

Lecture 14: Regular Expressions

Ling 1330/2330 Intro to Computational Linguistics
Na-Rae Han, 10/17/2023

Outline

- ▶ *Language and Computers*, Ch.4 Searching
 - ◆ 4.4 Searching semi-structured data with **regular expressions**
 - ◆ 4.41 Syntax of regular expressions

- ▶ Learning regular expressions
 - ◆ regex101 (real-time regex tester):
 - ◆ <https://regex101.com/>
 - ◆ Python Regex syntax reference:
<https://docs.python.org/3/library/re.html>
 - ◆ Regex tutorial:
https://gnosis.cx/publish/programming/regular_expressions.html
 - ◆ Na-Rae's Python 3 Notes on Regex:
<http://www.pitt.edu/~naraehan/python3/re.html>

Searching

- ▶ The perk of digital texts: they are *searchable*.
- ▶ The anti-perk of digital texts:
 - ◆ They often come in extremely large sizes.
 - ← Without means to search, they are unusable
 - ← Imagine the internet without Google/Bing...
- ▶ Searching in:
 - ◆ **Written texts:** is done, very efficiently
 - ◆ **Speeches:**
 - ◆ No native solution to searching in speech
 - ◆ Audio signals will first need to be converted to a text through speech recognition; and then search on written text

Searching for an expression

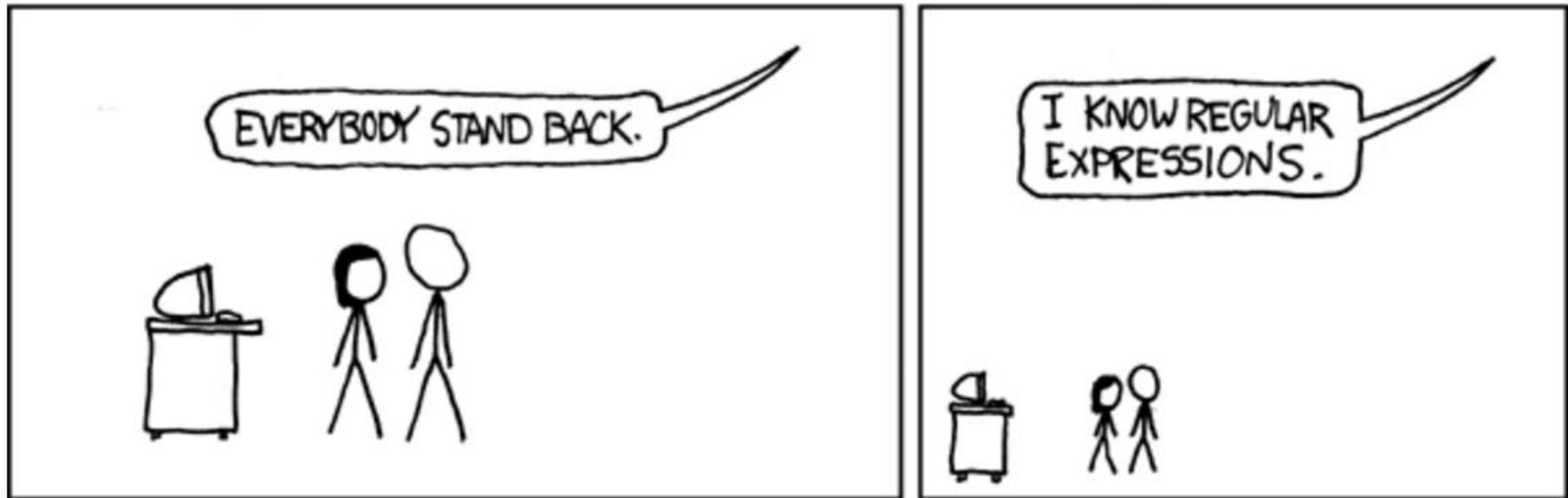
▶ Question:

- ◆ How would you find instances of *have been* in `austen-emma.txt`?
- ◆ How would you find *have been* along with its inflected varieties, i.e., *has been*, *had been*?
- ◆ You also want to allow *ever* or *never*, e.g., *has ever been*, *had never been*. How?
- ◆ More broadly, you want to find all instances of *have been*, with up to two words occurring between *have* and *been*. Can this be done with a single search?

