

# PROCESSING

EINE EINFÜHRUNG IN DIE INFORMATIK

Created by Michael Kirsch & Beat Rossmly

# INHALT

## 1. Stoff der Vorlesung

1. Processing Basics
2. float
3. for
4. else
5. mouse

## 2. Übung

1. Aufgabe 1
2. Aufgabe 2
3. Input

STOFF DER VORLESUNG

# PROCESSING BASICS

Einzeiliger Kommentar

```
// das ist ein einzeiliger Kommentar
```

Mehrzeiliger Kommentar

```
/* das ist  
ein mehrzeiliger  
Kommentar */
```

Inkrementieren

```
x = x+1;  
x += 1;  
x++;
```

Dekrementieren

```
x = x-1;  
x -= 1;  
x--;
```

Fensterbreite

```
w = width;
```

Fensterhöhe

```
h = height;
```

# FLOAT

- **float** zeichnet Variablen aus, die Gleitkommazahlen enthalten.
- Achtung: diese markieren - anders als im deutschsprachigen Raum üblich - ihre Nachkommastellen mit einem Punkt!
- Vorsicht beim Mischen von Datentypen ist geboten!

```
int a = 5;
int b = 2;

float z = a/b;
println(z); // -> 2.0 !!!

float c = 5;
float d = 2;

z = c/d;
println(z); // -> 2.5

int e = 5;
float f = 2;

z = e/f;
println(z); // -> 2.5 !!!

float g = 5;
int h = 2;

z = g/h;
println(z); // -> 2.5 !!!
```

# FOR

- **for** leitet eine Schleife ein.
- In den ( ) wird die Wiederholung des Rumpfes beschrieben. Angefangen bei **n** bis (hier) **5** im angegebenen Intervall (**++**).
- **{ }** enthalten die auszuführenden Befehle.

```
void setup () {  
    size(400,400);  
  
    for (int n=0; n<5; n++) {  
        float x = random(width);  
        float y = random(height);  
        ellipse(x,y,5,5);  
    }  
}
```

# ELSE

**if** bedingt das Ausführen einzelner oder mehrerer Befehle abhängig von einer Aussage.

---

```
if (x == 100) {  
    println("x ist genau 100");  
}
```

Ist die Bedingung von **if** nicht erfüllt, wird **else if** überprüft.

---

```
else if (x == 0) {  
    println("x ist 0");  
}
```

Die Befehle im Rumpf von **else** treten nur ein, sind die Aussagen von **if** und **else if** falsch.

```
else {  
    println("x ist weder 100 noch 0");  
}
```

# MOUSE

Die Variable **mousePressed** spiegelt den Status der linken Maustaste wieder. **true** für gedrückt und **false** für ungedrückt.

```
if (mousePressed) {  
  background(0);  
}
```

---

Ebenso enthalten **mouseX** und **mouseY** die aktuellen Koordinaten des Mauszeigers.

```
ellipse(mouseX, mouseY, 100,100);
```



ÜBUNG

# AUFGABE 1

Ein Quadrat wächst und schrumpft.

```
// deklariere die Variablen: x & y des Quadrats
// deklariere die Variablen: Seitenlänge und Faktor (anfangs 1 später -1)
void setup () {
  // lege Größe fest
  // initialisiere Variablen
}
void draw () {
  // verändere Seitenlänge um Faktor
  // überprüfe ob Seitenlänge zu groß/klein und reagiere
  // berechne Mittelpunkt des Quadrats
  // zeichne Quadrat
}
```

# AUFGABE 2

Finde alle 5 Fehler.

```
int x; v = 10;
void setup () {
    size(800,600);
    x = 50;
}
void draw () {
    x = x+v;
    if (x<0) {
        v = 10;
        x = 50;
    }
    if (x>750) {
v = -10;
x = 700 }
    if (v>0) {
        ellipse(x,300,100,100);
    }
    if (v<0) {
        rect(x,350,100,100);
    }
    background(0);
}
```

# INPUT

Versuche **mouseX** für Farben und Größen einzusetzen. Wie können Werte reguliert werden?

```
void setup () {  
  size(600,400);  
}  
  
void draw () {  
  background(0);  
  // z.B. setze Füllfarbe abhängig von Maus. Wie benutzen wir Werte > 255?  
  // zeichne Objekte  
}
```