# Listen

Listen stellen wie Strings eine sequentielle Anordnung dar. Während Strings jedoch aus einer Sequenz von beliebigen Zeichen aufgebaut sind, bestehen Listen aus einer Sequenz von beliebigen Objekten, also z.B. Integer-Zahlen, Float-Zahlen, Strings oder wieder Listen. Anders als Strings sind sie änderbar. Die Indizierung fängt bei 0 an! Das letzte Element in einer Liste hat dementsprechend die "Adresse" Anzahl-1

In [1]:
```
# Wie wir in dem Beispiel sehen, werden Listen mit eckigen Klammern generiert.
# Die Elemente einer Liste werden durch Kommata getrennt.

liste=[1,2,3]
```

In [2]:
```
liste1=['Eins', 'Zwei', 'drei', 4,5]
```

In [3]:
```
liste1[0]
```

Out[3]:
```
'Eins'
```

In [4]:
```
liste1[1:]
```

Out[4]:
```
['Zwei', 'drei', 4, 5]
```

In [5]:
```
liste1*2
```

Out[5]:
```
['Eins', 'Zwei', 'drei', 4, 5, 'Eins', 'Zwei', 'drei', 4, 5]
```

# Grundlegende Methoden von Listen

Listen in Python sind allerdings aus zwei guten Gründen flexibler als Arrays in anderen Sprachen: Sie haben keine fixierte Größe und sie haben keine Beschränkung der Objektart (Datentyp).

In [6]:
```
#Erstellung der Liste[0,0,0]:

liste_ph=[0]*3
liste_ph
```

Out[6]:
```
[0, 0, 0]
```

In [7]:
```
# oder so...

liste_bb=[0,0,0]
liste_bb
```

Out[7]:
```
[0, 0, 0]
```

In [8]:
```
# Ersetzen Sie bitte in dieser verschachtelten Liste mit Hilfe einer Python Anwe
# um stattdessen "Auf Wiedersehen" zu sagen.
```

```
meine_liste=[1, 2, [3, 4, 'Hallo']]
```

In [9]:
```
meine_liste[2][2]='Auf Wiedersehen'
meine_liste
```

Out[9]:
```
[1, 2, [3, 4, 'Auf Wiedersehen']]
```

In [10]:
```
# Sortieren Sie bitte folgende Liste aufsteigend

neue_liste=[6, 5, 4, 5, 3]
sorted(neue_liste)
```

Out[10]:
```
[3, 4, 5, 5, 6]
```

**oder...**

In [11]:
```
# mit Hilfe der Funktion 'sort'

neue_liste.sort()
neue_liste
```

Out[11]:
```
[3, 4, 5, 5, 6]
```

In [12]:
```
# Ermitteln der Anzahl der Listenelemente mit Hilfe der Funktion 'len'

len(neue_liste)
```

Out[12]:
```
5
```

In [15]:
```
# Mit der Funktion list(range(start, ende, schrittweite)) können Sie unkomplizie
# in einem bestimmten Wertebereich enthalten.

liste_3=list(range(10,101,10))
liste_3
```

Out[15]:
```
[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
```

# Bitte erstellen Sie ein Programm mit folgender Ausgabe

Bitte geben Sie den Startwert ein: 2

Bitte geben Sie den Endwert ein: 10

Bitte geben Sie die Schrittweite ein: 2

Ergebnis: [2, 4, 6, 8, 10]

In [19]:
```
# Benutzereingaben abfragen
startwert = int(input("Gib den Startwert ein: "))
endwert = int(input("Gib den Endwert ein: "))
schrittweite = int(input("Gib die Schrittweite ein: "))

# Liste erstellen und ausgeben
zahlenliste = list(range(startwert, endwert + 1, schrittweite))
print("Ergebnis:", zahlenliste)
```

```
Ergebnis: [2, 4, 6, 8, 10]
```

In [25]:
```
# Mit dem 'in' Operator kann man festestellen, ob ein bestimmtes Element in eine

moegliche_farben=["rot","grün","blau"]
farbe=input("Bitte geben Sie eine Farbe ein: ")
if farbe not in moegliche_farben:
    print("Diese Farbe ist nicht enthalten!")
else:
    print(farbe,'ist enthalten!')
```

```
blau ist enthalten!
```

In [42]:
```
# Die Methoden sort() und reverse()

inhalt=[5,23,45,15,22,54]
print(inhalt)
inhalt.sort()
print(inhalt)
inhalt.reverse()
print(inhalt)
```

```
[5, 23, 45, 15, 22, 54]
[5, 15, 22, 23, 45, 54]
[54, 45, 23, 22, 15, 5]
```

In [27]:
```
# Gegeben ist folgende Liste:

inhalt=[23.5,12, [19, "Hallo"], 12,45]
print(inhalt)
```

```
[23.5, 12, [19, 'Hallo'], 12, 45]
```

In [28]:
```
# Die Methode append()

inhalt.append(100)
print(inhalt)
```

```
[23.5, 12, [19, 'Hallo'], 12, 45, 100]
```

In [29]:
```
# Nochmal die Methode append()

inhalt[2].append(200)
print(inhalt)
```

```
[23.5, 12, [19, 'Hallo', 200], 12, 45, 100]
```

In [37]:
```
# Anzahl der Zahl (hier 12) in der Liste

anzahl=inhalt.count(19)
print(anzahl)
```

```
0
```

In [38]:
```
# Die Methode remove()

inhalt.remove(23.5)
print(inhalt)
```

```
[12, [19, 'Hallo', 200], 12, 45, 100]
```

In [ ]:
```
# Die Methode index()
```

```
print(inhalt)
ind=inhalt[1].index(200)
print(ind)
```

In [ ]:
```
# Die Methode insert()

inhalt.insert(3,77)
print(inhalt)
```

In [ ]:
```
# Die Funktion max()

Liste1=[10,20,40,70]
max(Liste1)
```

In [ ]:
```
# Die Funktion min()

Liste1=[10,20,40,70]
min(Liste1)
```

In [39]:
```
Orte=["Xanten", "Essen", "Aachen", "Wuppertal"]
min(Orte)
```

Out[39]: 'Aachen'

In [40]:
```
max(Orte)
```

Out[40]: 'Xanten'

# Übungsaufgaben

Schreiben Sie bitte ein Python-Programm, das eine Liste von Zahlen erstellt und dann folgende Aufgaben durchführt:

- Füge eine Zahl am Ende der Liste hinzu.
- Füge eine Zahl an einer bestimmten Position in der Liste hinzu.
- Entferne eine Zahl aus der Liste.
- Überprüfe, ob eine bestimmte Zahl in der Liste vorhanden ist.
- Berechne die Summe aller Zahlen in der Liste.

Gegeben ist folgende Liste: zahlen_liste = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]