Melissa Wei

Regex-Ed

Gee Li

Regex

Some people, when confronted with a problem, think

"I know, I'll use regular expressions."

Now they have two problems.

-Jamie Zawinski

Why Should I Care?

1. Wildcards * on Steroids
    - Regex can match almost anything
    - Perhaps 90% of what you do is 90% text processing.
2. Write Better Code
    - Parse strings easily to find or clean data with fewer lines of code.
    - It's faster to write and easier to debug and maintain. Honest!
3. Relevant Across Multiple Programs
    - Supported by TONS of programs (Java, Perl, Python, Ruby, PHP)
    - Many databases offer built-in regex features

ELI5: wtf is Regex??

Regex stands for "Regular Expression" and it is "a sequence of characters that define a search pattern." In other words, regular expressions are a way to say

"**match things** that show up **in a particular order**"

RegEx Character Class

Literal Characters:	[\ ^ $. | ? * + ( ) [{
Character Class:	[aeiou] [0-9] [0-9a-fA-F]
Shorthand Class:	\d [digit] \w [alphnum]| \s [ws]
Anchoring:	^ [start] | $ [end]
Grouping & Capturing:	set(?:value)?
Repetition:	[1-9][0-9]{3}

Regex & Ruby: PigLatinize

When words begin with consonant clusters, the whole sound is added to the end. How do we split a word on the first vowel that appears?

```
word = "skidamarink"
word.split(/([aeiou].*)/)
=> ["sk", "idamarink"]
```

Regex & Ruby: PigLatinize

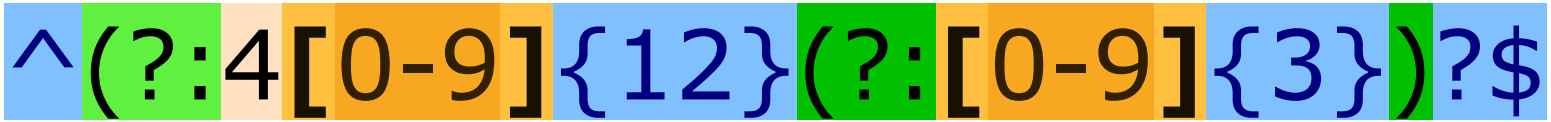
```
word.split(/([aeiou].*)/)
```

Regex between the `/`'s
`[aeiou]` look for the first instance of one of these characters
`.` matches any single character
`*` match 0 or more
`(...)` capture everything enclosed

[abc]	A single character of: a, b, or c	.	Any single character	(...)	Capture everything enclosed
[^abc]	Any single character except: a, b, or c	\s	Any whitespace character	(a|b)	a or b
[a-z]	Any single character in the range a-z	\S	Any non-whitespace character	a?	Zero or one of a
[a-zA-Z]	Any single character in the range a-z or A-Z	\d	Any digit	a*	Zero or more of a
^	Start of line	\D	Any non-digit	a+	One or more of a
$	End of line	\w	Any word character (letter, number, underscore)	a{3}	Exactly 3 of a
\A	Start of string	\W	Any non-word character	a{3,}	3 or more of a
\z	End of string	\b	Any word boundary	a{3,6}	Between 3 and 6 of a

options: i case insensitive m make dot match newlines x ignore whitespace in regex o perform #{...} substitutions only once

RegEx: Fun and Profit



What the hell is this?

RegEx: Validating Credit Card Numbers

Visa

```
^(?:4[0-9]{12}(?:[0-9]{3})?$
```

- Verifying a sequence of numbers
- Starts with 4
- Old: 13 digits | New: 16 Digits

RegEx: Validating Credit Card Numbers

```
^4[0-9]{12}(?:[0-9]{3})?$
```

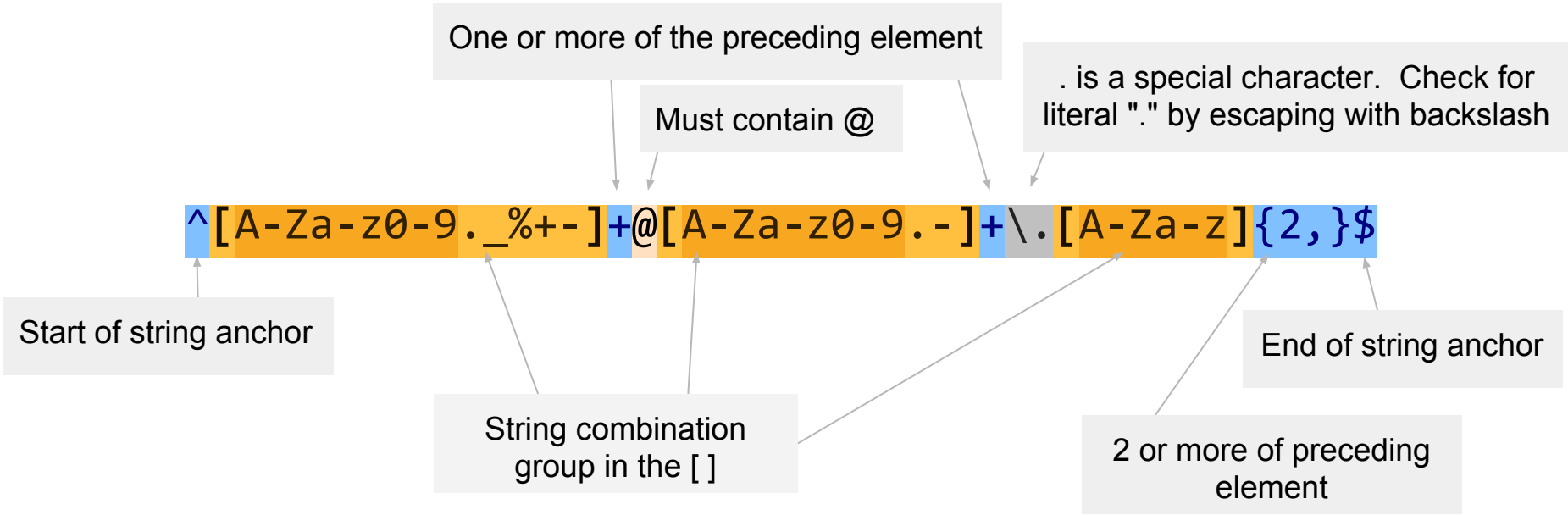
- `^4` Anchor matches start of string; number 4 start
- **1st Group:** `[0-9]`
  - Digits 0-9 `{12}` Repetition 12 times
- **2nd Group:** `(?:[0-9]{3})?`
  - `(?:)` Non-capturing group.
  - `?` Makes previous group optional.
  - `$` End of the matched group.

RegEx: Validating Credit Card Numbers

```
^(?:5[1-5][0-9]{2}|222[1-9]|22[3-9][0-9]|2[3-6]
[0-9]{2}|27[01][0-9]|2720)[0-9]{12}$
```

Understanding RegEx? Priceless

Email Validation



Resources

http://www.regular-expressions.info/

http://devdocs.io/ruby~2.3/regexp

https://ruby-doc.org/core-1.9.3/Regexp.html

http://www.explainxkcd.com/