

Codierung von Grafiken



- Grafiken werden meist als **Pixelgrafik** gespeichert
Dateiformate: jpg, png, gif, pbm ...
- Die Bilder bestehen dabei aus einzelnen Bildpunkten
- Die Angabe PPI bei Smartphones oder Bildschirmen gibt die Pixeldichte an

Nachteil: Pixelgrafiken können nicht ohne Verluste skaliert werden

Lösung: Vektorgrafiken wie SVG, Speicherung von Formen und deren gegenseitiger Lage



Bei **verlustbehafteten** Bildformaten gehen gegenüber **verlustfreien** Bildformaten Bildinformationen zugunsten des Speicherplatzes verloren. Zusätzlich ist bei beiden Formatarten oftmals Kompression möglich.

<i>Verlustbehaftete Pixelgrafikformate</i>	<i>Verlustfreie Pixelgrafikformate</i>
GIF: Maximal 256 Farben, ermöglicht Transparenz und Animationen	BMP: Maximal 16,7 Mio. Farben, keine (effiziente) Kompression
JPEG: Maximal 16,7 Mio. Farben, verschiedene Kompressionsstufen wählbar	PNG: Bis zu 281,4 Billionen Farben, verlustfreie Kompression, Transparenz.

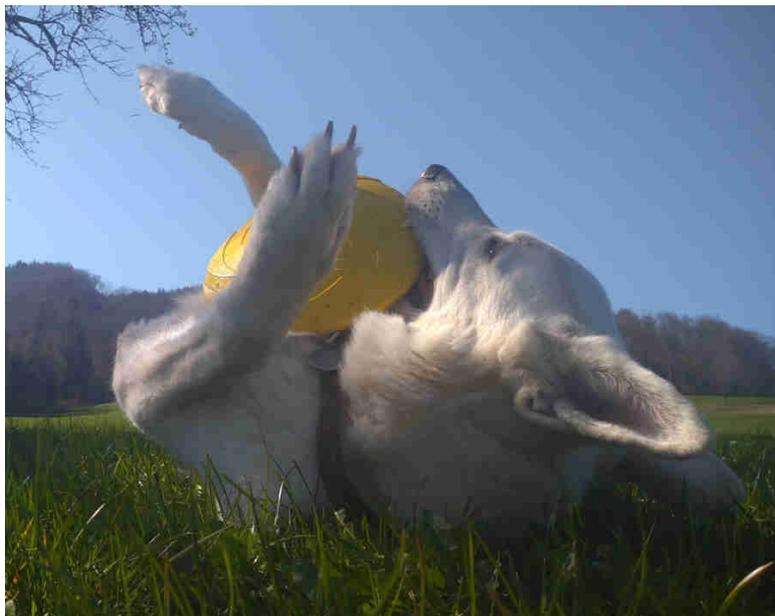
Binäre Codierung: Bilder



JPEG Q 80 / 203kb



JPEG Q 50 / 51kb



JPEG Q 20 / 20kb



PNG C9 / 880kb

Einfaches Pixel-Bildformat für s/w: Portable-Bitmap-Format (PBM)

