Python Kurzanleitung

Vorkurs 2025

Variablen & Typen

Zuweisung

```
zahl = 5
```

Grundtypen und Beispielwerte

```
int:   1, -2, 42
float: 1.2, 6e-10
str:   'Hello', "World"
bool:  True, False
list:  [1, 2, 3], ['a','b','c']
dict:  {'a': 1, 'b': 2}
```

Typ bestimmen

```
a = 0
print(type(a)) -> "<class 'int'>"
a = 'Lorem'
print(type(a)) -> "<class 'str'>"
```

Eingabe / Ausgabe

Konsolenausgabe

```
print('Lorem ipsum!')
print(5)
```

Konsoleneingabe

```
inp = input('Gib eine Zahl ein: ') -> str
num = int(inp) + 3                 -> int
```

Datei lesen

```
f = open('file.txt')
lines = f.readlines()
f.close()
```

Verzweigungen

Einfache Verzweigungen

```
if x == 5:
  print('case 1')
elif x == 3:
  print('case 2')
else:
  print('case 3')
```

Match anhand eines Wertes

```
match x:
  case 5:
    print('case 1')
  case 3:
    print('case 2')
  default:
    print('case 3')
```

Operatoren

Operatoren können entweder prefix auf einen Wert angewendet werden (z.B. `not <bool>`) oder infix zwischen zwei Werten (z.B. `<value1> + <value2>`). Arithmetische Operatoren arbeiten auf Zahlen (`int`, `float`)

Arithmetisch

Operator	Schreibweise
Addition:	`+`
Subtraktion:	`-`
Multiplikation:	`*`
Division:	`/`
Division (abgerundet):	`//`
Potenz:	`**`
Modulo:	`%`

Für alle Operatoren gibt es noch eine Variante mit Suffix `=` (z.B. `a += b`), welche eine Kurzform für `a = a + b` sind.

Vergleiche

Operator	Schreibweise
Gleich:	a `==` b
Ungleich:	a `!=` b
Kleiner als:	a `<` b
Kleiner gleich:	a `<=` b
Größer als:	a `>` b
Größer gleich:	a `>=` b

Vergleichsoperatoren arbeiten auf beliebigen vergleichenbaren Werten z.B. Zahlen und Strings.

Logisch

Operator	Schreibweise
Nicht:	`not` a
Oder:	a `or` b
Und:	a `and` b

Logische Operatoren arbeiten nur auf `bool` Werten.

Strings & Listen

Zugriff (int als Index, beginnend bei 0)

```
lst = [1, 3, 5]
lst[2] -> 5
```

Konkatenation

```
lst2 = lst + [4] -> [1, 3, 5, 4]
```

Länge

```
len('Lorem') -> 5
len(lst2) -> 4
```

Slices

```
lst2[0:3:2] -> [1, 5]
```

Enthalten

```
1 in lst2 -> True
6 in lst2 -> False
```

Strings

Formatierung

```
f'Heute ist der {tag}.{monat}.{jahr}'
```

Großbuchstaben:

```
'LoReM'.upper() -> 'LOREM'
```

Kleinbuchstaben:

```
'LoReM'.lower() -> 'lorem'
```

Ersetzen:

```
'lorem'.replace('ore', 'i') -> 'lim'
```

Listen

Definition

```
x = []
x = [5, 3, 2]
```

Element anhängen

```
x.append(y)
```

Letztes Element entfernen

```
x.pop()
```

Liste umkehren

```
x.reverse()
```

Liste sortieren

```
x.sort()
```

Konvertierung

Konvertierung von Primitiven (Beispiele)

```
int('5')      -> 5
int(5.37)     -> 5
float(5)      -> 5.0
str(1.3)      -> '1.3'
bool('True') -> True
```

Schleifen

Solange eine Bedingung gilt

```
while a < 20:
  print('Schleife läuft noch')
  a = (a + 3) * 2
```

Anhand einer Zahlensequenz:

```
for index in range(start, end, step):
  print(f'Wert an {index} ist {a[index]}')
```

Pro Wert eines Objektes:

```
for value in list:
  print(f'Wert: {value}')
```

Pro Wert und dessen Index:

```
for index,value in enumerate(list):
  print(f'Wert an {index} ist {value}')
```

Pro Wert am gleichen Index zweier Objekte (Listen müssen gleich lang sein)

```
for a,b in zip(list1, list2):
  print(f'Wertepaar: ({a}, {b})')
```

Funktionen

Definition

```
def fun(a, b):
  c = a + b
  return c
```

Aufruf

```
a, b = 5, 3
result = fun(a, b)
print(result) -> 8
```

Rekursion

```
def fun(n):
  if n == 0:
    return 1
  return fun(n-1) * 2
```