

## Übungsaufgaben: Array



**Aufgabe:** Bearbeiten Sie die Übungsaufgaben zur Datenstruktur - Array.



**Hinweise:**

- siehe Informationsblatt **Datenstruktur: Array** ([1\\_Informationsblatt\\_Array.pdf](#)).
- Falls Sie Hilfestellungen benötigen, löschen Sie die blauen Felder an den jeweiligen Aufgaben.

### Aufgabe 1: Erstellen eines Arrays

Erstellen Sie ein Array, mit dem Namen **reihe** und den Werten 10, 20, 30, 40, 50.

Geben Sie das Array als Ganzes aus.

**Hilfestellung:**

Ein Array kann in Python mit eckigen Klammern [] erstellt werden. Die Elemente werden durch Kommas getrennt. Um das Array auszugeben, können Sie die **print()**-Funktion verwenden.

#### Python Code (Lösung)

```
reihe = [10,20,30,40,50]
print(reihe)
```

### Aufgabe 2: Zugriff auf Elemente eines Arrays

Geben Sie das dritte Element, des Arrays **reihe** aus.

**Hilfestellung:**

In Python beginnt die Zählung von Array-Elementen bei 0. Um auf ein bestimmtes Element zuzugreifen, nutzen Sie die Syntax **array[index]**, wobei **index** die Position des Elements im Array angibt.

#### Python Code (Lösung)

```
reihe = [10,20,30,40,50]
print(reihe[2])
```

### Aufgabe 3: Array-Durchlauf mit Schleife

Geben Sie jedes Element, des Arrays **reihe** mithilfe einer Schleife einzeln aus.

#### **Hilfestellung:**

Mithilfe einer for-Schleife kann jedes Element eines Arrays der Reihe nach ausgegeben (durchlaufen) werden. Siehe im Informationsblatt **Zugriff auf die Elemente eines Arrays mit einer for-Schleife**

#### Python Code (Lösung)

```
reihe = [10,20,30,40,50]
for i in range(len(reihe)):
    print(reihe[i])
```

### Aufgabe 4: Länge eines Arrays ermitteln

Geben Sie die Länge (Anzahl der Elemente) des Arrays **reihe** aus.

#### **Hilfestellung:**

Die Länge eines Arrays können Sie mit der Funktion **len()** ermitteln. Diese gibt die Anzahl der Elemente im Array zurück.

#### Python Code (Lösung)

```
reihe = [10,20,30,40,50]
laenge = len(reihe)
print(laenge)
```

### Aufgabe 5: Einfügen eines Elements in ein Array

Schreiben Sie ein Programm, das die Zahl 60 am Ende des Arrays 10, 20, 30, 40, 50 hinzufügt und das neue Array ausgibt.

#### **Hilfestellung:**

Mit der Methode `append()` kannst du ein neues Element am Ende eines Arrays hinzufügen.  
Beachte, dass `append()` nur ein einzelnes Element auf einmal hinzufügt.

#### **Python Code (Lösung)**

```
reihe = [10,20,30,40,50]
reihe.append(60)
print(reihe)
```

### Aufgabe 6: Verändern von Elementen in einem Array

Erstelle ein Array mit den Zahlen 1, 2, 4, 4, 5 und ändere anschließend den Wert des dritten Elements auf 3. Geben Sie das neue Array als Ganzes aus.

#### **Hilfestellung:**

Verwende die Methode append(), um ein neues Element am Ende des Arrays hinzuzufügen. Um ein Element im Array zu verändern, greifst du über seinen Index zu und weist einen neuen Wert zu (z. B. array[index] = neuer\_wert). Denke daran, dass die Indizierung in Python bei 0 beginnt.

#### **Python Code (Lösung)**

```
reihe_neu = [1,2,4,4,5]
reihe_neu[2] = 3
print(reihe_neu)
```

### Aufgabe 7: Arrayelemente ausgeben

Erstellen Sie ein Array **zahlen**, das die Zahlen 3, 7, 1, 9 und 5 enthält. Greifen Sie auf das zweite und vierte Element des Arrays zu und speichere sie in den Variablen **zweites\_element** und **viertes\_element**.