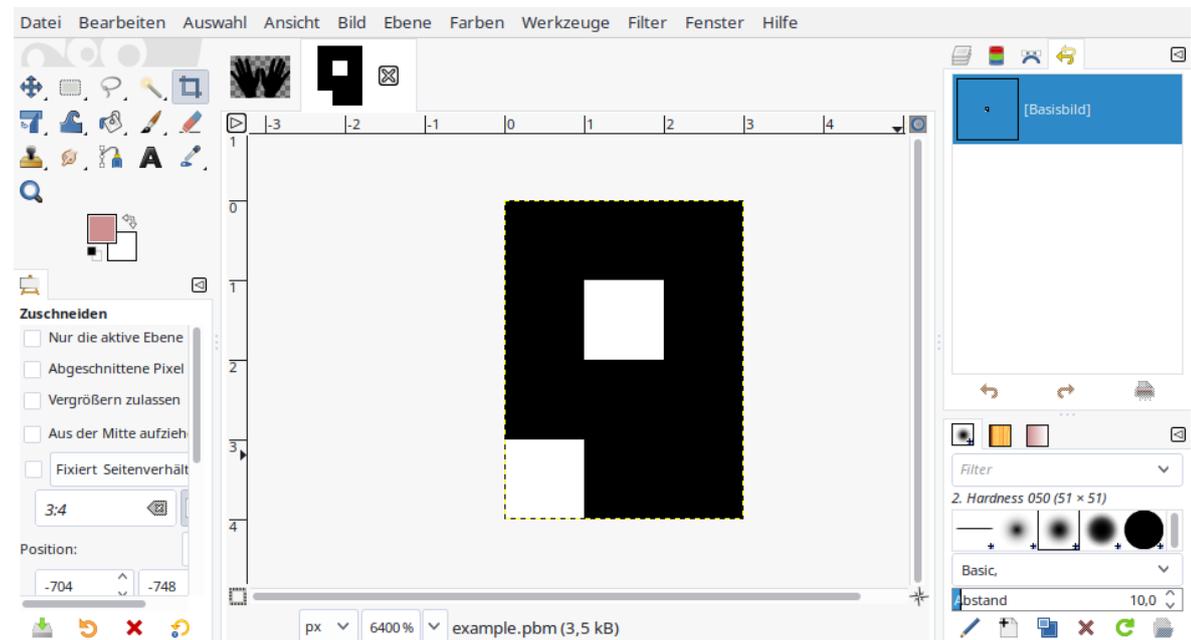
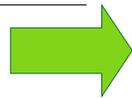
Kann mit einem Texteditor bearbeitet werden:

- Lege eine Datei example.pbm, beginnend mit P1
- Anschließend: „Spaltenzahl Zeilenzahl“, z.B. 3 4
- Damit hat man das Raster festgelegt: 12 Pixel in 4 Zeilen mit je 3 Spalten. An jede Pixelstelle schreibt man jetzt eine 1 für schwarz und eine 0 für weiss.

```
Bearbeiten Auswahl Ansicht Gehe zu Projekte LSP-Client Sitzung
neu Öffnen Speichern Speichern unter Rückgängig
b.pbm x
srv > samba > schools > default-school > teachers > sbel > b.pbm
1 | P1 3 4 1 1 1 1 0 1 1 1 0 1 1|
2 |
```

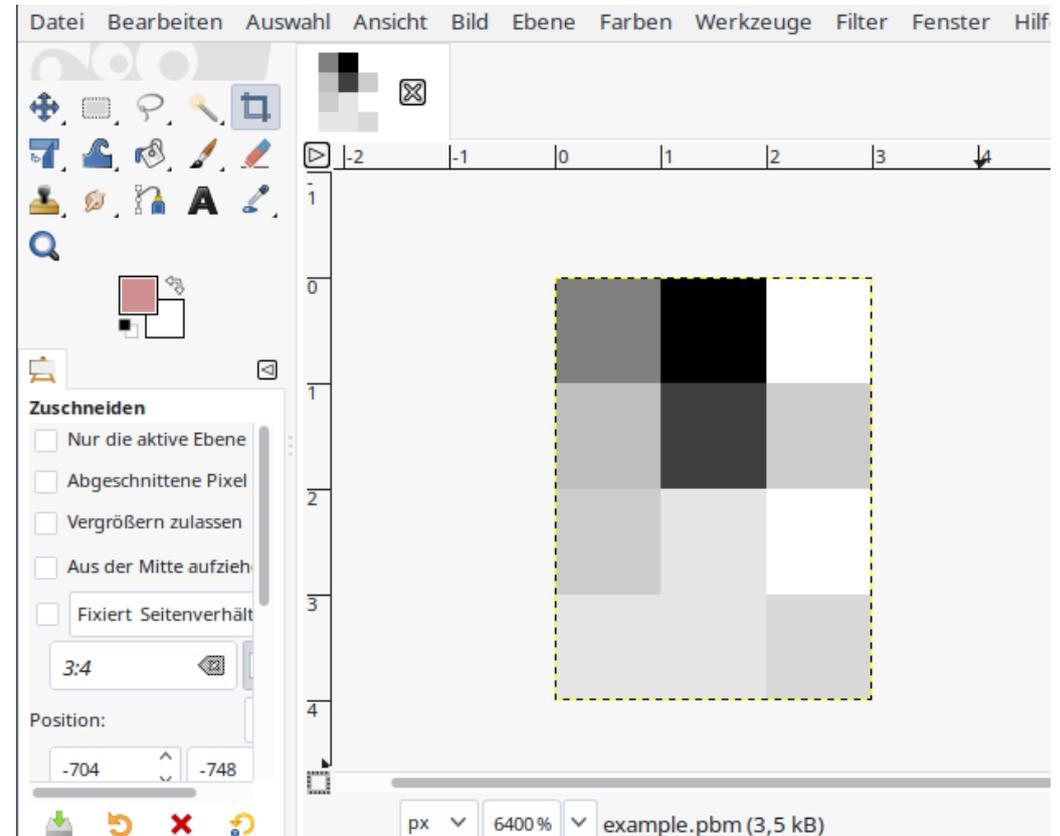
Oder:

```
b.pbm x
srv > samba > schools > defau
1 | P1
2 | 3 4
3 | 1 1 1
4 | 1 0 1
5 | 1 1 1
6 | 0 1 1
7 |
```



