

Einfache Datentypen

Ein Datentyp ist eine Angabe, welche Datenwerte eine Variable speichern kann, bzw. wie ein Variablenwert zu interpretieren ist.

Man unterscheidet

- einfache ("primitive") Datentypen und
- Referenztypen (Verweise auf Objekte (s.u.))

Bei den einfachen Datentypen unterscheidet man insgesamt 8 verschiedene Typen in 3 Gruppen:

1 Numerische Datentypen

Mit numerischen Datentypen kann gerechnet werden. Die jeweils angegebenen Wertebereiche werden vom Compiler überprüft. Es ist nicht möglich, Werte zuzuweisen, die außerhalb des gültigen Wertebereichs liegen.

Man unterscheidet:

1.1 Ganzzahlentypen

Ganzzahlentypen können nur ganze Zahlen aufnehmen, also keine Zahlen mit Nachkommastellen.

Je nach Datentyp ist der zur Verfügung gestellte Speicherplatz unterschiedlich groß, daraus folgt, dass mehr oder weniger große Zahlen gespeichert werden können.¹

Wertebereich von Ganzzahlentypen

Typ	Länge	Wertebereich
byte	8 Bit	-2^7 bis 2^7-1 (-128 bis 127)
short	16 Bit	-2^{15} bis $2^{15}-1$ (-32768 bis 32767)
int	32 Bit	-2^{31} bis $2^{31}-1$ (-2.147.483.648 bis 2.147.483.647)
long	64 Bit	-2^{63} bis $2^{63}-1$ (-9.223.372.036.854.775.808 bis 9.223.372.036.854.775.807)

Beispiel:

```
public class VarDeklaration
{
    public static void main( String[] args)
    {
        int iZahl;                                // Deklaration
        iZahl = 10;                                // Initialisierung
        System.out.println("Der Wert von iZahl ist "+ iZahl); // Verwendung
    }
}
```

¹ Ganze Zahlen werden in Java in der binären Zweierkomplement Darstellung gespeichert

1.2 Gleitpunkttypen

Gleitpunkttypen (auch Fließkommatypen genannt) können Zahlen mit Nachkommastellen aufnehmen. Die Stellengenauigkeit ist dabei je nach Datentyp verschieden:

Wertebereich von Gleitpunkttypen²

Typ	Länge	Genauigkeit	Wertebereich
float	32 Bit	7 Stellen	-3,40282347 10 ³⁸ bis 3,40282347 10 ³⁸
double	64 Bit	15 Stellen	- 1,797693138462315750 10 ³⁰⁸ bis 1,797693138462315750 10 ³⁰⁸

Beispiel:

```
double dZahl = 3.567;      // Nachkommastellen werden mit dem englischen Dezimalpunkt  
                           // abgetrennt  
double dZahl2 = -1.345e5;  // Exponenten werden mit E oder e abgetrennt  
float fZahl = 3.567f;      // Achtung, Suffix f erforderlich
```

2 Zeichen-Datentyp: char

Eine Variable vom Typ char kann genau ein Zeichen speichern. Der char-Typ verwendet zur binären Darstellung der Zeichen das UNICODE-Format UTF-16, so dass für ein Zeichen 16 Bit benötigt werden. Damit können alle Unicode-Zeichen dargestellt werden.³

Auf der Basis des Typs char können weitere Texttypen, wie z.B. Zeichenketten, die sog. Strings, gebildet werden.

Typ	Speicherung	Wertebereich
char	2 Byte	alle UniCode-Zeichen bzw. ganze Zahlen im Bereich 0 ... 65535

Beispiel:

```
char cZeichen = 'a'; // Zeichen werden in einfache Hochkommata eingeschlossen
```

3 Logischer Datentyp: boolean

Um reine zweiwertige Aussagen zu verarbeiten verwendet man den Typ boolean. Variablen von diesem Typ können nur die Werte true oder false annehmen, also keine Zwischenwerte.

Typ	Speicherung	Wertebereich
boolean	1 Byte	true, false

Beispiel:

```
boolean bGefunden = true;
```

² Gleitpunktzahlen werden in Java in der binären Mantissen-Exponent-Darstellung der Norm IEEE 754 gespeichert

³ Genau genommen ist char ein Ganzzahl Datentyp der positive Zahlen im Bereich 0 .. 65535 aufnehmen kann. Umgekehrt können auch in den Ganzzahl Datentypen short, int und long Unicode-Zeichen gespeichert werden.

4 Ergänzung: der „Datentyp“ String

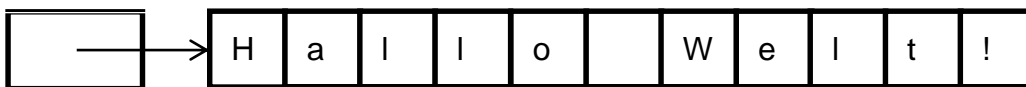
Für die Aufnahme von Zeichenketten ("Texte") existiert in Java die Klasse String. Es handelt sich hierbei jedoch nicht um einen einfachen Datentyp, sondern eine Klasse im Sinne der objektorientierten Programmierung.

Da jedoch die Verarbeitung von Texten in vielen Java-Programmen eine wichtige Rolle spielt, nimmt diese Klasse eine gewisse Sonderstellung ein und die Entwickler von Java haben ihm gewisse Vereinfachungen gegenüber anderen Klassen eingeräumt. So ist es möglich, eine String-Variable wie eine einfache Variable zu deklarieren, was mit anderen Objekten von Klassen nicht geht. Näheres dazu beim Thema objektorientierte Programmierung.

Beispiel:

```
String sText = "Hallo Welt"; // Strings müssen immer in doppelten Hochkommata  
                             // eingeschlossen werden
```

sText



sText ist eine sogenannte Verweis- bzw. Referenzvariable. Strings werden in Objekten gespeichert. Auf Objekte wird, im Gegensatz zu einfachen Variablen, immer indirekt über eine Referenzvariable zugegriffen.