

Lern-PDF: Zahlen aus Strings wie "High (90)" extrahieren – einfach erklärt

Ziel: Was lernst du hier?

In vielen Programmen (wie deinem GUI-Tool mit Comboboxen) gibt es **Texte mit Zahlen in Klammern**, zum Beispiel:

- "High (90)"
- "Low (30)"
- "Max(9)"

Du willst aus diesen Texten die **Zahl (z. B. 90)** herausholen, um sie als **echten Zahlenwert (Integer)** zu benutzen.

🔍 Warum macht man das?

Diese Zahlen brauchst du zum Beispiel, um:

- die **Qualität von Bildern** festzulegen
- die **Kompressionsstufe** zu setzen
- bestimmte Einstellungen wie Lautstärke oder Bitrate festzulegen

Die Texte wie "High (90)" sind **für den Nutzer gedacht**, du brauchst aber **nur die Zahl** daraus für dein Programm.

Methode: String zerschneiden mit `.find()` und Slicing

Beispiel:

```
text = "High (90)"
start = text.find("(") + 1
end = text.find(")")
number_str = text[start:end] # ergibt: "90"
```

Erklärung:

- `find("(")` gibt die Position von `(` zurück
- `+1`, damit die Klammer nicht mitkommt
- `find(")")` gibt das Ende der Zahl an

- `text[start:end]` schneidet die Zahl aus dem String

Jetzt hast du die Zahl als **String**: `"90"`

Umwandeln in Integer (Zahl):

```
if number_str.isdigit():  
    number = int(number_str)
```

Vorteil:

- Jetzt kannst du mathematisch damit arbeiten (z. B. `if number > 80:`)

Bonus: `.zfill()` für führende Nullen

Falls du z. B. für Dateinamen **immer 3-stellige Zahlen** willst:

```
padded = number_str.zfill(3) # ergibt: "090"
```

Beispiel:

```
filename = f"bild_{padded}.png" # "bild_090.png"
```

Vorteil:

- Einheitliche Dateinamen: "bild_001.png", "bild_099.png"
- Sauber sortierbar nach Namen

Zusammenfassung

Zweck	Methode	Ergebnis
Zahl aus "High (90)" holen	<code>text[text.find("(")+1:text.find(")"]]</code>	"90" (String)
In echte Zahl umwandeln	<code>int(number_str)</code>	90 (Integer)
Führende Nullen (optional)	<code>number_str.zfill(3)</code>	"090"

Tipps speziell für dein Projekt

- Verwende diese Methode **direkt nach dem `.get()`** der Combobox.

- Falls die Zahl fehlt oder falsch ist, kannst du einen Standardwert setzen:

```
if not number_str.isdigit():
    number = 75 # Standardwert
```

- Du kannst die Funktion auch verpacken:

```
def extract_number(text):
    if "(" in text and ")" in text:
        val = text[text.find("(")+1:text.find(")")]
        if val.isdigit():
            return int(val)
    return 75
```

Dann kannst du schreiben:

```
wert = extract_number(self.png_quality.get())
```

Mit diesem Wissen kannst du ganz einfach und sicher Zahlen aus nutzerfreundlichen Texten extrahieren – perfekt für dein Bild- und Audioprojekt! Du brauchst dafür **keine zusätzlichen Module**, nur Basisfunktionen von Python.