

GRUNDLAGEN DIGITALER GESTALTUNG 2

2. SEMESTER | SOMMERSEMESTER 2010 | KURS 4

.....
weitere basics der programmierung: eigene funktionen und klassen erstellen
.....

Um mehr die Möglichkeiten des Programmierens aus-
zuschöpfen, und den Code besser zu strukturieren wird
bei diesem Kurs gezeigt, wie man eigene Funktionen und
Klassen anlegen kann.

Eine eigene Funktion anzulegen kann zwei gute Gründe
haben. Zum einen eine sinnvolle Programmereinheit ab-
zugrenzen, zum anderen eine Rechenaufgabe erledigen zu
lassen, und einen Wert zurückzuliefern.

Beispiel 1 (Funktion dient zur Abgrenzung uns Strukturie-
rung)

```
void setup(){  
  size(300,300);  
  smooth();  
}  
void draw(){  
  background(255);  
  ellipseMousePos();  
}  
//Function  
void ellipseMousePos(){  
  ellipse(width/2, height/2, 20,20);  
}
```

Pro Frame wird die Funktion ellipseMousePos() aufgerufen, und
ein Kreis gezeichnet.

GRUNDLAGEN DIGITALER GESTALTUNG 2

2. SEMESTER | SOMMERSEMESTER 2010 | KURS 4

.....

weitere basics der programmierung: eigene funktionen und klassen erstellen

.....

Eine Funktion kann auch einen Wert zurückgeben, wie zum Beispiel ein Boolean. Ein Boolean hat nur die Zustände „true“ oder „false“.

Beispiel 2 (Funktion liefert einen Wert zurück)

```
int mysize = 100;
int xPos;
int yPos;
boolean selected;
void setup(){
  size(300,300);
  xPos = width/2;
  yPos = height/2;
  smooth();
}
void draw(){
  background(0);
  //function wird aufgerufen, Werte werden übergeben
  incircle(mouseX, mouseY);
  if(selected){
    fill(255,0,0);
  }else{
    fill(255);
  }
  ellipse(xPos, yPos, mysize, mysize);
}
```

```
//function
boolean incircle(int _x, int _y){
  float mydist = dist(xPos, yPos, _x, _y);
  if(mydist<mysize/2){
    selected = true;
    return true;
  }else{
    selected = false;
    return false;
  }
}
```

GRUNDLAGEN DIGITALER GESTALTUNG 2

2. SEMESTER | SOMMERSEMESTER 2010 | KURS 4

.....

weitere basics der programmierung: eigene funktionen und klassen erstellen

.....

Hier ist ein Beispiel für eine Funktion, die etwas zeichnet, was aus mehreren Objekten besteht und die Werte nutzt, die wir durch den Aufbau der Funktion definieren. Zum Beispiel die x und y Position, sowie die Größe des Objekts:

```
void setup(){
  size(800, 400);
  background(100);
  smooth();
  noStroke();
}
```

```
void draw(){
  background(100);
  //aufruf der function
  drawFlower(20, 200, 40);
  drawFlower(40, 190, 50);
  drawFlower(100, 80, 120);
  drawFlower(120, 100, 80);
}
```

```
//function
void drawFlower(int xPos, int yPos, float mySize){
  fill(200,255, 200);
  rect(xPos-1, yPos, 2,height-yPos);
  fill(100,220,100,100);
  ellipse(xPos,yPos, mySize, mySize);
  fill(0,255,0);
  ellipse(xPos,yPos, mySize/2, mySize/2);
  fill(255);
  ellipse(xPos,yPos, mySize/4, mySize/4);
}
```

Aufgabe 4.1
entwickle mit Hilfe der for-Schleife und einer Funktion eine Grafik, zum Beispiel ein „Blumenfeld“

GRUNDLAGEN DIGITALER GESTALTUNG 2

2. SEMESTER | SOMMERSEMESTER 2010 | KURS 4

.....
weitere basics der programmierung: eigene funktionen und klassen erstellen
.....

Klassen in der Programmierung zu erstellen ist sehr hilfreich und erschließt weitere Möglichkeiten, Ideen einfach umzusetzen.

Um eine Klasse zu erklären könnte man zum Beispiel in die Natur schauen. Ein Baum hat tausende Blätter. Sie sind von dem Aufbau und der Struktur gleich: Sie sind grün haben einen Stil und eine Form. Wenn man sie genauer betrachtet sind sie zwar alle unterschiedlich und individuell, doch vom Aufbau her folgen sie alle einem Bauplan.

Genauso verhält es sich mit einer Klasse innerhalb der Programmierung, man definiert die grundlegenden Eigenschaften und kann unzählige Objekte erstellen, die dieser Klasse entsprechen.

Beispiel nächste Seite:

GRUNDLAGEN DIGITALER GESTALTUNG 2

2. SEMESTER | SOMMERSEMESTER 2010 | KURS 4

.....
 weitere basics der programmierung: eigene funktionen und klassen erstellen

Beispiel Code in dem eine Klasse verwendet wird:

```

Flower f1,f2,f3, f4; //die Klasse heißt Flower und
//drei Objekte mit dem Namen f1, f2, f3 werden erstellt
void setup(){
  noStroke();
  smooth();
  size(800,400);
  //in der Klammer werden alle Argumente angegeben
  //die gebraucht werden
  f1 = new Flower(20, 200, 40);
  f2 = new Flower(40, 190, 50);
  f3 = new Flower(100, 80, 120);
  f4 = new Flower(120, 100, 80);
}
void draw(){
  background(100);
  f1.display();
  f2.display();
  f3.display();
  f4.display();
}

class Flower{
  //innerhalb der Klasse verhält es sich ähnlich, als würde
  ein

```

```

//neues Programm begonnen
int xPos;
int yPos;
float mySize;
//in der folgenden Zeile werden alle Werte angenommen,
//die angegebenen wurden
//und sie werden mit den Variablen innerhalb der Klasse
//gleichgesetzt.
Flower(int thexPos, int theyPos, float theSize){
  xPos = thexPos;
  yPos = theyPos;
  mySize = theSize;
}

//eine Funktion die in der Hauptklasse aufgerufen wird
void display(){
  fill(200,255, 200);
  rect(xPos-1, yPos, 2,height-yPos);
  fill(100,220,100,100);
  ellipse(xPos,yPos, mySize, mySize);
  fill(0,255,0);
  ellipse(xPos,yPos, mySize/2, mySize/2);
  fill(255);
  ellipse(xPos,yPos, mySize/4, mySize/4);
}
}

```

GRUNDLAGEN DIGITALER GESTALTUNG 2

2. SEMESTER | SOMMERSEMESTER 2010 | KURS 4

.....
Aufgaben bis zum 10.5
.....

Aufgabe 4.1

**entwickle mit Hilfe der for-Schleife und einer Funktion
eine Grafik, zum Beispiel ein „Blumenfeld“**

Aufgabe 4.2

**entwickle eine Klasse, und zeichne mehrere Objekte,
die dieser Klasse entstammen**