



Einführung in
Regular
Expressions

Thomas
Helmke

Einleitung

Grundlagen

Zeichenauswahl

Zeichen zählen

Fortgeschrittene
Techniken

Variablen

Weitere Infos

Einführung in Regular Expressions

Thomas Helmke

00.00.2018



1 Einleitung

2 Grundlagen

- Zeichenauswahl
- Zeichen zählen

3 Fortgeschrittene Techniken

- Variablen

4 Weitere Infos



Einführung in
Regular
Expressions

Thomas
Helmke

Einleitung

Grundlagen

Zeichenauswahl

Zeichen zählen

Fortgeschrittene
Techniken

Variablen

Weitere Infos

Einleitung



What sorcery is this?

Einführung in
Regular
Expressions

Thomas
Helmke

Einleitung

Grundlagen

Zeichenauswahl

Zeichen zählen

Fortgeschrittene
Techniken

Variablen

Weitere Infos

- $\wedge([\backslash w-]+(?:\backslash.[\backslash w-]+)*)$
 $@((?:[\backslash w-]+\backslash.)*\backslash w[\backslash w-]\{0,66\})\backslash.([a-z]\{2,6\}(?:\backslash.[a-z]\{2\})?)\$$
- $\wedge(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\backslash.$
 $(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\backslash.$
 $(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\backslash.$
 $(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\$$
- $\wedge((?=\backslash S*?[A-Z])(?=\backslash S*?[a-z])(?=\backslash S*?[0-9])\{6,\})\backslash S\$$



What sorcery is this?

Einführung in
Regular
Expressions

Thomas
Helmke

Einleitung

Grundlagen

Zeichenauswahl

Zeichen zählen

Fortgeschrittene
Techniken

Variablen

Weitere Infos

- `^([\w-]+(?:\.[\w-]+)*)
@((?:[\w-]+\.)*\w[\w-]{0,66})\.([a-z]{2,6}(?:\.[a-z]{2})?)?$`
- `^(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\
(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\
(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\
(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)$`
- `^((?=\S*[A-Z])(?=\S*[a-z])(?=\S*[0-9])).{6,})\S$`



What sorcery is this?

Einführung in
Regular
Expressions

Thomas
Helmke

Einleitung

Grundlagen

Zeichenauswahl

Zeichen zählen

Fortgeschrittene
Techniken

Variablen

Weitere Infos

- $\wedge([\backslash w-]+(?:\backslash. [\backslash w-]+)*)$
 $@((?:[\backslash w-]+\backslash.)*\backslash w[\backslash w-]\{0,66\})\backslash.([a-z]\{2,6\}(?:\backslash. [a-z]\{2\})?)\$$
- $\wedge(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\backslash.$
 $(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\backslash.$
 $(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\backslash.$
 $(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\$$
- $\wedge((?=\backslash S*?[A-Z])(?=\backslash S*?[a-z])(?=\backslash S*?[0-9])\{6,\})\backslash S\$$



What sorcery is this?

Einführung in
Regular
Expressions

Thomas
Helmke

Einleitung

Grundlagen

Zeichenauswahl

Zeichen zählen

Fortgeschrittene
Techniken

Variablen

Weitere Infos

- $\wedge([\backslash w-]+(?:\backslash. [\backslash w-]+)*)$
 $@((?:[\backslash w-]+\backslash.)*\backslash w[\backslash w-]\{0,66\})\backslash.([a-z]\{2,6\}(?:\backslash. [a-z]\{2\})?)\1
- $\wedge(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\backslash.$
 $(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\backslash.$
 $(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\backslash.$
 $(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\2
- $\wedge((?=\backslash S*?[A-Z])(?=\backslash S*?[a-z])(?=\backslash S*?[0-9])\{6,\})\backslash S\3

¹Emailadresse

²IPv4 Adresse

³Passwort, min. 6 Zeichen, 1 Gross, 1 Klein, 1 Ziffer



Theoretischer Hintergrund

Einführung in
Regular
Expressions

Thomas
Helmke

Einleitung

Grundlagen

Zeichenauswahl

Zeichen zählen

Fortgeschrittene
Techniken

Variablen

Weitere Infos

Definition von Wikipedia

Ein regulärer Ausdruck ist in der theoretischen Informatik eine Zeichenkette, die der Beschreibung von Mengen von Zeichenketten mit Hilfe bestimmter syntaktischer Regeln dient.