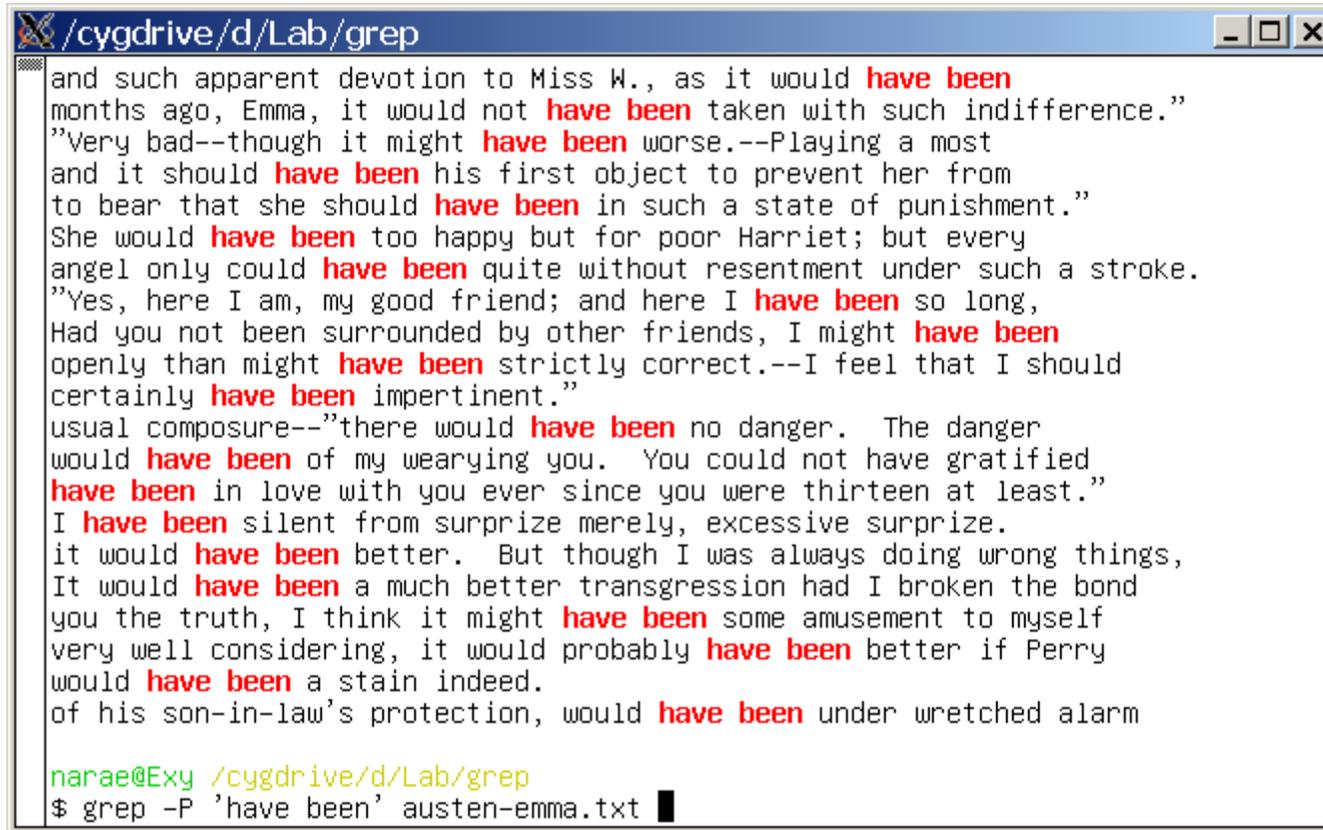
▶ Answer:

- ◆ YES, they can be done, using **regular expressions**.

-
- ▶ <https://www.explainkcd.com/wiki/index.php/208>: Regular Expressions



Searching, literally

A terminal window titled '/cygdrive/d/Lab/grep' showing the output of a grep command. The text is from 'austen-emma.txt' and contains several instances of the phrase 'have been' highlighted in red. The terminal prompt shows the user 'narae@Exy' and the command '\$ grep -P 'have been' austen-emma.txt'.

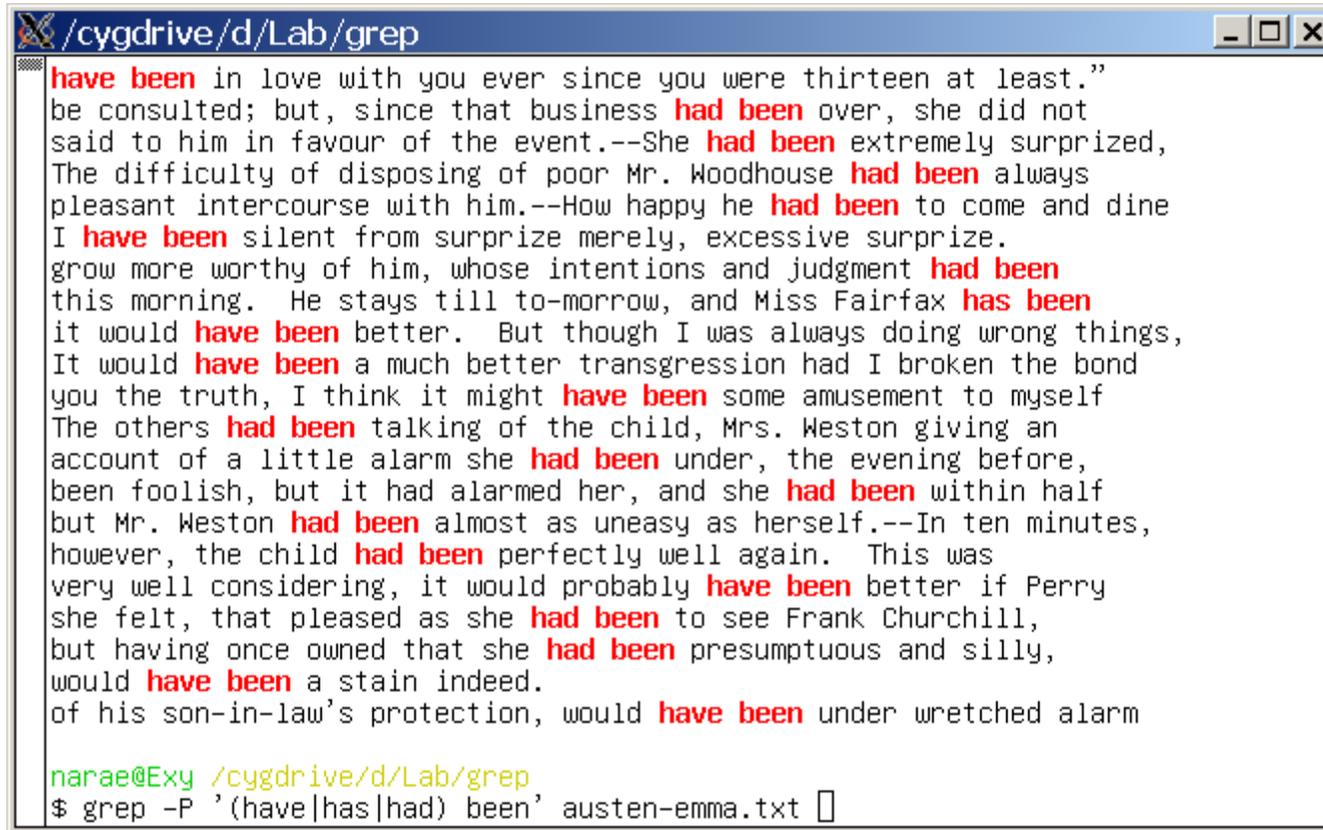
```
/cygdrive/d/Lab/grep
and such apparent devotion to Miss W., as it would have been
months ago, Emma, it would not have been taken with such indifference."
"Very bad--though it might have been worse.--Playing a most
and it should have been his first object to prevent her from
to bear that she should have been in such a state of punishment."
She would have been too happy but for poor Harriet; but every
angel only could have been quite without resentment under such a stroke.
"Yes, here I am, my good friend; and here I have been so long,
Had you not been surrounded by other friends, I might have been
openly than might have been strictly correct.--I feel that I should
certainly have been impertinent."
usual composure--"there would have been no danger. The danger
would have been of my wearying you. You could not have gratified
have been in love with you ever since you were thirteen at least."
I have been silent from surprize merely, excessive surprize.
it would have been better. But though I was always doing wrong things,
It would have been a much better transgression had I broken the bond
you the truth, I think it might have been some amusement to myself
very well considering, it would probably have been better if Perry
would have been a stain indeed.
of his son-in-law's protection, would have been under wretched alarm

narae@Exy /cygdrive/d/Lab/grep
$ grep -P 'have been' austen-emma.txt
```

