

Bildverarbeitung

- [Das Portable Anymap Bildformat](#)
- [Filter für das Portable Anymap Format mit python](#)
- [Farbmischung](#)

Das Portable Anymap Bildformat

Das Portable Anymap Bildformat gibt es in drei Versionen. Diese Bilder lassen sich mit einem Texteditor erstellen. Jedes Bildformat beginnt mit einer „magischen Zahl“ zur Identifizierung des Bildtyps, gefolgt von der Auflösung des Bildes und evtl. der Anzahl der Grau-, bzw. der Farbstufen. Die drei Bildarten lassen sich so definieren:

Schwarz-Weiß Bild

```
P1
11 7
1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0
```

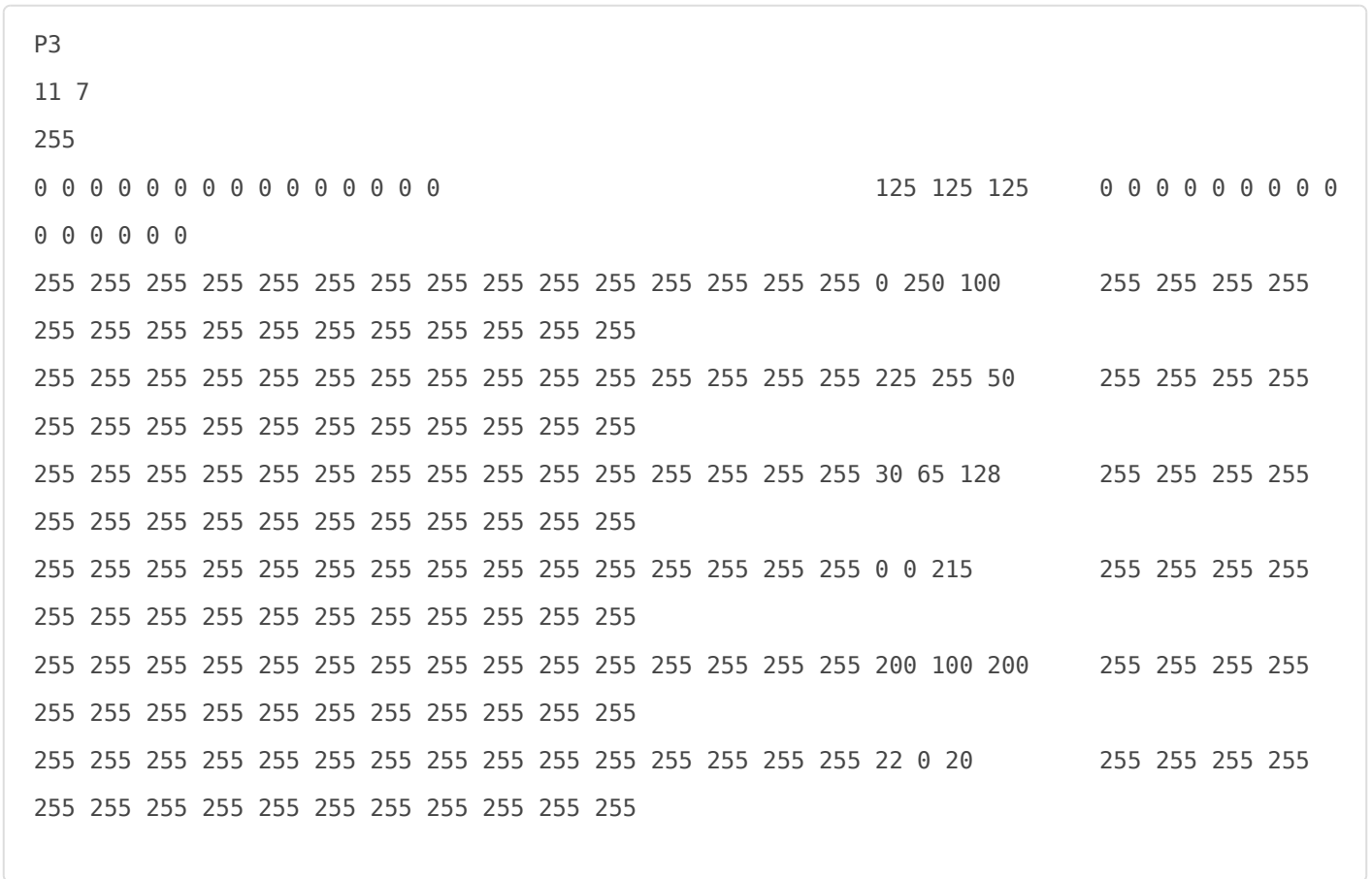
Dieses Bild stellt ein schwarzes T auf weißem Hintergrund dar.