Einführung in
Regular
Expressions

Thomas
Helmke

Einleitung

Grundlagen

Zeichenauswahl

Zeichen zählen

Fortgeschrittene
Techniken

Variablen

Weitere Infos

Grundlagen



Einführung in
Regular
Expressions

Thomas
Helmke

Einleitung

Grundlagen

Zeichenauswahl

Zeichen zählen

Fortgeschrittene
Techniken

Variablen

Weitere Infos

Zeichenauswahl



Ein oder mehr Zeichen

Einführung in
Regular
Expressions

Thomas
Helmke

Einleitung

Grundlagen

Zeichenauswahl

Zeichen zählen

Fortgeschrittene

Techniken

Variablen

Weitere Infos

- `/thomas/` findet „thomas“ als wort
- `/[thomas]/` findet „t“ oder „h“ oder „o“ oder „m“ oder „a“ oder „s“
- `/[a-z]/` findet einen beliebigen Kleinbuchstaben
- `/[0-9]/` findet eine beliebige Zahl



Ein oder mehr Zeichen

Einführung in
Regular
Expressions

Thomas
Helmke

Einleitung

Grundlagen

Zeichenauswahl

Zeichen zählen

Fortgeschrittene
Techniken

Variablen

Weitere Infos

- `/thomas/` findet „thomas“ als wort
- `/[thomas]/` findet „t“ oder „h“ oder „o“ oder „m“ oder „a“ oder „s“
- `/[a-z]/` findet einen beliebigen Kleinbuchstaben
- `/[0-9]/` findet eine beliebige Zahl



Ein oder mehr Zeichen

Einführung in
Regular
Expressions

Thomas
Helmke

Einleitung

Grundlagen

Zeichenauswahl

Zeichen zählen

Fortgeschrittene
Techniken

Variablen

Weitere Infos

- `/thomas/` findet „thomas“ als wort
- `/[thomas]/` findet „t“ oder „h“ oder „o“ oder „m“ oder „a“ oder „s“
- `/[a-z]/` findet einen beliebigen Kleinbuchstaben
- `/[0-9]/` findet eine beliebige Zahl



Ein oder mehr Zeichen

Einführung in
Regular
Expressions

Thomas
Helmke

Einleitung

Grundlagen

Zeichenauswahl

Zeichen zählen

Fortgeschrittene
Techniken

Variablen

Weitere Infos

- `/thomas/` findet „thomas“ als wort
- `/[thomas]/` findet „t“ oder „h“ oder „o“ oder „m“ oder „a“ oder „s“
- `/[a-z]/` findet einen beliebigen Kleinbuchstaben
- `/[0-9]/` findet eine beliebige Zahl



Vordefinierte Zeichengruppen

Einführung in
Regular
Expressions

Thomas
Helmke

Einleitung

Grundlagen

Zeichenauswahl

Zeichen zählen

Fortgeschrittene
Techniken

Variablen

Weitere Infos

- `/\d/` findet eine beliebige Zahl (digit)
- `/\w/` findet Zahlen, Buchstaben und Unterstriche (word)
- `/\s/` findet Leerzeichen, Tabs, Zeilenumbruch etc. (whitespace)
- Großbuchstabe funktioniert als Negierung



Vordefinierte Zeichengruppen

Einführung in
Regular
Expressions

Thomas
Helmke

Einleitung

Grundlagen

Zeichenauswahl

Zeichen zählen

Fortgeschrittene

Techniken

Variablen

Weitere Infos

- `/\d/` findet eine beliebige Zahl (digit)
- `/\w/` findet Zahlen, Buchstaben und Unterstriche (word)
- `/\s/` findet Leerzeichen, Tabs, Zeilenumbruch etc. (whitespace)
- Großbuchstabe funktioniert als Negierung



