

Zuordnung von Variablen

```
x = 5
print(x)
5
```

Ausgaben und Platzhalter sind im Text durch *kursive Schrift* gekennzeichnet.

Verschiedene print-Varianten

```
de = "Deutschland"
de_pop = 82.67
mil = "Millionen"
de_cap = "Berlin"
cap_pop = 3.47
cap = "Hauptstadt"
```

Typkonvertierung

```
print("In "+de+" leben "+str(de_pop)+mil+".")
In Deutschland leben 82.67 Millionen.
```

Benutzung von Kommas

```
print("In "+de_cap+", wohnen "+str(cap_pop)+mil+".")
In Berlin wohnen 3.47 Millionen Menschen
```

Einfügen mit .format()

```
print("{}s {} ist {}".format(de, cap, de_cap))
Deutschlands Hauptstadt ist Berlin.
```

Datentypen

String	"Hello", 'Night', "20", ""Multilines""
Integer	7, -132, 42
Float	7.32, 42.0, - 17.65
Boolean	True, False

Typumwandlung

str()	Variable zu string
int()	Variable zu integer
float()	Variable zu float
bool()	Variable zu boolean

Berechnen von Variablen

x + 7	12	Summe
x - 7	-2	Subtraktion
x * 3	15	Multiplikation
x ** 4	625	Potenzierung
x % 2	1	Restberechnung
x / 2	2.5	Division

Kurzschreibweise: x op= y
Beispiel: x += 2 erhöht x um 2

Vergleichoperatoren

x < y	Kleiner	x <= y	Kleiner gleich
x > y	Größer	x >= y	Größer gleich
x == y	Gleich	x != y	Ungleich

Geben einen booleschen Wert aus

Boolesche Operatoren

x and y	x or y	not y
---------	--------	-------

Bedingungen

If Bedingung

```
if Bedingung 1:
    Anweisung1
elif Bedingung 2:
    Anweisung2
else:
    Anweisung3
```

Beispiel:

```
if x < 0:
    print (" x ist kleiner als null.")
elif x == 0:
    print ("x ist gleich null.")
elif x == 1:
    print ("x ist gleich eins:")
else:
    print ("x ist größer als eins.")
```

Schleifen

While Schleife

```
while Bedingung:
    Anweisung
```

For Schleife

```
for i in *range(start,end[, step]):
    Anweisung
```

Methoden für Strings

string.upper()	String zu Großschrift
string.lower()	String zu Kleinschrift
string.count("x")	Zählt Stringelemente
string.strip()	Löscht Leerzeichen im String
string.replace("x","y")	Ersetzt alle x für y

Methoden für Listen

Input	Output
meine_liste.index(a)	Index von Element a
meine_liste.count(a)	Zählt ein Element
meine_liste.append(a)	Fügt Element a hinzu
meine_liste.remove(a)	Löscht Element a
meine_liste.reverse()	Keht die Liste um
meine_liste.sort()	Sortiert die Liste

Indexing und Slicing

meine_liste[1]	Element mit Index 1
meine_liste[-2]	Vorletztes Element
meine_liste[1:3]	Element mt Index 1 bis 3
meine_liste[1:]	Alle Elemente nach Index 0
meine_liste[:4]	Alle Elemente vor Index 4
meine_liste[:]	Kopiert alle Elemente

Die Indexierung von Listen startet bei 0.
Beispiel:
meine_liste = ["Eins", "Zwei", "Drei"]
meine_liste[0]
Eins



Nützliche Funktionen

help()	Zeigt Hilfe zum gesuchten Begriff an
type()	Gibt den Datentyp eines Objektes oder Variablen zurück
input()	Gibt die Möglichkeit einer Eingabe vom Benutzer
len()	Gibt die Länge eines Strings, Tupels oder Liste zurück
round()	Rundet einen Float
sum()	Gibt die Summe aller Elemente zurück
max()	Gibt das größte Element zurück
min()	Gibt das kleinste Element zurück

Exception Handling

```
try:  
    Anweisung  
except ValueError as e:  
    print e  
except (TypeError, AnotherError):  
    Anweisung
```

Falls innerhalb des **try** ein Fehler auftritt, der zum Abbruch des Programmes führen würde, wird das entsprechende **except** ausgeführt.

Jupyter Notebook Keyboard Shortcuts

Strg + A	Alles markieren
Strg + Z	Undo
Strg + Y	Redo
Strg + Enter	Zelle ausführen
Strg + Links	Ein Wort nach links
Strg + Rechts	Ein Wort nach rechts
Strg + Oben	Gehe zum Zellenanfang
Strg + Unten	Gehe zum Zellenende

Pakete

Import von Paketen

```
import numpy  
import pandas as pd
```

Selektiver Import

```
from matplotlib import pyplot
```