Analog Graustufenbilder

- Start P2 → Graustufen,
- Spaltenzahl Zeilenzahl, z.B. 3 4
- Maximalwert für „Weiss“ (1-255)
- Ins Raster an jede Pixelstelle schreibt man jetzt einen Wert zwischen 0 (schwarz) und dem gewählten Maximalwert (weiss).



```
srv > samba > schools > default-sch  
1 P2  
2 3 4  
3 20  
4 10 0 20  
5 15 5 16  
6 16 18 20  
7 18 18 17
```



Exkurs : Farben - es gibt mehrere Modelle, hier nur RGB

Prinzip: Additive Farbmischung aus **R**ot, **G**ruen und **B**lau.

Vordergrund

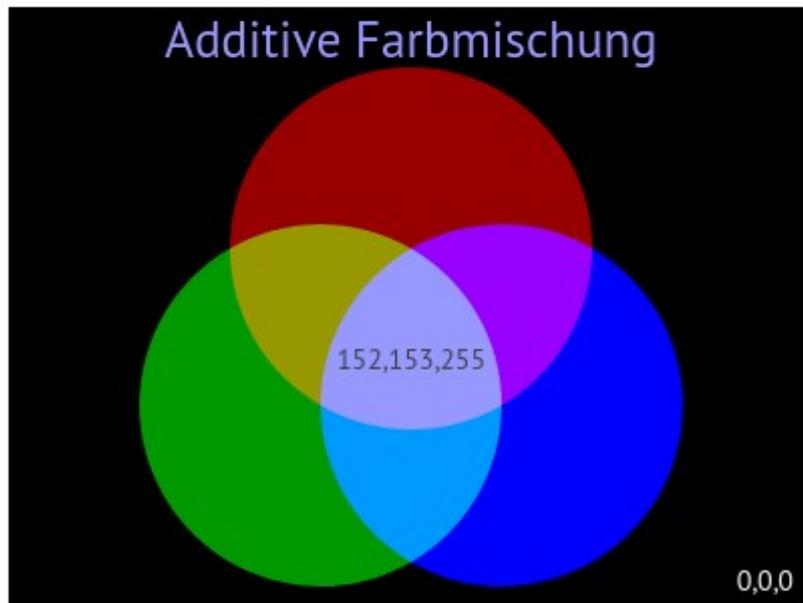
Rot	<input type="range" value="152"/>	<input type="text" value="152"/>	<input type="text" value="98"/>
Grün	<input type="range" value="153"/>	<input type="text" value="153"/>	<input type="text" value="99"/>
Blau	<input type="range" value="255"/>	<input type="text" value="255"/>	<input type="text" value="FF"/>

Hintergrund

Rot	<input type="range" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Grün	<input type="range" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
Blau	<input type="range" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>

