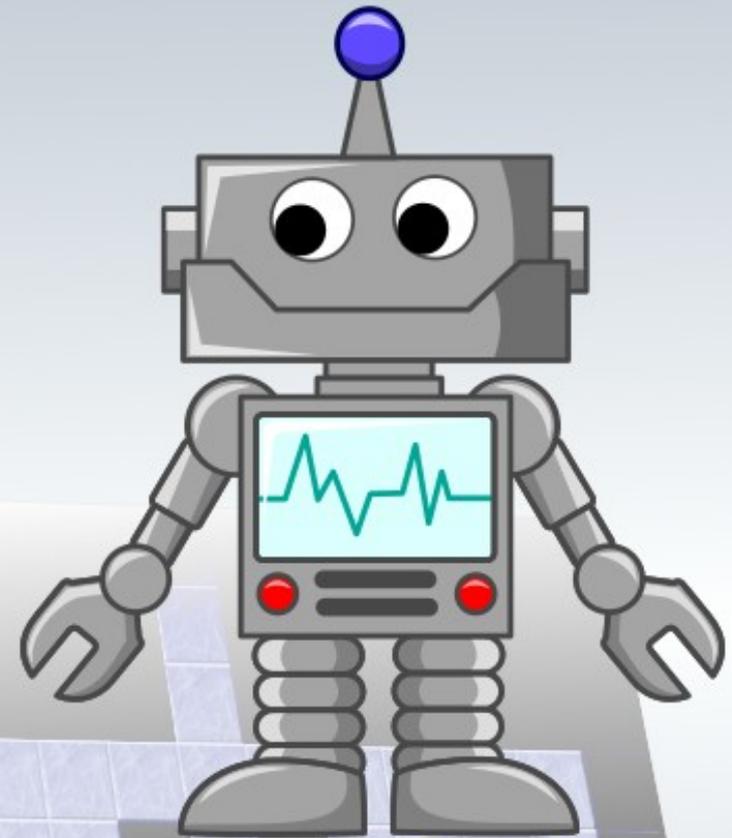


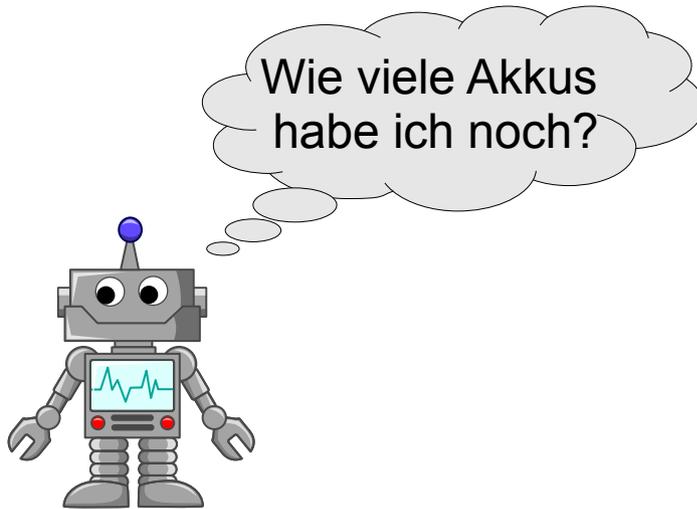
# Einführung in die Programmierung

## Variablen





Möchte man sich beim Programmieren etwas merken, benutzt man Variablen:

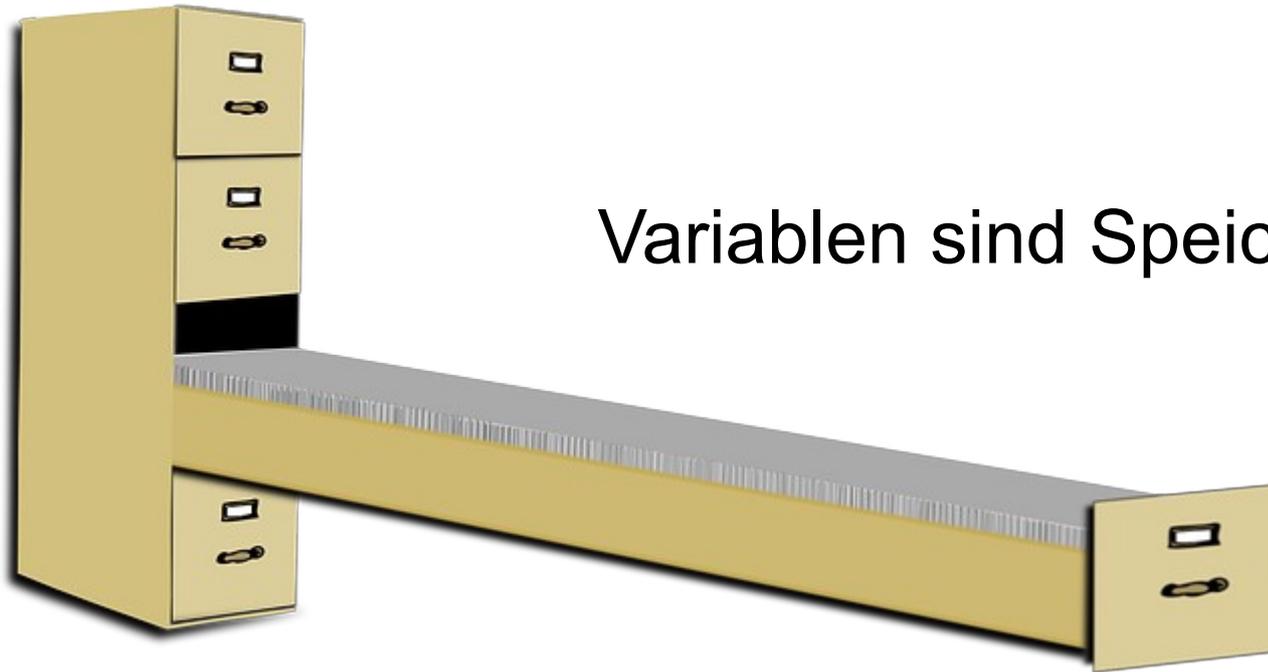


```
int anzahl;  
anzahl = 1;
```



```
boolean an;  
an = true;
```

Roboterszenario: Bildquellen der Objekte: siehe Bildnachweise.html



Variablen sind Speicher!

Bildquelle: Schublade, OpenClipart-Vectors (Lizenz: Pixabay Lizenz), via Pixabay, URL:  
<https://pixabay.com/de/kabinett-schublade-datei-157671/>

Man kann das, was man sich merken möchte, z.B. in einer Schublade speichern.  
Diese beschriftet man dann mit einem (aussagekräftigen) Namen.



In Schubladen kann man so einiges rein tun. Doch nicht immer passt das, was man speichern möchte, auch in jede beliebige Schublade!



Quelle Zechnall



Daher gibt es für unterschiedliche Zwecke den jeweils passenden Behälter => Typ der Variable.



Quelle Eimer, Efraimstochter (Lizenz: Pixabaylizenz), via Pixabay, URL: <https://pixabay.com/de/photos/eimer-strand-sand-sandeleimer-138880/>



Quelle Löffel, Monouso (Lizenz: Pixabaylizenz) via Pixabay, URL: <https://pixabay.com/it/photos/cucchiaio-design-degustazione-1199434/>



Quelle Markt, Hans (Lizenz: Pixabaylizenz) via Pixabay, URL: <https://pixabay.com/de/photos/markt-lebensmittel-früchte-äpfel-59074/>



<b>Typ</b>	<b>von</b>	<b>bis einschließlich</b>	<b>Größe</b>
<b>byte</b>	-128	127	8 Bit
<b>short</b>	-32.768	32.767	16 Bit
<b>int</b>	-2.147.483.648	2.147.483.647	32 Bit
<b>long</b>	-9.223.372.036.854.775.808	9.223.372.036.854.775.807	64 Bit
<b>boolean</b>	false	true	1 Bit
<b>float</b>	1.401298464324817E-45	3.4028234663852886E38	32 Bit
<b>double</b>	4.9E-324	1.7976931348623157E308	64 Bit
<b>char</b>	\u0000	\uFFFF	16 Bit
<b><u>String</u></b>	Gruppe von Zeichen, Wort oder Text (Satz)		variabel