Graustufenbild

```
P2
11 7
15
15 15 15 15 15 15 15 15 15 15 15
0 0 0 0 0 13 0 0 0 0 0
0 0 0 0 0 11 0 0 0 0 0
0 0 0 0 0 9 0 0 0 0 0
0 0 0 0 0 8 0 0 0 0 0
0 0 0 0 0 6 0 0 0 0 0
0 0 0 0 0 4 0 0 0 0 0
```

Dieses Bild stellt ein T auf weißem Hintergrund da, dessen Bein nach unten hin immer heller wird.

Farbbild



Dieses Bild stellt ein farbiges T dar.

Filter für das Portable Anymap Format mit python

```
from tkinter import *
from PIL import Image, ImageTk
# Ändere nichts oberhalb dieser Zeile

# Schreibe hier den Filter für P1 Dateien
def filter_p1(n):
    pass

# Schreibe hier den Filter für P2 Dateien
def filter_p2(n):
    pass

# Schreibe hier den Filter für P3 Dateien
def filter_p3(n):
    pass

# Ändere nichts unterhalb dieser Zeile.
new_image = ""
filename= input("Bitte Dateinamen angeben: ")
file = filename
newfile = file.split(".")[0]+ "_filtered." + file.split(".")[1]

with open(file, "r") as f:
    file = f.read()
    success = False
    if "P1" in file:
        success = True
        filetype = "P1"
    elif "P2" in file:
        success = True
        filetype = "P2"
```

```

elif "P3" in file:
    success = True
    filetype = "P3"
if not success:
    print("Could not determine filetype. Abort.")
    quit()
file_content = file.split("\n")
image_data = []
value = 0
for l in file_content:
    if not l.startswith("#"):
        match value:
            case 0:
                new_image = new_image + l + "\n"
                value += 1
            case 1:
                new_image = new_image + l + "\n"
                resolution = l
                value += 1
            case 2:
                if filetype != "P1":
                    new_image = new_image + l + "\n"
                    depth = l
                else:
                    image_data += filter(lambda x: x != "", l.split(" "))
                    value += 1
            case _:
                image_data += filter(lambda x: x != "", l.split(" "))

#print(image_data)
columns = int(resolution.split(" ")[0])
if filetype == "P3":
    columns = columns * 3
count = 1
for i in image_data:
    if filetype == "P1":
        if i != "":
            new_image = new_image + str(filter_p1(int(i))) + ""
    elif filetype == "P2":
        if i != "":

```

```

        if "\n" in i:
            print("I: "+i)
            new_image = new_image + str(filter_p2(int(i))) + " "
    elif filetype == "P3":
        if i != "":
            new_image = new_image + str(filter_p3(int(i))) + " "
    if count%columns == 0:
        new_image = new_image + "\n"
    count += 1
f.close()

with open(newfile, "w") as nf:
    nf.write(new_image)
    nf.close()

root = Tk()

# Read the Image
original_image = Image.open(filename)
filtered_image = Image.open(newfile)
# Resize the image using resize() method
w = int(resolution.split(" ")[0])
h = int(resolution.split(" ")[1])
scale=10
if w < 300:
    original_image = original_image.resize((w*scale, h*scale))
    filtered_image = filtered_image.resize((w*scale, h*scale))

oimg = ImageTk.PhotoImage(original_image)
fimg = ImageTk.PhotoImage(filtered_image)
# create label and add resize image
label1 = Label(root, image=oimg)
label1.image = oimg
label1.pack()
label2 = Label(root, image=fimg)
label2.image = fimg
label2.pack()
# Execute Tkinter
root.mainloop()
print("Bye, bye")

```


Farbmischung

[Farbpalettengenerator](#)