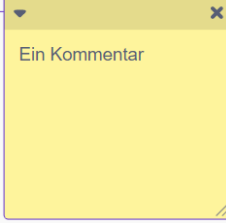
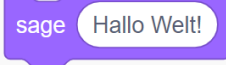


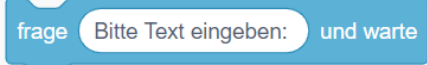
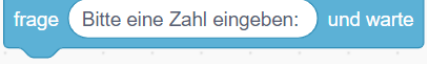
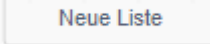
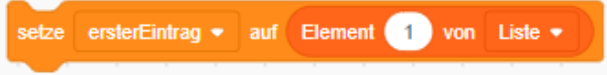

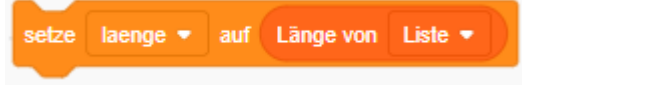
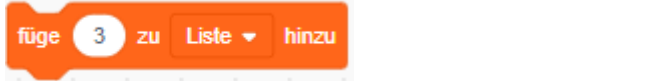

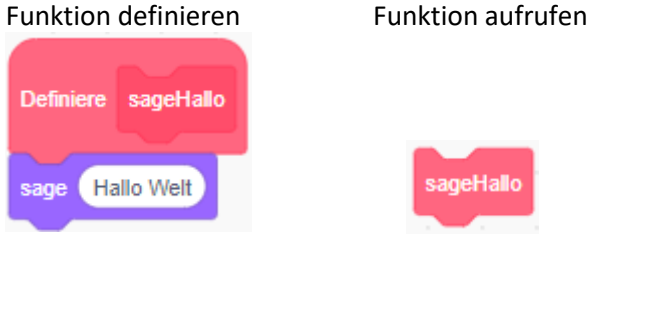
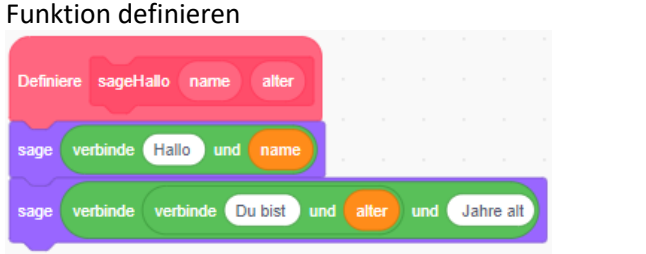
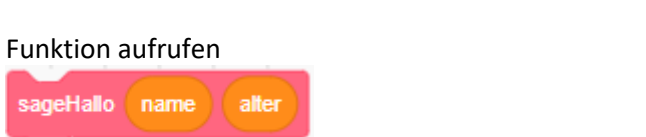



Befehlsübersicht

Beschrieb	Phyton (Textbasiert)	Scratch (Blockbasiert)
Einen Kommentar einfügen	<pre># Ein Kommentar, der bis ans Zeilenende geht """ Ein Kommentar, der über mehrere Zeilen geht. """</pre>	
Einen Text ausgeben	<pre>print ("Hallo Welt")</pre>	
Den Wert einer Variablen ausgeben	<pre>name = "Mark" print ("Hallo") print (name) oder name = "Mark" print ("Hallo", name)</pre>	
Texte kombinieren Texte kann man mit dem '+'-Operator aneinander anfügen.	<pre>begrueßung = "Hallo " text = begrueßung + "Mark" print(text)</pre>	
Einen Text einlesen	<pre>eingabeText = input ("Bitte Text eingeben: ")</pre>	
Eine Zahl einlesen Die Zahl muss mittels int(...) in eine ganze Zahl oder mittels float(...) in eine Kommazahl konvertiert werden	<pre>eingabeText = input("Bitte eine Zahl eingeben: "); eingabeZahl = int(eingabeText)</pre>	


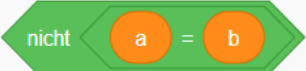
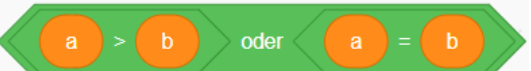
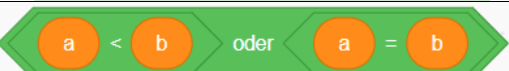
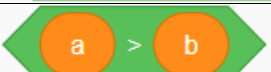
<p>Eine Bedingung stellen Eine Bedingung ist entweder Wahr (True) oder Falsch (False)</p>	<pre>bedingung = 10 > 20</pre>	
<p>Eine Bedingung verneinen Umdrehen einer Bedingung</p>	<pre>bedingung = 10 > 20 gegenteil = not bedingung</pre>	
<p>for-Schleife Eine for-Schleife wird verwendet um eine Liste, ein Text (ein Text ist nur eine Liste von Buchstaben), o.ä. zu durchzulaufen. Der eingerückte Code wird für jedes Listenelement einmal ausgeführt.</p>	<pre>liste = [1, 2, 3, 4] for zahl in liste: print("Zahl:", zahl) # Hier gehts normal weiter (ACHTUNG: nicht eingerückt)</pre>	<p>Gibt es in Scratch nicht</p>
<p>if-Abfragen Eine if-Abfrage überprüft, ob eine Bedingung Wahr oder Falsch ist. Alles was nach der if-Abfrage mit einem Tab eingerückt ist gehört zum Körper der if-Abfrage und wird nur ausgeführt, wenn die Bedingung Wahr ist</p>	<pre>wert = 10 # Eine Variable mit einem beliebigen Wert (hier 10) if wert > 20: # wird NUR ausgeführt, wenn Bedingung wahr ist print("Der Wert ist größer als 20") # (ACHTUNG: Einrückung) # Hier gehts weiter (ACHUTUNG: Nicht eingerückt)</pre>	
<p>if-else-Abfragen Mit einer if-else Abfrage kann man auch auf eine nicht erfüllte Bedingung mit dem 'else'-Zweig reagieren. Es wird immer entweder der if-Zweig oder der else-Zweig ausgeführt.</p>	<pre>wert = 10 # Eine Variable mit einem beliebigen Wert (hier 10) if wert > 20: # wird NUR ausgeführt wenn Bedingung erfüllt ist print("Der Wert ist größer als 20") else: # wird NUR ausgeführt wenn Bedingung nicht erfüllt ist print("Wert ist kleiner oder gleich 20.")</pre>	