In der Konsole / Kommandozeile soll folgende Ausgabe erscheinen:

```
Kommandozeile ×

>>> %Run -c $EDITOR_CONTENT

Zweites Element: 7
Viertes Element: 9

>>>
```

### Python Code (Lösung)

```
# Array erstellen
zahlen = [3, 7, 1, 9, 5]
# Auf das zweite und vierte Element zugreifen
zweites_element = zahlen[1]
vierthes_element = zahlen[3]

print("Zweites Element:", zweites_element)
print("Viertes Element:", vierthes_element)
```

### Aufgabe 8: Summe berechnen

Schreiben Sie ein Programm, das die Summe eines Arrays, welches Zahlen als Werte enthält, berechnet und zurückgibt.

Beispiel:

```
Kommandozeile x
>>> %Run -c $EDITOR_CONTENT
Das Array hat die Werte: [3, 7, 1, 9, 5]
Die Summe des Arrays ist: 25
>>>
```

### Python Code (Lösung)

```
# Array erstellen
zahlen = [3, 7, 1, 9, 5]
# Variable
summe = 0
# Summe berechnen
for i in range(len(zahlen)):
    summe = summe + zahlen[i]

print("Das Array hat die Werte:", zahlen)
print("Die Summe des Arrays ist:", summe)
```

### Aufgabe 9: Durchschnitt berechnen

Schreiben Sie ein Programm, das den Durchschnitt eines Arrays, welches Zahlen als Werte enthält berechnet und zurückgibt.

Beispiel:

```
Kommandozeile x
>>> %Run -c $EDITOR_CONTENT
Das Array hat die Werte: [3, 7, 1, 9, 5]
Der Durchschnitt des Arrays ist: 5.0
>>>
```

### Python Code (Lösung)

```
# Array erstellen
zahlen = [3, 7, 1, 9, 5]

# Variable
summe = 0
durchschnitt = 0
anzahl = len(zahlen)

# Summe berechnen
for i in range(len(zahlen)):
    summe = summe + zahlen[i]

durchschnitt = summe/anzahl

print("Das Array hat die Werte:", zahlen)
print("Der Durchschnitt des Arrays ist:", durchschnitt)
```

## Aufgabe 10: Maximum ermitteln

Schreiben Sie ein Programm, das aus einem Array, welches positive Zahlen als Werte enthält, den größten Wert ermittelt und ausgibt.

Beispiel:

```
Kommandozeile x
>>> %Run -c $EDITOR_CONTENT
Das Array hat die Werte: [3, 7, 1, 9, 5]
Der größte Wert des Arrays ist: 9
>>>
```

### Python Code (Lösung)

```
# Array erstellen
zahlen = [3, 7, 1, 9, 5]

# Variable
max = 0

# max ermitteln
for i in range(len(zahlen)):
    if zahlen[i] > max:
        max = zahlen[i]

print("Das Array hat die Werte:", zahlen)
print("Der größte Wert des Arrays ist:", max)
```

### Aufgabe 11: Anzahl der Tage mit Fieber zählen

Über 7 Tage wurden von einem Patienten die folgenden Körpertemperaturen gemessen:  
36.5, 37.0, 38.2, 37.8, 36.9, 38.5, 37.3

Schreiben Sie ein Programm, das die Messungen in einem Array speichert und die Anzahl der Tage zählt, an denen die Temperatur über **38°C** lag. Speichere das Ergebnis in der Variable **fieber\_tage** und gib es am Ende aus.

Beispiel:

```
Kommandozeile ×

>>> %Run -c $EDITOR_CONTENT

Temperaturen [36.5, 37.0, 38.2, 37.8, 36.9, 38.5, 37.3]
Anzahl der Tage mit Fieber: 2

>>>
```

**Hinweis:**

- Verwende eine Schleife und erhöhe fieber\_tage um 1, wenn eine Temperatur über 38°C ist.

### Python Code (Lösung)

```
temperaturen = [36.5, 37.0, 38.2, 37.8, 36.9, 38.5, 37.3]
fieber_tage = 0
for temp in temperaturen:
    if temp > 38.0:
        fieber_tage += 1

print(f"Anzahl der Tage mit Fieber: {fieber_tage}")
```