

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>
<code>dir(modulname)</code> zeigt alle Befehle im Modul an	
<code>help(befehl)</code> 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>randomint(a,b)</code>	Zufallszahl $a \leq x \leq b$
<code>randrange(a,b)</code>	Zufallszahl $a < x < b$
<code>randrange(b)</code>	Zufallszahl $0 \leq x < b$
<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):
    #Anweisung
    #Anweisung
    return result #optional
```

Selektionen

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



Bedingungen

<	kleiner als	a < 10
>	grösser als	b>4
==	gleich	c=='yes'
<=	kleiner gleich	d<=5
>=	grösser gleich	e<=7
!=	ungleich	g!='no'
'in'	in	'x' in 'mexico'
'not in'	nicht in	y not in 'mexico'

Zeichenketten (Strings)

str.lower()	in Kleinbuchstaben umwandeln
str.upper()	in Grossbuchstaben umwandeln
str.replace(old,new)	old durch new ersetzen
str.split()	Teilt den String auf
str[1:5]	Zeichen 1-5 anzeigen
list(str)	erzeugt eine Buchstabenliste

Strings Slicing

0	1	2	3	4	5	6	7	8	9	10	11
M	o	n	t	y		P	y	t	h	o	n
-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

[6:10] (highlighted)

[-12:-7] (highlighted)

Iterationen

for-Schleifen

```
for item in list:
    #An wei sungen für item
    #An wei sungen für item`
#Anwei sungen nach der Schleife
for i in range(n):
    #An wei sungen n mal Wieder holen
```

while Schleife

```
while bedingung:
    #An wei sungen
```

range(n) = [0,1,2 ,3,..., n-1] Liste mit den ersten n Zahlen
 break beendet die Schleife. continue beendet den aktuellen Durchlauf

Arbeiten mit Listen

len(my List)	Länge von myList
myList[i]	i-tes Element der Liste
myList [i:j]	Ausschnitt von i bis j
x in myList	True wenn x in myList ist
myList.append(x)	x myList anhängen
myList.insert(i,x)	x vor der Stelle i einfügen
myList [i]=x	Element i ersetzen
myList.remove(x)	entfernt x aus myList
myList.pop([i])	entfernt das i-te Element
myList=[]	

Dictionaries

len(dict)	Länge von dict
del dict[key]	löscht den Schlüssel key
dict.keys()	Liste von Schlüsseln
key in dict	Wahr wenn es den Schlüssel gibt
dict = {key: value, }	

Exception Handling

```
try:
    #An wei sungen
except exception type as var:
    #An wei sungen
finally:
    #An wei sungen
```

Dateien

open(file ,mode)	Datei öffnen
f.read()	liest den ganzen File
f.readline()	liest eine Zeile
f.readlines()	liest alle Zeilen
for line in f:	Zeile für Zeile durchgehen
f.write(prompt)	schreibt in die Datei
f.close()	schliesst die Datei
mode: 'r' lesen, 'w' schreiben, 'r+' lesen und schreiben, 'a' anhängen	
readlines()	erzeugt eine Liste von Zeilen

