

Zuordnung von Variablen	Berechnen von Variablen	Methoden für Strings																												
<pre>x = 5 print(x) 5</pre> <p>Ausgaben und Platzhalter sind im Text durch <i>kursive Schrift</i> gekennzeichnet.</p>	<table> <tr> <td>x + 7</td><td>12</td><td>Summe</td></tr> <tr> <td>x - 7</td><td>-2</td><td>Subtraktion</td></tr> <tr> <td>x * 3</td><td>15</td><td>Multiplikation</td></tr> <tr> <td>x ** 4</td><td>625</td><td>Potenzierung</td></tr> <tr> <td>x % 2</td><td>1</td><td>Restberechnung</td></tr> <tr> <td>x / 2</td><td>2.5</td><td>Division</td></tr> </table> <p>Kurzschreibweise: x op= y Beispiel: x += 2 erhöht x um 2</p>	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	<table> <tr> <td>string.upper()</td><td>String zu Großschrift</td></tr> <tr> <td>string.lower()</td><td>String zu Kleinschrift</td></tr> <tr> <td>string.count("x")</td><td>Zählt Stringelemente</td></tr> <tr> <td>string.strip()</td><td>Löscht Leerzeichen im String</td></tr> <tr> <td>string.replace ("x", "y")</td><td>Ersetzt alle x für y</td></tr> </table>	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
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																												
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																													
Verschiedene print-Varianten	Vergleichoperatoren	Methoden für Listen																												
<pre>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",cap_pop,mil".") In Berlin wohnen 3.47 Millionen Menschen Einfügen mit .format() print("{0}s {1} ist {2}.".format(de,cap,de_cap)) Deutschlands Hauptstadt ist Berlin.</pre>	<table> <tr> <td>x < y</td><td>Kleiner</td><td>x <= y</td><td>Kleiner gleich</td></tr> <tr> <td>x > y</td><td>Größer</td><td>x >= y</td><td>Größer gleich</td></tr> <tr> <td>x == y</td><td>Gleich</td><td>x != y</td><td>Ungleich</td></tr> </table> <p>Geben einen boolschen Wert aus</p>	x < y	Kleiner	x <= y	Kleiner gleich	x > y	Größer	x >= y	Größer gleich	x == y	Gleich	x != y	Ungleich	<table> <tr> <th>Input</th><th>Output</th></tr> <tr> <td>meine_liste.index(a)</td><td>Index von Element a</td></tr> <tr> <td>meine_liste.count(a)</td><td>Zählt ein Element</td></tr> <tr> <td>meine_liste.append(a)</td><td>Fügt Element a hinzu</td></tr> <tr> <td>meine_liste.remove(a)</td><td>Löscht Element a</td></tr> <tr> <td>meine_liste.reverse()</td><td>Kehrt die Liste um</td></tr> <tr> <td>meine_liste.sort()</td><td>Sortiert die Liste</td></tr> </table>	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()	Kehrt die Liste um	meine_liste.sort()	Sortiert die Liste		
x < y	Kleiner	x <= y	Kleiner gleich																											
x > y	Größer	x >= y	Größer gleich																											
x == y	Gleich	x != y	Ungleich																											
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()	Kehrt die Liste um																													
meine_liste.sort()	Sortiert die Liste																													
Datentypen	Boolsche Operatoren	Indexing und Slicing																												
<table> <tr> <td>String</td><td>"Hello", 'Night', "20", """Multilines""""</td></tr> <tr> <td>Integer</td><td>7, -132, 42</td></tr> <tr> <td>Float</td><td>7.32, 42.0, - 17.65</td></tr> <tr> <td>Boolean</td><td>True, False</td></tr> </table>	String	"Hello", 'Night', "20", """Multilines""""	Integer	7, -132, 42	Float	7.32, 42.0, - 17.65	Boolean	True, False	<table> <tr> <td>x and y</td><td>x or y</td><td>not y</td></tr> </table>	x and y	x or y	not y	<table> <tr> <td>meine_liste[1]</td><td>Element mit Index 1</td></tr> <tr> <td>meine_liste[-2]</td><td>Vorletztes Element</td></tr> <tr> <td>meine_liste[1:3]</td><td>Element mt Index 1 bis 3</td></tr> <tr> <td>meine_liste[1:]</td><td>Alle Elemente nach Index 0</td></tr> <tr> <td>meine_liste[:4]</td><td>Alle Elemente vor Index 4</td></tr> <tr> <td>meine_liste[:]</td><td>Kopiert alle Elemente</td></tr> </table> <p>Die Indexierung von Listen startet bei 0.</p>	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					
String	"Hello", 'Night', "20", """Multilines""""																													
Integer	7, -132, 42																													
Float	7.32, 42.0, - 17.65																													
Boolean	True, False																													
x and y	x or y	not y																												
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																													
Typumwandlung	Bedingungen																													
<table> <tr> <td>str()</td><td>Variable zu string</td></tr> <tr> <td>int()</td><td>Variable zu integer</td></tr> <tr> <td>float()</td><td>Variable zu float</td></tr> <tr> <td>bool()</td><td>Variable zu boolean</td></tr> </table>	str()	Variable zu string	int()	Variable zu integer	float()	Variable zu float	bool()	Variable zu boolean	<p>If Bedingung</p> <pre>if Bedingung 1: Anweisung1 elif Bedingung 2: Anweisung2 else: Anweisung3</pre> <p>Beispiel:</p> <pre>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.")</pre>	<p>Beispiel:</p> <pre>meine_liste = ["Eins" , "Zwei" , "Drei"] meine_liste[0] Eins</pre>																				
str()	Variable zu string																													
int()	Variable zu integer																													
float()	Variable zu float																													
bool()	Variable zu boolean																													
Schleifen	While Schleife																													
	<p>While Schleife</p> <pre>while Bedingung: Anweisung</pre> <p>For Schleife</p> <pre>for i in *range(start,end[, step]): Anweisung</pre>																													



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

Pakete

Import von Paketen

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

Selektiver Import

```
from matplotlib import pyplot
```

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



By [The_one_and_only](#)
cheatography.com/the-one-and-only/

Not published yet.
Last updated 9th August, 2018.
Page 2 of 2.

Sponsored by [Readability-Score.com](#)
Measure your website readability!
<https://readability-score.com>