

Java

Cheat Sheet / Spickzettel



Primitive Datentypen

Datentyp	Größe	Bereich
<i>byte</i>	8	-128..127
<i>short</i>	16	-32,768..32,767
<i>int</i>	32	-2,147,483,648.. 2,147,483,647
<i>long</i>	64	-9,223,372,036,854,775,808.. 9,223,372,036,854,775,807
<i>float</i>	32	3.4e-0.38.. 3.4e+0.38
<i>double</i>	64	1.7e-308.. 1.7e+308
<i>char</i>	16	Gesamtes Unicode-Zeichenset
<i>Boolean</i>	1	Wahr, Falsch

Operatoren

Operator-Typ	Operatoren
Rechenoperator	+, -, *, /, %, ?
Zuweisung	=, +=, -=, *=, /=, %=, &=, ^=, =, <<=, >>=, >>>=
Bitwise	^, &,
Logisch	&&,
Relational	<, >, <=, >=, ==, !=
Shift	<<, >>, >>>
Ternär	?:
Unär	++x, -x, x++, x-, +x, -x, !, ~

Variablen

Es gibt 3 Variablentypen:

1. Lokale Variable
2. Instanz-Variable
3. Statische Variable

```
{public | private} [static] type varName [= ausdruck | wert];
```

Methoden

1. Methoden-Initialisierung
2. Methoden-Aufruf

```
{public | private} [static] {type | void} methodenName(arg1, ..., argN) {Befehle;}
```

Datentypen konvertieren

// Erweitern (byte<short<int<long<float<double)

```
int i = 10; // int-> long
long l = i; // automatische Typenkonvertierung
```

// Eingrenzen

```
double d = 10.02;
long l = (long)d; // explizite Typisierung
```

// Numerische Werte zu String konvertieren

```
String str = String.valueOf(value);
```

// String zu numerischem Wert konvertieren

```
int i = Integer.parseInt(str);
double d = Double.parseDouble(str);
```

Benutzereingabe

Java bietet drei Möglichkeiten, eine Eingabe vom Benutzer/von der Konsole zu übernehmen:

1. Verwendung der BufferedReader-Klasse
2. Scanner-Klasse verwenden
3. Konsolen-Klasse

// **BufferedReader**

```
BufferedReader reader = new BufferedReader(new  
InputStreamReader(System.in));  
String name = reader.readLine();
```

// **Scanner**

```
Scanner in = new Scanner(System.in);  
String s = in.nextLine();  
int a = in.nextInt();
```

// **Console**

```
String name = System.console().readLine();
```

Grundlagen eines Java-Programms

Ein Basisprogramm in Java besteht aus mindestens den folgenden Komponenten:

1. Klassen & Objekte
2. Methoden
3. Variablen

// **Standard Java-Programm**

```
public class HalloWelt {  
    public static void main(String[ ] args) {  
        System.out.println("Hallo LerneProgrammieren");  
    }  
}
```

Java-Programm kompilieren

Du musst dein Java-Programm unter dem Namen der Klasse speichern, welche die main()-Methode enthält (**klassenName.java**)

Ruft den Compiler mit dem javac-Befehl auf:

```
javac klassenName
```

Führe das Programm mit dem jeweiligen Code aus:

```
java klassenName
```

Schleifen

```
// for  
for (bedingung) {  
    Befehle;  
}
```

```
// for each  
for (int i: einArray) {}
```

```
// while  
while (bedingung) {  
    Befehle;  
}
```

```
// do while  
do {  
    Befehle;  
} while(bedingung)
```

If-Abfragen

```
//if Befehl  
if (bedingung) {  
    Befehle;  
}
```

```
//if-else Befehl  
if (bedingung) {  
    Befehle;  
} else {  
    Befehle;  
}
```

```
//switch Befehl  
switch (var) {  
    case 1:  
        Befehle;  
        break;  
    default:  
        Befehle;  
        break;  
}
```

Eindimensionale Arrays (1-D)

```
// Initialisierung  
type[] varName= new type[size];
```

```
// Deklaration  
type[] varName= new type[] {wert1, wert2,...};
```

Mehrdimensionale Arrays (2-D)

```
// Initialisierung  
type[][] varName = new dataType[zeile][spalte];
```

```
// Deklaration  
type[][] varName = {{wert1, wert2...},{wert1, wert2...}..};
```

