

Drawbot Cheatsheet

Kurzbefehle

`cmd r` Programm ausführen

`cmd alt 5` Zeile ausrücken, `cmd alt 6` Zeile einrücken

`cmd alt 8` Zeile aktivieren, `cmd alt 9` Zeile deaktivieren («auskommentieren»)

`cmd Pfeiltaste` Wert ausgewählter Zahl ändern

Kommentare

`# Das ist ein Kommentar`

```
""  
Und das ist ein mehrzeiliger Kommentar.  
Wenn es mal etwas mehr sein soll.  
""
```

Variablen

`foo = 12` Wertzuweisung

`foo = foo + 1` Wert in Variable um 1 erhöhen

`foo += 1` Wert in Variable um 1 erhöhen (Kurzschreibweise)

`foo = foo - 1` Wert in Variable um 1 reduzieren

`foo -= 1` Wert in Variable um 1 reduzieren (Kurzschreibweise)

Datentypen

`3` Integer (Ganze Zahlen)

`3.5` Float (Fließkommazahl)

`'Anna'` String (Zeichenkette)

`True` und `False` Boolean (Wahr oder Falsch)

`['Anna', 'Paul']` List (Beliebige lange Liste)

Rechnen

`+` Addition

`-` Subtraktion

`*` Multiplikation

`/` Division, mit Resultat mit Kommastellen

`//` Division, mit ganzer Zahl als Resultat (gerundet)

`%` Modulo (Rest der Division)

Rechnen mit Strings

`'la' + 'la'` Output: lala

`3 * 'la'` Output: lalala

`3 * 'la' + 'li'` Output: lalalali

Vergleichsoperatoren

`a == b` a ist gleich b

`a != b` a ist ungleich b

`a > b` a ist grösser als b

`a >= b` a ist grösser gleich b

`a < b` a ist kleiner als b

`a <= b` a ist kleiner gleich als b

Vergleichsoperatoren eignen sich zur Formulierung von Bedingungen.

```
a = 2
b = 3
if a < b:
    print(a, 'ist kleiner als', b)
```

Nützliche Python-Funktionen

`print()` Output in Konsolen-Fenster schreiben

`randint(A, B)` zufällige ganze Zahl zwischen A und B

`random()` zufällige Zahl zwischen 0 und 1

`range(7)` generiert die Liste `[0, 1, 2, 3, 4, 5, 6]`

`range(4, 7)` generiert die Liste `[4, 5, 6]`

`range(2, 7, 2)` generiert die Liste `[2, 4, 6]`

Die wichtigsten Drawbot-Funktionen

`newPage(w, h)` generiert eine neue Seite in Print-Dokumenten, bzw. ein neues Frame in Animationen. Angaben in Zahlen ergeben Punkte für PDF, bzw. Pixel für GIF, PNG etc. Möglich sind auch Angaben wie `"A4"` oder `"A0Landscape"`, siehe [Drawbot-Doku](#).

`width()` liefert die Breite der Hintergrundfläche

`height()` liefert die Höhe der Hintergrundfläche

`oval(x, y, w, h)` zeichnet ein Oval, bzw. einen Kreis, wenn $w = h$

`rect(x, y, w, h)` zeichnet ein Rechteck

`line((x1, y1), (x2, y2))` zeichnet eine Linie. Die Funktion `line()` braucht als Argumente **zwei Klammern** mit den Positionen zweier Punkte.

`font("Times-Italic", 200)` stellt Font und Schriftgrösse ein. Als Name muss der sog. «Postscript-Name» des Fonts angegeben werden. Das ist meist der Dateiname ohne Endung, kann aber auch in der Schriftsammlung unter den Infos zum Font nachgeschaut werden.

`fill(0)` stellt die Füllfarbe ein. Werte von 0–1.

- Ein einziger Wert bezieht sich auf **Graustufen**
- zwei Werte auf **Graustufen und Transparenz**
- drei Werte auf **RGB**
- vier Werte auf **RGB und Transparenz**.
- zum Umrechnen 255-Stufige RGB-Werte **durch 255 teilen**
- `None` ist auch ein gültiger Wert.

`stroke()` stellt Linienfarbe ein. Wenn nicht angegeben, auf `stroke(None)` eingestellt.

`strokeWidth()` Die Strichstärke in Punkt.

Schleifen

Die for-Schleife durchläuft eine vorgegebene Liste. Mittels der Iterationsvariable kann auf die einzelnen Elemente zugegriffen werden.

```
for text in ["lorem", "ipsum"]:  
    print(text)
```

Output: lorem ipsum

```
for i in range(5):  
    print(i)
```

Output: 0 1 2 3 4

Bedingungen

Die if-Anweisung wird benutzt, um eine Bedingung zu überprüfen. Falls diese Bedingung erfüllt ist, wird ein Block von Anweisungen ausgeführt (der so genannte if-Block), andernfalls wird ein anderer Block von Anweisungen ausgeführt (elif oder else)

Allgemeine Form:

```
if bedingung1:  
    anweisungen1  
elif bedingung2:  
    anweisungen2  
else:  
    anweisungen3
```

Beispiel:

```
if random() > 0.66:  
    print("Schere")
```

```
elif random() > 0.33:  
    print("Stein")  
else:  
    print("Papier")
```

Reservierte Wörter

... die nicht als Variablen verwendet werden dürfen:

```
and, del, from, not, while, as, elif, global, or, with, assert, else,  
if, pass, yield, break, except, import, print, class, exec, in,  
raise, continue, finally, is, return, def, for, lambda, try
```