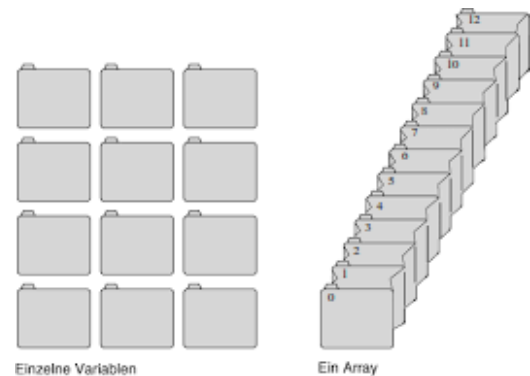


## Dynamische Reihung

Zur Speicherung und Verarbeitung vieler gleichartig strukturierter Daten, verwendet man nicht einzelne Variablen, sondern sogenannte Datenstrukturen. Eine dieser Datenstrukturen ist die dynamische Reihung (dynamisches Array). Eine Reihung besteht aus Elementen, in welchen Daten (desselben Typs, also z. B. nur ganze Zahlen) abgespeichert werden. Die einzelnen Elemente kann man sich somit wie Variablen mit einem Wert vorstellen. Sie werden jedoch nicht über einen Variablennamen referenziert, sondern durch den Namen der Reihung und ihrer Position in der Reihung, dem sogenannten Index. Ist die Größe der Reihung variabel und der Zugriff auf die Reihung frei wählbar, spricht man von einer dynamischen Reihung.



## Methoden für dynamische Reihungen

Auf dynamische Reihungen können wir die nachfolgenden Methoden anwenden:

Methode	Beschreibung
<code>DynArray()</code>	Eine leere dynamische Reihung wird angelegt.
<code>isEmpty()</code>	Gibt den Wert wahr zurück, wenn die Liste leer ist, (ansonsten den Wert falsch).
<code>append(&lt;value&gt;)</code>	Fügt ein Element mit dem Wert <value> ans Ende der dynamischen Reihung an.
<code>getItem(&lt;index&gt;)</code>	Gibt den Wert des Elements an der Position <value> zurück.
<code>insertAt(&lt;index&gt;,&lt;value&gt;)</code>	Ein neues Element mit dem Wert <value> wird in die dynamische Reihung an die Position <index> eingefügt, die nachfolgenden Elemente werden jeweils um eine Position nach hinten verschoben.
<code>setItem(&lt;index&gt;,&lt;value&gt;)</code>	Der Wert des Elements an der Position <index> wird auf den Wert <value> gesetzt.
<code>delete(index)</code>	Das Element an der Position <index> wird entfernt. Alle nachfolgenden Elemente werden um eine Position nach vorne geschoben.
<code>getLength()</code>	Die Anzahl der Elemente der dynamischen Reihung wird zurückgegeben.

## Aufgaben

### Aufgabe 1

Erläutern Sie den Beispielcode unten und was dessen Ausführung bewirkt. Stellen Sie dazu den Inhalt der dynamischen Reihung nach der Ausführung jeder Codezeile dar.

```
reihung = DynArray()
if reihung.isEmpty():
    print("Hallo")
reihung.append(5)
reihung.append(1)
reihung.insertAt(1,4)
reihung.setItem(2,0)
reihung.delete(0)
if not reihung.isEmpty():
    print(reihung.getItem(reihung.getLength()-2))
```

### Aufgabe 2

Gegeben sei eine dynamische Reihung, in welcher das Alter einer Gruppe von Personen enthalten ist. Diese wird z. B. durch den nachfolgenden Programmcode erzeugt.

```
alter = DynArray()
alter.append(21)
alter.append(20)
alter.append(33)
alter.append(30)
alter.append(27)
alter.append(28)
alter.append(31)
alter.append(19)
```

- Implementieren Sie eine Methode *GetDynArray()*, mit der alle Elemente der dynamischen Reihung untereinander ausgegeben werden.
- ✂ **Hinweis:** Speichern Sie diese und auch die nachfolgenden Methoden in einer Datei *methoden.py*, damit Sie diese in Zukunft in anderen Programmen weiter verwenden können.
- Implementieren Sie eine Methode *GetMinimum()*, mit der das kleinste Element der dynamischen Reihung ausgegeben wird.
- Implementieren Sie eine Methode *GetMaximum()*, mit der das größte Element der dynamischen Reihung ausgegeben wird.

### Aufgabe 3

- Stellen Sie einen der vorgestellten Sortieralgorithmen in schriftlich verbalisierter Form dar.
- Überführen Sie den Algorithmus anschließend in ein Struktogramm und stellen Sie dieses dar.
- Implementieren Sie den Algorithmus anschließend in Python und testen Sie ihn an der dynamischen Reihung aus Aufgabe 2.