Vordefinierte Zeichengruppen

Einführung in
Regular
Expressions

Thomas
Helmke

Einleitung

Grundlagen

Zeichenauswahl

Zeichen zählen

Fortgeschrittene
Techniken

Variablen

Weitere Infos

- `/\d/` findet eine beliebige Zahl (digit)
- `/\w/` findet Zahlen, Buchstaben und Unterstriche (word)
- `/\s/` findet Leerzeichen, Tabs, Zeilenumbruch etc. (whitespace)
- Großbuchstabe funktioniert als Negierung



Vordefinierte Zeichengruppen

Einführung in
Regular
Expressions

Thomas
Helmke

Einleitung

Grundlagen

Zeichenauswahl

Zeichen zählen

Fortgeschrittene

Techniken

Variablen

Weitere Infos

- `/\d/` findet eine beliebige Zahl (digit)
- `/\w/` findet Zahlen, Buchstaben und Unterstriche (word)
- `/\s/` findet Leerzeichen, Tabs, Zeilenumbruch etc. (whitespace)
- Großbuchstabe funktioniert als Negierung



Einführung in
Regular
Expressions

Thomas
Helmke

Einleitung

Grundlagen

Zeichenauswahl

Zeichen zählen

Fortgeschrittene
Techniken

Variablen

Weitere Infos

Zeichen zählen



Quantifier

Einführung in
Regular
Expressions

Thomas
Helmke

Einleitung

Grundlagen

Zeichenauswahl

Zeichen zählen

Fortgeschrittene
Techniken

Variablen

Weitere Infos

- `/a?/` null- oder einmal
- `/a+/` mindestens einmal
- `/a*/` beliebig oft
- `/a{n}/` genau n-mal
- `/a{n,}/` mindestens n-mal
- `/a{n,m}/` mindestens n-mal, maximal m-mal
- `/a{0,m}/` maximal m-mal



Quantifier

Einführung in
Regular
Expressions

Thomas
Helmke

Einleitung

Grundlagen

Zeichenauswahl

Zeichen zählen

Fortgeschrittene
Techniken

Variablen

Weitere Infos

- `/a?/` null- oder einmal
- `/a+/` mindestens einmal
- `/a*/` beliebig oft
- `/a{n}/` genau n-mal
- `/a{n,}/` mindestens n-mal
- `/a{n,m}/` mindestens n-mal, maximal m-mal
- `/a{0,m}/` maximal m-mal



Quantifier

Einführung in
Regular
Expressions

Thomas
Helmke

Einleitung

Grundlagen

Zeichenauswahl

Zeichen zählen

Fortgeschrittene
Techniken

Variablen

Weitere Infos

- `/a?/` null- oder einmal
- `/a+/` mindestens einmal
- `/a*/` beliebig oft
- `/a{n}/` genau n-mal
- `/a{n,}/` mindestens n-mal
- `/a{n,m}/` mindestens n-mal, maximal m-mal
- `/a{0,m}/` maximal m-mal



- `/a?/` null- oder einmal
- `/a+/` mindestens einmal
- `/a*/` beliebig oft
- `/a{n}/` genau n-mal
- `/a{n,}/` mindestens n-mal
- `/a{n,m}/` mindestens n-mal, maximal m-mal
- `/a{0,m}/` maximal m-mal



Quantifier

Einführung in
Regular
Expressions

Thomas
Helmke

Einleitung

Grundlagen

Zeichenauswahl

Zeichen zählen

Fortgeschrittene
Techniken

Variablen

Weitere Infos

- `/a?/` null- oder einmal
- `/a+/` mindestens einmal
- `/a*/` beliebig oft
- `/a{n}/` genau n-mal
- `/a{n,}/` mindestens n-mal
- `/a{n,m}/` mindestens n-mal, maximal m-mal
- `/a{0,m}/` maximal m-mal



