4 }
```

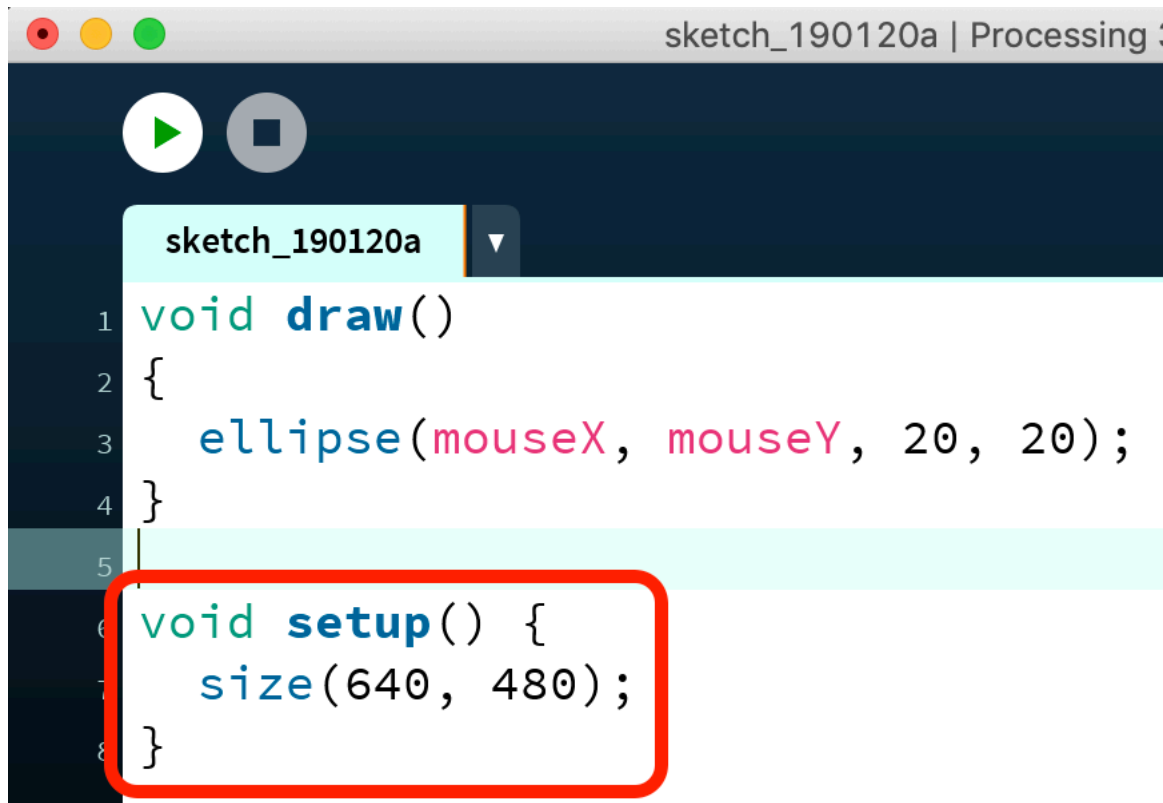
Die sgn. **Methode** „draw()“ wird dabei im Hintergrund immer und immer wieder aufgerufen bis Du selbst Dein Programm beendest (durch Klick auf den **Stop**-Knopf).

