

## **Outline**

Programming

- Regex

Practicalities

- Quiz review

Cool media art pieces

Prof. Angela Chang

Lecture 9: Regular Expressions

Fall 2017. Oct 4

# **CODE, CULTURE, AND PRACTICE**

# REGEX

Regular Expressions: search patterns  
language of symbols that describes a text pattern.

**Literal** matches- strings match exactly  
"The cow, camel, and cat communicate" -walk it through and backtracks

**Metacharacters--** characters with special meaning

. \* + [abc] ^ # ! ? (abc) \d(2,3)

They transform literal characters into powerful expressions. they can have more than one meaning (used in context

<http://regexpal.com>

# Investigating a file of text

Download Pride and Prejudice:

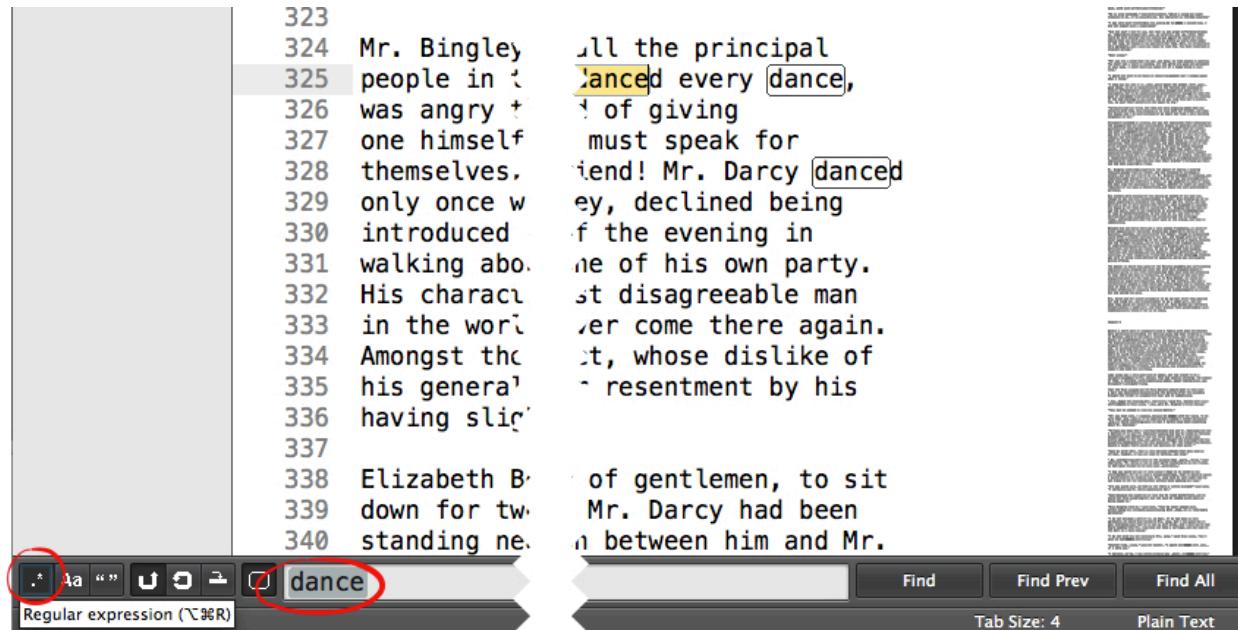
<http://www.gutenberg.org/files/1342/1342.txt>

you might have to visit this link twice to get the file

Save ASCII version to your computer (e.g. ⌘+S on Ctrl+S)

Open it in your text editor (Geany, TextWrangler, Notepad++, Sublime)

Find how many occurrences of the pattern ‘dance’ exists (⌘+F)

