

Laden einer Bibliothek

Importanweisung	Befehlsaufruf
<code>import modulname</code>	<code>modulname.befehl()</code>
<code>import modul as short</code>	<code>short.befehl()</code>
<code>from modul import *</code>	<code>befehl()</code>

`dir(modulname)` zeigt alle Befehle im Modul an
`help(befehl)` zeigt die Hilfe für den Befehl an

Module

<code>random</code>	Zufallszahlen
<code>math</code>	Mathematik
<code>turtle</code>	Turtle-Graphik
<code>os</code>	Betriebssystem
<code>sys</code>	Console

Zufallszahlen

<code>randint(a,b)</code>	Zufallszahl $a \leq x \leq b$
<code>randrange(a,b)</code>	Zufallszahl $a < x < b$
<code>randrange(b)</code>	Zufallszahl $0 < x < b$
<code>random()</code>	Zufallszahl 0.0 - 1.0

Turtlegraphik

<code>fd(s)</code>	Vorwärts s Einheiten
<code>bk(s)</code>	Rückwärts s Einheiten
<code>rt(a)</code>	Drehen nach Rechts
<code>lt(a)</code>	Drehen nach Links
<code>pu()</code>	nicht Zeichnen
<code>pd()</code>	ab jetzt Zeichnen
<code>circle(r)</code>	Kreis mit Radius r
<code>fillcolor(f)</code>	Füllfarbe setzen
<code>begin_fill()</code>	ab jetzt Ausfüllen
<code>end_fill()</code>	Ausfüllen beenden
<code>home()</code>	zum Startpunkt
<code>reset()</code>	neu Anfangen

s: Anzahl Pixel
a: Winkel in Grad
f: Farbe z.Bsp. 'red'

Ein- und Ausgabe

```
var = int(input('prompt'))
var = float(input('prompt'))
print('prompt')
print('text %s text' %(var))
print('text {} text'.format(var))
%s Platzhalter Textvariable, %d Platzhalter Zahlvariable
```

Operatoren

<code>x+y</code>	Addition	<code>x-y</code>	Subtraktion
<code>x*y</code>	Multiplikation	<code>x/y</code>	Division
<code>x%y</code>	Modulo	<code>x**y</code>	x^y
<code>x//y</code>	Division ohne Rest		

Datentypen

Integer	-25, 23
Float	-2.34, 65.3
String	'Hello', "World", ""multiline""
Boolean	True, False
List	[value, ...]
Tupel	(value, ...) ¹
Dictionary	{key:value,...}
Set	{value, value,...} ²

¹ Klammern optional
² `set()` erzeugt eine leere Menge

Funktionen

```
def funktionsname(Var1, Var2=4):
    #Anweisungen
    #Anweisungen
    return result #optional
```

Selektionen

```
if bedingung:
    #Anweisungen, falls bedingung erfüllt ist
elif bedingung2:
    #Anweisungen
else:
    #Anweisungen
```