Starte nun Dein Programm erneut durch Klick auf den **Play**-Knopf. Wenn Du nun im angezeigten Fenster die Maus bewegt, solltest lauter Kreise gezeichnet werden:



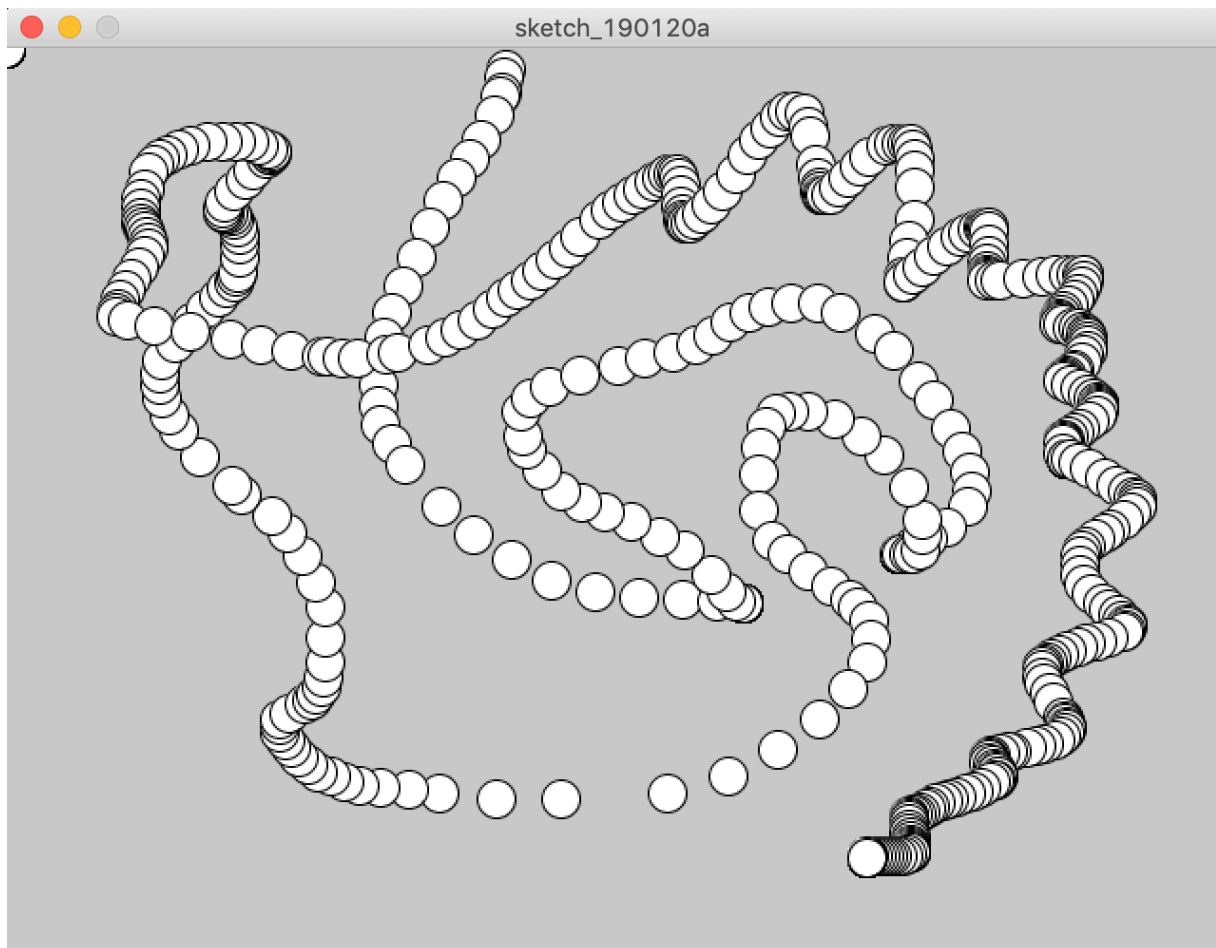
Schritt 7

Leider ist unser Fenster recht klein. Lass uns diese nun wie folgt ändern:



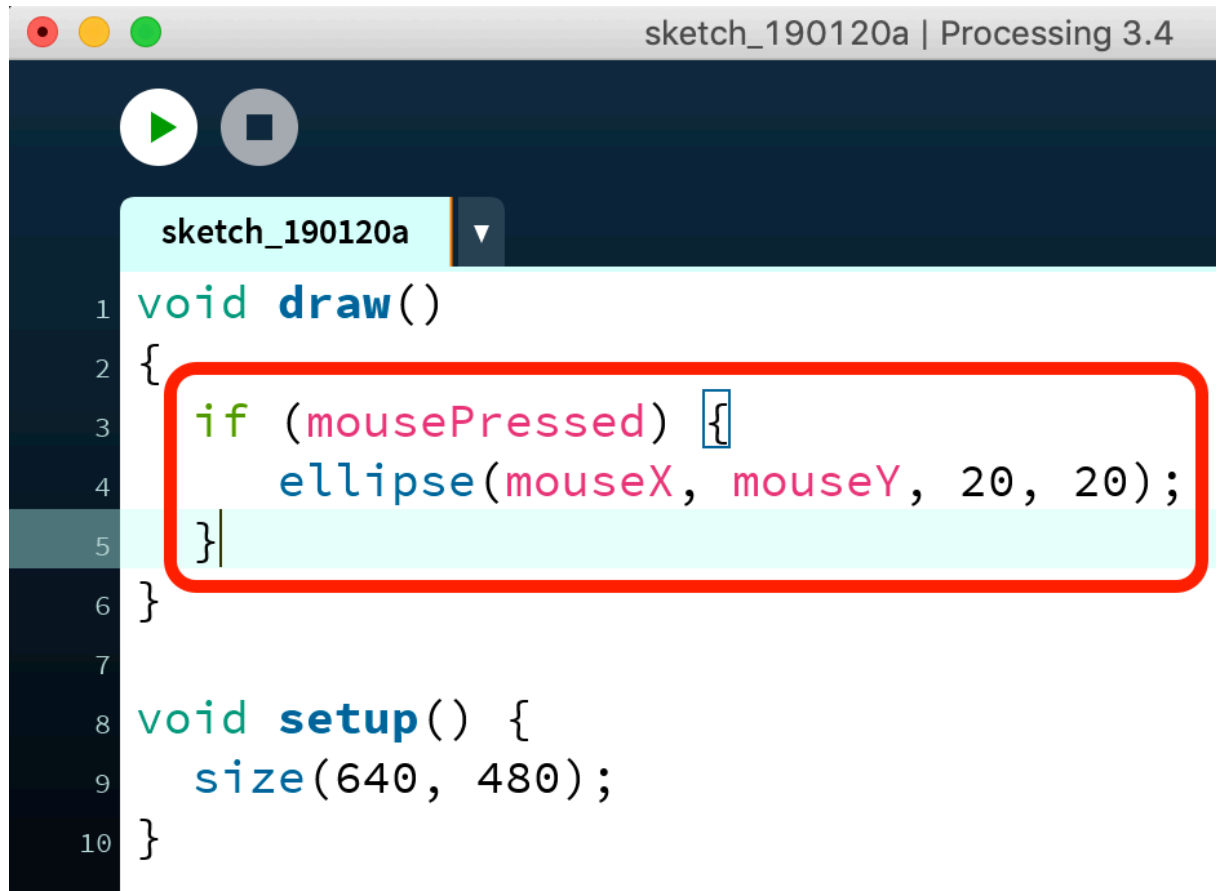
```
sketch_190120a | Processing :  
  
1 void draw()  
2 {  
3   ellipse(mouseX, mouseY, 20, 20);  
4 }  
5  
6 void setup() {  
7   size(640, 480);  
