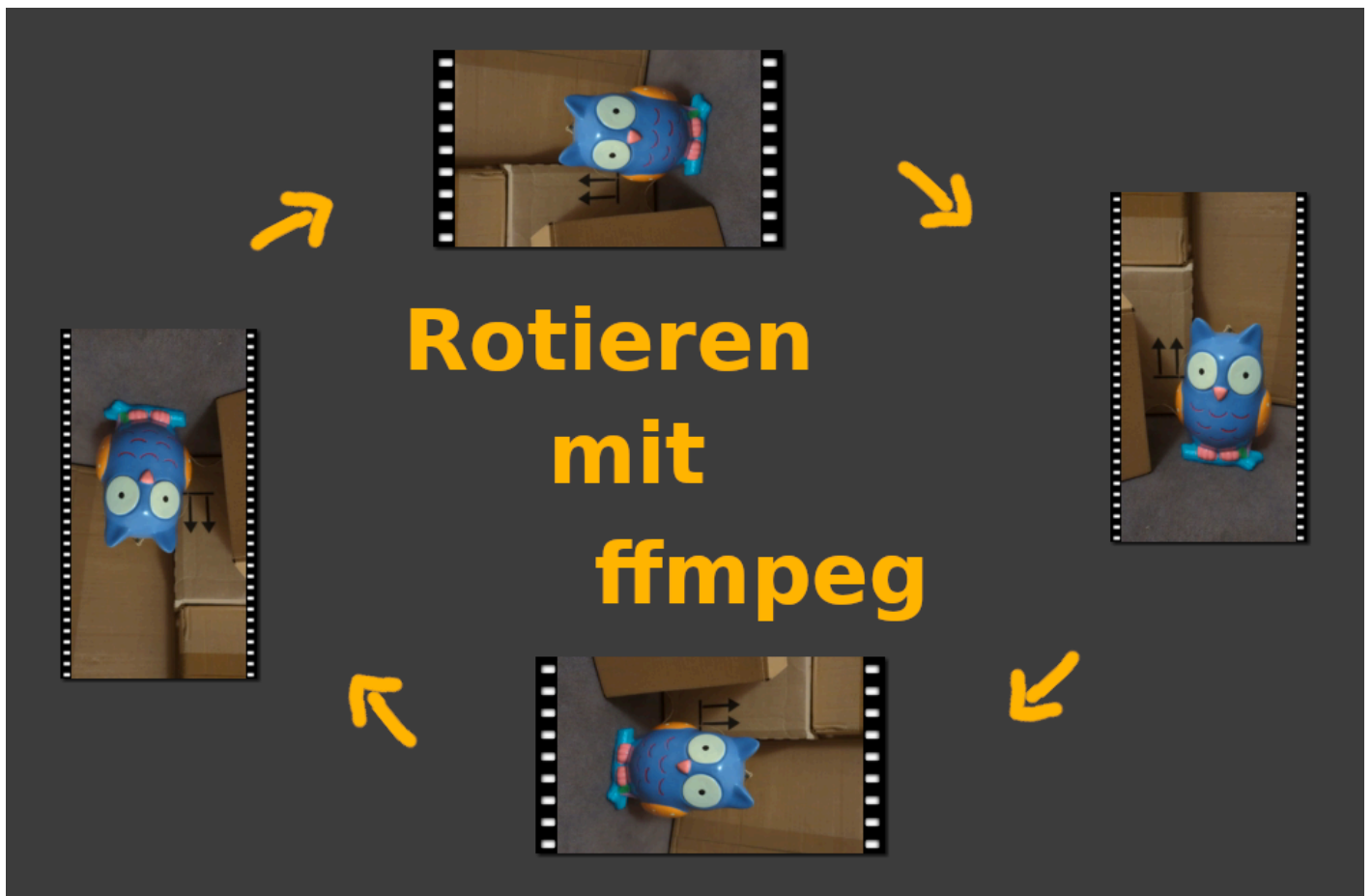


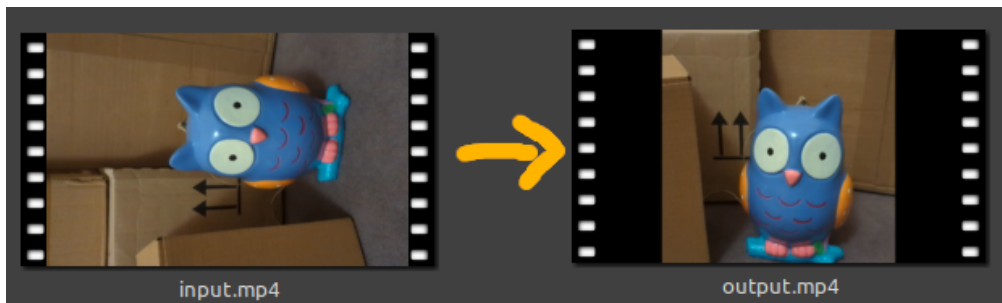
# Rotieren eines Videos mit ffmpeg



Um ein Video zu rotieren, gibt es zwei Hauptmöglichkeiten:

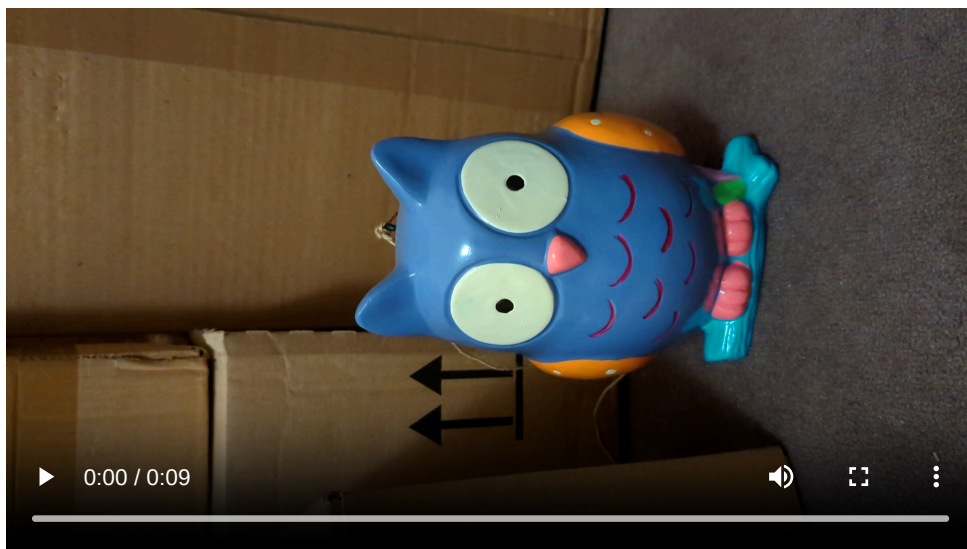
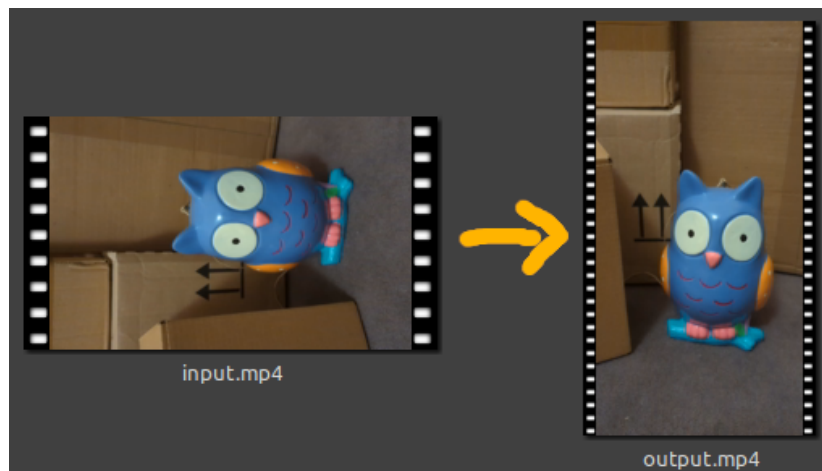
1. Neu rendern (langsamer aber zuverlässig)
  - 90° (90° mit dem Uhrzeigersinn)
  - -90° (90° gegen den Uhrzeigersinn)
  - 180°
2. Nur die MetaDaten anpassen (sehr viel schneller aber fehleranfälliger)
  - 90° (90° mit dem Uhrzeigersinn)
  - -90° (90° gegen den Uhrzeigersinn)
  - 180°

Und wir wollen nicht nur das Bild im Video rotieren. Weil dann Teile des Bildes abgeschnitten werden und schwarze Balken an den Rändern entstehen:



Um das zu vermeiden, wollen wir das gesamte Video rotieren, zusammen mit seiner Ausrichtung. Das heist, aus einem horizontalen Video wird ein vertikales Video. Oder aus einem vertikalen wird ein horizontales Video.

