Arrays

```
float noten[] = {1, 1.5, 1.25};
```

noten[0]	1	4 Byte
noten[1]	1.5	4 Byte
noten[2]	1.25	4 Byte

Die Grösse der einzelnen Elemente entsprechen dem Platzbedarf des Elementtyps.

```
noten[3] = 1.75;
```

Der Zeile Nummer 3 wird ein Wert zugewiesen. Der vorherige Inhalt der Zeile wird überschrieben.

```
float wert = noten[2];
```

Der Wert der Zeile 2 wird ausgelesen.

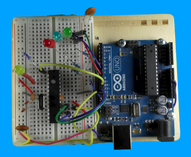
```
float wert = noten[5];
```

Es wird ein Wert ausgelesen, der gar nicht im Array vorhanden ist. Das ist nicht verboten, gibt aber einen völlig sinnlosen Wert zurück!

```
noten[5] = 3;
```

Es wird ein Wert in einen Speicherbereich geschrieben, der nicht zum Array gehört. Das wird vom Programm nicht bemerkt und kann zu Fehlern oder Abstürzen führen!

Die erste Zeile eines Arrays ist immer die Nummer 0.



Arrays

```
float noten[5] = {1, 1.5, 1.25};
```

noten[0]	1	4 Byte
noten[1]	1.5	4 Byte
noten[2]	1.25	4 Byte
noten[3]	undefiniert	4 Byte
noten[4]	undefiniert	4 Byte

Die Grösse der einzelnen Elemente entsprechen dem Platzbedarf des Elementtyps.

Nicht initialisierte Zeilen haben einen undefinierten Inhalt.

```
noten[3] = 1.75;
```

Der Zeile Nummer 3 wird ein Wert zugewiesen. Der vorherige Inhalt der Zeile wird überschrieben.

```
float wert = noten[2];
```

Der Wert der Zeile 2 wird ausgelesen.

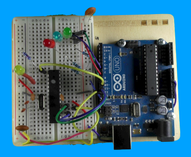
```
float wert = noten[5];
```

Es wird ein Wert ausgelesen, der gar nicht im Array vorhanden ist. Das ist nicht verboten, gibt aber einen völlig sinnlosen Wert zurück!

```
noten[5] = 3;
```

Es wird ein Wert in einen Speicherbereich geschrieben, der nicht zum Array gehört. Das wird vom Programm nicht bemerkt und kann zu Fehlern oder Abstürzen führen!

Die erste Zeile eines Arrays ist immer die Nummer 0.



Übergabe an Funktionen

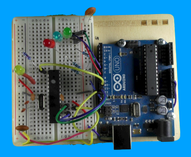
```
int zahl = 5;

void print(int i) {
   Serial.println(i);
   i = 7;
}

void setup() {
  print(zahl);
  Serial.println(zahl);
}
```

Hier wird der Wert der Variablen **zahl** an die Funktion **print** übergeben und in die neue lokale Variable **i** kopiert. Wir können die Variable **i** in der Funktion ändern, das hat aber keinen Einfluss auf die Variable **zahl**.

Ausgabe des Programms:

5

5



Übergabe an Funktionen

```
float noten[] = {1, 1.5, 1.25};
```

Zeiger (Pointer)

noten[0]	1	4 Byte
noten[1]	1.5	4 Byte
noten[2]	1.25	4 Byte

Das Array wird im Speicher angelegt und belegt dort einen bestimmten Speicherplatz.

```
float noten[] = {1, 1.5, 1.25};

void print(float n[]) {
   Serial.println(n[1]);
   n[1] = 2;
}

void setup() {
  print(noten);
  Serial.println(noten[1]);
}
```

Hier wird die Speicher - Adresse des Arrays übergeben. Eine solche Adresse ist immer 4 Byte lang. Normalerweise spricht man von einem Zeiger oder Pointer. Wichtig ist, dass dabei keine Kopie der Daten angelegt wird. Beide Variablen (**noten** und **n**) benutzen denselben Speicherplatz. Deshalb wirken sich Änderungen innerhalb der Funktion auch auf das Array ausserhalb aus.

Ausgabe des Programms:

1.5

2