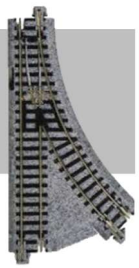
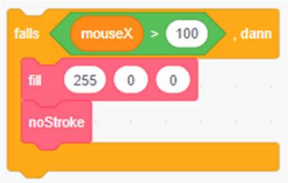
Bedingungen und Verzweigungen



Verzweigungen und Bedingungen kennst du bereits aus Scratch:

Bei einer `if`-Verzweigung („falls … dann“) entscheidet eine Bedingung, ob ein Block ausgeführt oder ausgelassen wird.



wird zu:

```
if (mouseX > 100) {
  fill(255, 0, 0);
  noStroke();
}
```

Bei einer `if-else`-Verzweigung („falls … dann … sonst“) entscheidet eine Bedingung, welcher von zwei Blöcken ausgeführt wird.

wird zu:

```
if (mouseX > 100) {
  fill(255, 0, 0);
  noStroke();
}
else {
  fill(0, 255, 0);
  strokeWeight(3);
}
```

1 Probiere die `if`-Verzweigung und die `if-else`-Verzweigung aus. Probiere verschiedene ⚔ Bedingungen und eine oder mehrere verschiedene 🗐 Anweisungen für die Blöcke aus.

Sketch Kap3_A1a

```
void setup() {
  size(400, 400);
}

void draw() {
  background(255);
  if ( ⚔ ) {
    🗐
  }
}
```

⚔ `mouseX > 200`
⚔ `mouseY < 200`
⚔ `mousePressed`
⚔ `mouseX == 200`
⚔ `keyPressed`
⚔ `key == 'a'`

🗐 `circle(mouseX, mouseY, 10);`
🗐 `square(mouseX, mouseY, 10);`
🗐 `fill(255,0,0);`
🗐 `fill(0,255,0);`
🗐 `noFill();`

Sketch Kap3_A1b

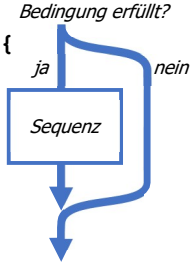
```
void setup() {
  size(400, 400);
}

void draw() {
  background(255);
  if ( ⚔ ) {
    🗐
  }
  else {
    🗐
  }
}
```

Syntax von Verzweigungen und Bedingungen

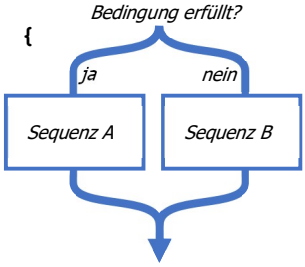
if-Verzweigung:

```
if ( Bedingung ) {
  Sequenz
}
```



if-else-Verzweigung:

```
if ( Bedingung ) {
  Sequenz A
}
else {
  Sequenz B
}
```



Vergleichsoperatoren:

`a > b`	größer
`a < b`	kleiner
`a >= b`	größer gleich
`a <= b`	kleiner gleich
`a != b`	ungleich
`a == b`	gleich

Beachte das doppelte Gleichzeichen!

Einrückungen bei Blöcken

Zur Übersichtlichkeit rückt man alle Anweisungen in einem Block ein.

Der Editor von Processing verwendet dazu zwei Leerzeichen.

Du solltest dich an diese Einrückungen halten, damit dein Programmcode übersichtlich bleibt!

Übersichtlich!

```
void draw() {
  background(255);
  if (mousePressed) {
    fill(255, 0, 0);
  }
  else {
    fill(0, 255, 0);
  }
  circle(mouseX, mouseY, 50);
}
```

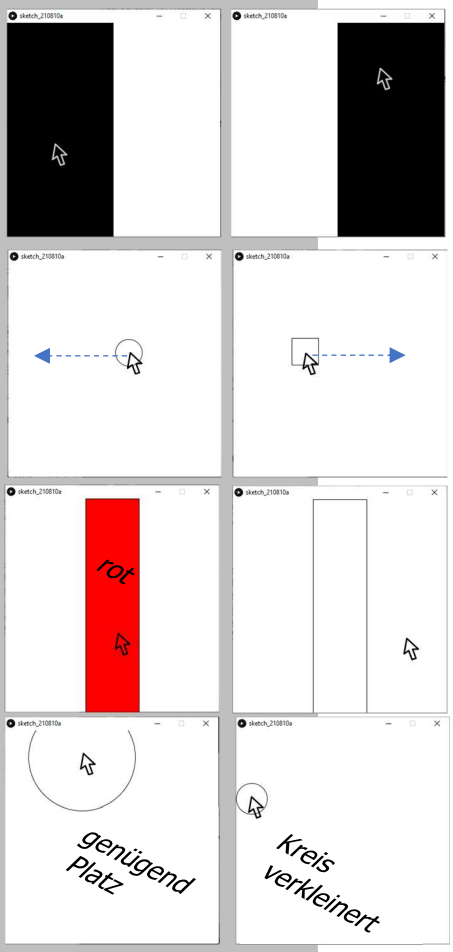
Chaos!

```
  void draw() {
background(255);
if (mousePressed) {
fill(255, 0, 0);
    }else {
    fill(0, 255, 0); }
  circle(mouseX, mouseY, 50);
        }
```