Dazu wird sowohl das Bild im Video rotiert, aber auch das SeitenVerhältnis des Videos angepasst:

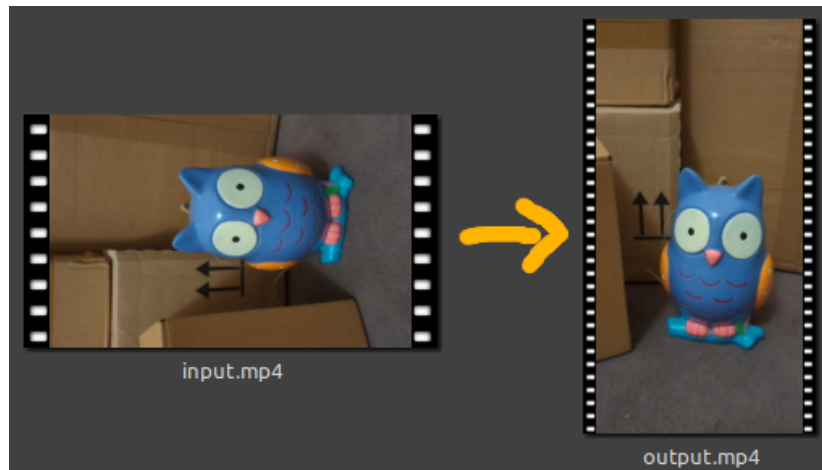


## 1. Neu rendern (langsamer aber zuverlässig)

Wenn wir das Video rotieren und dafür komplett neu rendern, dann dauert das etwas länger, als wenn wir nur MetaDaten dafür anpassen würden.

Aber dafür können wir sicher sein, dass jedes Video-Programm mit dem neuen Video klar kommt und es gibt keine Probleme.

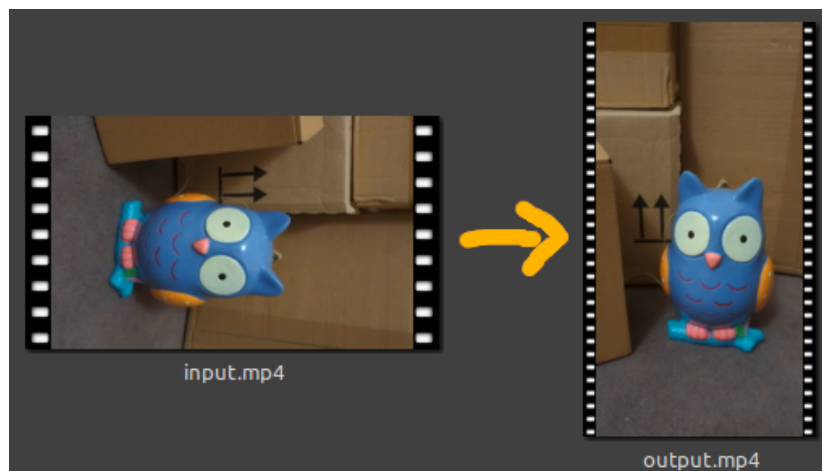
## 90° (90° mit dem Uhrzeigersinn)



```
ffmpeg -i "input.mp4" -vf "rotate=90*PI/180:oh=iw:ow=ih" "output.mp4"
```

- `-i "input.mp4"`: Gibt die Input-Datei an.
- `-vf "rotate=90*PI/180:oh=iw:ow=ih"`: Gibt den Video-Filter "rotate" an. Der Winkel muss hier in "rad" (Bogenmaß) angegeben werden.
  - `90*PI/180` "rad" entsprechen 90°.  
(`PI` entspricht 180°, `PI/180` entspricht dann 1° und `90*PI/180` entspricht dann 90°)
  - `oh=iw` bedeutet, dass die Output-Höhe der Input-Breite entsprechen soll, um die Ausrichtung des Videos zu ändern. (`oh` steht für "output-height" und `iw` für "input-width")
  - `ow=ih` bedeutet, dass die Output-Breite der Input-Höhe entsprechen soll, um die Ausrichtung des Videos zu ändern. (`ow` steht für "output-width" und `ih` für "input-height")
- `"output.mp4"`: Gibt die Output-Datei an.