8 }
```

Starte Dein Programm erneut durch Klick auf den **Play**-Knopf. Nun sollte das Fenster deutlich größer sein:



Schritt 8

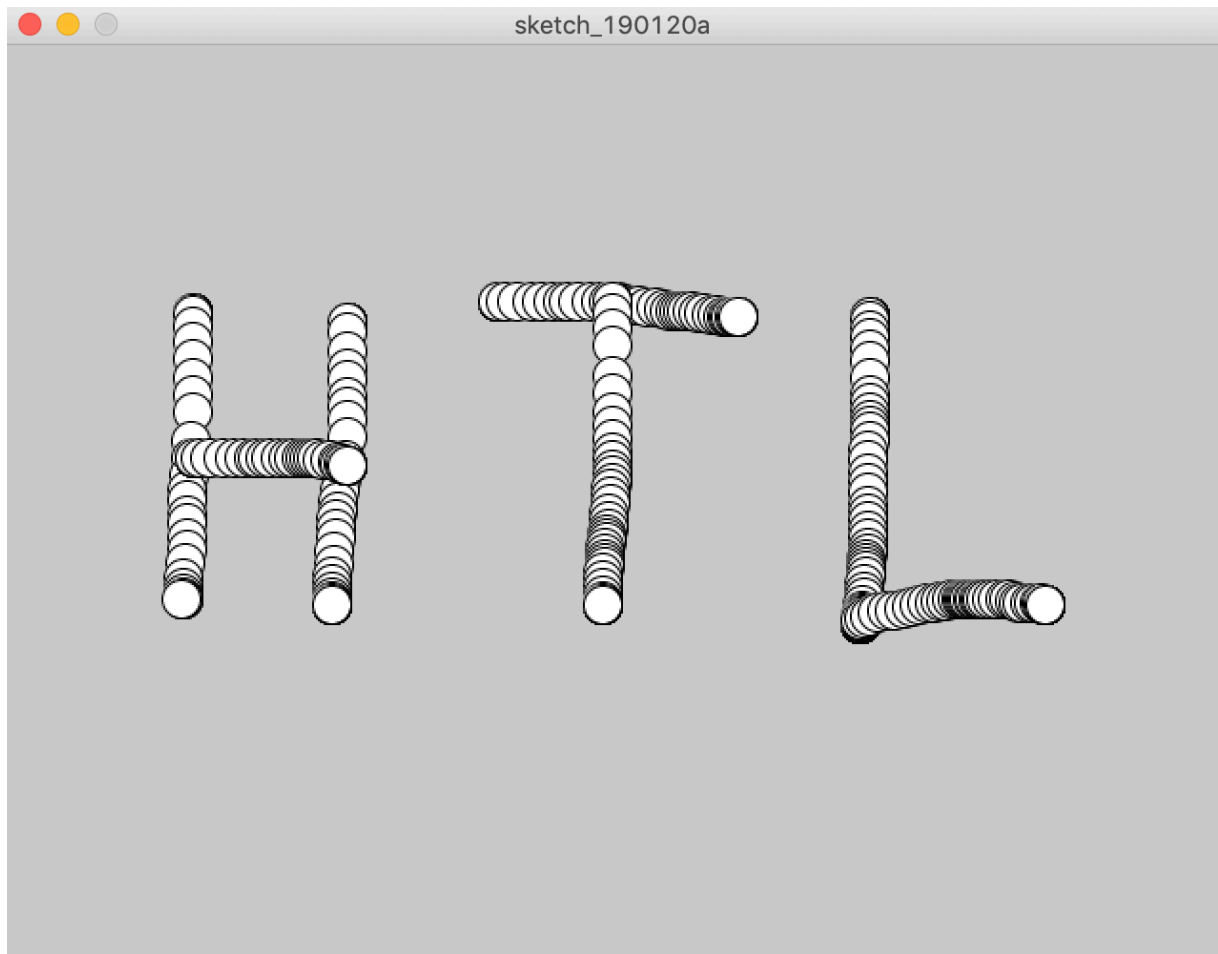
Eigentlich möchten wir, dass nur dann ein Kreis gezeichnet wird, wenn wir eine **Maustaste** drücken. Lass uns dies nun in Dein Programm mit einbauen:

The image shows a screenshot of the Processing IDE interface. At the top, the window title is "sketch_190120a | Processing 3.4". Below the title bar, there are two circular buttons: a play button (green triangle) and a stop button (grey square). The main area is a code editor with a dark background. The code is as follows:

```
1 void draw()  
2 {  
3   if (mousePressed) {  
4     ellipse(mouseX, mouseY, 20, 20);  
5   }  
6 }  
7  
8 void setup() {  
9   size(640, 480);  
10 }
```

A red rectangular box highlights the code block from line 3 to line 5, which contains the conditional logic for drawing the circle when the mouse is pressed.

