PROFESSOR ERKLÄRBÄR

Serie Informatik · Folge 7 · Grundlagen

INFORMATIK

LEHRERINFO ZUM VIDEO

Programmieren leicht gemacht

FACH	STUFE	DAUER	VIDEO
Informatik	Klasse 7–9	12 min	youtube.com/@Erklaerbaer

01 Lernziele

Nach dem Video können die Schülerinnen und Schüler …

- erklären, was eine Programmiersprache ist und warum Hochsprachen übersetzt werden,
- ein erstes Python-Programm mit `print` schreiben, ausführen und kommentieren,
- Variablen anlegen, ändern und sinnvoll benennen; die Zuweisung von rechts nach links lesen,
- die Datentypen `int`, `float`, `str` und `bool` unterscheiden,
- mit `+ - * / // % **` rechnen und Eingaben mit `input()` lesen und umwandeln,
- Fehlermeldungen lesen und SyntaxError, NameError, TypeError und ValueError beheben.

02 Voraussetzungen

Algorithmus und Ablaufplan (Folge 6); dass der Prozessor nur Maschinenbefehle versteht (Folge 5). Keine Programmiererfahrung nötig. Werkzeug: Thonny (kostenlos, thonny.org) oder ein Online-Editor ohne Konto.

03 Ablauf des Videos

ZEIT	KAPITEL	INHALT UND BEISPIEL
0:00	Einstieg	print("Hallo Welt") · print(15 * 52) · Ident · Fahrplan · Ziel: Taschenrechner 0.1
0:34	Warum Programmiersprachen?	Maschinensprache gegen Hochsprache · JavaScript, Java/Kotlin, C++, Scratch · Python
1:14	Das Werkzeug	Thonny oder Online-Editor · Code oben, Konsole unten, Startknopf
1:39	Hallo Welt	print Stück für Stück · Anführungszeichen = Text · ohne sie NameError · Kommentar mit #
2:12	Variablen	Kiste score · = heisst «bekommt den Wert» · score = score + 10 · Namensregeln PAUSE-FRAGE x = x * 2 + 1 · Merksatz

3:32	Datentypen	int, float (3.5), str, bool · type(42) gegen type("42") · Merksatz
4:31	Rechnen	Operatoren · 7 / 2, 7 // 2, 7 % 2 · Taschengeld 15 · 52 = 780 PAUSE-FRAGE 17 // 5 und 17 % 5
5:40	Eingaben	input und F-String (Hallo Mia!) · Alter in Tagen · "3" + "4" = "34" · int() · Merksatz
6:42	Fehlermeldungen lesen	von unten lesen · SyntaxError, NameError, TypeError, ValueError · Zeilennummer
7:24	Projekt-Baustein	Taschenrechner 0.1: zwei Zahlen mit float(input()), Summe 5.0
7:55	Typische Fehler	vier Fehlvorstellungen (siehe 04)
8:39	Zusammenfassung	Kernaussagen · zurück zu Hallo Welt
9:14	Quiz	4 Fragen · Endkarte Folge 8

04 Typische Fehlvorstellungen

- = heisst «ist gleich» – in Python heisst es «bekommt den Wert» (Zuweisung); verglichen wird mit == (Folge 8).
- `input()` liefert eine Zahl – immer Text; erst `int()`/`float()` machen eine Zahl daraus.
- Kommazahlen schreibt man im Code mit Komma – im Code mit Punkt: `3.5`.
- `score` und `Score` sind dieselbe Variable – Python unterscheidet Gross- und Kleinschreibung.

05 Einsatz im Unterricht

Einstieg: Alle schreiben in Thonny `print("Hallo Welt")` und danach ihren eigenen Satz – ein Erfolgserlebnis in der ersten Minute.

Fehlerjagd: Code mit je einem Syntax-, Namens-, Typ- und Wertfehler austeilen; die Klasse liest die Meldungen von unten und korrigiert.

Kisten-Spiel: Beschriftete Schachteln mit Zetteln für die Variablen; Zuweisungen von rechts nach links «ausführen».

Sicherung: Aufgaben 3 und 4 als Hausaufgabe, Aufgabe 6 als Mini-Projekt.

Hinweis: Die Fehlermeldungen entsprechen Python ab Version 3.10 (z. B. «'(' was never closed»); ältere Versionen formulieren teils anders. Das Alter in Tagen ist ohne Schaltjahre gerechnet. Variablennamen sind im Video englisch, wie im Berufsalltag üblich; deutsche Namen ohne Umlaute funktionieren genauso.

06 Quellen

- Python Software Foundation: The Python Tutorial (docs.python.org/3/tutorial), Kapitel 3 «An Informal Introduction to Python»
- Python-Dokumentation: Built-in Functions (print, input, int, float, type) und Built-in Exceptions
- Thonny – Python IDE for beginners, Universität Tartu (thonny.org)
- Lehrplan 21, Medien und Informatik: Bereich «Informatik – Programmieren» (MI.2.2)

NAME _____ KLASSE _____ DATUM _____

Programmieren leicht gemacht

1 Merksätze ergänzen ●○○

Eine _____ ist ein Name für einen gespeicherten Wert; = _____ zu.

`input()` liefert immer _____ – zum Rechnen erst mit _____ oder _____ umwandeln.

2 Welcher Datentyp? ●○○

Gib den Typ **an**: a) `17` b) `"17"` c) `17.0` d) `False` e) `"False"` f) `-3`

3 Variablen von Hand ●●○

Gib nach jeder Zeile den Wert von `a` und `b` **an**: `a = 4 · b = a + 3 · a = a * b · b = b - 1 · a = a // b`

4 Rechnen mit Python ●●○

Berechne ohne Computer: a) `2 + 3 * 4` b) `(2 + 3) * 4` c) `20 / 8` d) `20 // 8` e) `20 % 8` f) `2 ** 5`

5 Fehler finden ●●○

Nenne den Fehlertyp und **korrigiere**: a) `print("Hallo)` b) `Print("Hallo")` c) `alter = input("Alter? ") + 1` d) `zahl = int("zwölf")`

6 Celsius in Fahrenheit

Schreibe ein Programm, das eine Temperatur in Grad Celsius einliest und in Fahrenheit ausgibt (F = C · 1.8 + 32). **Teste** es mit 0, 37 und 100.

Lösungen

1 Merksätze ergänzen

Variable · weist · Text · int() · float()

2 Welcher Datentyp?

a) int · b) str · c) float · d) bool · e) str · f) int

3 Variablen von Hand

a = 4, b – · a = 4, b = 7 · a = 28, b = 7 · a = 28, b = 6 · a = 4, b = 6 (28 // 6 = 4)

4 Rechnen mit Python

a) 14 (Punkt vor Strich) · b) 20 · c) 2.5 · d) 2 · e) 4 · f) 32

5 Fehler finden

a) SyntaxError – Anführungszeichen schliessen: `print("Hallo")` · b) NameError – Python unterscheidet Gross/Klein: `print` · c) TypeError – Text + Zahl: `alter = int(input("Alter? ")) + 1` · d) ValueError – `int()` versteht nur Ziffern: `int("12")`

6 Celsius in Fahrenheit

`c = float(input("Grad Celsius: "))` · `f = c * 1.8 + 32` · `print(f"{c} °C sind {f} °F")` → 0 °C = 32 °F · 37 °C = 98.6 °F (Python zeigt 98.60000000000001 – ein Rundungsfehler, siehe Folge 2 Advanced) · 100 °C = 212 °F

Q Quiz im Video

1 B (str) · 2 B (immer Text) · 3 B (1) · 4 B (x wird um 1 grösser)