

ITS Teil 4: Processing

Fenstergröße

Das Fenster wird dabei von der oberen linken Seite aus betrachtet, dort liegt der Nullpunkt. Wichtig!! Bei einem 200 px breiten Fenster liegt der angezeigte Wertebereich bei 0 – 199, weil immer mit der 0 angefangen wird.

Zeichenfunktionen

point(x, y) → Zeichnet einen Punkt mit den Koordinaten x und y
line (x1, y1, x2, y2) → zeichnet eine Linie zwischen den Punkten (x1, y1) und (x2, y2)
rect (x, y, Breite, Höhe) → rectMode() steuert die Bedeutung von x, y (CORNER / CENTER)
ellipse (x, y, Breite, Höhe) → ellipseMode steuert die Bedeutung von x, y
size(x, y) → legt die Größe des Fensters fest (in Pixel)

Farben

colorMode() → Wählt einen Farbraum aus (z.B. RGB, HSV) → Über Argumente wird maximaler Wert für Parameter des Systems gesetzt
background () → Setzt die Hintergrundfarbe
fill() → Setzt die Füllfarbe
noFill() → keine Füllung
stroke() / noStroke() → Linienfarbe

Variablen

Jede Variable hat einen Datentyp, welcher sich während der Zeit nicht ändern kann. Jener wird vor der ersten Verwendung deklariert.

Bsp.:

```
int x; // Deklaration → Variable x ist vom Typ int (also eine ganze Zahl)  
x = 10;  
line (x, ...);
```

oder

```
int x = 10;
```

Datentypen

int	Ganzzahl (-2 ³¹ ... 2 ³¹ -1)
float	32-Bit Fließkommazahl
double	64-Bit Fließkommazahl
String	Text (i.d.R. in Anführungszeichen setzen)
boolien	Wahrheitswert (true / false)

Regeln:

- Variablen können nur Werte des gleichen Datentyps zugewiesen werden

Bsp.:

```
int x = 5;  
int a = 10;  
x = a;
```

```
float f = 1.5f;  
int x = a; //NICHT ERLAUBT!!!  
int x = (int)f;
```

- Eine Variable gilt bis zum Ende eines Blocks, in welchem sie definiert wurde (→ Block ist in der Regel eine Abgrenzung bei z.B. einer Schleife durch geschweifte Klammern)

Bsp.:

```
for(...) {  
    boolien b; //Deklaration von lokaler Variable b  
  
    }  
    b=false; //b kann global nicht verwendet werden
```

Operationen

Bsp:

```
int a = 5; //Addition  
int b = 6;  
int summe = a + b;  
summe = summe + 1;  
summe+=1;  
summe++;
```

```
int c = 10; //Modulo-Operation  
int rest = 10%3;
```

Verzweigungen

Bsp:

```
x = ...;  
if (x < 100) {  
    fill(255,0,0);  
}  
else {  
    fill(0,255,0);  
}
```

```
if(x < 100 && y < 100) {  
    ...  
}
```

&& → UND-Verknüpfung

|| → ODER-Verknüpfung

! → logisches NICHT

Schleifen

Aufbau der for-Schleife:

- Initialisierung (z.B. `int i = 10`) → vor erstem Schleifendurchlauf
- Bedingung (z.B. `i <= 10`) → vor jedem Schleifendurchlauf
- Änderung (z.B. `i++`) → nach jedem Schleifendurchlauf

Bsp.:

```
for (int i = 1; i <= 10; i++)    {  
...  
}
```