

# JavaC - CheatSheet

## Allgemein

### Primitive Datentypen

<b>boolean</b>	Boolescher Wert	true, false
<b>char</b>	Textzeichen (UTF16)	'a', 'B', '0', 'é' etc.
<b>byte</b>	Ganzzahl (8 Bit)	-128 bis 127
<b>short</b>	Ganzzahl (16 Bit)	-32'768 bis 32'767
<b>int</b>	Ganzzahl (32 Bit)	-2 <sup>31</sup> bis 2 <sup>31</sup> -1
<b>long</b>	Ganzzahl (64 Bit)	-2 <sup>63</sup> bis 2 <sup>63</sup> -1, 1L (L Suffix)
<b>float</b>	Gleitkommazahl(32 Bit)	0.1f, 2e4f (2*10 <sup>4</sup> )
<b>double</b>	Gleitkommazahl(64 Bit)	0.1, 2e4

**Überlauf/Unterlauf** ist in Java definiert. Zählt einfach, je nach dem, unten oder oben weiter. Bei Gleitkommazahlen wird 2\*1e308 zu POSITIVE\_INFINITY.

### Explizite Typkonversation

Nur C-Style Cast: (int)3.5; → 3

**TO DO: evtl. noch Arrays und Mehrdimensionale Arrays**

### TO DO: evtl. komplexe Datentypen

```
package ch.hsr.javac.ex1.task2;

public class ImageProcessing {
    //inverts the color of each pixel
    public static int[][] invert(int[][] pixels) {
        for(int row=pixels.length-1 ; row >= 0 ; --row){
            for(int col=pixels[0].length-1 ; col >= 0 ; --col){
                pixels[row][col] *= -1;
            }
        }
        return pixels;
    }

    public static int[][] rotate(int[][] pixels) {
        int[][] rotated = new int[pixels[0].length][pixels.length];
        for(int row = 0 ; row <= pixels.length-1 ; ++row){
            for(int col = 0 ; col <= pixels[0].length-1 ; ++col){
                rotated[col][pixels.length-1-row] = pixels[row][col];
            }
        }
        return rotated;
    }

    public static int[][] mirror(int[][] pixels) {
        for(int row = 0 ; row <= pixels.length - 1 ; ++row){
            for(int col = 0 ; col < pixels[0].length / 2 ; ++col){
                int temp = pixels[row][col];
                pixels[row][col] = pixels[row][pixels[0].length-1-col];
                pixels[row][pixels[0].length-1-col] = temp;
            }
        }
        return pixels;
    }

    string str = "I'm a string";
```