```
# Eingabe
s = input(),   k = int(input('Bitte Zahl eingeben'))

# Schleifen, Bedingungen
while bool:,  while x is not None:, while a:,
for i in range(6): ...,  range(2,5),  range(10,-1,-1)
if bool1: .... elif bool2: ... else: ...

# eingebaute Funktionen
ord('a'),  chr(97)                      # Codierungszahl
abs(x)                                  # Betrag

# Strings
s = '',  s = 'ab', s = "ab"
s1 + s2, s.upper(), s.lower(),          # returns: String
s.replace(s1,s2), ''.join(a), s.join(a)
s.strip(), s.strip('?.!;')
s.islower(), s.isupper(), s.isdigit()   # returns: bool
s.isalpha(), s.isalnum(), s1 in s, s1 not in s
s.startswith(s1), s.endswith(s1),
s.count(s1), len(s)                     # returns: int
s.index(s1),  s.index(s1,i,j) # suche in [i,j)
s.find(s1),s.find(s1,i,j)     # suche in [i,j), ggf. = -1
s.split(), s.split(s1)                  # returns: list
for c in s: print(c)                    # Schleifen
for i in range(len(s)): print(s[i])

# formatierter String:
s = 'Stadt {} hat {} Einwohner'.format(s,x)
s = 'Wert = {:5.2f}'.format(x)   # 5 Stellen mit Dezimalpunkt
          #  davon 2 hinterm Komma

# Tuple
t = (), t = (1,), t = (1,2,3)
len(t), max(t), min(t)
t[0], t[1:3], t[0:len(t):2]          # indexing und slicing
x in t, x not in t                   # returns: bool
t1 + t2, tuple(s), tuple(a)          # returns: tuple
for x in t: print(x)                 # Schleifen
for i in range(len(t)): print(a[i])

# Listen
a = [] , a = [0,1,2,3], a = [0] * 5
len(a), max(a), min(a)
a[0], a[1:3], a[0:len(a):2]          # indexing und slicing
x in a, x not in a                   # returns: bool
a.count(x), a.index(x)               # returns: int
a.index(x,i,j) # suche in [i,j)
a1 + a2, list(t), list(s)            # returns: list
sorted(a), sorted(a, reverse = True)
x = a.pop()                          # gibt letztes Element zurück
x = a.pop(k)                         # gibt k-tes Element zurück
a.append(x), a.extend(a1)            # returns: None
a.remove(x), a.insert(i,x)
a.reverse()
a.sort(), a.sort(reverse=True)
for x in a: print(x)                 # Schleifen
for i in range(len(a)): print(a[i])

# Dictionaries
m = {}, m = {'a' : 1, 'b' : 2}, m['c'] = 3,  x = m['b']
len(m), del m['a'], k in m, k not in m
m.keys(), m.values(), m.items()   # kann in list/tuple umgewandelt werden
for k in m: ...   for v in m.values(): ...   for k,v in m.items(): ...

# Sets
s = set(), s = {1,2,3}
len(s), x in s, x not in s
s <= t, s < t # Teilmengen
s | t, s & t  # Vereinigung, Schnittmenge
s - t, s^t    # Differenz, Entweder-oder

# Comprehensions
a = [i*i for i in range(10) if (i*i)%2==0]
m = {x:len(x) for x in a}
s = {i*i for i in range(10)}

# Stack, Queue, Heap
st = aList, st.append(x), x = st.pop()
from collections import deque
q = deque(aList), q.append(x), x = q.popleft()
from heapq import heapify, heappop, heappush
h = [], heapify(aList), heappush(h,x), x = heappop(h)
len(st), len(q), len(h), while st: ... while q: ... while h: ...

# Zufall
import random
random.randint(0,20)        # int Zufallszahl ∈ [0,20]
random.random()             # zufällige float ∈ [0,1)
random.uniform(0,10)        # zufällige float ∈ [0,10]
```

```
# Verschiedenes
a = [(1,13),(4,5),(3,6)]
a.sort(key = lambda x:x[1], reverse=True)  #a sortieren nach y-Koordinate
if .... : raise RuntimeError("Fehler .....")

# Klassen
class Person:
    def __init__(self,alter):
        self.alter = alter
    def __lt__(self,other):   # Instanzen vergleichen
        self.alter < other.alter
class Student(Person):
    def __init__(self,alter,fach):
        super().__init__(alter)
        self.fach = fach

# unsere selbstgebastelten ADTs
Eintrag: inhalt, next
Liste: anf, pos - empty, endpos, reset, advance, elem,
insert (neues wird akt. Elem), delete (Nachfolger wird akt. Elem.)
Keller: tp - empty, push, top, pop
Schlange: head, tail - empty, enq, deq, front
Knoten: inhalt, links, rechts
Baum: wurzel - empty, value, left, right
Suchbaum: lookup, insert, delete
```