`/have been/`

- ▶ *have been* as a literal string

'have been', 'has been', 'had been'



```
narae@Exy /cygdrive/d/Lab/grep
$ grep -P '(have|has|had) been' austen-emma.txt
have been in love with you ever since you were thirteen at least."
be consulted; but, since that business had been over, she did not
said to him in favour of the event.--She had been extremely surprized,
The difficulty of disposing of poor Mr. Woodhouse had been always
pleasant intercourse with him.--How happy he had been to come and dine
I have been silent from surprize merely, excessive surprize.
grow more worthy of him, whose intentions and judgment had been
this morning. He stays till to-morrow, and Miss Fairfax has been
it would have been better. But though I was always doing wrong things,
It would have been a much better transgression had I broken the bond
you the truth, I think it might have been some amusement to myself
The others had been talking of the child, Mrs. Weston giving an
account of a little alarm she had been under, the evening before,
been foolish, but it had alarmed her, and she had been within half
but Mr. Weston had been almost as uneasy as herself.--In ten minutes,
however, the child had been perfectly well again. This was
very well considering, it would probably have been better if Perry
she felt, that pleased as she had been to see Frank Churchill,
but having once owned that she had been presumptuous and silly,
would have been a stain indeed.
of his son-in-law's protection, would have been under wretched alarm

narae@Exy /cygdrive/d/Lab/grep
$ grep -P '(have|has|had) been' austen-emma.txt
```