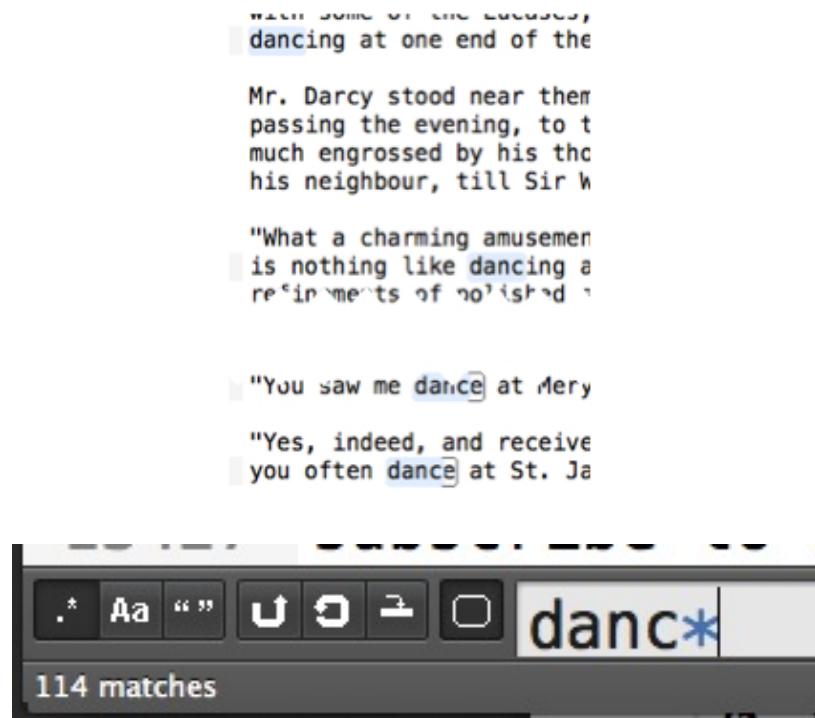


click regex search

64 occurrences of ‘dance’

# Wildcard character \*

type danc\* (asterisk) to get all verb forms and tenses of dance



\* means matching 0 or more times – dance, dancing, dances, danced, danc

# Simple text analysis

How vocalic is *Pride and Prejudice*?

vocalic | vō'kəlɪk, və- | adjective Phonetics  
of, relating to, or consisting of a vowel or |  
vowels.

Character class pattern [ ] - match any characters inside  
Define a character class of five vowels: **[aeiou]**



# Simple text analysis

How vocalic is *Pride and Prejudice*?

vocalic | vō'kälik, və- | adjective Phonetics  
of, relating to, or consisting of a vowel or |  
vowels.



case-sensitive search



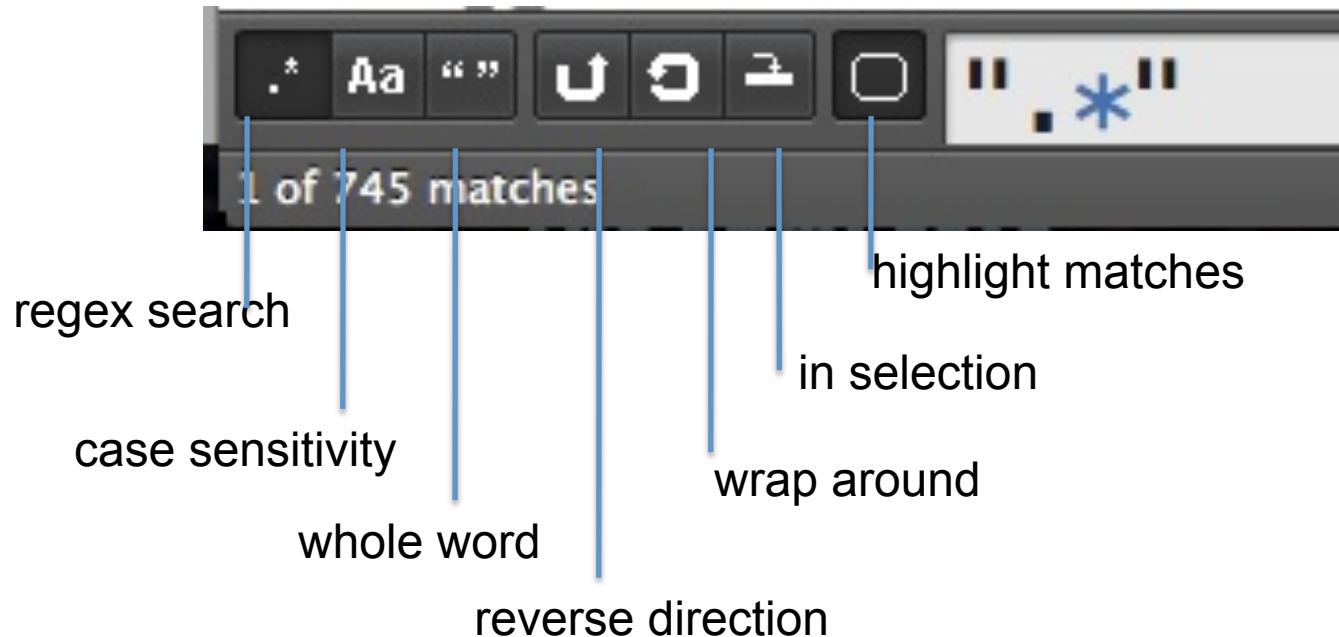
Case doesn't matter, so uncheck case sensitivity.

