

Variablen (Prinzip, Deklaration, Initialisierung, Zuweisung)

- Variablen sind Datenbehälter
- in diese können Werte gespeichert werden
- zwei wichtige Informationen, die eine Variable enthalten muss
 - Datentyp (bestimmt was in den Datenbehälter reingelegt werden darf)
 - Bezeichner (Name der Variable)

//Initialisierung einer Variable

```
int age = 23;
```

//Deklaration einer Variable

```
int age2;
```

//Werte an eine Variable zuweisen

```
age2 = 30;
```

- der Datentyp muss nur bei der Deklaration bzw. bei der Initialisierung mit angegeben werden (also nur einmal pro Variable)
- danach wird nur über den Bezeichner auf die Variable zugegriffen

Primitive Datentypen - Ein Überblick

-Datentypen geben Computer genaue Auskunft über den Inhalt von Variablen
=> ohne Datentypen könnte der Computer nicht vernünftig arbeiten

Übersicht über die 8 primitive Datentypen

Primitiver Datentyp	Größe	Wertebereich
boolean	undefiniert	true/false
char	16 Bit	0 bis +65.535
byte	8 Bit	-128 bis +127
short	16 Bit	-32.768 bis 32.767
int	32 Bit	-2^{31} bis $2^{31} - 1$
long	64 Bit	-2^{63} bis $2^{63} - 1$
float	32 Bit	1,40239846E-45 bis 3,40282347E+38
double	64 Bit	4,94065645841246544E-324 bis 1,79769131486231570E+308

Die Ganzzahl Datentypen (byte, short, int, long)

```
//Die Ganzzahl Datentypen
```

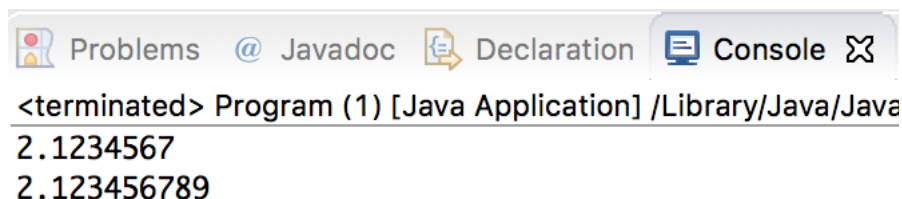
```
byte variable1;  
short variable2;  
int variable3;  
long variable4;
```

Die Fließkomma Datentypen (float, double)

```
//Die Fließkomma Datentypen
```

```
float variable1 = 2.123456789f;  
double variable2 = 2.123456789;
```

```
System.out.println(variable1);  
System.out.println(variable2);
```



```
<terminated> Program (1) [Java Application] /Library/Java/Java  
2.1234567  
2.123456789
```

- unterschiedliche Ausgabe aufgrund der unterschiedlichen Genauigkeit
- double besitzt doppelten Speicherplatz von float
- => höhere Genauigkeit als float

-alles was die Genauigkeit von float überschreitet, wird einfach abgeschnitten (siehe Ausgabe)

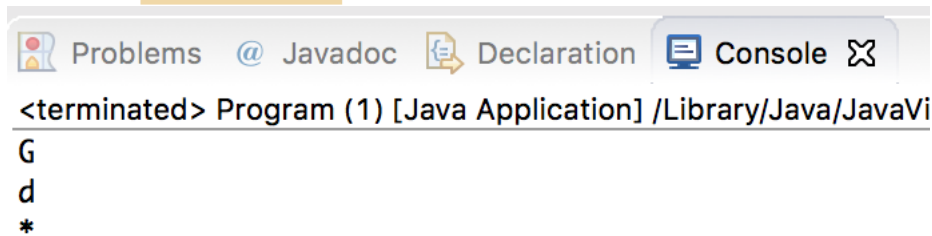
Der Datentyp char (Zeichenkonstanten)

-wird zum abspeichern einzelner Zeichen verwendet

Beachte: Zeichen werden intern nicht als Zeichen, sondern in Form einer Zahl abgespeichert (Stichwort: Addition von zwei Zeichen ist möglich)

Wichtig: einfache hochgestellte Anführungsstriche verwenden

```
char variable1 = 'G';  
char variable2 = 'd';
```

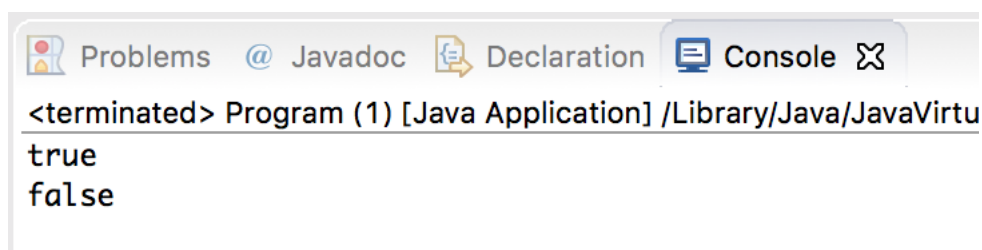


```
<terminated> Program (1) [Java Application] /Library/Java/JavaVi  
G  
d  
*
```

Der Datentyp boolean (Wahrheitswerte)

```
//Datentyp boolean  
boolean variable1 = true;  
boolean variable2 = false;
```

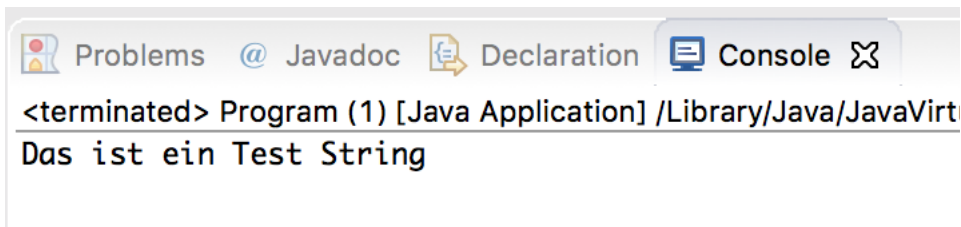
```
System.out.println(variable1);  
System.out.println(variable2);
```



```
<terminated> Program (1) [Java Application] /Library/Java/JavaVirtu  
true  
false
```

-wird erst im Modul Kontrollstrukturen bzw. Operatoren relevant

```
//Die Klasse String  
String variable1 = "Das ist ein Test String";  
System.out.println(variable1);
```



Wichtig: doppelte hochgestellte Anführungsstriche verwenden

- in Strings können mehrere Zeichen gespeichert werden und nicht nur ein Zeichen wie im Datentyp char
- komplexer Datentyp => erkennt man daran, dass Typ großen Anfangsbuchstaben besitzt
- Strings können auch verkettet werden

```
String variable1 = "Heute ist ein toller ";  
String verketteterString = variable1 + "Tag";  
  
System.out.println(verketteterString);
```