`/(have|has|had) been/`

- ▶ Allows inflected forms of *have*

Include *never* or *ever*



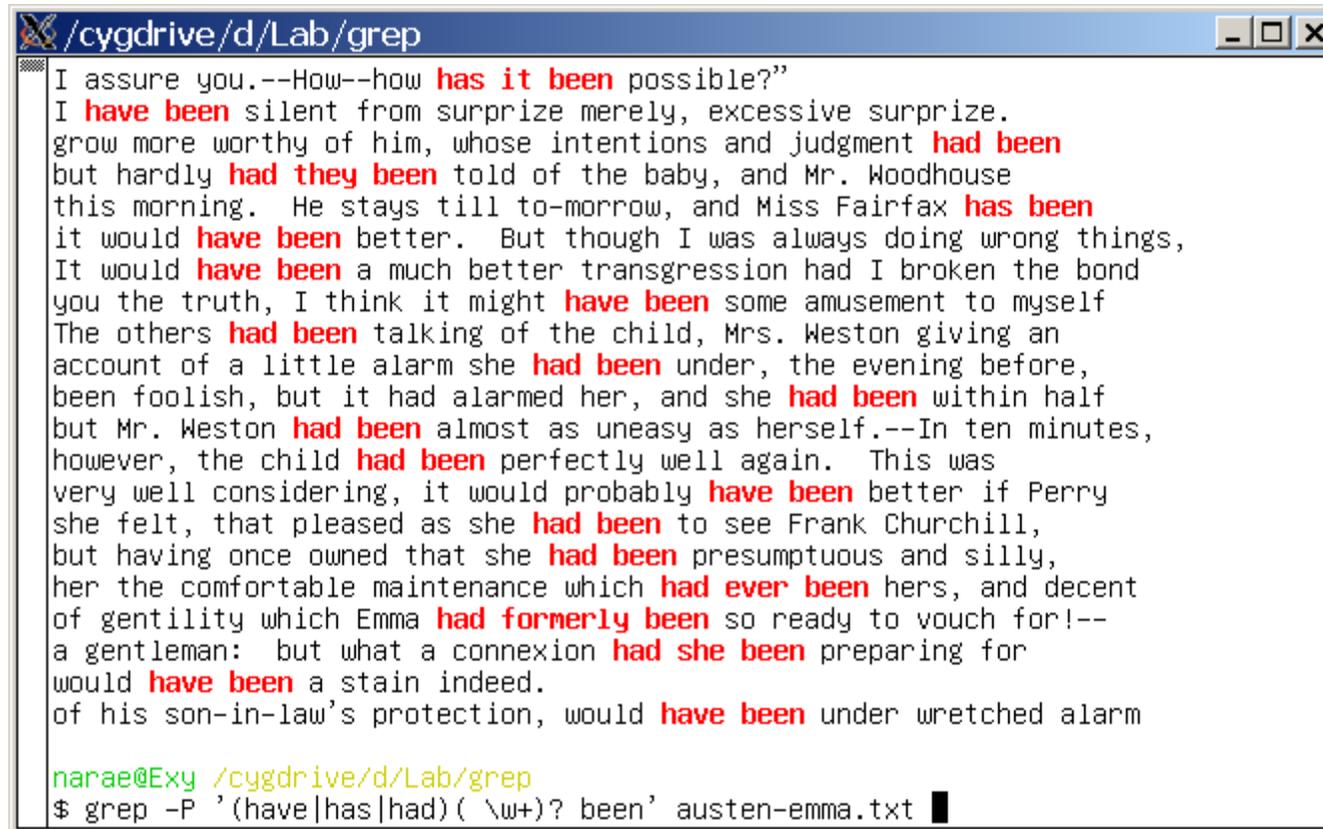
```
/cygdrive/d/Lab/grep
said to him in favour of the event.--She had been extremely surprized,
The difficulty of disposing of poor Mr. Woodhouse had been always
by herself--but even he had never been able to finish the subject
pleasant intercourse with him.--How happy he had been to come and dine
I have been silent from surprize merely, excessive surprize.
grow more worthy of him, whose intentions and judgment had been
this morning. He stays till to-morrow, and Miss Fairfax has been
it would have been better. But though I was always doing wrong things,
It would have been a much better transgression had I broken the bond
you the truth, I think it might have been some amusement to myself
The others had been talking of the child, Mrs. Weston giving an
account of a little alarm she had been under, the evening before,
been foolish, but it had alarmed her, and she had been within half
but Mr. Weston had been almost as uneasy as herself.--In ten minutes,
however, the child had been perfectly well again. This was
very well considering, it would probably have been better if Perry
she felt, that pleased as she had been to see Frank Churchill,
but having once owned that she had been presumptuous and silly,
her the comfortable maintenance which had ever been hers, and decent
would have been a stain indeed.
of his son-in-law's protection, would have been under wretched alarm

narae@Exy /cygdrive/d/Lab/grep
$ grep -P '(have|has|had)( n?ever)? been' austen-emma.txt
```

`/(have|has|had)(n?ever)? been/`

- ▶ Allows *never* or *ever* to intervene (along with a space!)

Any word in between



```
/cygdrive/d/Lab/grep
I assure you.--How--how has it been possible?"
I have been silent from surprize merely, excessive surprize.
grow more worthy of him, whose intentions and judgment had been
but hardly had they been told of the baby, and Mr. Woodhouse
this morning. He stays till to-morrow, and Miss Fairfax has been
it would have been better. But though I was always doing wrong things,
It would have been a much better transgression had I broken the bond
you the truth, I think it might have been some amusement to myself
The others had been talking of the child, Mrs. Weston giving an
account of a little alarm she had been under, the evening before,
been foolish, but it had alarmed her, and she had been within half
but Mr. Weston had been almost as uneasy as herself.--In ten minutes,
however, the child had been perfectly well again. This was
very well considering, it would probably have been better if Perry
she felt, that pleased as she had been to see Frank Churchill,
but having once owned that she had been presumptuous and silly,
her the comfortable maintenance which had ever been hers, and decent
of gentility which Emma had formerly been so ready to vouch for!--
a gentleman: but what a connexion had she been preparing for
would have been a stain indeed.
of his son-in-law's protection, would have been under wretched alarm

narae@Exy /cygdrive/d/Lab/grep
$ grep -P '(have|has|had)( \w+)? been' austen-emma.txt
```

