

# PROCESSING

WIR ZIEHEN UM!

Created by Michael Kirsch & Beat Rossmly

# INHALT

## 1. Rückblick

1. JAVA?

## 2. Theorie

1. Wir setzen unsere IDE auf
2. Wo bekomme ich diese her?
3. Was ist die richtige JDK?
4. Wir setzen alle IntelliJ auf
5. Wie schreibe ich ein Java Programm?
6. Wie kann ich Output erzeugen?
7. Wo sind Setup und Draw?
8. Wo deklariere ich Variablen?
9. Wo definiere ich Funktionen?
10. Static

## 3. Verknüpfung

RÜCKBLICK

JAVA?

THEORIE

# WIR SETZEN UNSERE IDE AUF

- Um auf einem Computer Software zu entwickeln benötigen wir eine "Integrated Development Environment".
- Diese erlaubt uns Code zu schreiben, diesen zu überprüfen und zu kompilieren.
- Die empfohlene IDE heißt: IntelliJ IDEA
- Jede andere Java IDE ist genauso verwendbar.

# WO BEKOMME ICH DIESE HER?

- [Download](#)
- [Setup Guide](#)

# WAS IST DIE RICHTIGE JDK?

- IntelliJ verwendet von Java mindestens Version 8 und davon das Java Development Kit (JDK)
- Ein JDK ist eine Java Paket, dass über die Ausführbarkeit von Java Applikationen auh das Entwickeln dieser ermöglicht.
- Meist reicht es also nicht aus, dass der eigene Computer bereits Java Code ausführen kann.
- [JDK Setup](#)



WIR SETZEN ALLE INTELLIJ AUF

# WIE SCHREIBE ICH EIN JAVA PROGRAMM?

- Der Startpunkt jedes Java Programms ist die "main" Methode der Hauptklasse.
- Diese hat immer den Rückgabebetyp: void
- Diese ist immer: public
- Diese ist immer: static
- Diese erwartet immer als Übergabewert ein Array vom Typ String

```
public class Main {  
    public static void main (String [] args) {}  
}
```

# WIE KANN ICH OUTPUT ERZEUGEN?

- Ausgaben in der Konsole kann ich über den Befehl "System.out.println();" erhalten.

```
public class Main {  
  
    public static void main (String [] args) {  
        System.out.println("!");  
    }  
  
}
```

# WO SIND SETUP UND DRAW?

- Wenn wir uns erinnern, dass der Startpunkt jedes Processing Sketchs die "setup" Methode ist und der Startpunkt jedes Java Programms die "main" Methode, dann ist "main" vergleichbar mit "setup".
- Wo findet aber "draw" statt?

```
public class Main {  
  
    public static void main (String [] args) {  
        // setup  
    }  
  
}
```

# WO SIND SETUP UND DRAW?

- Die Befehle in "setup" werden einmalig durchlaufen. Wie kann ich eine Wiederholung ähnlich zu "draw" erreichen?

```
public class Main {  
  
    public static void main (String [] args) {  
        // setup  
  
        // draw  
        for (int i=0; i<1000000000; i++) {  
            System.out.println(i);  
        }  
    }  
}
```

# WO DEKLARIERE ICH VARIABLEN?

- Das Deklarieren außerhalb der Klasse funktioniert nicht?
- Das Deklarieren innerhalb der Klasse funktioniert nicht?
- Warum funktioniert nur das Deklarieren innerhalb der "main" Methode?

```
// int i; <- funktioniert hier nicht

public class Main {

    // int i; <- funktioniert hier nicht

    public static void main (String [] args) {
        int i = 0;
    }

}
```

# WO DEFINIERE ICH FUNKTIONEN?

- Das Definieren außerhalb der Klasse funktioniert nicht?
- Warum funktioniert das Definieren innerhalb der Klasse nicht?

```
// void f () {} <- funktioniert hier nicht

public class Main {

    public static void main (String [] args) {
        f();
    }

    void f () {} // <- funktioniert hier nicht
}
```

# STATIC

- Klassen dienen als abstrakte Beschreibungen aber sind selbst erst verwendbar wenn wir ein Objekt davon erzeugt haben.
- "static" zeichnet Elemente der Klasse aus, die auch ohne erzeugtes Objekt verwendet werden können! Diese werden aber von allen Objekten der Klasse geteilt.

```
public class Main {  
  
    static int i;  
  
    public static void main (String [] args) {  
        i = 0;  
        f();  
    }  
  
    static void f () {}  
  
}
```



VERKNÜPFUNG

QUELLEN