We get an extra 4,423 matches.

So there are 209,723 vowels in *Pride and Prejudice*.

# Play with other settings

Some settings will change your results, others won't.



# Comparing files

Download *Leaves of Grass*

<http://www.gutenberg.org/cache/epub/1322/pg1322.txt>  
<http://www.gutenberg.org/files/1322/1322.txt>

Pride and Prejudice

Aa " " ↗ ↙ ↘ ↙  
30 of 209723 matches

.\* Aa " " ↗  
114 matches

the ever  
Darcy  
extreme. St  
made unl  
.\* Aa " " ↗  
124592 matches

Leaves of Grass

" " ↗ ↙ ↘ ↙ [aeiou]  
208186 matches

.\* Aa " " | danc\*  
74 matches

(On earth and  
heavens, )  
Urging slowly  
And waiting e  
.\* Aa " " ↗ [^\s]+  
124732 matches

Which is more  
vocalic?  
[aeiou]

How many times is  
dancing mentioned?  
danc\*

Which is wordier?  
[^s]+

"+ 1 or more sequence of  
^\s non-whitespace letters"  
^ not  
\s whitespace  
\S non-whitespace letters  
so \S+ == [^\s]+

# Building up a regex string

A first pass searching for

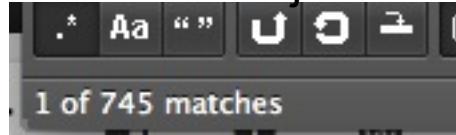
".\*"

"non newline, arbitrary characters inside straight double-quotation marked"

Create regex pattern for “one-line quotation”

1. Start and end with “ ” (double quote)
2. Something that’s not a newline newline .
3. Zero or more occurrences. \*

## Pride & Prejudice



"But it is not merely this affair," she continued, "on which my dislike is founded. Long before it had taken place my opinion of you was decided. Your character was unfolded in the recital which I received many months ago from Mr. Wickham. On this subject, what can you have to say? In what imaginary act of friendship can you here defend yourself? or under what misrepresentation can you here impose upon others?"

"You take an eager interest in that gentleman's concerns," said Darcy, in a less tranquil tone, and with a heightened colour.

"Who that knows what interest in him?"

"His misfortunes!" re have been great indee

"And of your inflictions,

This regex expression doesn’t work very well:

- Line breaks cause long quotations to be counted.
- Most of the quotes for Leaves of Grass are in Project Gutenberg text.

Also, this regex is greedy— “.\*” counts two quotes as one.

*These four words are indeed a double-quotation mark followed by a bunch of arbitrary characters, followed by a double quotation mark.*

## Leaves of Grass



(Control+G to go to line.6477 in Leaves of Grass.

6477	That matter of Troy and Achilles'
6478	Placard "Removed" and "To Let" or
6479	Repeat at Jerusalem, place the no
6480	Mount Moriah,

# Changing the quotation regex

A new one-line quotation search using a pattern that has:

1. “ (double quote mark) at the start
2. [ ^” ] Followed by *anything* that is not a double quote mark
  - \* zero or more occurrences any characters (except newline).
1. and ends with a double quote mark “

Pride & Prejudice

745 vs. 1776 matches

The screenshot shows a search interface with a toolbar at the top containing various icons. The search bar contains the regular expression pattern `"",*"`. Below the search bar, it says "401 of 745 matches". The main area displays a list of 745 matches, with the first few lines shown:

7490	"Blame you! Oh, no."
7491	"But you blame me for it."
7492	"But you blame me for it."
7493	"No--I do not know that."
7494	"But you _will_ know it day."

At the bottom of the interface, another search bar contains the pattern `"[^"]*"`, and it says "969 of 1776 matches".