## Deklaration

int zahl;

double kommaZahl;

String satz;

boolean aussage;

char buchstabe;

Syntax:

Variablentyp

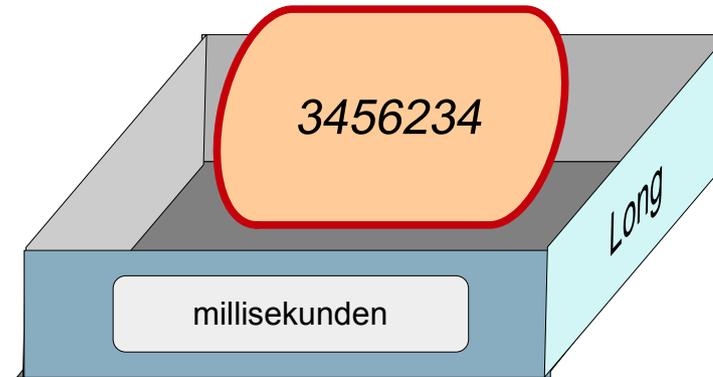
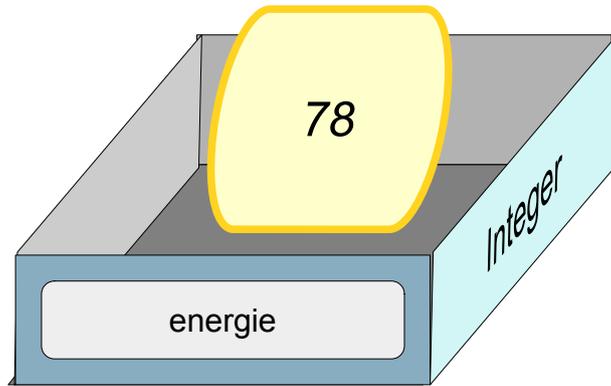
Variablenname  
(mit Kleinbuchstaben beginnen)

## Initialisierung

```
zahl = 5;
```

```
satz = "Hallo";
```

```
kommaZahl = 3.4;
```





## Lokale Variablen

- zum Speichern von Zwischenergebnissen
- gelten nur innerhalb einer Methode
- am Anfang einer Methode deklarieren und initialisieren

## Objektvariablen = Attribute = Felder

- zum dauerhaften Speichern
- gelten innerhalb des ganzen Objekts
- am Anfang der Klasse deklarieren
- im Konstruktor initialisieren

## Parameter

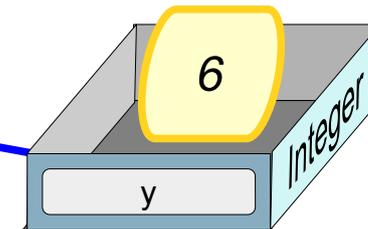
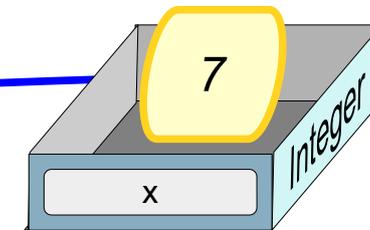
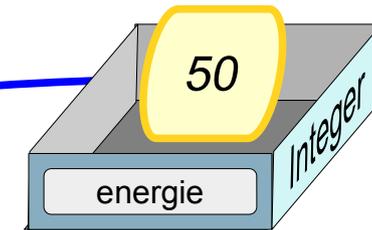
- zur Übergabe von Zusatzinformationen an Methoden
- gelten nur innerhalb der Methode
- im Kopf der Methode deklarieren (in ()-Klammer)
- werden beim Methodenaufruf initialisiert



## Objektvariablen = Attribute = Felder

AB1 : AB1

private int energie	50
private ArrayList<Gegenstand> inventar	
private GreenfootImage originalImage	
<b>int x</b>	<b>7</b>
int y	6
private int mySequenceNumber	90
private int lastPaintSequenceNumber	91
int rotation	0
World world	
private GreenfootImage image	
private Object data	
private Rect boundingRect	
private int[] boundingYs	



Roboterszenario: Bildquellen der Objekte: siehe Bildnachweise.html