„HTML-Farbe:“

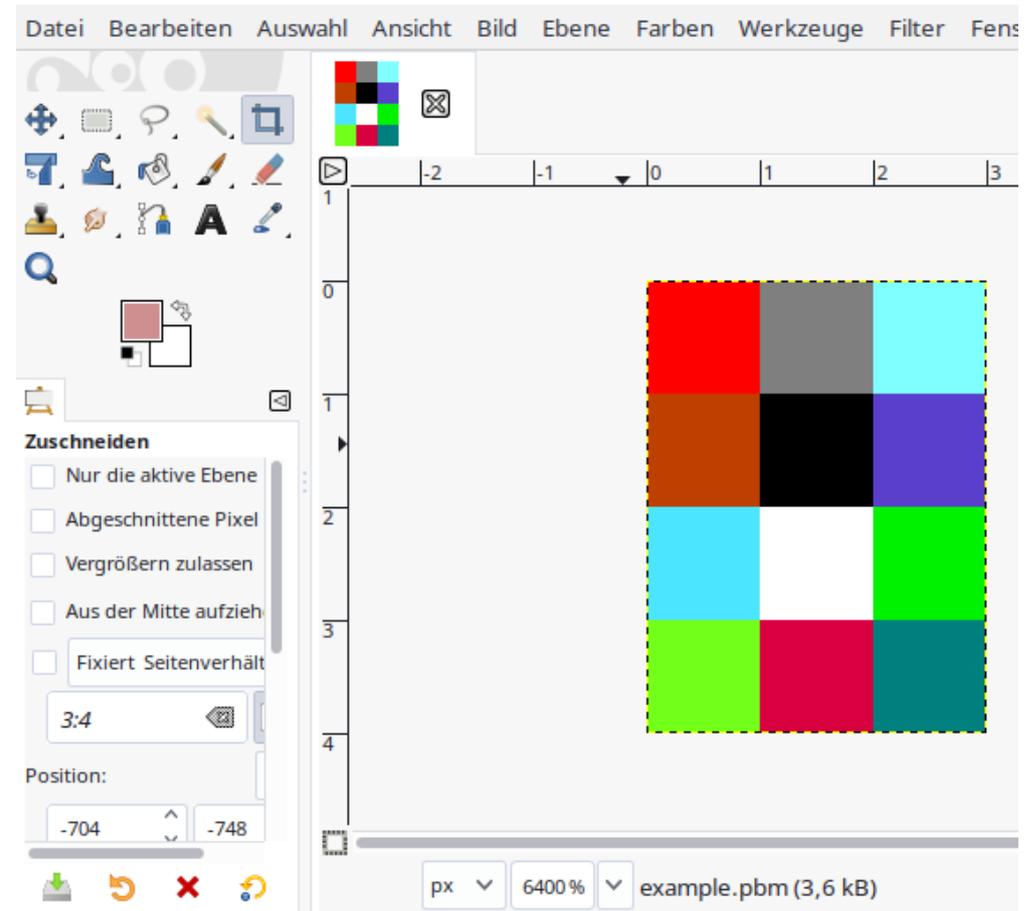
#9899FF



<https://www.colorzilla.com/firefox/>

PBM Farbbilder

- Start P3 → farbig,
- Spaltenzahl Zeilenzahl, z.B. 3 4
- Maximalwert für die Farbwerte R,G,B (1-255)
- Ins Raster an jede Pixelstelle schreibt man jetzt **drei** Werte: zwischen 0 und dem gewählten Maximalwert, das ergibt den RGB Wert, der dargestellt wird.
- Meist wird als Maximalwert 255 gewählt (1Byte), damit ergeben sich 255x 255x255 verschiedene Farben.
(ca. 16Mio – wer Windows 3.1 noch kennt, dem kommt das bekannt vor...)



```
< > b.pbm x
srv > samba > schools > default-school > teachers > sbel > b.pbm
1 P3
2 3 4
3 20
4 10 0 0 10 10 10 10 20 20
5 15 5 0 0 0 0 7 5 16
6 6 18 20 20 20 20 0 19 0
7 9 20 2 17 0 5 0 10 10
```