Zipf's law : frequency $\propto \frac{1}{\text{rank}}$

↓ word → most used ranking (the in 1)

Properties of language

Regular expression

Text Normalization

Edit Distance

3000 words will cover 95% of everyday writing - Robert Charles

Synsets

Stop words : Removing them, meaning remains same

Why regular expression ?

(

String search in Google doc - Primitive string result ✗

b : start

n : end

: String search ✓ → using regular exp.

re python library : regular exp.

>> import re

re.findall (regex, text) → find bubblegum

re.sub(regex, repl, text) → Replace (search substring)

s = "Jack is a boy"

Index of a

/a/_a

re.search("(^|\s)a", s)

↳ /a/_a

whitespace

re.search('a', s).span() → Return index of 'a' (only 1 instance)

findall + search → Return all indexes of 'a' (all instances)

[{"indices : m.span()" for m in (re.finditer('a', s))]

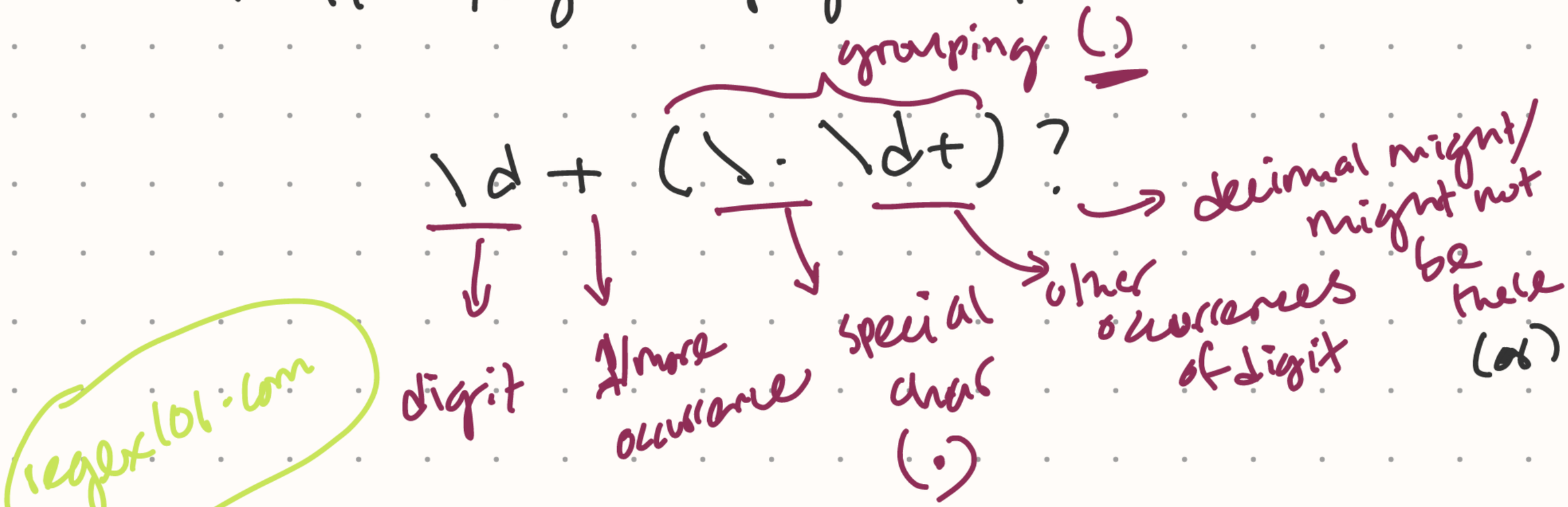
\d → any digit

\d+

metacharacters → For this reason regular exp is used instead of string search

s = "Price? 45 or 55.33?"

To specify any digit using regular exp → \d



re.findall("\d+(\.\d+)?", s) → Find all numbers from string s → 45 55.33

s = "Price? The price of 1 turkey is 30$."

only product prices → $

(\d+(\.\d+)?(?=\$))

Positive Lookahead ($30)

Negative lookahead → does not end w/ $ (unit)

Positive Lookbehind ($30)

Negative Lookbehind → Do not want sth that ends w/ sth ($)