

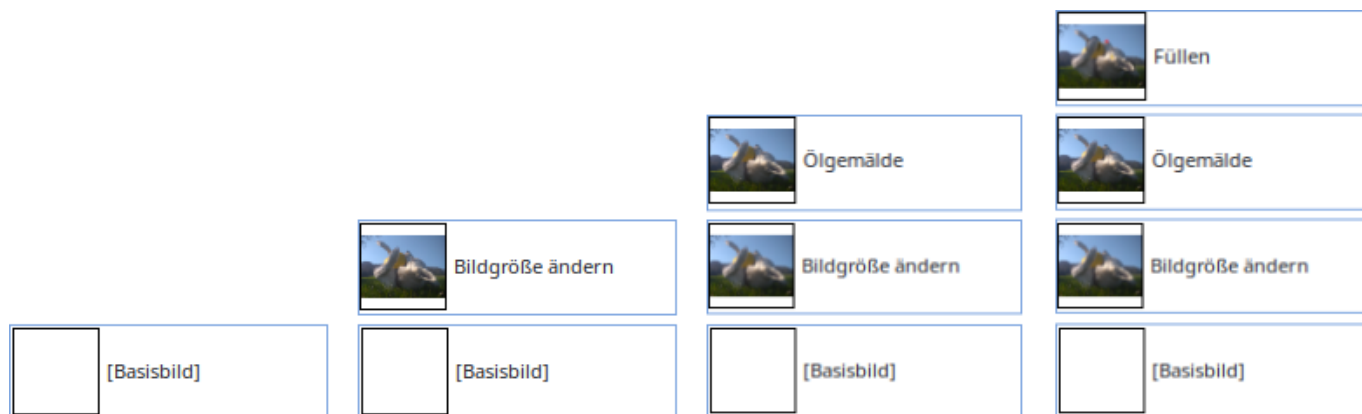
Zurück! Die Rückgängig Funktion

Wer kennt das nicht, einmal die Katze über die Tastatur gerannt und schon ist das Bild komplett im Eimer:



Wir alle wissen, was da hilft: STRG+Z - Rückgängig!

Welches Prinzip steckt hinter dieser Funktion? Das Programm speichert zunächst alle Arbeitsschritte - es stapelt Sie einfach aufeinander:



Mit der Funktion "Rückgängig" kann man die **letzte Anweisung** zurücknehmen. Woher weiß das Programm, was die letzte Anweisung war? Ganz einfach, das war stets die Anweisung die ganz oben auf dem **Stapel** der Anweisungen liegt.

Um die Anweisung rückgängig zu machen, wird sie einfach vom Stapel **heruntergenommen**, eine neue Anweisung wird stets **oben auf den Stapel gelegt**. Wenn mehrerer Arbeitsschritte zurückgenommen werden sollen, werden diese **nacheinander** oben vom Stapel genommen - bis dieser **leer** ist, dann kann nichts mehr rückgängig gemacht werden.

Eine solche lineare Datenstruktur heißt **Stapel**:

- Ein Stapel funktioniert nach dem Prinzip LIFO: Last In First Out
- Elemente können stets nur oben auf den Stapel gelegt werden.
- Nur auf das oberste Element kann zugegriffen werden.
- Nur das oberste Element kann vom Stapel entfernt werden.
- Wenn der Stapel leer ist, kann kein weiteres Element vom Stapel entfernt werden.

[Weiter zur Implementation >>>](#)

Material

| | | | |
|---|-----------|------------|-------|
| img_20210404_151748.jpg | 228.0 KiB | 06.10.2021 | 16:44 |
| stack.odp | 2.2 MiB | 06.10.2021 | 18:42 |
| stack.pdf | 508.8 KiB | 06.10.2021 | 18:42 |
| stack_gimp.png | 58.9 KiB | 06.10.2021 | 16:50 |

From:
<https://info-bw.de/> -

Permanent link:
<https://info-bw.de/faecher:informatik:oberstufe:adt:stack:zuerueck:start>

Last update: **04.12.2023 14:39**