Quantifier

Einführung in
Regular
Expressions

Thomas
Helmke

Einleitung

Grundlagen

Zeichenauswahl

Zeichen zählen

Fortgeschrittene
Techniken

Variablen

Weitere Infos

- `/a?/` null- oder einmal
- `/a+/` mindestens einmal
- `/a*/` beliebig oft
- `/a{n}/` genau n-mal
- `/a{n,}/` mindestens n-mal
- `/a{n,m}/` mindestens n-mal, maximal m-mal
- `/a{0,m}/` maximal m-mal



Quantifier

Einführung in
Regular
Expressions

Thomas
Helmke

Einleitung

Grundlagen

Zeichenauswahl

Zeichen zählen

Fortgeschrittene
Techniken

Variablen

Weitere Infos

- `/a?/` null- oder einmal
- `/a+/` mindestens einmal
- `/a*/` beliebig oft
- `/a{n}/` genau n-mal
- `/a{n,}/` mindestens n-mal
- `/a{n,m}/` mindestens n-mal, maximal m-mal
- `/a{0,m}/` maximal m-mal



Lazy or Greedy?

Einführung in
Regular
Expressions

Thomas
Helmke

Einleitung

Grundlagen

Zeichenauswahl

Zeichen zählen

Fortgeschrittene
Techniken

Variablen

Weitere Infos

- `/a*/` Greedy Quantifier: findet so viele Zeichen wie möglich
- `/a*/?` Lazy Quantifier: findet so wenig Zeichen wie möglich



Lazy or Greedy?

Einführung in
Regular
Expressions

Thomas
Helmke

Einleitung

Grundlagen

Zeichenauswahl

Zeichen zählen

Fortgeschrittene
Techniken

Variablen

Weitere Infos

- `/a*/` Greedy Quantifier: findet so viele Zeichen wie möglich
- `/a*?/` Lazy Quantifier: findet so wenig Zeichen wie möglich



Einführung in
Regular
Expressions

Thomas
Helmke

Einleitung

Grundlagen

Zeichenauswahl

Zeichen zählen

Fortgeschrittene
Techniken

Variablen

Weitere Infos

Fortgeschrittene Techniken



Einführung in
Regular
Expressions

Thomas
Helmke

Einleitung

Grundlagen

Zeichenauswahl

Zeichen zählen

Fortgeschrittene
Techniken

Variablen

Weitere Infos

Variablen



Variablen I

Einführung in
Regular
Expressions

Thomas
Helmke

Einleitung

Grundlagen

Zeichenauswahl

Zeichen zählen

Fortgeschrittene
Techniken

Variablen

Weitere Infos

- Werte können in Variablen gespeichert werden
- Variablen können mathematisch verrechnet werden

```
xsize = 3.5;  
ysize = xsize;  
zsize = 0.5*xsize;
```



Variablen I

Einführung in
Regular
Expressions

Thomas
Helmke

Einleitung

Grundlagen

Zeichenauswahl

Zeichen zählen

Fortgeschrittene
Techniken

Variablen

Weitere Infos

- Werte können in Variablen gespeichert werden
- Variablen können mathematisch verrechnet werden

```
xsize = 3.5;  
ysize = xsize;  
zsize = 0.5*xsize;
```




Variablen I

Einführung in
Regular
Expressions

Thomas
Helmke

Einleitung

Grundlagen

Zeichenauswahl

Zeichen zählen

Fortgeschrittene
Techniken

Variablen

Weitere Infos

- Werte können in Variablen gespeichert werden
- Variablen können mathematisch verrechnet werden

```
xsize = 3.5;  
ysize = xsize;  
zsize = 0.5*xsize;
```



Einführung in
Regular
Expressions

Thomas
Helmke

Einleitung

Grundlagen

Zeichenauswahl

Zeichen zählen

Fortgeschrittene
Techniken

Variablen

Weitere Infos

Weitere Infos



Weitere Infos

Einführung in
Regular
Expressions

Thomas
Helmke

Einleitung

Grundlagen

Zeichenauswahl
Zeichen zählen

Fortgeschrittene
Techniken

Variablen

Weitere Infos

- <https://github.com/Syralist/hshb-pres-regex>
- <https://regex101.com/>
- <https://regexr.com/>