

Regular Expression

(Regex)

Warum brauchen wir Reguläre Ausdrücke?

- Um automatisch nach den gewünschten Inhalten/Dateien/... zu suchen
- Um zu prüfen, ob Benutzereingaben korrekt sind
z.B. bei der Angabe der Email-Adresse

Regex

[A-Za-z]	Groß-/ Kleinbuchstabe
[aA]	a oder A
\d	Ziffer
\w	Groß-/ Kleinbuchstabe oder Ziffer
\s	Whitespace wie Leerzeichen, \n, \t, \r
\D	not a digit
\S	not a Whitespace
\W	not a word
.	Jedes beliebige Zeichen

Regex

\wedge	Zeilenanfang
$\$$	Zeilenende
$X?$	0-1 mal X
X^*	0-n mal X
X^+	1-n mal X
$X\{n\}$	genau n mal X
$X\{n, \}$	mindestens n mal X
$X\{n,m\}$	zwischen n und m mal X
$[a-z\&\&[\wedge abc]]$	alle Zeichen von d bis z

Regex in Java

Mit Hilfe eines Pattern-Objektes gibt man das gesuchte Muster an z.B. `A(\w)*d`, damit suchen wir alle Ausdrücke nach dem Schema A... d wie Android, Abend, Ackerland, Ad, A99678d.

Um in einem String ein „\“ zu schreiben, muss man “\\“ schreiben.

```
Pattern pattern = Pattern.compile("A(\\w)*d");
```

Regex in Java

```
String text = "Android Ad A00343d Ackerland Abenteuer bald Abend";

//nach diesem Schema wird gesucht
Pattern pattern = Pattern.compile("A(\\w)*d");

//wende das Schema auf die Variable text an
Matcher matcher = pattern.matcher(text);

//solange es noch ein Wort in der Variablen text gibt, auf das das Schema passt
while(matcher.find()) {

    //gibt das nächste Wort aus, auf das das Schema passt
    System.out.println(matcher.group());
}
```


Welche und wie viele Wörter findet er hier?

```
String text = "Android Ad A00343d Ackerland Abenteuer bald Abend";

//nach diesem Schema wird gesucht
Pattern pattern = Pattern.compile("A.*d");

//wende das Schema auf die Variable text an
Matcher matcher = pattern.matcher(text);

//solange es noch ein Wort in der Variablen text gibt, auf das das Schema passt
while(matcher.find()) {

    //gibt das nächste Wort aus, auf das das Schema passt
    System.out.println(matcher.group());
}
```

Welche und wie viele Wörter findet er hier?

```
String text = "Android Ad A00343d Ackerland Abenteuer bald Abend";

//nach diesem Schema wird gesucht
Pattern pattern = Pattern.compile("A.*d");

//wende das Schema auf die Variable text an
Matcher matcher = pattern.matcher(text);

//solange es noch ein Wort in der Variablen text gibt, auf das das Schema passt
while(matcher.find()) {

    //gibt das nächste Wort aus, auf das das Schema passt
    System.out.println(matcher.group());
}
```

Regex gibt immer das größte Wort aus, das trifft. Der ganze String fängt mit A an und hört mit d auf.

Aufgabe

Filtert aus einem Text

1) euren Namen

2) alle Wörter, die mit S oder s anfangen und mit e aufhören

Aufgabe

Lasst uns jeweils das letzte Wort eines Satzes filtern, wenn es mit a-k oder A-K anfängt.

Tipp: „.“ steht bei Regex für jedes beliebige Zeichen, wenn wir den normalen Punkt meinen müssen wir \. schreiben

Aufgabe

Lasst uns aus einem Logtext alle Zeiten rausfiltern wie

00:37:10 - Pizza bestellt

01:10:56 - Pizzalieferung ist da!

08:10:00 - Aufstehen

08:30:45 - Schon wieder Hunger

Tipp: Möchte man in Regex zwei Schemata zulassen, macht man das so: ((möglichkeit1)|(möglichkeit2))

Links

<https://www.vogella.com/tutorials/JavaRegularExpressions/article.html>

<https://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html>