

Was ist Java

Java läuft auf mehr als *3 Billionen* Maschinen.

Es wird für *Google's Android OS*, *verschiedene Desktop/Web Anwendungen*, und vieles mehr benutzt.

Java ist wie jede Programmiersprache sehr groß, hat eine riesige Dokumentation und tausende Tutorial-Videos/Bücher! **Das heißt dass man Java nicht "Über das Wochenende" lernen kann!**

Hallo, Welt!

Das erste Programm in jeder Programmiersprache ist das sogenannte *Hello World*, welches einfach einen Text beim Start des Programmes ausgibt.

In Java sähe dies wie folgt aus:

```
class MyClass {
    public static void main(String[] args) {
        System.out.println("Hello World");
    }
}
```

In Java muss jede Zeile Code in einer **Klasse** sein, in unserem Fall heißt Sie: **MyClass**.

Auch hat jedes Programm eine Start-Methode namens **main()**

Was dieses `public static void` und `String[] args` bei der Methode bedeutet, erfahrt Ihr in "Variablen & Methoden".

Mit **System.out.println([Nachricht]);** gibt man etwas in die Konsole aus.

Alle Klassen/Methoden sowie andere Fluss-Kontroll Strukturen müssen innerhalb einer `{}` sein. Alles andere (*Statements*) muss ein **;(Semikolon)** am Ende haben.

Kommentare

Kommentare in Java sehen wie folgt aus:

```
// Einzeiliger Kommentar
/* Mehrzeiliger
 * Kommentar
 */
/** Mehrzeiliger Dokumentations-
 * Kommentar
 */
```

Variablen

Variablen speichern den Wert ihres Typen ab.

Jede Variable hat einen Namen, einen Typen (zB. `String`) sowie ein Wert (zB. `"Hey"`).

Hier ein Beispiel:

```
String name = "Dave";
int age = 16;
```

`name = 16;` > **Fehler:** Die Variable `name` ist vom Typ `String`, also kann sie auch nicht auf eine Zahl gesetzt werden!

Primitive Variablen, welche nicht gesetzt (initialisiert) wurden, haben einen gewissen Standard-Wert. (Meißt 0)

Bei Objekt-Variablen wie `String` oder `Scanner` ist der Standard-Typ **null**.

D.h. wenn man mit einem Objekt was machen will, welches `null` ist, wird eine **NullPointerException** geworfen.

Primitive Datentypen

Name	Größe	Beispiel
byte	8 bits	Sub-ID / Auswahl / Stack-Größe / Alter
short	16 bits	Item-ID
int	32 bits	
long	64 bits	Zeit-angabe (<i>Siehe System.currentTimeMillis()</i>)
boolean	1 bit	Ja/Nein
char	16 bits unicode	Buchstabe
float	32 bit decimal	Fließkomma-Zahlen (Yaw/Pitch-Blickrichtung)
double	64 bit decimal	Große Komma-Zahlen

Primitive Operatoren

- + Addition (Zusammen-Rechnen)
- Subtraktion (Weg-Rechnen)
- / Dividation (Wie oft `x` in `y` reinpasst)
- * Multiplikation (`x` mal `y`)
- % Modulo (Rest von `x / y`)

So ziemlich jede Sprache bietet diese Primitiven Rechnungs-Typen. Auf Klammer-Punkt-Strich wird geachtet!



String

length()	Gibt die Länge der Zeichenkette
s.charAt(index)	Gibt den Buchstaben an der Stelle <code>index</code>
s.substring(start, end)	Gibt die Zeichenkette mit <code>start</code> -weggeschnittenen Zeichen.
s.toLowerCase()	Gibt die Zeichenkette, in der alle Buchstaben klein sind
s.toUpperCase()	Gibt die Zeichenkette, in der ALLE BUCHSTABEN GROß sind
s.indexOf(c)	Gibt den Ort (Index) des ersten Buchstabens namens <code>c</code>
s.split(regex)	Gibt eine String-Array zurück, wessen Abschnitte durch <code>regex</code> bestimmt/geteilt wurde
s.trim()	Gibt die Zeichenkette ohne Leere-Zeichen zurück
s.equals(s2)	Gibt zurück, ob <code>s1</code> gleich ist wie <code>s2</code>

String (cont)

s.equals(s2)	Gibt zurück, ob <code>s1</code> gleich ist wie <code>s2</code> (Groß/Kleinschreibung wird nicht beachtet)
--------------	---

String ist ein Datentyp für eine **Zeichenkette** in Java, welches schon viele nützliche Funktionen bereitlegt.

Ein String kann ganz einfach mit `"[Text]"` erstellt werden:

```
String helloWorld = "Hello
Welt";
```

Nutzer-Eingabe

```
class NutzerEingabe {

    public static void main(String[] args){
        // Neue Instanz von "Scanner"
        erstellen
        Scanner myScanner = new
        Scanner();
        System.out.println("Gib irgend
        etwas ein!");

        // Wartet bis der Scanner
        einen Input bekommt, fängt
        diesen ab und setzt ihn auf die
        Variable "line"
        String line = myScanner.nextLine();
        System.out.println("Deine
        Eingabe: " + line);
    }
}
```

<https://docs.oracle.com/javase/7/docs/api/java/util/Scanner.html>