Tipp: Der Editor von Processing kann die Einrückungen automatisch korrigieren:

Menü: Bearbeiten Autoformatierung
Tastenkombination: STRG + T

2 Einfache Verzweigungen: Erstelle jeweils einen passenden Sketch. Verwende eine Fenstergröße von 400x400 Pixel.



a) Das Fenster ist horizontal in der Mitte geteilt. Diejenige Seite, auf der sich der Mauszeiger befindet, wird schwarz eingefärbt, die andere Seite weiß.
Verwende: `rect`, `fill`, `mouseX`
Probiere auch verschiedene Varianten aus: Du kannst auch nur die `rect`*-Anweisungen oder nur die* `fill`*-Anweisungen in den* `if`*-Block schreiben.*

b) Wenn der Mauszeiger nach rechts bewegt wird, dann erscheint ein Kreis um den Mauszeiger. Wird er nach links bewegt, dann erscheint ein Quadrat um den Mauszeiger. Steht er still, wird alles schwarz.
Verwende: `square`, `circle`, `mouseX`, `pmouseX`
Die Position des Mauszeigers aus dem vorherigen Frame wird von Processing als Wert in der Variablen `pmouseX` *gespeichert.*

c) Der Bereich in der Mitte wird rot, wenn sich der Mauszeiger darin befindet.
Tipp: Überlege dir, welche Bedingungen für `mouseX` *gelten müssen. Verwende dann zwei ineinander verschachtelte if-Blöcke. Wir lernen später noch eine geschicktere Möglichkeit kennen.*

d) Zeichne um den Mauszeiger einen Kreis mit Radius 200, falls in x-Richtung genügend Platz vorhanden ist. Wenn der Mauszeiger zu Nahe am linken oder rechten Rand ist, soll der Kreis kleiner werden, dass er den Rand genau berührt.

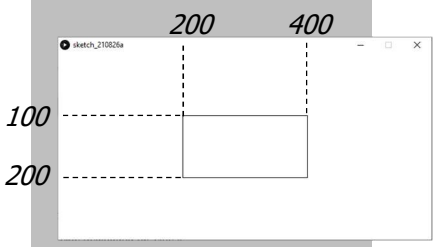
Weitere Teilaufgaben findest du bei den Lösungssketchen.

Logische Operatoren für Bedingungen

Mit den logischen Operatoren UND (AND), ODER (OR), NICHT (NOT) kann man zusammengesetzte Bedingungen formulieren. Bei komplexen Zusammensetzungen können Klammern notwendig sein.

Scratch:			
Java:	A && B	A || B	! A

3 Button
Der Programmcode rechts zeichnet ein Rechteck, das als Button dienen soll.

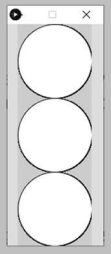


a) Formuliere eine Bedingung für eine `if`-Verzweigung, um zu entscheiden, ob der Mauszeiger sich in dem Rechteck befindet. Färbe in diesem Fall den Button grün ein.

b) Ergänze dein Sketch um eine weitere Verzweigung, dass der Button rot eingefärbt wird, wenn zusätzlich eine Maustaste gedrückt wird.

Sketch Kap3_A3

```
void setup() {
  size(600, 300);
}
void draw() {
  background(255);
  rect(200, 100, 200, 100);
}
```



4 Ampel
Der Programmcode rechts zeichnet drei Kreise übereinander. Ergänze Verzweigungen, dass je nach y-Position der Maus das jeweilige Ampelsegment farbig eingefärbt wird.

Sketch Kap3_A4

```
void setup() {
  size(100, 300);
}
void draw() {
  circle(50, 50, 100);
  circle(50, 150, 100);
  circle(50, 250, 100);
}
```