

Verkettete Liste

```
class Element:
    def __init__(self, wert):
        self.inhalt = wert
        self.naechster = None

element1 = Element(10)
element2 = Element(20)
element3 = Element(30)
element1.naechster = element2
element2.naechster = element3

def druckeListe(element):
    while element != None:
        print(element.inhalt)
        element = element.naechster

def anhaengen(wert):
    element=element1
    while element.naechster != None:
        element = element.naechster
    element.naechster=Element(wert)

def einfuegen(wert):
    element=element1
    while element.inhalt < wert:
        temp1=element
        element = element.naechster
    temp2=Element(wert)
    temp2.naechster=temp1.naechster
    temp1.naechster=temp2

anhaengen(40); anhaengen(50); anhaengen(60); einfuegen(55); einfuegen(15)
druckeListe(element1)
```

alternativ

```
def druckeListe(element):
    print(element.inhalt)
    if element.naechster != None:
        druckeListe(element.naechster)

def druckeListe(element):
    if element.naechster != None:
        druckeListe(element.naechster)
    print(element.inhalt)
```

Das Programm dient nur dem Einstieg. Es kann verbessert werden.

Verkettete Liste

```
class element:  
    def __init__(self, wert):  
        self.inhalt = wert  
        self.zeiger = None
```

```
root=element(1)  
zuletzt=root
```

```
def erzeuge():  
    global zuletzt  
    for n in range(2,11):  
        neu=element(n)  
        print(neu.inhalt)  
        zuletzt.zeiger=neu  
        zuletzt=neu
```

```
def drucke():  
    global root  
    while root.zeiger!= None:  
        print(root.inhalt)  
        root=root.zeiger  
    print(root.inhalt)
```

```
erzeuge()  
drucke()
```

Verkettete Liste

```
class element:  
    def __init__(self, wert):  
        self.inhalt = wert  
        self.zeiger = None
```

```
anfang=None
```

```
def erzeuge():  
    global anfang  
    for n in range(1,11):  
        neu=element(n)  
        print(neu.inhalt)  
        neu.zeiger=anfang  
        anfang=neu
```

```
def drucke():  
    global anfang  
    print(anfang.inhalt)  
    while anfang.zeiger!= None:  
        anfang=anfang.zeiger  
        print(anfang.inhalt)
```

```
erzeuge()  
drucke()
```

Binärbaum

```
class Element:
    def __init__(self, wert):
        self.inhalt = wert
        self.links = None
        self.rechts = None

element1 = Element(10)
element2 = Element(20)
element3 = Element(30)
element4 = Element(40)
element5 = Element(50)

element1.links = element2
element1.rechts = element3
element2.links = element4
element2.rechts = element5

def druckeListe(element):
    print(element.inhalt)
    if element.links != None:
        druckeListe(element.links)
    if element.rechts != None:
        druckeListe(element.rechts)

druckeListe(element1)
```

alternativ

```
def druckeListe(element):
    if element.links != None:
        druckeListe(element.links)
    if element.rechts != None:
        druckeListe(element.rechts)
    print(element.inhalt)
```

```
def druckeListe3(element,stufe):
    if element.rechts != None:
        druckeListe3(element.rechts,stufe+1)
    print(' '*stufe + str(element.inhalt))
    if element.links != None:
        druckeListe3(element.links,stufe+1)
```

```
druckeListe3(element1,0)
```