`/(have|has|had)(\w+)? been/`

- ▶ Allows any single word (along with a space) to intervene

More intervening words

```
/cygdrive/d/Lab/grep
for much to have been done, even had his time been longer.--He had
the declaration, that had I not been convinced of her indifference,
never have allowed me to send it, had any choice been given her.--
had _you_ not been in the case--I should still have distrusted him.”

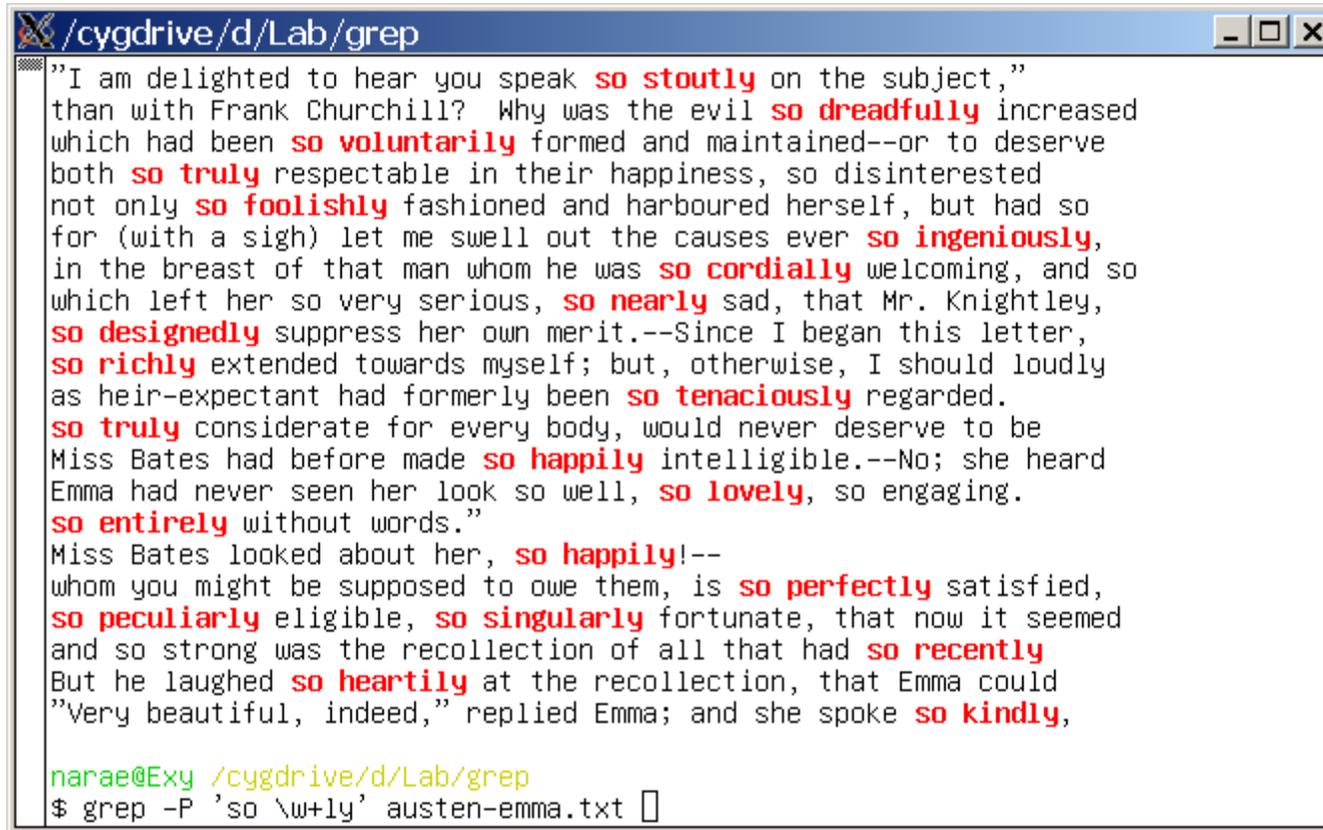
narae@Exy /cygdrive/d/Lab/grep
$ grep -P '(have|has|had)( \w+){2,4} been' austen-emma.txt
Sixteen years had Miss Taylor been in Mr. Woodhouse's family,
Harriet, she found, had never in her life been within side the Vicarage,
Perhaps she might have passed over more had his manners been
at this moment; never had his smile been stronger, nor his eyes
openly as he might have done had her father been out of the room,
to her and the most soothing to him, had in all likelihood been
No second meeting had there yet been between him and Emma.
How the trampers might have behaved, had the young ladies been
to see what had not yet been seen, the old Abbey fish-ponds;
mind as every thing had so long been, and was very much pleased
for much to have been done, even had his time been longer.--He had
the declaration, that had I not been convinced of her indifference,
never have allowed me to send it, had any choice been given her.--
had _you_ not been in the case--I should still have distrusted him.”

narae@Exy /cygdrive/d/Lab/grep
$ grep -P '(have|has|had)( \w+){2,4} been' austen-emma.txt
```

`/(have|has|had)(\w+){2,4} been/`

- ▶ With 2-4 intervening words (along with a space!)

That is so ...ly

A screenshot of a terminal window with a blue title bar containing the path `/cygdrive/d/Lab/grep`. The terminal displays the output of a `grep` command. The text is from Jane Austen's *Emma* and contains several words ending in `-ly` that have been highlighted in red. At the bottom of the terminal, the prompt `narae@Exy /cygdrive/d/Lab/grep` is shown, followed by the command `$ grep -P 'so \w+ly' austen-emma.txt` and a cursor.

```
/cygdrive/d/Lab/grep
"I am delighted to hear you speak so stoutly on the subject,"
than with Frank Churchill? Why was the evil so dreadfully increased
which had been so voluntarily formed and maintained--or to deserve
both so truly respectable in their happiness, so disinterested
not only so foolishly fashioned and harboured herself, but had so
for (with a sigh) let me swell out the causes ever so ingeniously,
in the breast of that man whom he was so cordially welcoming, and so
which left her so very serious, so nearly sad, that Mr. Knightley,
so designedly suppress her own merit.--Since I began this letter,
so richly extended towards myself; but, otherwise, I should loudly
as heir-expectant had formerly been so tenaciously regarded.
so truly considerate for every body, would never deserve to be
Miss Bates had before made so happily intelligible.--No; she heard
Emma had never seen her look so well, so lovely, so engaging.
so entirely without words."
Miss Bates looked about her, so happily!--
whom you might be supposed to owe them, is so perfectly satisfied,
so peculiarly eligible, so singularly fortunate, that now it seemed
and so strong was the recollection of all that had so recently
But he laughed so heartily at the recollection, that Emma could
"Very beautiful, indeed," replied Emma; and she spoke so kindly,

narae@Exy /cygdrive/d/Lab/grep
$ grep -P 'so \w+ly' austen-emma.txt
```

