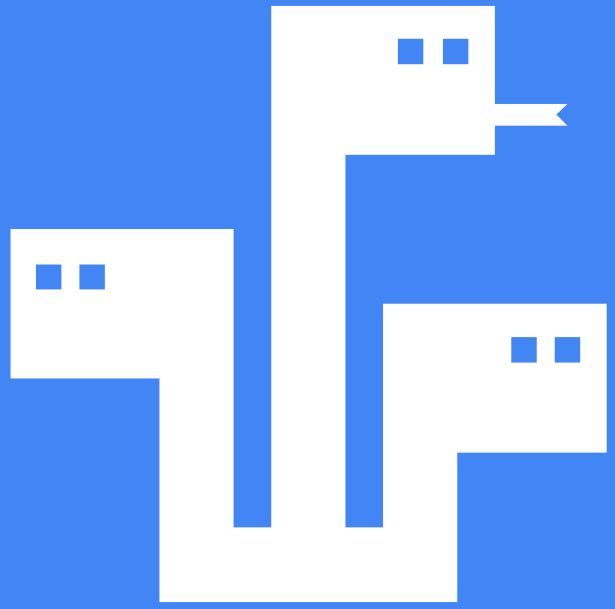
Regex

Regular **Ex**pressions

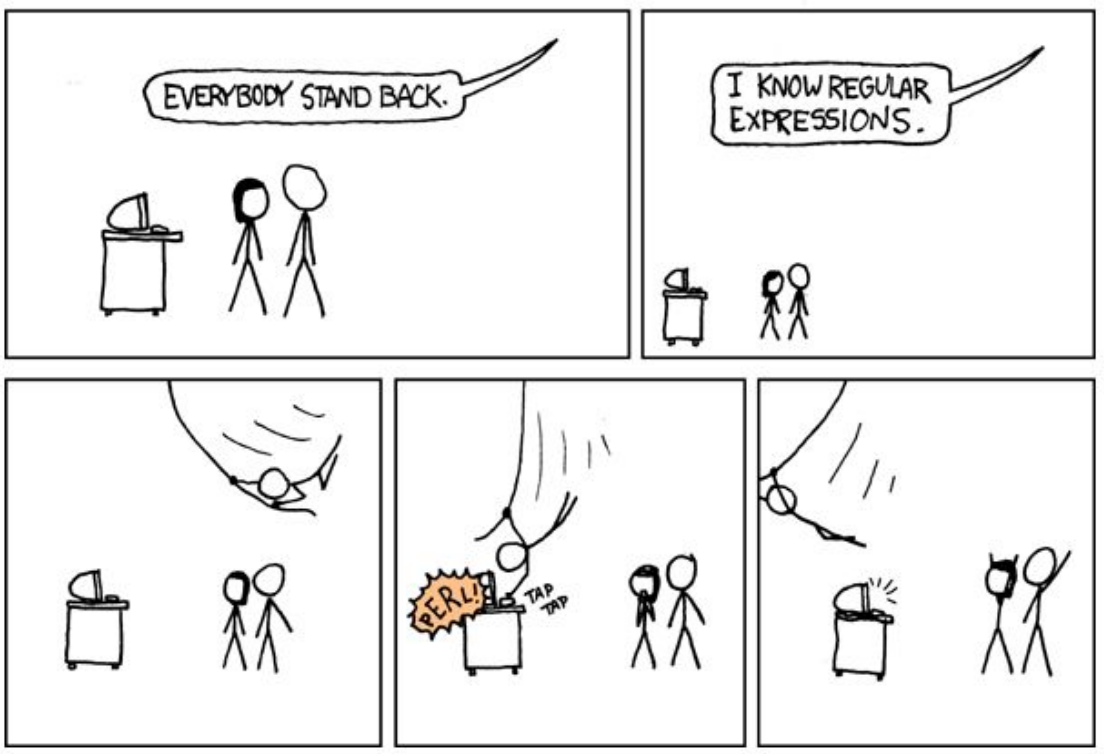
Python - Nick Reynolds
May 12, 2017



Regular expressions are a powerful language for matching text patterns.

Used in any language

- Data validation
- Web scraping
- Data wrangling
- Simple parsing
- *etc!*



Source: https://xkcd.com/208/

Basics

- Most of the time, characters match themselves!

```
import re

s = "Christmas is December 25th, that's soon!"

results = re.findall(r'a', s)
print(results)
# ['a', 'a']

results = re.findall(r'th', s)
print(results)
# ['th', 'th']
```

Square Brackets

- Sometimes you need a bit more structure though

```
[abcd]
```

Square brackets say, match any character in this set

```
[a-z0-9]
```

You can also specify ranges using dash

```
[^a-z]
```

Square brackets with a caret ^ at the start says ‘not in this set’

Sequences

- These make life easier, rather than specifying every digit

```
import re

s = "Christmas is December 25th, that's soon!"

results = re.findall(r'\d', s)
print(results)
# ['2', '5']
```

Character	Legend
\d	Any decimal digit
\D	Any non-digit character
\s	Any whitespace character
\S	Any non-whitespace character
\w	Any alphanumeric character
\W	Any non-alphanumeric character

Repeating Things

- Patterns aren't patterns without repetition

```
[0-9]+
```

Plus matches **one or more** occurrences

```
[0-9]*
```

Asterisk matches **zero or more** occurrences

```
[0-9]?
```

Question marks matches **zero or one** occurrences

```
[0-9]{1,2}
```

Braces are used for specifying **exactly** how many times something should repeat {min, max}

Brackets and or's

- Brackets can be used to separate logic i.e. create a group
- Or indicates the pattern must match one or the other

```
(abc)|(dfg)
```

Would match abc or dfg

```
(abc)|([0-9])
```

Would match abc or a single number

Time to string it all together!

Practical

References

- https://docs.python.org/3/howto/regex.html
- https://thenounproject.com/