

# Processing

## Anleitung

### Wichtige Funktionen

<b>setup()</b>	<p>Die Funktion setup() wird zu Beginn des Programms aufgerufen. Sie eignet sich gut für Einstellungen:</p> <hr/> <pre>function setup() {   createCanvas(640, 360); // setze Größe des Fensters }</pre>
<b>draw()</b>	<p>In jeder Sekunde wird 60 mal gemacht, was wir in der draw() Funktion schreiben. Häufig entscheiden wir hier, was auf dem Bildschirm zu sehen ist:</p> <hr/> <pre>function draw() {   background(51); // Hintergrund dunkel (0=schwarz 255=weiß)   fill(255, 0, 0); // Alles soll rot werden   rect(20, 20, 20, 20); // zeichne kleines Viereck }</pre>

### Hilfreiche Funktionen

<b>print()</b>	<p>Damit kann man sich Variablen anzeigen lassen.</p> <hr/> <pre>function draw() {   print(mouseX); }</pre>
<b>createCanvas(width, height)</b>	<p>Setzt die Größe des Fensters. "width" ist die Breite und "height" die Höhe.</p> <hr/> <pre>function setup() {   createCanvas(800, 600); }</pre>
<b>frameRate()</b>	<p>Setzt, wie oft in der Sekunde die draw() Funktion ausgeführt wird.</p> <hr/> <pre>function setup() {   frameRate(60);   println(frameRate); }</pre>

## Zeichnen

<b>rect(x, y, width, height)</b>	Zeichnet ein Rechteck in das Fenster ein. x und y bestimmen die Position und width und height die Größe des Vierecks.
<b>circle(x, y, radius)</b>	Zeichnet einen Kreis in das Fenster ein. x und y bestimmen die Position und der Radius die Größe des Kreises.
<b>color(r, g, b)</b>	<p>Mit der Funktion color() kann man eine neue Farbe erstellen. Die Farbe besteht aus den drei Anteilen <b>Rot</b>, <b>Grün</b> und <b>Blau</b>. 0 ist der kleinste Wert und 255 der größte.</p> <hr/> <pre>color rot = color(255, 0, 0); void draw() {   fill(rot); // alles wird Rot gezeichnet }</pre>
<b>fill(color)</b> <b>stroke(color)</b>	<p>Diese Funktionen setzen die Füllfarbe sowie die Randfarbe von Formen (zum Beispiel rect())</p> <hr/> <pre>void draw() {   fill(rot); // setze Füllfarbe auf Rot   stroke(blau); // setze Randfarbe auf Blau   rect(10, 10, 20, 20); // Zeichne Viereck }</pre>
<b>background(color)</b>	<p>Diese Funktion bestimmt die Hintergrundfarbe.</p> <hr/> <pre>void draw() {   background(51); // alles wird dunkel grau }</pre>
<b>strokeWeight()</b>	<p>Damit kann eingestellt werden, wie dick die Linien sind, die gezeichnet werden sollen.</p> <hr/> <pre>void setup() {   strokeWeight(0); // keine Linien   strokeWeight(1); // dünne Linien   strokeWeight(8); // dicke Linien }</pre>

## Interaktion mit dem Benutzer

<p><b>mouseX</b> <b>mouseY</b></p>	<p>Die Variablen mouseX und mouseY beinhalten die Position der Maus.</p> <hr/> <pre>void draw() {   fill(255, 0, 0);   rect(mouseX, mouseY, 10, 10); }</pre> <hr/> <p>Dieser Code zeichnet ein Viereck dorthin, wo sich die Maus befindet.</p>
<p><b>keyPressed()</b> <b>keyReleased()</b></p>	<p>Mit der Funktion keyPressed() kann man abfragen, ob eine Taste gedrückt wurde. Genauso kann mit der Funktion keyReleased() herausgefunden werden, ob eine Taste losgelassen wurde.</p> <hr/> <pre>function keyPressed() {   print(key);   if (key == 'a') {     print("Die Taste a wurde gedrückt");   } }  function keyReleased() {   print(key);   if (key == 'a') {     print("Die Taste a wurde losgelassen");   } }</pre>

## Programmieren

<b>Variablen</b>	<p>Manchmal muss man Informationen über längere Zeit speichern. Dafür können Variablen genutzt werden:</p> <hr/> <pre>let rot = color(255, 0, 0); let position = 0;  function draw() {   fill(rot);   rect(position, position, 10, 10);   position += 5; // Position wird verändert }</pre> <hr/> <p>Um eine Variable zu erstellen, muss man das Wort "let" verwenden. Die Position speichert beispielsweise die aktuelle Position des Vierecks, die Variable rot speichert die Farbe rot.</p>
<b>Verzweigungen</b>	<p>Verzweigungen können verwendet werden, um nur unter bestimmten Bedingungen etwas zu tun.</p> <hr/> <pre>let rot = color(255, 0, 0); let position = 0;  function draw() {   fill(rot);   rect(position, position, 10, 10);   position += 5; // Position wird verändert   if (position &gt; 100) { // &lt;- Verzweigung     position = 0;   } }</pre> <hr/> <p>In diesem Beispiel wird die Position auf 0 zurückgesetzt, wenn sie größer als 100 ist.</p>
<b>random(start, end)</b>	<p>Erzeugt eine zufällige Zahl zwischen start und end.</p> <hr/> <pre>void draw() {   let random_color = color(     random(0, 255),     random(0, 255),     random(0, 255)   );   background(random_color); }</pre>