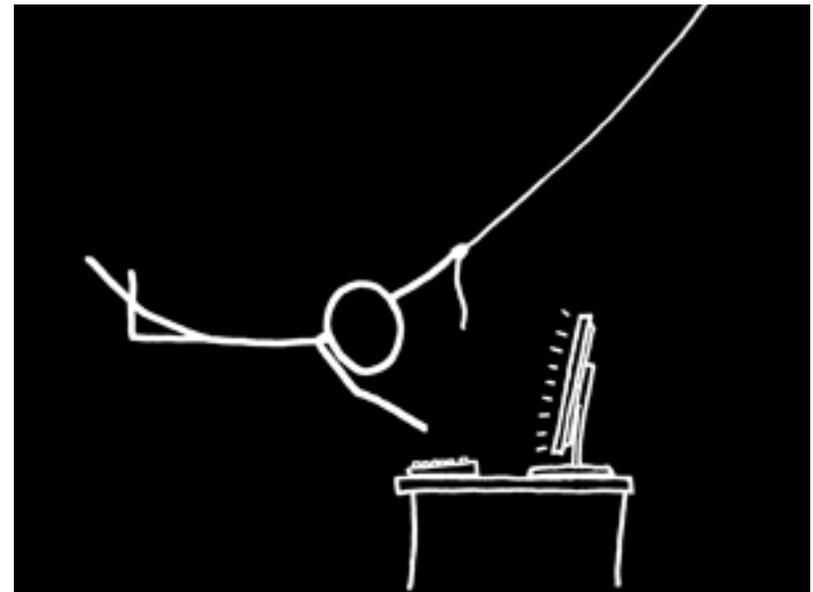

Regular Expressions

<http://xkcd.com/208/>

```
/Everybody stand back/  
I know regular expressions
```



Overview

- Regular expressions (a.k.a. regex, regexp or RE) are essentially a tiny, highly specialized programming language
 - embedded inside Python, Perl, Java, php and other languages
- Can use this little language to specify the rules for a pattern to match any set of possible strings
 - Sentences, e-mail addresses, ads, dialogs, etc
- “Does this string match the pattern?”, or “Is there a match for the pattern anywhere in this string?”
- Regular expressions can also be used as a language generator; regular expression languages are the first in the Chomsky hierarchy

Useful for Matching Text

- A language for specifying patterns in text
- Examples
 - Matching names like “Jane Q. Public”):
`/\b[A-Z] [a-z]+ +[A-Z] \. +[A-Z] [a-z]+\b/`
 - Matching all email addresses, with patterns like
.....@.....edu
.....@.....gov
.....@.....com
 - Matching all URLs
 - A fairly predictable set of characters & symbols selected from a finite set (e.g. a-z, www, http, ~, /)
 - And many others!

In these slides, we use the (Perl) convention that regular expressions are surrounded by / - Python uses “

Regular Expressions as a formal language

- In language theory, Regular Expressions specify a language that can be recognized by Finite State Automata
a.k.a. Finite Automaton, Finite State Machine, FSA or FSM
 - An abstract machine which can be used to implement regular expressions (etc.).
 - Has a finite number of states, and a finite amount of memory (i.e., the current state).
 - Can be represented by directed graphs or transition tables
- The regular languages are the first in the Chomsky hierarchy (context-free languages and context-sensitive languages are the next)
- Regular languages are exactly the set of languages recognized by finite automata

Introduction to the notation of RE

- Talk by Dan Jurafsky
- This introduction to RE is part of the publically available lectures from the Coursera course in Natural Language Processing with Dan Jurafsky and Chris Manning:
<https://class.coursera.org/nlp/lecture/index>