

Funktionen in Processing

Funktionen übernehmen Aufgaben und
strukturieren

Wir benutzen das schon!

- Funktionen sind Befehle, die komplexere Aufgaben ausführen.
- Erkennbar beim Aufruf durch die Klammern am Ende
- Bsp:
 - `size(500,500);`
 - `circle(200,300,50);`

Eine Unterscheidung

- Zur Verfügung stehen Befehle und Anfragen
- Befehle führen eine Aktion aus wenn man sie aufruft
- Bei Definition erkennbar am Schlüsselwort **void**
- Anfragen prüfen etwas oder berechnen etwas und **geben das Ergebnis zurück**
- Bei Definition erkennbar an Schlüsselwörtern
 - boolean
 - int
 - float
 - ...

Definition einer Funktion

- Die Definition beginnt mit einem Schlüsselwort (**void** für Befehle oder der Datentyp für Anfragen)
- Funktionen haben immer einen eindeutigen Namen
- Danach wird in einer Klammer definiert ob und welche Werte übergeben werden
- Danach kommt in { } die eigentliche Befehlsfolge

Beispiel: ballSpringt()

```
void ballSpringt() {  
    ballX = random(0,500);  
    ballY = random(0,500);  
}
```

Ablauf eines Aufrufs

```
float ballX;  
float ballY;  
  
void setup() {  
    size(500,500);  
    frameRate(5);  
    ballX = 0;  
    ballY = 0;  
}  
  
void draw() {  
    ballSpringt();  
  
    background(100);  
    circle(ballX, ballY, 50);  
}  
  
void ballSpringt() {  
    ballX = random(0,500);  
    ballY = random(0,500);  
}
```

Funktionen mit Zusatzinformation

- Manche Funktionen benötigen Zusatzinformationen um ihre Aufgabe ausführen zu können
- Beispiele:
 - circle erwartet eine x- und eine y-Koordinate und den Kreisradius
- Diese Zusatzinformationen werden in der Klammer übergeben.