

CheatSheet für Reguläre Ausdrücke (Regex)

Regex-Objekt anlegen

```
import re # Das Regex-Modul einbinden

win_code_pattern_re = re.compile(r"\d{3}-\d{3}-\d{4}") # Definition des Regex-Musters

mo = win_code_pattern_re.search(text) # im String text nach dem Muster suchen, MatchObjekt

# mo != None      -> Muster wurde gefunden
# mo = None       -> Muster wurde NICHT gefunden
```

Auf Gruppen zugreifen

<code>mo.group(1)</code>	Zugriff auf die erste (!) Gruppe
<code>mo.group(0)</code> oder <code>mo.group()</code>	Zugriff auf das komplette Muster (ohne Gruppierung)

HINWEIS: Rückgabe von `group` ist ein String! Wenn ihr rechnen wollt, müsst ihr das konvertieren (z.B. `int()`)

Zeichen-Klassen

<code>\d</code>	Eine beliebige numerische Ziffer von 0 bis 9.
<code>\D</code>	Jedes Zeichen, das keine numerische Ziffer von 0 bis 9 ist.
<code>\w</code>	Ein beliebiger Buchstabe, eine numerische Ziffer oder der Unterstrich. (Stellen Sie sich dies als passende „Wort“-Zeichen vor.)
<code>\W</code>	Jedes Zeichen, das kein Buchstabe, keine numerische Ziffer oder ein Unterstrich ist.
<code>\s</code>	Jedes Leerzeichen, jeder Tabulator und jeder Zeilenumbruch. (Stellen Sie sich dies als passende „Leerzeichen“ vor.)
<code>\S</code>	Jedes Zeichen, das nicht ein Leerzeichen, ein Tabulator oder ein Zeilenumbruch ist.

Wiederholungen

<code>*</code>	0...n Wiederholungen
<code>+</code>	1...n Wiederholungen
<code>{3}</code>	Genau 3 Wiederholungen
<code>{3,5}</code>	3 bis einschließlich 5 Wiederholungen

HINWEIS: Wiederholung funktionieren sowohl innerhalb einer Gruppe für Zeichen oder für die gesamte Gruppe, z.B. `(\d{1,2})` oder `(\d){1,2}`.