Strings erstellen

```
String str1 = "Hallo"; // String als Literal erstellen
```

```
String str2 = new String("LerneProgrammieren"); // String mit „new“ Keyword erstellen
```

Schleifen-Beispiele

Erstellen einer Fibonacci-Reihe:

```
for (i = 1; i <= n; ++i){
    System.out.print(t1 + " + ");
    int summe = t1 + t2;
    t1 = t2;
    t2 = summe;
}
```

Erstellen eines Pyramidenmusters:

```
k = 2*n - 2;

for(i=0; i<n; i++){
    for(j=0; j<k; j++){
        System.out.print(" ");
    }
    k = k - 1;
    for(j=0; j<=i; j++ ){
        System.out.print("* ");
    }
    System.out.println();
}
```

Array-Beispiele

Array mit Zufallswerten erstellen:

```
double[] arr = new double[n];

for (int i=0; i<n; i++){
    a[i] = Math.random();
}
```

Größten Wert in einem Array finden

```
double max = 0;

for(int i=0; i<arr.length(); i++) {
    if(a[i] > max) max = a[i];
}
```

Werte in einem Array umkehren:

```
for(int i=0; i<(arr.length())/2; i++) {
    double temp = a[i];
    a[i] = a[n-1-i];
    a[n-1-i] = temp;
}
```

Häufig benötigte String-Methoden

str1 == str2 // Adresse vergleichen;

String neuerString = str1.equals(str2); // Zwei Werte vergleichen

String neuerString = str1.equalsIgnoreCase() // Werte vergleichen, ignoriert Case

neuerString = str1.length() // String-Länge berechnen

neuerString = str1.charAt(i) // Zeichen extrahieren (i)

neuerString = str1.toUpperCase() // String in Großbuchstaben

neuerString = str1.toLowerCase() // String in Kleinbuchstaben

neuerString = str1.replace(alterWert, neuerWert) //Suchen und Ersetzen

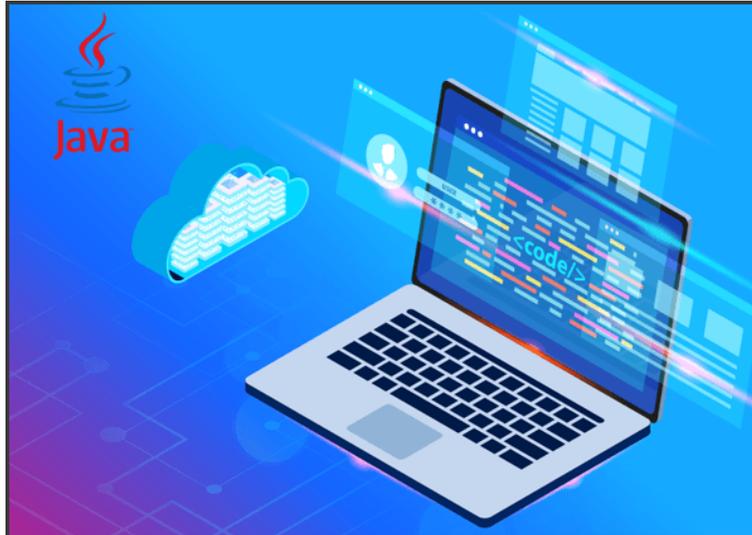
neuerString = str1.trim() // Leerzeichen entfernen

neuerString = str1.contains("Wert123"); // Auf vorhandenen Wert prüfen

neuerString = str1.toCharArray(); // String in Array konvertieren

neuerString = str1.isEmpty(); //Prüft, ob String leer ist

neuerString = str1.endsWith(); // Prüft, ob String mit einem bestimmten Suffix endet



Java Schnellstart

Lerne eine der beliebtesten Sprachen und programmiere Apps, Webseiten sowie Spiele.

- Ohne Vorkenntnisse
- Keine Installationen notwendig

Jetzt noch leichter **Java lernen** mit dem [LerneProgrammieren Java-Schnellstart](#).
Ein Kurs für alle, die frustfrei Java lernen und Apps programmieren wollen.

- Ohne Vorkenntnisse
- Bequem im Browser lernen
- Baue echte Apps, Spiele und Projekte

[Hier klicken](#)

