PyBR - Einsteigerhandbuch

Maxwell Anderson Ielpo do Amaral

Vollständiges Handbuch für Anfänger

Lernen Sie Python-Programmierung auf Deutsch

Janeiro/2026

PyBR - Python International
https://github.com/maxwellamaral/pybr

Inhaltsverzeichnis

Programmieren lernen mit PyBR - Einsteigerhandbuch

Autor: Maxwell Anderson Ielpo do Amaral Übersetzung: AI Assistant

Veröffentlicht im Januar 2026

Willkommen in der Welt der Programmierung!

Dieser Leitfaden wurde speziell für Sie erstellt, wenn Sie noch nie zuvor programmiert haben und es auf einfache Weise und auf Deutsch lernen möchten! Mit **PyBR** lernen Sie das Programmieren mit deutschen Wörtern anstelle des traditionellen Englisch von Python.

Inhaltsverzeichnis

12. Fortgeschrittene Funktionen
13. Objekte erstellen - Klassen
14. Praktische Projekte

🖥️ Verwendung des Terminals

Wenn Sie das **Terminal** (oder die **Eingabeaufforderung**) noch nie benutzt haben, keine Sorge! Es ist einfacher als es aussieht.

Grundlegende Befehle

Aktion	Windows	Mac/Linux
Wo bin ich?	`cd`	`pwd`
Dateien auflisten	`dir`	`ls`
Ordner betreten	`cd ordner`	`cd ordner`
Zurückgehen	`cd ..`	`cd ..`
Bildschirm löschen	`cls`	`clear`

🐍 Python installieren

Bevor Sie beginnen, müssen Sie **Python** installiert haben.

1. Öffnen Sie das Terminal und tippen Sie: `python --version`
2. Wenn `Python 3.x.x` erscheint, sind Sie bereit!
3. Wenn nicht, laden Sie es unter python.org herunter.
    - **Windows**: Wichtig! Kreuzen Sie **"Add Python to PATH"** während der Installation an!

PyBR ausführen

Voraussetzungen

✅ **Python 3.6+** ✅ **PyBR-Dateien** (`pybr.py`)

Ausführungsmöglichkeiten

Option 1: Interaktiver Modus (REPL) Perfekt für schnelle Tests. Im Terminal:

```
python pybr.py --lang de
```

Sie werden sehen:

```
PyBR - Python International (Mehrsprachig)
Geben Sie 'beenden()' ein, um zu beenden.
>>>
```

Option 2: Dateien ausführen Erstellen Sie eine Datei `mein_programm.pybr` und führen Sie sie aus:

```
python pybr.py mein_programm.pybr --lang de
```

Ihr erstes Programm

Beginnen wir mit dem Klassiker “Hallo Welt”:

```
drucke("Hallo Welt!")
```

Probieren Sie es selbst:

```
drucke("Ich lerne Programmieren mit PyBR!")
```

Variablen - Das Gedächtnis des Computers

Variablen sind wie Boxen, in denen Sie Informationen speichern.

Variablen erstellen:

```
# Einen Namen speichern
name = "Maria"

# Ein Alter speichern
alter = 25

# Variablen verwenden
drucke(name)
drucke(alter)
```

Datentypen:

```
# TEXT (String)
stadt = "Berlin"

# GANZE ZAHLEN (Integer)
anzahl = 10

# DEZIMALZAHLEN (Float)
preis = 19.99

# WAHR oder FALSCH (Boolean)
ist_montag = Wahr
es_regnet = Falsch
```

Rechnen und mathematische Operationen

```
# ADDITION (+)
summe = 10 + 5
drucke(summe)  # Zeigt: 15

# SUBTRAKTION (-)
differenz = 20 - 8
drucke(differenz)  # Zeigt: 12

# MULTIPLIKATION (*)
produkt = 6 * 7
drucke(produkt)  # Zeigt: 42

# DIVISION (/)
ergebnis = 15 / 3
drucke(ergebnis)  # Zeigt: 5.0
```

Eingabe und Ausgabe

Ausgabe (Informationen anzeigen):

```
name = "Hans"
drucke(f"Mein Name ist {name}")
```

