GB Englischer Begriff bzw. Pythonbefehl	DE Deutscher Ausdruck	Erklärung / Beispiel
`print()`	ausgeben / Ausgabe	Gibt Text oder Werte auf dem Bildschirm aus (`print("Hallo")`)
`input()`	eingeben / Eingabe	Liest Benutzereingabe ein (`name = input("Name? ")`)
`Int,integer`	ganze Zahl / Ganzzahl	Datentyp für Zahlen ohne Nachkommastellen (`x = int("5")`)
`float`	Kommazahl / Gleitkommazahl	Zahl mit Dezimalstellen (`pi = 3.14`)
`Str, string`	Text / Zeichenkette	Datentyp für Texte (`name = "Lina"`)
`bool`	Wahrheitswert	`True` (wahr) oder `False` (falsch)
Variable	Speichername / Platzhalter für Werte	z. B. `alter = 12`

`if`	wenn-dann	Bedingte Anweisung:
		`if zahl > 10:`
		`  print("zahl ist größer als ", 10)`
		Weiters wird `zahl > 10` Bedingung genannt

`If-else`	wenn-dann,
	sonst
		Verzweigung: Bedingte Anweisung mit Alternative if zahl > 10:
		`  print("zahl ist größer als ", 10)`
		else:
		`  print("zahl ist kleiner oder gleich als ", 10)`

`elif`	wenn-dann,
	sonst, wenn-dann,
	sonst
		Mehrfachverzweigung: mit vielen Alternativen
		`If zahl > 10:`
		`  print("zahl:", zahl, "ist größer als", 10)`
		`elif zahl == 10:`
		`  print("zahl:", zahl, "ist gleich", 10)`
		`else:`
		`  print("zahl:", zahl, "ist kleiner als", 10)`

`while`	solange	Schleife: Wiederholt Code, solange Bedingung wahr ist – bis der Benutzer 15 eingibt

GB Englischer Begriff bzw. Pythonbefehl	DE Deutscher Ausdruck	Erklärung / Beispiel
		`While input("Gib 15 ein: ") != 15:` `  Print("Du hast nicht 15 eingegeben.)`
`list`	Liste	Sammlung von Werten: `früchte = ["Apfel", "Banane"]`. Denke an einen Rucksack mit Fächern.

`for`	für jedes	Schleife: Wiederhole Code für jeden Wert in der Liste – für jede Frucht im Rucksack.
		`for frucht in ["Apfel", "Banane"]:`
		`  print("was ich heute esse: ", frucht)`

`range()`	Bereich / Zahlenreihe	Erzeugt Zahlenliste: • `range(5)` → [0, 1, 2, 3, 4] • `range(1, 5)` → [1, 2, 3, 4] • `range(3, 5)` → [3, 4]

`for mit range()`	zähle von … bis …	Schleife: Wir starten mit 0 und hören bei 4 auf zu Zählen.
		`for i in range(5):`
		`  print("ich zaehle von 0 bis 4: ", i)`

`len()`	Länge	Gibt die Anzahl der Elemente zurück: `len(früchte)`
`def`	Definiere eine Funktion	Erstellt eine Funktion: `def begrüsse():`
`return`	zurückgeben	Gibt ein Ergebnis aus einer Funktion zurück
`import`	einbinden / importieren	Z. B. zum Laden von Modulen `import random` random.nextint(5) # erzeugt zahlen von 0 bis 5.
`from und import`	einbinden / importieren	Z. B. zum speziellen Laden von Modulen • `from random import nextint` • `from random import *` nextint(5) # kein random. Mehr notwendig
Kommentar	Notiz / Hinweis im Code	Mit `#` markiert, wird vom Programm ignoriert

GB Englischer Begriff bzw. Pythonbefehl	DE Deutscher Ausdruck	Erklärung / Beispiel
`type()`	Typ anzeigen	Zeigt den Datentyp eines Werts an: `type(42)` → `<class 'int'>`
`modulo %`	Rest / Modulo / geteilter Rest	z. B. `10 % 3 = 1`, weil 10 = 3×3 + 1
Operator	Rechenzeichen / Vergleichszeichen	+, -, *, /, ==, <, > usw.
random	zufällig	`Wird für die Erzeugung von Zufallszahlen verwendet (z. B. Würfelsimulation)`