

# Reguläre Ausdrücke (regular expressions)

`/pattern/modifiern`

## delimiter

Als Delimiter darf jedes nicht alphanumerische Zeichen außer `\` verwendet werden. Gängige Delimiter sind: `/ @ # ~ ! & =`

## pattern

### Metacharaktere

`.` jedes beliebige Zeichen, außer Zeilenumbruch  
`^` Anfang des Strings  
`$` Ende des Strings  
`|` Alternativen  
`()` Zusammenfassen von Teilen des pattern; erzeugt eine Rückwärtsreferenz  
`[]` siehe Zeichenklassen  
`{}` siehe Quantoren

### Zeichenlitterale

Zeichen, die direkt (wörtlich, literal) übereinstimmen müssen, werden direkt notiert.  
*Bsp.: `/foo/` => der Ausdruck muss die Zeichenkette foo enthalten*

### Zeichenklassen [characters]

Alle erlaubten Zeichen (engl. characters) sind zwischen `[` und `]` anzugeben.  
*Bsp.: `[abc]` => Die Zeichen a, b und c sind erlaubt.*  
Ein vorangestelltes `^` negiert die Zeichenklasse.  
*Bsp.: `[^abc]` => Die Zeichen a, b und c sind verboten, alle anderen sind erlaubt.*  
Ein `-` zwischen zwei Zeichen gibt eine Reichweite an.  
*Bsp.: `[a-c0-2]` => Die Zeichen a, b, c, 0, 1 und 2 sind erlaubt.*

### Rückwärtsreferenzen (backreferences)

`$n` ist Referenz der n. Klammergruppe

### Erweitertes Pattern

`(?:pattern)` Erzeugt keine backreference  
`(?!pattern)` Negiert das `pattern`; wirkt sich auf das Vorhergehende `pattern` aus.  
`(?<!pattern)` Negiert das `pattern`; wirkt sich auf das Nachfolgende `pattern` aus.

### Quantoren {n,m}

`*` Null oder mehr Vorkommen  
`+` Ein oder mehr Vorkommen  
`?` Null oder ein Vorkommen  
`{n}` Genau n Vorkommen  
`{n,}` n oder mehr Vorkommen  
`{,m}` m oder weniger Vorkommen  
`{n,m}` n bis m Vorkommen.  
Wird dem Quantor ein `?` nachgestellt, erzeugt dies ein nicht gieriges Verhalten.

## modifiern

**i** Keine Unterscheidung zwischen Groß- und Kleinschreibung  
**s** . trifft auch auf einen Zeilenumbruch (`\n`) zu.  
**m** Die Zeichenkette wird zeilenweise gelesen

*Bsp.: `/^A$/i` trifft auf die Zeichenkette a zu.*  
*Bsp.: `/^ab.cd$/s` trifft auf die Zeichenkette ab\ncd zu.*  
*Bsp.: `/^ab$/m` trifft auf die Zeichenkette ab\ncd zu.*