



Python Grundlagen – Übersicht

print()

Ausgabe

Gibt Text oder Werte auf dem Bildschirm aus.

Beispiel:

```
print("Hallo")
```

input()

Eingabe

Liest Benutzereingabe ein.

Beispiel:

```
name = input("Name?")
```

int / integer

Ganze Zahl

Datentyp für ganze Zahlen ohne Nachkommastellen.

Beispiel:

```
x = int("5")
```

float

Kommazahl / Gleitkommazahl

Zahl mit Dezimalstellen.

Beispiel:

```
pi = 3.14
```

str / string

Text / Zeichenkette

Datentyp für Texte.

Beispiel:

```
name = "Lina"
```

bool

Wahrheitswert

Hat nur die Werte True (wahr) oder False (falsch).

Variable

Platzhalter

Speichert einen Wert.

Beispiel:

```
alter = 12
```



if

Bedingte Anweisung

Führt Code nur aus, wenn eine Bedingung erfüllt ist.

Beispiel:

```
if zahl > 10:  
    print("Zahl ist größer als 10")
```

if-else

Verzweigung

Führt eine Alternative aus, wenn die Bedingung nicht erfüllt ist.

Beispiel:

```
if zahl > 10:  
    print("größer")  
else:  
    print("kleiner oder gleich")
```

elif

Mehrfachverzweigung

Erlaubt mehrere Bedingungen nacheinander zu prüfen.

Beispiel:

```
if zahl > 10:  
    print("größer")  
elif zahl == 10:  
    print("gleich")  
else:  
    print("kleiner")
```

while

Schleife (solange)

Wiederholt Code, solange eine Bedingung wahr ist.

Beispiel:

```
while input("Gib 15 ein: ") != "15":  
    print("Du hast nicht 15 eingegeben.")
```

list

Liste

Sammlung von Werten.

Beispiel:

```
fruechte = ["Apfel", "Banane"]
```



for

Schleife (für jedes)

Wiederholt Code für jedes Element einer Liste.

Beispiel:

```
for frucht in ["Apfel", "Banane"]:  
    print("Ich esse:", frucht)
```

range()

Zahlenbereich

Erzeugt eine Liste von Zahlen.

Beispiel:

range(5) → [0, 1, 2, 3, 4]

len()

Länge

Gibt die Anzahl der Elemente zurück.

Beispiel:

len(fruechte) → 2

def

Funktion definieren

Erstellt eine eigene Funktion.

Beispiel:

```
def begruesse():  
    print("Hallo!")
```

return

Rückgabewert

Gibt ein Ergebnis aus einer Funktion zurück.

import

Module laden

Lädt externe Module.

Beispiel:

```
import random  
print(random.randint(0, 5))
```

from ... import

Gezielter Import

Lädt bestimmte Teile eines Moduls.

Beispiel:

```
from random import randint  
print(randint(0, 5))
```



Kommentar

Notiz im Code

Mit # markiert, vom Programm ignoriert.

Beispiel:

```
# Das ist ein Kommentar
```

type()

Datentyp anzeigen

Zeigt den Typ eines Werts.

Beispiel:

```
type(42) → <class 'int'>
```

modulo %

Restwert

Berechnet den Rest einer Division.

Beispiel:

```
10 % 3 = 1
```

Operator

Rechen- / Vergleichszeichen

Verwendet für Berechnungen und Vergleiche.

Beispiele: +, -, *, /, ==, <, >

random

Zufallszahlen

Erzeugt Zufallswerte.

Beispiel:

```
import random
```

```
print(random.randint(1,6))
```