`/so \w+ly/`

- ▶ so followed by a word ending in `-ly`

grep and regular expressions

▶ grep

- ◆ **Global Regular Expression Print**
 - ◆ A command-line utility that searches plain-text data for *lines* matching a regular expression pattern
 - ◆ Comes standard in Unix, Linux, Mac OS-X
 - ◆ Some ports available for Windows (install [git Bash](#))
 - ◆ Variants:
 - ◆ `egrep` ("extended", same as `grep -E`), `fgrep`
 - ◆ What I am using here is in fact `grep -P --color`
 - `-P` means perl-style regular expression notation, which is also what Python uses
- ```
grep -P '(have|has|had)(\w+)? been' austen-emma.txt
```
- `-P` is not available on **Macs**; use `grep -E` or `pcgrep` (perl-compatible re grep) instead

# Regular expressions

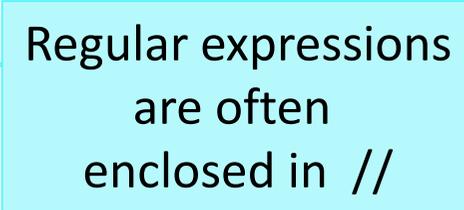
---

## ▶ Regular expression

- ◆ A compact representation of a set of strings

`/(have|has|had)( n?ever)? been/` describes:

- have been*
- has been*
- had been*
- have ever been*
- has ever been*
- had ever been*
- have never been*
- has never been*
- had never been*



Regular expressions  
are often  
enclosed in //

- ◆ The set of strings can be infinite in size.
- ◆ Serves as a pattern for search.

# Practice

## ► regex101

- ◆ A real-time regular expression tester
- ◆ <https://regex101.com/>
- ◆ Select "python" flavor →

### FLAVOR

</> pcre (php)

</> javascript

</> python ✓

Online regex tester and debugger - x +  
https://regex101.com

regular expressions @regex101 donate contact bug reports & feedback wiki

REGULAR EXPRESSION 21 matches, 142 steps (~93ms)  
r" \w{5,} " gm

TEST STRING SWITCH TO UNIT TESTS >  
Colorless wee-little green ideas sleep and eat furiously for the 1000th time.  
Colorless Wee-Little Green Ideas Sleep And Eat Furiously For The 1000th Time.  
COLORLESS WEE-LITTLE GREEN IDEAS SLEEP AND EAT FURIOUSLY FOR THE 1000TH TIME.

EXPLANATION  
▼ "\w{5,}" gm  
▼ \w{5,} matches any word character (equal to [a-zA-Z0-9\_])  
{5,} Quantifier — Matches between 5 and unlimited times, as many times as possible, giving back as needed (greedy)  
▼ Global pattern flags  
g modifier: global. All matches (don't return after first match)

MATCH INFORMATION  
Match 1  
Full match 0-9 `Colorless`  
Match 2  
Full match 14-20 `little`  
Match 3  
Full match 21-26 `green`  
Match 4

QUICK REFERENCE  
Search reference  
all tokens Any single character .  
common tokens Any whitespace character \s  
general tokens Any non-whitespace charac... \S  
Any digit \d

# Regex demo

---

▶ A snippet from 'Fox in Sox':

◆ <https://sites.pitt.edu/~naraehan/python3/text-samples.txt>

- ◆ /e/
- ◆ /ea/
- ◆ /ew/
- ◆ /e+/
- ◆ /ee|ea|ew/
- ◆ /e./
- ◆ /f.e/
- ◆ /[aeiou]/
- ◆ /[aeiou][aeiou]/
- ◆ /[aeiou]+/

- ◆ /[a-z]/
- ◆ /[A-Z]/
- ◆ /[A-Za-z]/
- ◆ /\w/
- ◆ /\W/
- ◆ /\s/
- ◆ /\S/
- ◆ /. /
- ◆ /.+ /



