

Cheatsheet 1 - Grundfiguren

Grundstruktur eines Programmes

```
void setup() {           //Voreinstellungen
  size(500, 500);       //Fenstergröße: Länge=500, Breite=500
}

void draw() {
  //die für das Zeichnen benötigten Code-Teile
}
```

Grundfiguren

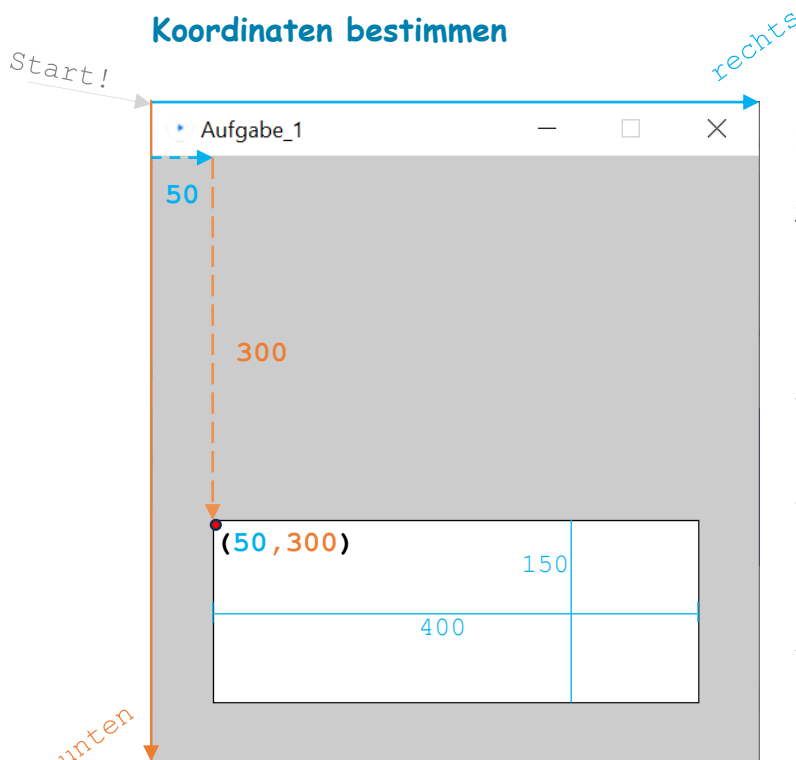
```
rect(x, y, b, h); // Rechteck: linken oberer Eckpunkt (x, y);
                  // Breite=b; Höhe=h

triangle(x1, y1, x2, y2, x3, y3); // Dreieck: Punkt 1=(x1, y1),
                                     Punkt2(x2, y2) und
                                     Punkt3(x3, y3)

circle(x, y, d); // Kreis: Mittelpunkt=(x, y), Durchmesser= d
```

Anweisungen in Programmen werden Zeile für Zeile von oben nach unten ausgeführt. Bereits gezeichnete Formen können so von anderen überdeckt werden.

Koordinaten bestimmen



1. Wir starten immer links oben.
2. Möchten wir den roten Eckpunkt erreichen, gehen wir **50 Schritte nach rechts** und **300 Schritte nach unten**.
3. Somit haben wir den Punkt **(50,300)** erreicht.
4. Das Rechteck hat eine Länge von 400 und eine Höhe von 150.
5. Schreibe die Werte hinein:
`rect(50,300,400,150);`

Farben

Die Werte in den Klammern geben den Rotanteil, den Grünanteil und den Blauanteil an.

```
fill(255, 0, 0); // Füllfarbe: rot
fill(0, 255, 0); // Füllfarbe: grün
fill(0, 0, 255); // Füllfarbe: blau
background(255, 0, 0); // Hintergrundfarbe: rot
```

Vordefinierte Variablen

Eine Variable ist ein Platzhalter, der Informationen enthält, z.B. Zahlen oder Buchstaben. Diese Variablen **existieren** bereits, die kannst du im Programm aufrufen.

