

# Variablen in Java

---

## 1. Einführung in Variablen in Java

In Java sind Variablen grundlegende Bausteine, die zur Speicherung von Daten dienen, die im gesamten Programm abgerufen und manipuliert werden können. Variablen in Java sind Container, die Datenwerte enthalten. Jede Variable hat einen spezifischen Datentyp, der definiert, welche Art von Wert sie speichern kann. Diese Datentypen reichen von primitiven Typen wie Ganzzahlen, Gleitkommazahlen, Zeichen und Booleans bis hin zu komplexen Objekten, die aus Klassen erstellt werden. Java erzwingt eine strikte Typprüfung, was bedeutet, dass jede Variable vor der Verwendung mit einem spezifischen Datentyp deklariert werden muss. Dies stellt die Datenintegrität sicher und reduziert Laufzeitfehler.

Wichtige Punkte:

- **Deklaration:** Jede Variable muss vor der Verwendung mit einem Datentyp deklariert werden.
- **Initialisierung:** Variablen können bei der Deklaration initialisiert werden.
- **Gültigkeitsbereich (Scope):** Variablen in Java haben einen Gültigkeitsbereich, der bestimmt, wo im Code auf sie zugegriffen werden kann.
- **Lebensdauer:** Die Lebensdauer einer Variable wird durch ihre Deklaration bestimmt und besteht nur innerhalb dieses Bereichs.

## 2. Beispielcode für Variablen in Java

Hier ist ein Beispiel in Java, das zeigt, wie verschiedene Variablentypen deklariert und initialisiert werden.

```
public class VariableExamples {  
    public static void main(String[] args) {  
        // Numerische Variablen  
        int age = 25; // Integer-Typ  
        double salary = 45000.75; // Double-Typ  
  
        // Textvariable  
        String name = "Alice"; // String-Typ für Text  
  
        // Boolean-Variable  
        boolean isStudent = true; // Boolean-Typ  
  
        // Variablen ausgeben  
        System.out.println("Name: " + name);  
        System.out.println("Alter: " + age);  
        System.out.println("Gehalt: " + salary);  
        System.out.println("Ist Student: " + isStudent);  
    }  
}
```

## 3. Datentypen in Java (JDK 23)

Die folgende Tabelle gibt einen umfassenden Überblick über die in JDK 23 verfügbaren Datentypen, kategorisiert nach Art. Jeder Datentyp hat spezifische Grenzen und beansprucht eine bestimmte Menge an Speicher, wie in der Tabelle dargestellt.

Kategorie	Datentyp	Gültige Grenzen	Speichergröße	Beispieldeklaration in Java
Numerisch	<code>byte</code>	-128 bis 127	1 Byte	<code>byte b = 100;</code>
	<code>short</code>	-32,768 bis 32,767	2 Bytes	<code>short s = 20000;</code>
	<code>int</code>	-2,147,483,648 bis 2,147,483,647	4 Bytes	<code>int i = 500000;</code>
	<code>long</code>	-9,223,372,036,854,775,808 bis 9,223,372,036,854,775,807	8 Bytes	<code>long l = 5000000000L;</code>
	<code>float</code>	Ungefähr $\pm 3.4e-038$ bis $\pm 3.4e+038$	4 Bytes	<code>float f = 3.14f;</code>
	<code>double</code>	Ungefähr $\pm 1.7e-308$ bis $\pm 1.7e+308$	8 Bytes	<code>double d = 3.14159;</code>
Text	<code>char</code>	Einzelnes Unicode-Zeichen	2 Bytes	<code>char c = 'A';</code>
	<code>String</code>	Zeichenfolge	Variiert (Objekt)	<code>String s = "Hello World";</code>
Boolean	<code>boolean</code>	<code>true</code> oder <code>false</code>	1 Bit	<code>boolean flag = true;</code>

## 4. Ungültige Variablenzuweisungen

In Java hat jeder Datentyp spezifische Anforderungen und Einschränkungen. Hier sind einige häufige ungültige Variablenzuweisungen, die in Java Fehler verursachen würden:

- **Ungültige `float`-Zuweisung (fehlendes 'f' Suffix):**

```
float f = -35.0; // Fehler: Float-Literal erfordert ein 'f'-Suffix für den float-Typ
```

**Korrektur:**

```
float f = -35.0f;
```

- **Außerhalb des Bereichs `byte`-Zuweisung:**

```
byte b = 150; // Fehler: Wert überschreitet den Bereich von byte (-128 bis 127)
```

**Korrektur:**

```
byte b = 127;
```

- **Außerhalb des Bereichs `short`-Zuweisung:**

```
short s = 40000; // Fehler: Wert überschreitet den Bereich von short  
(-32,768 bis 32,767)
```

**Korrektur:**

```
short s = 32767;
```

- **Ungültige `char`-Zuweisung (mehrere Zeichen):**

```
char c = 'AB'; // Fehler: char kann nur ein einzelnes Zeichen speichern
```

**Korrektur:**

```
char c = 'A';
```

- **Ungültige `boolean`-Zuweisung (numerischer Wert):**

```
boolean flag = 1; // Fehler: boolean kann nur true oder false sein
```

**Korrektur:**

```
boolean flag = true;
```

## 5. Übungen mit verschiedenen Variablentypen

1. **Übung 1:** Deklariere eine `int`-Variable namens `year` und setze sie auf das aktuelle Jahr. Gib den Wert der Variable aus.
2. **Übung 2:** Deklariere eine `double`-Variable namens `temperature` und setze sie auf einen Wert mit einer Dezimalstelle. Gib die Temperatur aus.

3. **Übung 3:** Erstelle eine `String`-Variable namens `city` und setze sie auf den Namen deiner Heimatstadt. Zeige den Stadtnamen an.
4. **Übung 4:** Definiere eine `boolean`-Variable namens `isAvailable` und setze sie auf `false`. Gib den Verfügbarkeitsstatus aus.
5. **Übung 5:** Deklariere eine `char`-Variable namens `grade` und setze sie auf einen Buchstaben zwischen 'A' und 'F'. Gib die Note aus.