

14. ANIMIERTE BILDER

■ DU LERNST HIER...

wie du mit beliebigen, lokal gespeicherten Bildern Computeranimationen erstellst. Alle Bilder, die in den Musterbeispielen verwendet werden, kannst du im voraus von [hier](#) herunterladen.

■ MUSTERBEISPIELE

Flackerfreie Animation

Im Abschnitt *Computeranimation* hast du animierte Figuren so erstellt, dass du die vorhergehende Figur zum Löschen mit weisser Farbe übermaltest. Häufig ist dies nicht möglich, insbesondere bei der Animationen mit Bildern oder wenn das Animationsfenster einen Hintergrund enthält. In diesem Fall musst du jeweils das ganze Animationsfenster löschen. Den Animationsablauf kannst du so beschreiben:

Wiederhole:

- ganzes Animationsfenster löschen
- ev. Hintergrund zeichnen
- Bild zeichnen
- einige Millisekunden warten
- neue Bildposition festlegen

Da nach dem Löschen des Fensters ein Moment lang ein leeres Fenster sichtbar ist, ergibt sich bei diesem Verfahren oft ein unschönes Flackern. Um eine ruckfreie Computeranimation zu erhalten, es wichtig, dass das Löschen und Zeichnen in einem **Bildspeicher** (Bildbuffer) erfolgt und das Bild erst danach als Ganzes im Animationsfenster dargestellt ("gerendert") wird. Man spricht auch von einer **Doppelbufferung**.

In unserer Turtlegrafik wird die Doppelbufferung wie folgt eingeschaltet: Mit `makeTurtle(autoRefresh = False)` deaktivierst du das automatische Rendern und die Turtle wird versteckt. Alle Zeichnungsoperationen werden nachfolgend zuerst in einem Bildbuffer durchgeführt und erst mit dem Befehl `refresh()` im Animationsfenster angezeigt.

Bilder laden

Um ein Bild in der Animation zu verwenden, musst du die Bilddatei (im Format PNG oder GIF) vor der Programmausführung vom lokalen Datenträger in den Online-Editor laden. Dazu klickst du auf Schaltfläche *Bilder laden* und wählst die Bilddatei aus.



Im ersten Beispiel zeichnest du ein drehendes Rad. Ein passendes Bild kannst du von [hier](#) herunterladen. Speichere die Datei *wheel.png* in einem beliebigen Ordner auf deiner Festplatte. Um es in den Editor laden, klickst du auf *Bilder laden* und wählst die Bilddatei aus.

Obschon also die Turtle verborgen ist, dient sie bei Animationen immer noch als "Leitfigur". Mit dem Befehl **drawImage()** wird das Bild mit der Bildmitte an der aktuellen Turtle-Position in der Blickrichtung der Turtle in den Bildbuffer kopiert.



Wie oben dargelegt, löschst du in der Animationsschleife zuerst mit **clear()** das Fenster, kopierst das geladene Bild mit **drawImage()** in den Bildbuffer und renderst es mit **refresh()** im Animationsfenster. Du wartest 10 Millisekunden und drehst dann die Turtle und damit auch das Bild um 1° nach rechts.

Programm: [▶ Online-Editor](#)

```
from gturtle import *  
  
makeTurtle(autoRefresh = False)  
  
while True:  
    clear()  
    drawImage()  
    refresh()  
    delay(10)  
    right(1)
```

[▶ In Zwischenablage kopieren](#)

Mehrere Bilder verwenden

Du kannst auch mehrere Bilder in den Editor laden und eine animierte Figur, wie in einem Trickfilm bewegen. Im nächsten Programm verwendest du 8 Bilder (*pony_0.gif*, *pony_1.gif*, ..., *pony_7.gif*). Du kannst diese als ZIP-Datei von [hier](#) herunterladen und in irgendeinem Ordner auspacken. Den Bildern wird beim Laden ein Index in alphabetischer Reihenfolge des Dateinamens zugewiesen (hier 0, 1, 2, ..., 7), den du in **drawImage(i)** verwenden kannst.

In deiner Animation werden die 8 Pony-Bilder mit einer **for-Schleife** der Reihe nach angezeigt. Die Turtle als Leitfigur bewegt sich mit jedem neuen Bild 5 Schritte vorwärts. Sie startet rechts und wenn sie den linken Rand erreicht, wird sie und damit das Bild wieder an den rechten Rand versetzt.

Programm: [▶ Online-Editor](#)

```
from gturtle import *  
  
makeTurtle(autoRefresh = False)  
setX(270)  
left(90)  
while True:  
    for i in range(8):  
        clear()  
        drawImage(i)  
        refresh()  
        delay(100)  
        forward(5)  
        if getX() < -270:  
            setX(270)
```

[▶ In Zwischenablage kopieren](#)



Hintergrundbild hinzufügen

Ein feuerspeiender Drache fliegt über einen Dschungel-Hintergrund. Zur Verfügung stehen dir 8 Drachenbilder (*dragon_0.gif, ..., dragon_7.gif*) und das Hintergrundbild *jungle.png*, die du als ZIP-Datei von [hier](#) herunterladen kannst. Hier verwendest den Befehl **drawImage()** mit 4 Parametern: *drawImage(x, y, dir, index)*, wo *x, y* die Koordinaten der Bildmitte, *dir* die Richtung des Bildes und *index* der Bildindex sind.

Da der ganze Bildschirm in der Animationsschleife gelöscht wird, musst du auch das **Hintergrundbild** (Index 8) bei jedem Schleifendurchgang neu zeichnen.

Programm: [▶ Online-Editor](#)

```
from gturtle import *

makeTurtle(autoRefresh = False)
x = -270
y = 60
while True:
    for i in range(8):
        clear()
        drawImage(0, 0, 0, 8)
        drawImage(x, y, -10, i)
        refresh()
        delay(100)
        x += 10
        y += 2
    if x > 270:
        x = -270
        y = 60
```

[▶ In Zwischenablage kopieren](#)



■ MERKE DIR...

Mit der Schaltfläche **Bilder laden** kannst du lokal gespeicherte Bilder (im Format PNG oder GIF) in der Editor laden und in deinem Programm verwenden. Der Befehl **drawImage()** kann keinen, einen oder 4 Parameter haben. Bei fehlenden Parametern wird das Bild mit dem Index 0 an der aktuellen Position der Turtle angezeigt. Um die Bilder zu bewegen kannst du die Turtle als Leitfigur verwenden, oder du legst die Position und die Richtung der Bilder mit den Parametern im Befehl **drawImage(x, y, dir, index)** fest.

Um eine flackerfreie Animation zu erhalten, deaktivierst du in *makeTurtle()* mit **autoRefresh = False** das automatische Rendern und renderst den Bildbuffer mit **refresh()** auf dem Bildschirm.

■ ZUM SELBST LÖSEN

1. Das Postauto soll sich wiederholt von links nach rechts bewegen. Die Bilddatei kannst du [hier](#) herunterladen oder mit rechtem Klick auf das nebenstehende Bild und der Option *Grafik speichern unter* auf die Festplatte kopieren.



2. Ergänze die Pony-Animation mit einem Hintergrundbild. Du kannst alle Bilder als ZIP-Datei von [hier](#) herunterladen.



3. Stelle mit den fünf Bildern *person_0.png*, *person_1.png*,

person_2.png, *person_3.png* und *person_4.png* eine von links nach rechts laufende Person mit einem blauen Hintergrund dar. Die Bilder kannst du als ZIP-Datei von [hier](#) herunterladen.