**NOT [A-z]!!**

# Regex demo

---

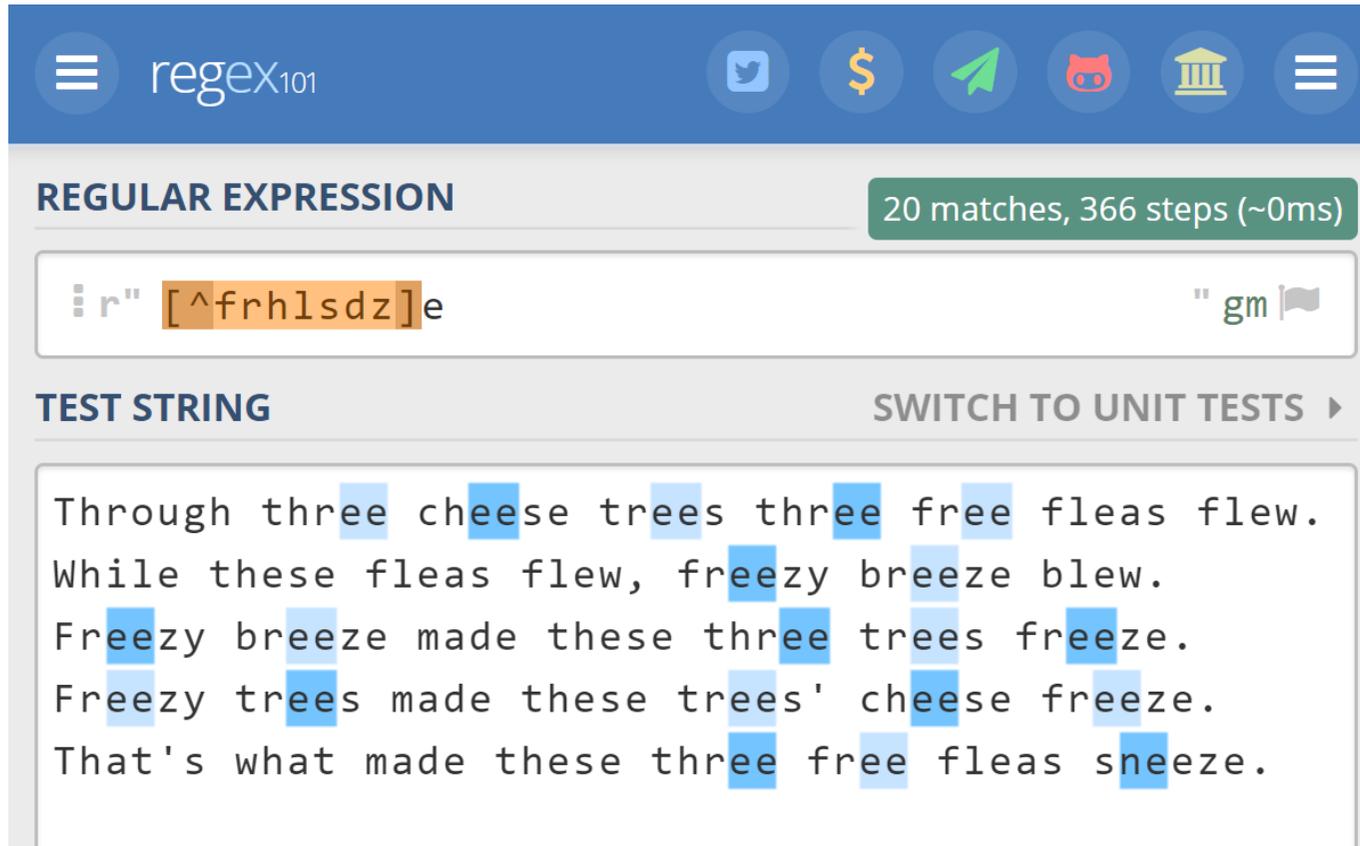
▶ A snippet from 'Fox in Sox':

◆ <https://sites.pitt.edu/~naraehan/python3/text-samples.txt>

- ◆ Words (no symbols) `/[A-Za-z]+/` or `/\w+/  
◆ Capitalized words /\b[A-Z]\w+/  
◆ Words ending in ee /\w*ee\b/  
◆ Words that contain ee /\w*ee\w*/  
◆ Words that do not contain e /\b[^e ]+\b/  
◆ Words that are 4 chars long /\b\w\w\w\w\b/ or /\b\w{4}\b/  
◆ e and any character before it other than f? and r? and h? and l? /[^frhl]e/`

If it matches  
newline, use `\s`  
instead of space

# Regexing with Dr. Seuss



The screenshot shows the regex101 website interface. At the top, there is a blue header with the site logo and several social media icons. Below the header, the main content area is divided into sections. The first section is titled "REGULAR EXPRESSION" and contains the regex `r"[^frhlsdz]e` with flags `gm`. To the right of the input field, a green badge indicates "20 matches, 366 steps (~0ms)". Below this is the "TEST STRING" section, which contains a Dr. Seuss poem. The words "three", "cheese", "trees", "free", and "fleas" are highlighted with blue boxes, indicating they are matches for the regex. The poem text is: "Through three cheese trees three free fleas flew. While these fleas flew, freezy breeze blew. Freezy breeze made these three trees freeze. Freezy trees made these trees' cheese freeze. That's what made these three free fleas sneeze."

# Syntax of regular expressions (1)

---

## ► Literals, concatenation, alternation

| RE            | What                       | Matches                 |
|---------------|----------------------------|-------------------------|
| <b>a</b>      | A single literal character | <i>a</i>                |
| <b>ab</b>     | Concatenation              | <i>ab</i>               |
| <b>ab xyz</b> | Alternation                | <i>ab</i> or <i>xyz</i> |

## ► A single character in a **set** []

| RE                           | What                 | Matches                                                                                      |
|------------------------------|----------------------|----------------------------------------------------------------------------------------------|
| <b>[aeiou]</b>               | Character set        | Any single character in the set, i.e., <i>a</i> , <i>e</i> , <i>i</i> , <i>o</i> or <i>u</i> |
| <b>[a-z]</b><br><b>[0-9]</b> | Character range      | Any single character in the range                                                            |
| <b>[^aeiou]</b>              | "Negative" character | Any single character that is NOT in the set                                                  |

# Syntax of regular expressions (2)

---

## ► Predefined character sets

| RE              | What                           | Matches                                          |
|-----------------|--------------------------------|--------------------------------------------------|
| <code>\d</code> | any digit                      | any single digit: <i>0, 1, 3, ..., 9</i>         |
| <code>\D</code> | any non-digit                  | any single char that's not one of above          |
| <code>\s</code> | any whitespace character       | space, tab, new-line character, etc.             |
| <code>\S</code> | any non-whitespace character   | any single char that's not one of the above      |
| <code>\w</code> | any alphanumeric character     | <i>a, b, A, Z, 0, 1, 9, _</i> (underscore)       |
| <code>\W</code> | any non-alphanumeric character | any single character that's not one of the above |