## -90° (90° gegen den Uhrzeigersinn)

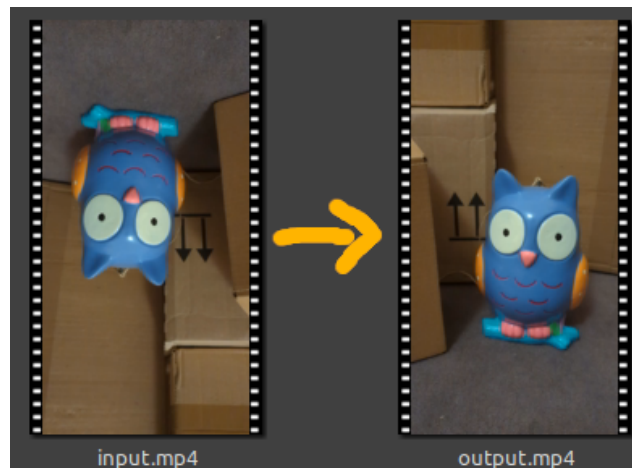


```
ffmpeg -i "input.mp4" -vf "rotate=-90*PI/180:oh=iw:ow=ih" "output.mp4"
```

- `-90*PI/180` "rad" entsprechen -90°.  
(`PI` entspricht 180°, `PI/180` entspricht dann 1° und `-90*PI/180` entspricht dann 90°)

- `oh=iw` bedeutet, dass die Output-Höhe der Input-Breite entsprechen soll, um die Ausrichtung des Videos zu ändern. (`oh` steht für "output-height" und `iw` für "input-width")
- `ow=ih` bedeutet, dass die Output-Breite der Input-Höhe entsprechen soll, um die Ausrichtung des Videos zu ändern. (`ow` steht für "output-width" und `ih` für "input-height")

180°



```
ffmpeg -i "input.mp4" -vf "rotate=180*PI/180" "output.mp4"
```

- `180*PI/180` "rad" entsprechen 180°. (`PI` entspricht 180°, `PI/180` entspricht dann 1° und `180*PI/180` entspricht dann 180°)
- `oh=iw` und `ow=ih` entfallen, da die Ausrichtung des Videos bei einer Drehung um 180° gleich bleibt.

## 2. Nur die MetaDaten anpassen (sehr viel schneller aber fehlerAnfälliger)

Jetzt wollen wir nur die Metadaten so anpassen, dass das Video gedreht werden soll. Das geht sehr viel schneller, weil das Video dafür nicht neu gerendert werden muss, sondern fast komplett kopiert werden kann.

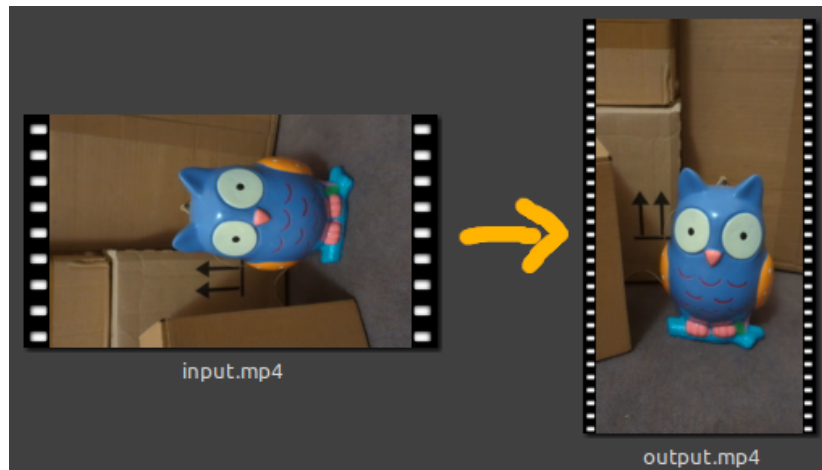
**Achtung:** Das kann zu Fehlern in einigen Video-Programmen führen, wenn diese die Metadaten nicht beachten!

Wenn man das Video aber nur selbst verwenden will und die eigene verwendete Software damit klar kommt, kann das eine sehr schnelle Alternative sein.

