

Eingabe

```
eingabe = input("Eingabe: ")
# Datentyp ist immer(!) String
# evtl. Casting zu anderem
Datentyp
try:
    eingabe = int(eingabe)
except ValueError as err:
    print("Fehler!")
    print("Mel dung: {}".format(err))
```

Variablen

Kleinschreibung

Trennung mehrerer Wörter durch Unterstrich

Keine Ziffer am Beginn, nur Buchstabe oder Unterstrich

Sprechende Namen verwenden

Kommentare

Einzeilige Kommentare werden durch das Hash-Zeichen (Raute) eingeleitet.
Mehrzeilige Kommentare werden mit drei Anführungszeichen begonnen und müssen auch wieder mit drei Anführungszeichen beendet werden.

```
# Dies ist ein Kommentar
""" Dies ist ein
    langer Kommentar
"""
```

Fehlerbehandlung

```
try:
    # Anweisungen
except ExceptionType as err:
    # Anweisungen
    # Zugriff auf Fehlermeldung: err
```

ExceptionType immer explizit angeben

Operatoren

x+y	Addition
x-y	Subtraktion
x*y	Multiplikation
x/y	Division
x%y	Modulo
x**y	Exponentiation

Bedingungen

<	kleiner als	wert < 10
>	größer als	wert > 5
==	gleich	wert == 23
<=	kleiner gleich	5 <= 7
>=	größer gleich	23 >= 22
!=	ungleich	78 != 93
in	(enthalten) in	b in "Libelle"
not in	nicht (enthalten) in	y not in "Vogel"

Ausgabe

```
print("Hello World")
print("Hello", "John")
name = "John"
print("Hello", name)
print("Hello {}".format(name))
print() # erzeugt leere Zeile
und Zeilen umbruch
```

Wiederholungen / Schleifen

for-Schleife

```
container = [1,2,3]
for element in container:
    print(element)
```

while-Schleife

```
x = 5
while x > 0:
    print("x: {}".format(x))
    x-=1
```

for-Schleife hat enumerate-Erweiterung zum Mitzählen der Indexposition
Gefahr von Endlosschleifen beim Einsatz von while

Slicing bei Sequenzen

	0	1	2	3	4	5	6	7	8	9	10	11
	M	o	n	t	y		P	y	t	h	o	n
	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1
	[6:10]						[-12:-7]					



By Tobias Siebenlist (t7l)
cheatography.com/t7l/

Published 14th April, 2016.
Last updated 13th May, 2016.
Page 1 of 2.

Sponsored by CrosswordCheats.com
Learn to solve cryptic crosswords!
<http://crosswordcheats.com>

Laden und Verwenden eines Moduls		Arbeiten mit Strings		Arbeiten mit Listen	
Importanweisung	Befehlsaufruf	<code>len(mystr)</code>	Anzahl der Zeichen von mystr	<code>len(my_list)</code>	Anzahl der Elemente
<code>import modulname</code>	<code>modulname.methode()</code>	<code>mystr.lower()</code>	Umwandlung zu Kleinbuchstaben	<code>mylist[i]</code>	i-tes Element der Liste
<code>dir(modulname)</code>	zeigt alle Methoden eines Moduls an	<code>mystr.upper()</code>	Umwandlung zu Großbuchstaben	<code>mylist[i:j]</code>	Ausschnitt von i bis j
<code>help(methode)</code>	zeigt Hilfe für eine Methode an	<code>mystr.replace(old,new)</code>	old durch new ersetzen	<code>mylist[i:j:2]</code>	nur jedes zweite Element
Funktionen		<code>mystr.split(char)</code>	teilt an char auf, erzeugt Liste	<code>x in mylist</code>	True, wenn x in mylist ist
<code>def fname(param1, param2):</code>	<code> ergebnis = param1 + param2</code>	<code>mystr[1:5]</code>	Zeichen 1-5 ausschneiden	<code>mylist.append(x)</code>	x an mylist anhängen
<code> return ergebnis</code>		<code>mystr[1:5:2]</code>	nur jedes zweite Zeichen	<code>mylist[i] = x</code>	Element an Stelle i ersetzen
Bedingte Abfragen		Arbeiten mit Dateien		<code>"c".join(mylist)</code>	erzeugt String, c verbindet
<code>if bedingung:</code>	<code> # Anweisungen</code>	<code>with open(fname, mod) as var:</code>	<code> # Anweisungen, z.B.</code>	Arbeiten mit Dictionaries	
<code>elif andere_bedingung:</code>	<code> # Anweisungen</code>	<code> c = var.read()</code>	<code> cl = var.readlines()</code>	<code>len(mydict)</code>	Anzahl der Einträge
<code>else:</code>	<code> # Anweisungen</code>	<code> var.write(string_content)</code>	<code>read()</code> liest Inhalt komplett in einen String ein	<code>del mydict[key]</code>	löscht Schlüssel key
beliebig viele elif-Abschnitte möglich else erhält keine Bedingung Abarbeitung von oben nach unten		<code>readlines()</code> liest Inhalt zeilenweise in eine Liste ein	<code>write(string_content)</code> schreibt Stringinhalt in Datei	<code>list(mydict.keys())</code>	Liste aller Schlüssel
Datentypen				<code>list(mydict.values())</code>	Liste aller Werte
Integer	-25, 34			<code>list(mydict.items())</code>	Liste von Tupeln
Float	-2.53, 43.1			<code>key in mydict</code>	True, wenn key vorhanden
String	"Hello", "World"				
Boolean	True, False				
List	[value, ...]				
Tupel	(value, ...)				
Dictionary	{key:value, ...}				



By Tobias Siebenlist (t7l)
cheatography.com/t7l/

Published 14th April, 2016.
Last updated 13th May, 2016.
Page 2 of 2.

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
<http://crosswordcheats.com>