`mouseX` (= enthält die aktuelle x-Koordinate)

`mouseY` (= enthält die aktuelle y-Koordinate)

```
void draw() {
  circle(mouseX, 200, 100); // Kreis bewegt sich links und rechts
}
```

Man verwendet statt fixen Zahlen die Variable `mouseX`, um die x-Koordinate laufend zu ändern.

Cheatsheet 2 - Verzweigungen und eigene Variablen

Um Codestellen nur unter bestimmten Bedingungen auszuführen, können wir `if`-Anweisungen nutzen. Wenn wir wollen, dass eine andere Codestelle ausgeführt wird, können wir `else` - Blöcke nutzen.

```
if (Bedingung) {  
  // optionaler Code  
}  
  
else {  
  // optionaler Code  
}
```

Um in Processing Werte miteinander zu vergleichen, werden Vergleichsoperatoren verwendet. Diese haben die gleiche Bedeutung, wie du bereits aus dem Mathematik - Unterricht kennst.

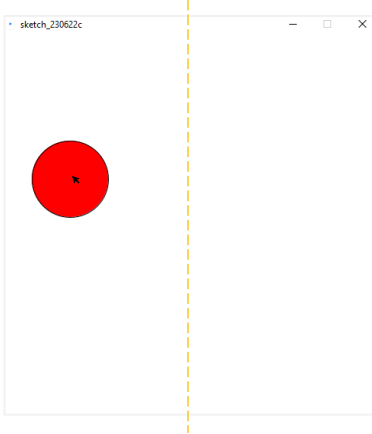
Bezeichnung	Vergleichsoperator	Bezeichnung	Vergleichsoperator
kleiner	<	größer	>
kleiner gleich	<=	größer gleich	>=
gleich	==	ungleich	!=

Beispiel: Wenn sich der Mauszeiger auf der linken Fensterhälfte befindet, dann soll der Kreis rot werden, sonst grün. Die Fenstergröße ist 500 × 500.

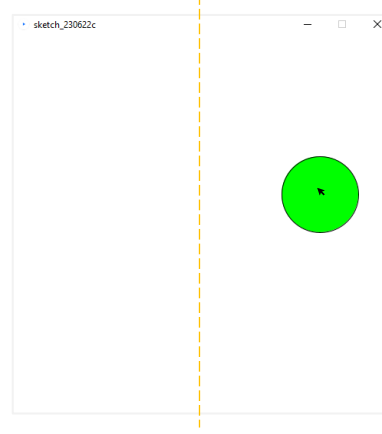
```
void setup() {  
  size(500, 500);           // Fenstergröße: 500 × 500  
}  
  
void draw() {  
  background(255);         // Hintergrund: weiß  
  circle(mouseX, mouseY, 100);  
  
  if (mouseX < 250) {      // Bedingung mit Vergleichsoperator  
    fill(255, 0, 0);       // Fall 1: roter Kreis  
  } else {  
    fill(0, 255, 0);      // Fall 2: grüner Kreis  
  }  
}
```

Hier siehst du eine graphische Veranschaulichung. Ist der Mauszeiger links, wird der Kreis rot, sonst grün.

Fall 1:



Fall 2:



Variablen

Variablen werden verwendet, um mit diesen zu rechnen und um sie an mehreren Stellen im Programm wieder zu verwenden. Es gibt vordefinierte Variablen und eigene Variablen.

Neue vordefinierte Variablen

Diese Variablen **existieren** bereits, die kannst du im Programm aufrufen.

`mousePressed` (= gibt an, ob Mauszeiger gedrückt ist)

`keyPressed` (= gibt an, ob eine beliebige Taste gedrückt ist)

`key` (= Zeichen, welches als letztes gedrückt wurde)

Eigene Variablen

Diese Variablen **existieren nicht**, die musst du selbst erstellen. Du wählst den Datentyp und den Namen deiner Variable. Ein Datentyp gibt den Typ deiner Variable an, z.B. Zahl. Um eine ganzzahlige Variable zu verwenden, verwende `int`.

```
int size; // Hier erstellst du eine Variable namens size
size = 50; // size hat nun den Wert 50
```

Du kannst auch mit Variablen rechnen und sie aufrufen.

```
circle(100, 100, size); // Kreis: Durchmesser=50
int number = size + 20; // number = 70
```