Python Sequenzen (C) 2020 T.Birnthaler OSTC GmbH
==========

Doku --> http://docs.python.org/3/library/stdtypes.html#sequence-types-list-tuple-range

Sequenzen sind eine "geordnete" Folge von Elementen, jedem Element wird eine Ganzzahl als "Index" zugeordnet (beginnend mit 0), Ã¼ber den es angesprochen werden kann. Der Index lÃ¤uft aufsteigend von 0 bis LÃ¤nge der Sequenz minus 1 bzw. absteigend von minus 1 bis minus LÃ¤nge der Sequenz:

```
  s1 = "hallo"                       s2 = ""     # String + leerer Strin
  t1 = ( True, "hallo", 3.14 )       t2 = ()     # Tupel + leerer Tupel
  l1 = [ 123, 456, 789 ]             l2 = []     # Liste + leere Liste

   01234  # Positive Indices
  "hallo" # Sequenz (str)
      -1  # Negative Indices
     -2   #
    -3    #
   -4     #
  -5      #
```

Die Zugriffsgeschwindigkeit auf ein bestimmtes Element Ã¼ber seinen Index ist unabhÃ¤ngig von der Sequenz-LÃ¤nge immer gleich schnell ("RANDOM ACCESS").

Die Zugriffsgeschwindigkeit auf einen bestimmten Element-Wert ist abhÃ¤ngig von der Sequenz-GrÃ¶Ã-^_e und immer langsam (erfordert das Durchsuchen der gesamten Sequenz)

Folgende Operationen werden von (fast) allen Sequenz-Typen (str, tuple, list, bytes, bytearray) unterstÃ¼tzt (sowohl mutable als auch immutable). S und T sind Sequenzen des gleichen Typs, I, J, K, N sind Ganzzahlen und E ist ein beliebiges Objekt, das die BeschrÃ¤nkungen der Sequenz S erfÃ¼llt.

Operation	Bedeutung
E in S	Element E in S vorhanden?
E not in S	Element E in S nicht vorhanden?
S + T	Verkettung von S und T
S * N N * S	Sequenz N-mal verketten
S[I]	Element mit Index I (Start bei 0)
S[I:J]	Slice von I bis J-1 (mit Schrittweite 1)
S[I:J:K]	Slice von I bis J-1 mit Schrittweite K
len(S)	LÃ¤nge von S (Anzahl Elemente)
min(S)	Kleinstes Element in S (gemÃ¤Ã-^_ Vergleich "<")
max(S)	GrÃ¶Ã-^_tes Element in S (gemÃ¤Ã-^_ Vergleich "<")
S.index(E[,I[,J]])	Index des 1. Elements gleich E in S (von Index I bis J-1; Standard: 0 und len(S)-1)
S.count(E)	Anzahl Vorkommen von Element E in S

Immutable Sequenzen

Folgende Operation wird von Immutable Sequenzen (z.B. str, tuple, bytes) zusÃ¤tzlich zu obigen unterstÃ¼tzt (S ist eine immutable Sequenz):

Operation	Bedeutung
hash(S)	Hashwert erstellen

Mutable Sequenzen

Folgende Operationen werden von mutable Sequenzen (z.B. list, bytearray) zusÃ¤tzlich zu den allgemeinen Sequenz-Operationen unterstÃ¼tzt (S ist eine mutable Sequenz, T ist eine Sequenz oder ein Iterator des gleichen Typs, I, J, K, N sind Ganzzahlen und E ist ein beliebiges Objekt, das die BeschrÃ¤nkungen der Sequenz S erfÃ¼llt):

Operation	Bedeutung
S[I] = E	Element an Position I durch E ersetzen
S[I:J] = T	Elemente von I..J-1 durch Inhalt von Iterable T ersetz.

del s[I:J]	Elemente von I..J-1 aus Liste entfernen
S[I:J:K] = T	Elemente s[I:J:K] durch Inhalt von Iterable T ersetzen
del s[I:J:K]	Elemente s[I:J:K] aus Liste entfernen
S.append(E)	E am Sequenz-Ende anhÃ¤ngen: S[len(S):len(S)] = [E]
S.extend(T) s += T	Elemente von T anhÃ¤ngen: S[len(S):len(S)] = T
S.insert(I, E)	Element E an Position I einfÃ¼gen: s[I:I] = [E]
E = S.pop([I])	Element an Position I (STD: 0) entfernen und zurÃ¼ck
S.remove(E)	Erstes Element mit Wert E entfernen (Suche!)
S *= N	Sequenz N-mal vervielfachen + verketten
S.reverse()	Reihenfolge der Elemente umdrehen
S.clear()	Sequenz leeren (alle Elemente entfernen): del S[:]
S2 = S.copy()	Shallow/Flache Kopie erzeugen: S[:]

HINWEIS: Indexwerte I und J sowie Schrittweite K dÃ¼rfen auch negativ sein.

list-Sequenz

Folgende Operation wird von Listen zusÃ¤tzlich zu den allgemeinen
Sequenz-Operationen und den Operationen fÃ¼r mutable Sequenzen unterstÃ¼tzt:

Operation	Bedeutung
L.sort(key,reverse)	Elemente aufsteigend sortieren (gemÃ¤ÃM-^_ Vergleich "<") (key=MAPFUNC, reverse=True -> absteigend)