

<b>Addieren und Subtrahieren</b>	2 + 3 9 - 8	Mit + und — kann man addieren oder subtrahieren.
<b>Multiplizieren und Dividieren</b>	3 * 5 5 / 3	Mit * und / kann man multiplizieren oder dividieren.
<b>Variablen erstellen</b>	a = 24 begrueßung = "Wilkommen!" b = a + 10	Mit dem = Operator kann man Variablen erstellen und ihnen einen Wert geben.
<b>Vergleichsoperatoren</b>	> (groesser als), < (kleiner als), >= (groesser gleich), <= (kleiner gleich), == (gleich), != (ungleich)	Vergleichsoperatoren zum Benutzen in if-Anweisungen oder while-Schleifen.
<b>if-Anweisungen</b>	if a == b: print("a ist gleich b") else: print("a ist ungleich b")	Anweisungen werden bei dem Erfüllen von Bedingungen ausgeführt.
<b>Funktionen aufrufen</b>	print("Hallo") meinefunktion(5, "hallo", True)	Aufrufen von Funktionen. Argumente werden mit Komma getrennt.
<b>Liste</b>	liste1 = ["Emelent 1", 2, "Element 3"] liste1[0] liste1[0] = "Element 1"	Erstellen einer Liste, Abrufen und Verändern eines Elements (Index fängt bei 0 an).
<b>Tuple</b>	tuple1 = (1, "Hallo", a)	So wie eine Liste aber unveränderbar.
<b>Dictionary</b>	dict1 = { "foo":bar, x:bar2, 1234:"hallo" } dict1["foo"]	Eine Liste, in der Einträge mit einem "Key" aufgerufen werden, anstatt mit einem Index.
<b>while-Schleife</b>	while True: [...] while a < b: [...]	Schleife, die läuft <b>während</b> eine Bedingung wahr ist.
<b>for-Schleife</b>	for x in liste1: print x for x in range(0, 10): [...]	Schleifen die <b>für</b> jedes Element in einer Liste ausgeführt werden. Bei range() wird die letzte Zahl nicht ausgegeben (z.B. hier 0, 1, 2 .. 8, 9)
<b>Importieren von Modulen</b>	import random from time import sleep	Mit import [...] oder from [...] import [...] kann man Module importieren um zusätzliche Funktionen zu kriegen.
<b>Generieren von Zufallszahlen</b>	import random zahl1 = random.randint(0, 5) print zahl1	Mit random.randint() kann man zufällige Ganzzahlen generieren, inklusive der ersten und der letzten (also z.B. hier kann alles von 0 bis 5 herauskommen).
<b>Warten</b>	from time import sleep time.sleep(5)	Um eine Pause im Programm zu machen kann man time.sleep() benutzen. Das Argument ist in Sekunden angegeben.