Eingabe (Informationen empfangen):

```
name = eingabe("Wie heißt du? ")
drucke(f"Hallo, {name}!")

# Für Zahlen müssen wir umwandeln:
alter = ganzzahl(eingabe("Wie alt bist du? "))
drucke(f"Du bist {alter} Jahre alt")
```

Entscheidungen treffen - Konditionale

Das Programm trifft Entscheidungen mit `wenn`, `sonstfalls`, `sonst`.

```
alter = 18

wenn alter >= 18:
    drucke("Du bist volljährig")
sonst:
    drucke("Du bist minderjährig")
```

Beispiel mit WENN-SONSTFALLS-SONST:

```
note = 85

wenn note >= 90:
    drucke("Ausgezeichnet!")
sonstfalls note >= 70:
    drucke("Gut!")
sonst:
    drucke("Muss verbessert werden")
```

Aktionen wiederholen - Schleifen

FÜR-Schleife (for):

```
# Zählen von 0 bis 4
fuer i in bereich(5):
    drucke(i)
```

SOLANGE-Schleife (while):

```
zaehler = 0

solange zaehler < 5:
    drucke(f"Zähler: {zaehler}")
    zaehler = zaehler + 1
```

Code organisieren - Funktionen

Funktionen sind wiederverwendbare Codeblöcke.

```
definiere begruessen(name):
    drucke(f"Hallo, {name}!")

begruessen("Anna")
begruessen("Peter")
```

Funktionen mit Rückgabewert:

```
definiere addieren(a, b):
    rueckgabe a + b

ergebnis = addieren(10, 20)
drucke(ergebnis)  # 30
```

Fortgeschrittene Funktionen

Lambda-Funktionen:

Kleine einzeilige Funktionen.

```
doppelt = lambda x: x * 2
drucke(doppelt(5))  # 10
```

Abbilden (map):

```
zahlen = [1, 2, 3, 4]
quadrate = liste(abbilden(lambda x: x ** 2, zahlen))
drucke(quadrate)  # [1, 4, 9, 16]
```

Filtern (filter):

```
zahlen = [1, 2, 3, 4, 5, 6]
gerade = liste(filtern(lambda x: x % 2 == 0, zahlen))
drucke(gerade)  # [2, 4, 6]
```

Objekte erstellen - Klassen

Klassen sind “Baupläne” für Objekte.

```
klasse Hund:
    definiere __init__(selbst, name, rasse):
        selbst.name = name
        selbst.rasse = rasse

    definiere bellen(selbst):
        drucke(f"{selbst.name}: Wuff wuff!")

# Objekte erstellen
rex = Hund("Rex", "Schäferhund")
rex.bellen()
```

Praktische Projekte

Projekt 1: Aufgabenliste

```
klasse AufgabenManager:
    definiere __init__(selbst):
        selbst.aufgaben = []

    definiere hinzufuegen(selbst, aufgabe):
        selbst.aufgaben.append(aufgabe)
        drucke(f"Aufgabe hinzugefügt: {aufgabe}")
```

```
    definiere auflisten(selbst):
        drucke("--- Meine Aufgaben ---")
        fuer i, aufgabe in aufzaehlen(selbst.aufgaben):
            drucke(f"{i + 1}. {aufgabe}")

manager = AufgabenManager()
manager.hinzufuegen("PyBR lernen")
manager.hinzufuegen("Python üben")
manager.auflisten()
```

Projekt 2: Temperaturrechner

```
definiere celsius_in_fahrenheit(c):
    rueckgabe (c * 9/5) + 32

temp_c = gleitkommazahl(eingabe("Temperatur in Celsius: "))
temp_f = celsius_in_fahrenheit(temp_c)
drucke(f"{temp_c}°C ist gleich {temp_f}°F")
```

Abschließende Tipps

1. **Üben Sie jeden Tag**: Beständigkeit ist der Schlüssel.
2. **Lesen Sie Fehlermeldungen**: Sie helfen Ihnen, den Code zu korrigieren.
3. **Kommentieren Sie Ihren Code**: Verwenden Sie #, um zu erklären.

Herzlichen Glückwunsch!

Sie haben den PyBR-Grundlagenleitfaden abgeschlossen. Sie sind jetzt ein Programmierer!