**Achtung:** Es kann sein, dass z.B. ein Hochkant-Video (Portrait-Video) in seinen MetaDaten schon den Rotations-Wert 90 oder -90 hat! Dann müssen wir ein wenig rumprobieren, welcher der Werte der passende für uns ist: `0`, `90`, `-90` oder `180`.

Das kann man ja aber einfach ausprobieren, weil die Methode ja sowieso so schnell geht.

## 90° (90° mit dem Uhrzeigersinn)

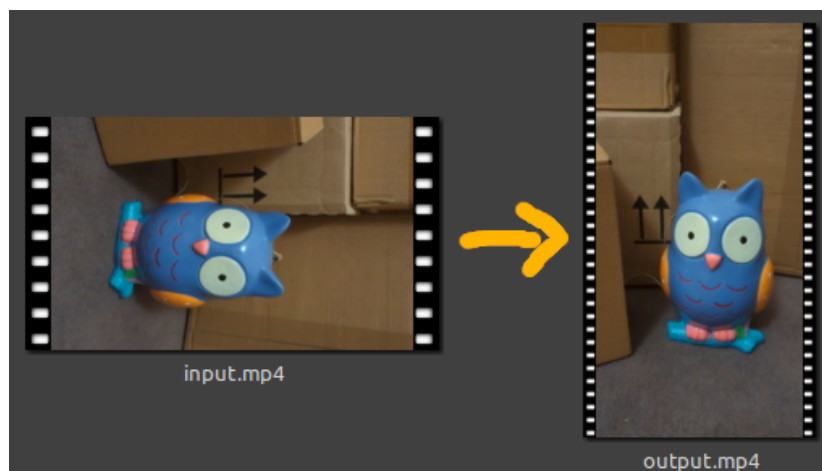


```
ffmpeg -i "input.mp4" -map_metadata 0 -metadata:s:v rotate="-90" -codec copy "output.mp4"
```

- `-map_metadata 0` bedeutet, dass die MetaDaten von der 1. Input-Datei übernommen werden sollen.
- `-metadata:s:v rotate="-90"` bearbeitet die MetaDaten:
  - `-metadata:s:v` bedeutet, dass die MetaDaten für UnterTitel und das Video bearbeitet werden sollen (`s` steht für "subTitles" und `v` für "video")
  - `rotate="-90"` steht dafür, dass die Kamera beim Filmen um -90° gedreht war .. und deshalb das Video bei der darstellung um +90° gedreht werden soll.

**Achtung:** Wir schreiben **-90** um das Video **+90°** zu rotieren!

## -90° (90° gegen den Uhrzeigersinn)



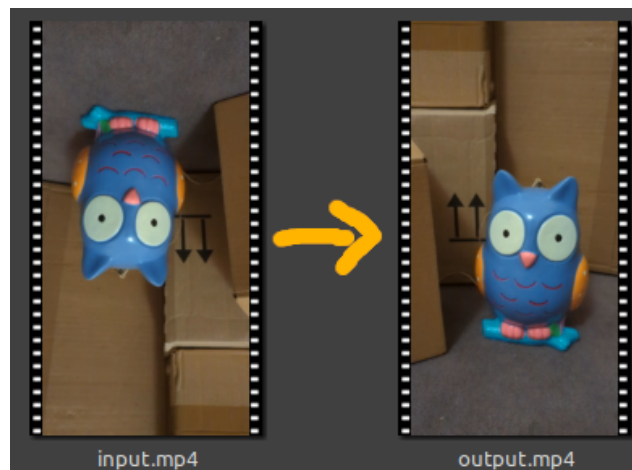
```
ffmpeg -i "input.mp4" -map_metadata 0 -metadata:s:v rotate="90" -codec copy "output.mp4"
```

- `rotate="90"` steht dafür, dass die Kamera beim Filmen um +90° gedreht war .. und deshalb das Video bei der darstellung um -90° gedreht werden soll.



**Achtung:** Wir schreiben **90** um das Video **-90°** zu rotieren!

**180°**



```
ffmpeg -i "input.mp4" -map_metadata 0 -metadata:s:v rotate="180" -codec copy "output.mp4"
```

- `rotate="180"` steht dafür, dass die Kamera beim Filmen um  $+180^\circ$  gedreht war .. und deshalb das Video bei der darstellung um  $-180^\circ$  gedreht werden soll.

Hier ist das Vorzeichen egal, weil eine Drehung um  $+180^\circ$  das Selbe ist wie um  $-180^\circ$ .

