

Python Cheat-Sheet

# Kommentar	Der Hashtag leitet eine Kommentarzeile ein. Diese Zeile wird nicht ausgeführt!
:	Leitet einen Anweisungsblock ein: Alle Zeilen, die danach eingerückt sind, gehören zum Anweisungsblock, z.B. zu der Funktion oder Prozedur.
<pre>a = 42 print(a) # erhoeht a um 2 a = a+2 print(a)</pre>	Variablen legt man einfach fest, indem man ihnen einen Wert zuweist. Variablenamen sind beliebig, müssen aber mit einem Buchstaben anfangen. Bei Zuweisungen ist immer der Variablenname links und der (neue) Wert rechts.
<pre>a = 42 print("DIE Antwort", a)</pre>	Schlüsselwort für die Ausgabe an die Konsole . Was ausgegeben wird, steht in der Klammer. Mehrere Ausgaben können durch Komma getrennt werden.
<h3>Funktionen / Prozeduren</h3>	
<pre>def addieren(a,b): ergebnis = a+b return ergebnis</pre>	Schlüsselwort, um eine Funktion oder Prozedur zu definieren: - Eine Funktion <u>berechnet</u> etwas; das Ergebnis wird mit <code>return</code> zurückgegeben. - Eine Prozedur <u>tut</u> etwas (z.B. verändert Variablen). Es gibt kein <code>return</code> . - Prozeduren und Funktionen können Parameter (einer oder mehrere) mitgegeben werden. Die Parameter stehen in Klammern hinter dem Namen der Prozedur bzw. Funktion. Innerhalb der Prozedur bzw. Funktion kann mit dem Parameter gearbeitet werden. Funktionen/Prozeduren ohne Parameter: leere Klammern.
<pre>ergebnis = addieren(3,5) print(ergebnis)</pre>	Funktionen und Prozeduren kann man wie rechts aufrufen . Man muss die richtige Zahl von Parametern angeben. <u>Bei Funktionen muss man sich für den Rückgabewert interessieren!</u> Der Rückgabewert wird hier in der Variable <code>ergebnis</code> gespeichert.
<h3>Bedingungen</h3>	
<pre>x = 3 y = 5 if x == y: print("gleich!") else: print("ungleich!")</pre>	Bedingung . Wenn die Bedingung erfüllt ist, dann wird das ausgeführt, was in dem Anweisungsblock nach dem Doppelpunkt steht. else bedeutet "sonst".

Schleifen	
<pre>for i in range(1,10): zahl = i*2 print(zahl)</pre>	<p>for ist das Schlüsselwort für eine Zähl-Schleife: Der folgende Anweisungsblock wird wiederholt ausgeführt. Die Zähl-Schleife hat eine Zähl-Variable, in diesem Fall i. Diese wird bei jedem Durchlauf eins hochgezählt. Mit in range(...) gibt man den Bereich an. Die erste Zahl ist einschließlich, <u>die zweite Zahl ist ausschließlich!</u></p>
<pre>zahl = 2 while zahl < 1000: # Zahl verdoppeln zahl = zahl * 2 print(zahl)</pre>	<p>while ist das Schlüsselwort für eine bedingte Schleife: Die Schleife wird so lange ausgeführt, wie die Bedingung erfüllt ist. <u>In der Schleife muss man dafür sorgen, dass die Bedingung irgendwann mal erfüllt werden kann!</u> Sonst hat man eine sogenannte <i>Endlosschleife</i>.</p>
Listen	
<pre>liste = [2, 5, -3, 8] print(liste)</pre>	<p>Listen werden mit eckigen Klammern definiert.</p>
<pre># index 0 1 2 3 liste = [2, 5, -3, 8] zahl = liste[2] print(zahl) # gibt -3 aus</pre>	<p>Mit <code>liste[2]</code> kann man in <code>liste</code> das Element am Index 2 (d.h. an der Position 2) ansprechen (und für andere Indices entsprechend). <u>Der Index startet immer bei 0.</u></p>
<pre>liste = [2, 5, -3, 8] a = max(liste) b = min(liste) c = len(liste) print(a,b,c) # gibt 8 -3 4 aus</pre>	<p>Es gibt vorgefertigte Funktionen für Liste, die Maximum Minimum Länge ermitteln.</p>
<pre>liste = [2, 5, -3, 8] a = 7 if a in liste: print(a, "dabei") else: print(a, "nicht dabei")</pre>	<p>Mit in kann man überprüfen, ob ein Wert in einer Liste enthalten ist.</p>
<pre>liste = [2, 5, -3, 5, 8] a = 7 liste.append(a) b = 5 liste.remove(b) print(liste)</pre>	<p>Mit <code>append</code> und <code>remove</code> kann man Elemente zur Liste hinzufügen bzw. löschen. <code>append</code> fügt hinten hinzu. <code>remove</code> löscht <u>nur das erste Vorkommen</u> des Elements! Aufgerufen werden die Funktionen mit Listenname Punkt <code>append</code> (bzw. <code>remove</code>).</p>
<pre># index 0 1 2 3 liste = [2, 5, -3, 8] position = liste.index(-3) print(position)</pre>	<p>Mit dem Schlüsselwort <code>index</code> kann man den Index (= die Position) eines Elementes in der Liste herausfinden. <u>Der Index wird immer von 0 an gezählt.</u></p>
<pre># index 0 1 2 3 liste = [2, 5, -3, 8] liste.insert(2,42) print(liste) # [2, 5, 42, -3, 8]</pre>	<p>Mit <code>insert</code> kann man an einer gewünschten Position ein Element in eine Liste einfügen. Als Parameter wird erst der Index angegeben und dann der Wert, der eingefügt werden soll.</p>