# Syntax of regular expressions (3)

---

## ▶ Any single character

| RE | What                                                           | Matches                  |
|----|----------------------------------------------------------------|--------------------------|
| .  | Any single character<br>except for the new line character '\n' | a, b, A, 1, 9, %, !, ... |

## ▶ Place indicators

- ◆ These have zero width– they do *not* match any character themselves

| RE | What                | Example matches |                                                                 |
|----|---------------------|-----------------|-----------------------------------------------------------------|
| ^  | Beginning of string | /^a/            | matches <i>a, ab, abc</i><br>does not match <i>ba, bac</i>      |
| \$ | End of string       | /a\$/           | matches <i>a, ba, bca</i><br>does not match <i>ab, bac</i>      |
| \b | Word boundary       | /ed\b/          | matches <i>ed</i> in 'worked', 'worked?'<br>but not 'education' |

# Syntax of regular expressions (4)

---

## ► Counters

| RE            | What                                    | Example matches          |                                       |
|---------------|-----------------------------------------|--------------------------|---------------------------------------|
| <b>?</b>      | Optionality: 0 or 1                     | <code>/n?ever/</code>    | <i>ever, never</i>                    |
| <b>*</b>      | Kleene star; any number (0 to infinity) | <code>/no*/</code>       | <i>n, no, noo, nooo, noooooo, ...</i> |
| <b>+</b>      | at least one (1 to infinity)            | <code>/no+/</code>       | <i>no, noo, nooo, nooooooo, ...</i>   |
| <b>{n}</b>    | exactly <i>n</i>                        | <code>/yes{3}/</code>    | <i>yesss</i>                          |
| <b>{n, }</b>  | at least <i>n</i>                       | <code>/yes{3, }/</code>  | <i>yesss, yessss, yessssss, ...</i>   |
| <b>{n, m}</b> | between <i>n</i> and <i>m</i>           | <code>/yes{2, 5}/</code> | <i>yess, yesss, yessss, yesssss</i>   |

# Syntax of regular expressions (5)

---

## ► Escaped characters

- ◆ Special characters in RE: ., ?, +, \*, (, ), [, ], {, }, -, |, ^, \$, \
- ◆ What if we need to match these characters, literally?
- ◆ Use a **backslash** "`\`" to escape

| RE               | What       | Matches                     |
|------------------|------------|-----------------------------|
| <code>\.</code>  | escaped .  | . (actual period character) |
| <code>\?</code>  | escaped ?  | ? (actual question marker)  |
| <code>\\$</code> | escaped \$ | \$ (actual dollar sign)     |
| <code>\\</code>  | escaped \  | \ (actual backslash)        |

# Operator precedence

---

▶ In algebra:

- ◆  $10 + 2 \times 3 = 16$  ← not 36.  $\times$  has precedence over  $+$
- ◆  $(10 + 2) \times 3 = 36$  ← precedence superseded using  $()$

▶ RE operators also have precedence.

- ◆  $/ab|cd/$  matches *ab* and *cd*
- ◆  $/a(b|c)d/$  matches *abd*, *acd*

← Alternation " $|$ " has the lowest operator precedence

← Good idea to use  $()$  whenever using  $|$

# Practice

---

- ▶ First two paragraphs from Abraham Lincoln's Wikipedia entry:
  - ◆ [https://en.wikipedia.org/wiki/Abraham\\_Lincoln](https://en.wikipedia.org/wiki/Abraham_Lincoln)
- ▶ Compose regular expressions for:
  1. Words ending with *-y*
  2. Words starting with a capital letter and ending in *-ed*
  3. Vowel character clusters (2+ vowels)
  4. Lincoln's names (full name or last name only)
  5. Numbers
  6. Years
  7. Numbers followed by alphabetic letter(s): 1930s, 16th
  8. Dates (January 1, 1999) or months (January 1999)
  9. Capitalized words
  10. *the* and its next word
  11. hyphenated words

# Practice

---

- ▶ First two paragraphs from Abraham Lincoln's Wikipedia entry:

- ◆ [https://en.wikipedia.org/wiki/Abraham\\_Lincoln](https://en.wikipedia.org/wiki/Abraham_Lincoln)

- ▶ Compose regular expressions for:

1. `/\w+y\b/` or `/[A-Za-z]+y\b/` (Word boundary `\b` is needed)

2. `/\b[A-Z][a-z]*ed\b/`

3. `/[aeiou][aeiou]+/` or `/[aeiou]{2,}/`

4. `/(Abraham )?Lincoln/`

5. `/\d+/` or `/[0-9]+/`

6. `/\d\d\d\d/` or `/\d{4}/` or `/[0-9]{4}/`

7. `/\d+[a-z]+/`

8. `/[A-Z][a-z]+( \d\d?,)? \d\d\d\d/`

9. `/[A-Z][a-z]+/`

10. `/the \w+/`

11. `/\w+-\w+/`

**`/\bthe .../` is  
more precise**

**Will over-match:  
"In 1860"**

**Will over-match:  
"1000000"**

# Wrapping up

---

## ▶ Next class:

- ◆ Regex in Python
  - ◆ <https://sites.pitt.edu/~naraehan/python3/re.html>
- ◆ FSA (Finite-State Automata)

## ▶ Exercise 7 out

- ◆ Regexing Steve Jobs!
- ◆ **With regex, there is a HIGH chance of your solution being wrong in some way without you realizing it. Make sure to study the EXERCISE KEY.**