Exploring Regular Expression Usage and Context in Python

Carl Chapman, Kathryn T. Stolee*

Iowa State University, North Carolina State University

carlallenchapman@gmail.com, ktstolee@ncsu.edu

19 July, 2016

Why Regular Expressions?

- Regexes are everywhere! (we think...)

Why Regular Expressions?

- Regexes are everywhere! (we think...)
- Everyone writes regexes! (we think...)

Why Regular Expressions?

- Regexes are everywhere! (we think...)
- Everyone writes regexes! (we think...)
- Regexes are hard to read/write! (again, we think...)

Why Regular Expressions?

- So, we wanted to write a tool to support regex creation.

Why Regular Expressions?

- So, we wanted to write a tool to support regex creation.
- But...

Why Regular Expressions?

- So, we wanted to write a tool to support regex creation.
- But...

Regex feature usage references are missing!

Why Regular Expressions?

- So, we wanted to write a tool to support regex creation.
- But...

Regex feature usage references are missing!

- and...

Why Regular Expressions?

- So, we wanted to write a tool to support regex creation.
- But...

Regex feature usage references are missing!

- and...

We don't know how/when/why developers use regexes!

Research Goals

1. RQ1: In what contexts do professional developers use regular expressions?
2. RQ2: How is the re module used in Python projects?
3. RQ3: Which regular expression language features are most commonly used in Python?
4. RQ4: How behaviorally similar are regexes across projects?

Research Goals

1. RQ1: In what contexts do professional developers use regular expressions?
2. RQ2: How is the re module used in Python projects?
3. RQ3: Which regular expression language features are most commonly used in Python?
4. RQ4: How behaviorally similar are regexes across projects?

Regular Expressions: The Basics

- `(ab*c|yz*)$`
  - ✓ abbbbbbbc
  - ✓ y
  - ✓ abcy

Regular Expressions: The Basics

- `(ab*c|yz*)$`
  - ✓ abbbbbbbc
  - ✓ y
  - ✓ abcy
  - ✗ abcccc
  - ✗ yxw

Regular Expressions: The Basics

- `(ab*c|yz*)$`
  - ✓ abbbbbbbc
  - ✓ y
  - ✓ abcy
  - ✗ abcccc
  - ✗ yxw
- `(ab*c|yz*)`

Regular Expressions: The Basics

- `(ab*c|yz*)$`
  - ✓ abbbbbbbc
  - ✓ y
  - ✓ abcy
  - ✗ abcccc
  - ✗ yxw
- `(ab*c|yz*)`
  - ✓ abbbbbbbc
  - ✓ y
  - ✓ abcy
  - ✓ abcccc
  - ✓ yxw

In Python: Utilizations of the re module

function pattern flags

```
r1 = re.compile("(0|-?[1-9][0-9]*)$", re.MULTILINE)
```

function which function of the re module is called?

pattern string used to specify regex behavior

flags modifies the regex engine

Survey Context?

- 18 professional developers
- Small mobile payment management company
- 9 years average development experience

How frequently do developers use regexes?

- 50% – at least once per week
- Regexes are most frequently composed within command line and text editor tools
- 2 developers write more than 50 regexes in general programming languages (e.g., Java) annually
- Database queries using regexes were rare

Common regex activities

How often do you use regexes for...

Activity	Frequency
Locating content within a file or files	4.4
Capturing parts of strings	4.3
Parsing user input	4.0

