

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