<p>while-Schleife Eine while-Schleife führt den eingerückten Code aus, solange die Bedingung erfüllt ist.</p>	<pre>zahl = 1 while zahl < 10: # solange die Bedingung wahr ist... # ...wird dieser eingerückte Code wiederholt print("Zahl:", zahl) zahl = zahl + 1 # Zähler erhöhen # Hier gehts normal weiter (Achtung: nicht eingerückt)</pre>	
<p>Eine leere Liste erstellen</p>	<pre>liste = []</pre>	
<p>Eine Liste mit Einträgen erstellen</p>	<pre>liste = [1, 2, 3, 4, 5]</pre>	
<p>Einen Eintrag aus der Liste lesen Ein Wert aus der Liste wird über seinen Index, seine Position in der Liste, abgefragt Achtung: Das erste Element hat den Index 0</p>	<pre>liste = ["hallo", "test", "welt"] # Liste mit Einträgen erstellen ersterEintrag = liste[0] # der Erste Eintrag aus der Liste wird in der Variablen "ersterEintrag" gespeichert</pre>	

<p>Einen Eintrag aus der Liste lesen Eintrag vom Ende der Liste lesen Achtung: Das letzte Element hat den Index -1.</p>	<pre>liste = ["hallo", "test", "welt"] # Liste mit Einträgen erstellen letzterEintrag = liste[-1] # der Letzte Eintrag aus der Liste wird in der Variablen "letzterEintrag" gespeichert</pre>	
<p>Länge einer Liste ermitteln</p>	<pre>liste = [1, 2, 3, 4, 5] # Liste mit Einträgen erstellen laenge = len(liste) # die Länge der Liste wird in der Variablen "laenge" gespeichert</pre>	
<p>Einen Eintrag hinzufügen</p>	<pre>liste = [1, 2] # Liste mit Einträgen erstellen liste.append(3) # einen Eintrag anfügen</pre>	
<p>Einen Eintrag entfernen</p>	<pre>liste = [1, 2, 5, 6] # Liste mit Einträgen erstellen liste.remove(6) # löscht den Wert 6 aus der Liste</pre>	
<p>Eine Funktion definieren Eine Funktion ist ein Block Code, der vom Haupt-Programm getrennt definiert wird. Die Funktion wird erst dann ausgeführt, wenn man sie über ihren Namen aufruft. Dafür kann man eine Funktion so oft mal will verwenden. Variablen, die innerhalb einer Funktion definiert sind, sind nur innerhalb der Funktion verwendbar!</p>	<pre># Neue Funktion mit dem Namen 'sageHallo' definieren def sageHallo(): print("Hallo Welt") # Einrückung! # Funktion aufrufen (keine Einrückung) sageHallo()</pre>	<p>Funktion definieren Funktion aufrufen</p> 
<p>Eine Funktion mit Übergabeparametern definieren Funktion können Parameter (Variablen, die nur innerhalb der Funktion existieren) haben. Diese müssen beim Aufrufen mitübergeben werden.</p>	<pre># Funktion definieren def sageHallo(name, alter): print("Hallo", name) print("Du bist", alter, "Jahre alt") # Funktion aufrufen sageHallo("Mark", 22)</pre>	<p>Funktion definieren</p>  <p>Funktion aufrufen</p> 

<p>Zufallszahl Will man eine Zufallszahl erzeugen, kann man dies mit dem Modul random tun. Das Modul random muss einmal importiert werden, um verwendet werden zu können</p>	<pre>import random # Modul random importieren # generiert eine Zufallszahl zwischen 1 und 100 zufallsZahl = random.randint(1,100)</pre>	
<p>Aktuelles Datum Über das Modul datetime kann man das aktuelle Datum abfragen. Das Modul muss einmal importiert werden, um verwendet werden zu können.</p>	<pre>import datetime # Modul datetime importieren # aktuelles Datum abfragen heute = datetime.datetime.now() jahr = heute.year monat = heute.month tag = heute.day sekunde = heute.second</pre>	<p>Gibt es in Scratch nicht</p>
<p>Aktuelle Zeit in Sekunden Über das Modul time kann die aktuelle Zeit in Sekunden ermittelt werden. Das Modul muss einmal importiert werden, um verwendet werden zu können.</p>	<pre>import time # Modul time importieren # aktuelle Zeit in Sekunden sekunden = time.time()</pre>	<p>Gibt es in Scratch nicht</p>

Vergleichsmöglichkeiten

Beschrieb	Phyton	Scratch
a ist gleich b	<code>a == b</code>	
a ist nicht gleich b	<code>a != b</code>	
a ist grösser gleich b	<code>a >= b</code>	
a ist kleiner gleich b	<code>a <= b</code>	
a ist grösser als b	<code>a > b</code>	
a ist kleiner als b	<code>a < b</code>	