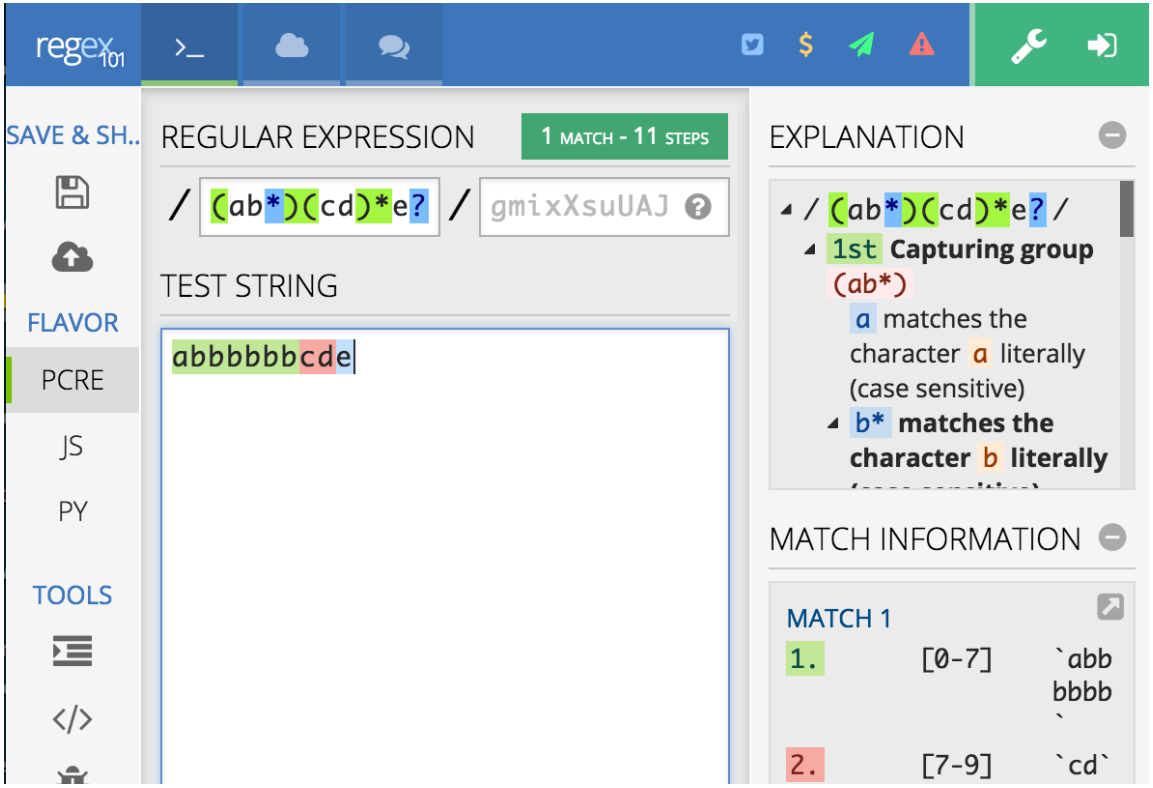
Key: 6 = very frequently, 5 = frequently, 4 = occasionally, 3 = rarely, 2 = very rarely, 1 = never

Testing regular expressions

Developers test regular expressions less often than other code.

Testing regular expressions

Developers test regular expressions less often than other code.



50% say they use testing tools like www.regex101.com

Pain Points

hard to compose (11 = 61%)

...very difficult to write them since I've never read up on them.

...trickiness to getting the expression right

Pain Points

hard to compose (11 = 61%)

...very difficult to write them since I've never read up on them.

...trickiness to getting the expression right

hard to read (7 = 39%)

long ones can be hard to read

Readability. Edge cases.

It is terrible to read (especially later after initial development)

Pain Points

hard to compose (11 = 61%)

...very difficult to write them since I've never read up on them.

...trickiness to getting the expression right

hard to read (7 = 39%)

long ones can be hard to read

Readability. Edge cases.

It is terrible to read (especially later after initial development)

inconsistency across implementations (3 = 17%)

Differences in implementation across languages

Some regexes work differently (or don't work) in some languages.

Recap

- Regexes are everywhere! (x% of Python projects studied)

Recap

- Regexes are everywhere! (x% of Python projects studied)
- Everyone writes regexes! (y% of surveyed developers write them weekly)

Recap

- Regexes are everywhere! (x% of Python projects studied)
- Everyone writes regexes! (y% of surveyed developers write them weekly)
- Regexes are hard to read/write! (this was reported as a pain point)

Recap

- Regexes are everywhere! (x% of Python projects studied)
- Everyone writes regexes! (y% of surveyed developers write them weekly)
- Regexes are hard to read/write! (this was reported as a pain point)

also...

- feature usage
- behavioral similarity
- many opportunities

Questions?

Katie Stolee – ktstolee@ncsu.edu