Starte nun wieder Dein Programm – nun sollten nur noch dann Kreise gezeichnet werden, wenn Du eine Maustaste drückst:

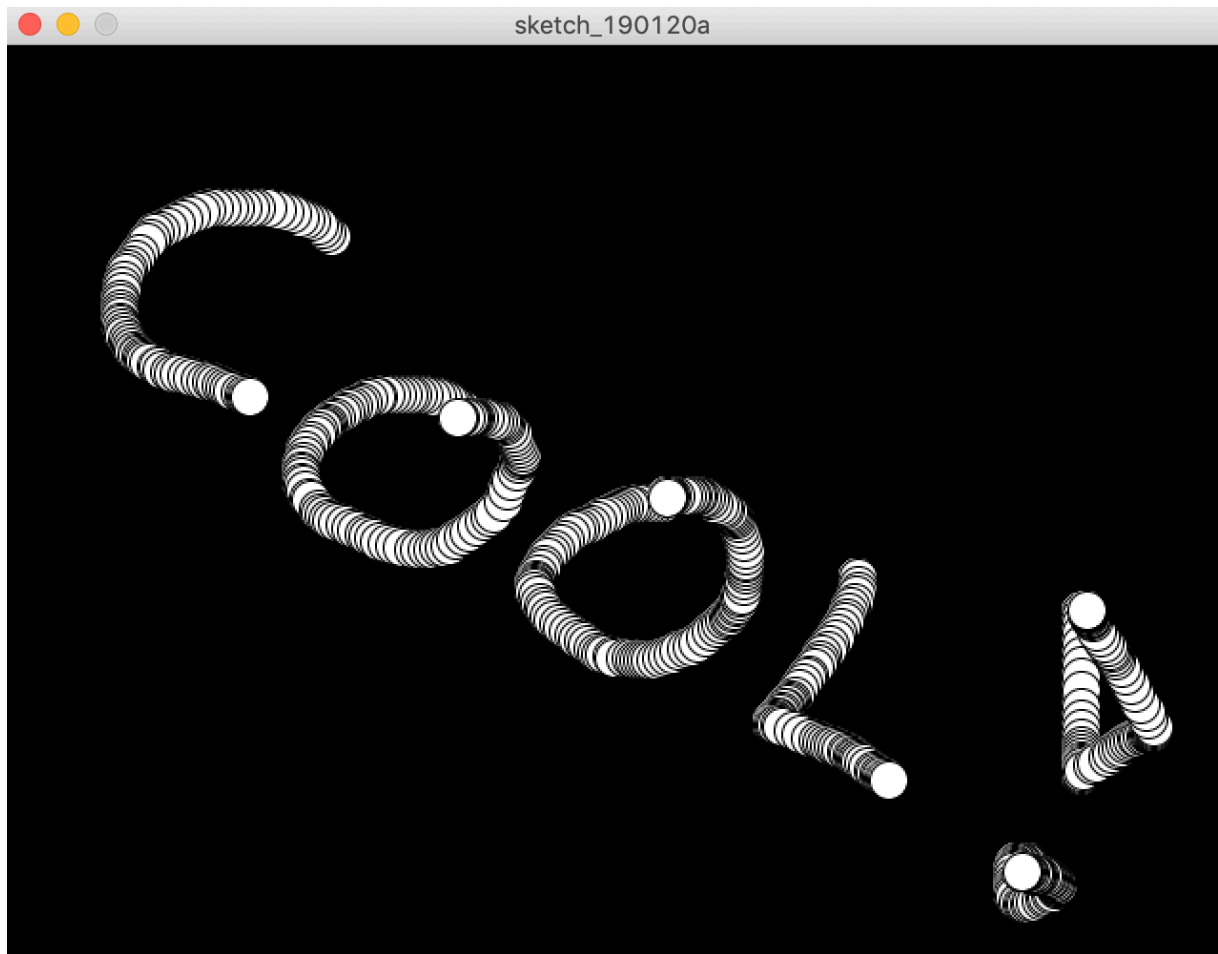


Schritt 9

Nun möchten wir das Programm so erweitern, dass mit der **linken** Maustaste gezeichnet und mit der **rechten** Maustaste das Fenster gelöscht werden kann. Außerdem möchten wir den Hintergrund auf **schwarz** ändern. Gib dazu folgenden Text ein:

```
sketch_190120a ▼
1 void draw()
2 {
3   if (mousePressed) {
4     if (mouseButton == LEFT) {
5       ellipse(mouseX, mouseY, 20, 20);
6     } else if (mouseButton == RIGHT) {
7       background(0);
8     }
9   }
10 }
11
12 void setup() {
13   size(640, 480);
14   background(0);
15 }
```

Nun kannst Du mit der linken Maustaste malen und mit der rechten Maustaste das Fenster löschen:



**Gratuliere, Du hast soeben Dein erstes Zeichenprogramm
in Java geschrieben!**