# Unterrichtsstunde Funktionen

## Übersicht:

In dieser Unterrichtsstunde soll den Schülern das Konzept von Funktionen beigebracht werden. Es wird beigebracht, was Funktionen sind, wie Funktionen definiert werden, was Rückgabewerte sind und worfür Funktionen verwendet werden können.

## Lernziel:

Nach der Stunde sollte jede/r eine eigene Funktion schreiben und aufrufen können. Konzepte wie Rückgabewert und Parameter sollten verstanden sein. Zusätzlich sollten Schüler verstehen, was Funktionen für Vorteile bieten um ihren eigenen Code mit funktionen strukturieren zu können.

## Zeitplan:

1. Was sind Funktionen? (5min)

2. Arten von Funktionen (10min)

3. Übungen (35min)

4. Vorteile von Funktionen (10min)

5. Übungen (30min)

## Theorie:

### **1. Was sind Funktionen?**

* Erwähnung bereits bekannter Funktionen:

|  |
| --- |
| void setup() { *// "Definition"* }  ellipse(250, 250, 50, 50); *// "Aufruf"* |

* Funktionen sind **Blöcke von Code**, die **aufgerufen** werden können
* Der Befehl ellipse(...); ist ein Aufruf einer Funktion, die Processing für uns definiert hat. Setup ist eine spezielle Funktion, die von Processing am Anfang des Programms aufgerufen wird.
* Neben den eingebauten Funktionen wie ellipse oder rect können wir auch unsere eigenen Funktionen schreiben und aufrufen!
* eigene Funktionen müssen **in setup oder in draw aufgerufen** werden

### 2. Arten von Funktionen *(functionExamples.pde)*

* Einfache Funktion

|  |
| --- |
| void setup() {  blueRect(); }  void blueRect() {  fill(0, 0, 255);  rect(20, 20, 50, 50); } |

* Funktion mit Parametern

|  |
| --- |
| void setup() {  blueRect(20, 20); }  void blueRect(float x, float y) {  fill(0, 0, 255);  rect(x, y, 50, 50); } |

* Funktion mit Rückgabewert

|  |
| --- |
| void setup() {  int fuenfmalfuenf = quadrat(5);   print(fuenfmalfuenf); }  int quadrat(int x) {  return x \* x; } |

### Wichtig: Funktionen mit Rückgabewert können wie Werte verwendet werden!

* **Typ am Anfang**, **“return” am Ende**
* Können auch mehrere returns haben!
* Zeit für Übungen 1 und 2!

### 3. Vorteile von Funktionen

* **Lesbarkeit**

Codeblöcken können Namen gegeben werden, sodass man sich besser im Programm zurecht findet und schneller erkennt, welcher Programmcode was macht

* **Wiederverwendbarkeit**

Codeblöcke, die man immer wieder an verschiedenen Stellen ausführen will, muss man nicht immer wieder neu schreiben: Man definiert sie in einer Funktion und verwendet diese wieder. Man spart sich Schreibarbeit und muss weniger tippen.

* **Struktur**

Programm wird in kleinere Unterprogramme runtergebrochen. Somit kann man größere Programme bauen: Indem viele kleine Programme zusammenarbeiten.

* Übungen 3 bis 5 + Besprechung

# Übungen Funktionen

## Übung 1:

a) Schreibe eine Funktion namens “kleinerRoterKreis” (ohne Parameter und ohne Rückgabewert) die einen kleinen roten Kreis malt. Rufe die Funktion auf.  
b) Erweitere die Funktion so, dass sie Parameter x und y entgegennimmt, und den kleinen roten Kreis an diesen Koordinaten malt.

## Übung 2:

a) Schreibe eine Funktion “istMausLinks”, die berechnet, ob der Mauszeiger links oder rechts im Sketch ist. Gib einen boolean (true oder false) als Rückgabewert: Wenn die Maus Links ist true, ansonsten false.

b) Nutze die Funktion, um den Hintergrund zu animieren: Wenn die Maus links ist, soll der Hintergrund schwarz sein, wenn die Maus rechts ist, weiß.

Übung 3:

Processing besitzt viele eingebaute Funktionen, die einen Rückgabewert haben. Es gibt z.B. sqrt(), das die Wurzel berechnet, oder sin(), was die Sinus-Funktion ist. Eine davon ist random(): random(int1, int2); gibt einen zufälligen Wert zwischen int1 und int2 zurück.

a) Verwende random(), um einen Sketch zu zeichnen, der eine Form an einer zufälligen Stelle malt!

b) Verwende sin(), um einen Sketch zu zeichnen, in dem eine Form hin und her schwingt!

## Übung 4:

Die Funktion triangle() nimmt drei Punkte triangle(x1, y1, x2, y2, x3, y3); und zeichnet ein Dreieck zwischen den Punkten. Erstelle mithilfe von triangle und rect deine eigene Funktion house(), die ein Haus zeichnet. Fang erst einmal an, eine house-Funktion ohne Parameter zu schreiben, und füge dann nacheinander Parameter hinzu für x, y, Höhe und Breite hinzu.

## Übung 5:

Schau nochmal über deinen alten Code und die Übungen, die du bereits erledigt hast. Gibt es Beispiele, die du mithilfe von Funktionen besser lösen/verbessern kannst?