

Eingabe & Ausgabe

Ausgabe von Daten mit print()

Oft werden Programme geschrieben, bei denen dem Benutzer Daten am Bildschirm ausgegeben werden. In Python kann man dazu die Funktion `print()` benutzen.

Die Funktion `print()` gibt das, was in ihren nachstehenden Klammern steht, auf dem Bildschirm aus. Dabei ist es wichtig, dass Texte in Anführungszeichen gesetzt werden, damit Python die Zeichen auch als Text erkennt und nicht als Python- bzw. Programm-Elemente interpretiert.

Erklärvideo

[python-ausgabe-print.mp4](#)

Beispiel

```
print("Hallo Du da!")
```

Kopiere den Beispielcode und füge ihn in den Editor ein. Führe das Programm aus!

Eingaben einsammeln mit input()

Mit der Input-Funktion können wir einen Benutzer zu einer Eingabe auffordern. Die Eingabe können wir dann einer Variable zuweisen und damit für eine spätere Verwendung speichern.

Erklärvideo

[python-eingabe-input.mp4](#)

Beispiel

```
anzahl_spalten = input("Wie viele Spalten soll die Grafik haben? ")  
print(anzahl_spalten)
```

Funktionsweise

- Der Interpreter sieht den Aufruf der `input`-Funktion und kopiert den Text der

Eingabeaufforderung (hier: Wie viele Spalten soll die Grafik haben?) und zeigt diesen in der Python-Shell an.

- Dann wartet der Interpreter auf eine Eingabe des Benutzers. Die Benutzereingabe über die Tastatur wird mit Drücken der Enter-Taste beendet.
- Anschließend wird die Benutzereingabe an den Code zurückgegeben.

Kopiere den Beispielcode und füge ihn in den Editor ein. Führe das Programm aus!

```
<html>
```

```
<iframe src=„https://trinket.io/embed/python/0b99211fef“ width=„100%“ height=„200“  
frameborder=„0“ marginwidth=„0“ marginheight=„0“ allowfullscreen></iframe>
```

```
</html>
```

Variablen, Eingabe & Ausgabe

Erklärvideo

[python-variablen-anwenden.mp4](#)

Rechenoperatoren

Zur Durchführung von Berechnungen stehen verschiedene Rechenoperatoren zur Verfügung:

Operator	Zeichen	Beispiel	Ergebnis
Addition	+	3 + 4	7
Subtraktion	-	5 - 11	-6
Multiplikation	*	3 * 6	18
Division	/	9 / 2	4.5
Potenz	**	3 ** 4	81

```
zahl_a = zahl_x + zahl_y  
zahl_b = zahl_x - 20  
zahl_c = zahl_x * zahl_y  
zahl_d = zahl_x / 5  
zahl_e = zahl_x ** 4
```

Aufgabe 1

1. Deklariere und initialisiere eine Variable mit dem Namen 'Wohnort' und weise ihr einen beliebigen Wert (natürlich einen Ortsnamen) zu. Gebe die Variable mit der print-Anweisung auf der Shell aus.
2. Deklariere und initialisiere eine Variable mit dem Namen 'PLZ' und fordere den Benutzer zu einer Eingabe auf. Gebe die Variable mit der print-Anweisung auf der Shell aus.
3. Deklariere und initialisiere eine Variable mit dem Namen 'Geburtsjahr', fordere den Benutzer zu einer Eingabe auf und weise den Eingabewert der Variable zu. Gebe das Alter (Ganzzahl) mit der print-Anweisung auf der Shell aus. Das Alter muss hierbei berechnet werden ;)

[Nutze hierzu den Editor.](#)

Aufgabe 2

Verändere den Programmcode so, dass ein Benutzer die Anzahl der Punkte bestimmen kann.

[Nutze hierzu den Editor.](#)

```
import turtle
import random

#Farben "#373f51" "#208aae" "#e43f6f"

turtle.speed(20)
turtle.penup()
turtle.goto(-100,100)

turtle.pendown()
for i in range(1,4):
    zufallszahl = random.randint(1, 100)
    if zufallszahl > 90:
        turtle.dot(80,"#373f51")
    elif zufallszahl > 10:
        turtle.dot(80,"#208aae")
    else:
        turtle.dot(80,"#e43f6f")
    turtle.penup()
    turtle.forward(85)
```

Aufgabe 3

Entwerfe ein Programm, das die Rechenoperationen plus, minus, mal, geteilt und hoch durchführt (siehe [Rechenoperatoren](#)).

Die Bildschirmausgabe sollte so aussehen:

```
Zahl1: 13, Zahl2: 7
-----
Addition: 20
Subtraktion: 6
Multiplikation: 91
Division: 1.8571428571428572
Potenz: 62748517
```

Editor

Hier geht es weiter...

[Funktionen ohne Parameter](#)

[\[informatik, arbeitsauftrag, computerkunst, lernpfad, python\]](#)

From: <https://herr-pfeiffer.de/unterrichtswiki/> - **Unterrichtswiki - Herr Pfeiffer**

Permanent link: <https://herr-pfeiffer.de/unterrichtswiki/informatik:computerkunst:eingabe-ausgabe?rev=1741442557>

Last update: **2025/03/08 15:02**