Leaves of Grass

13 vs. 25 matches

The screenshot shows a search interface with a toolbar at the top containing various icons. The search bar contains the regular expression pattern `"",*"`. Below the search bar, it says "6 of 13 matches". The main area displays a list of 13 matches, with the first few lines shown:

16922	learning, intuitions
16923	"Of all Geologies--Histo
16924	Metaphysics all,
16925	all are onw
16926	"Life, life an endless m
16927	duly over,)
16928	"The world, the race, th
16929	"All bound as is befitti

At the bottom of the interface, another search bar contains the pattern `"[^"]*"`, and it says "6 of 25 matches".

2x more one-liners found with `"[^"]*"'`

# Using ^ to exclude characters

Placard "Removed" and "To Let"

Problem: The regex ".\*" will grab two quotes in a single line.

Solution: we should restrict what is inside the quotation marks so that it excludes double-quotes.

" [^"] \* "

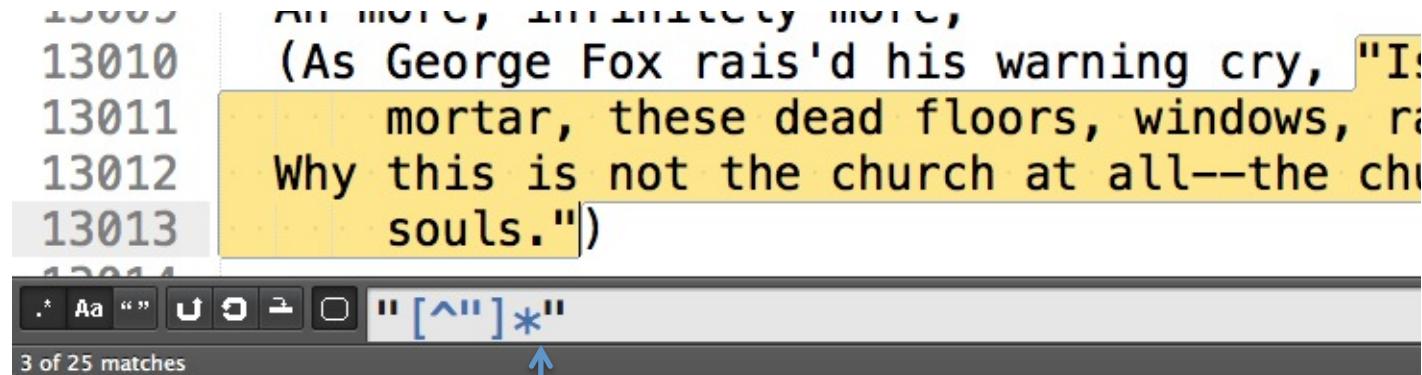
" start with a double quote  
[^"] ]any character that is NOT a double quote  
followed by zero or more non-whitespace characters  
" ending with a double quote

By excluding double-quotes inside our quotes, regex returns each quote separately. We go from 13 to 25 matches by being more specific.



# Multiline quotes

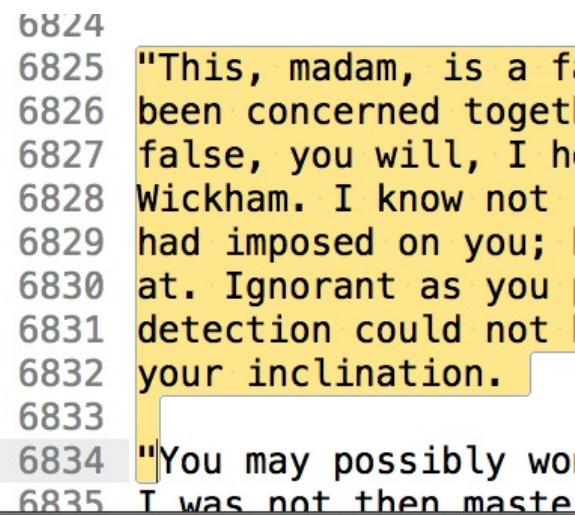
If we click “find next,” it turns out that our new regex will also find multiline quotes.



13000     all more, all suddenly more,  
13010     (As George Fox rais'd his warning cry, "I:  
13011         mortar, these dead floors, windows, ra  
13012         Why this is not the church at all--the ch  
13013         souls.")  
13014

.\* Aa "" ⌂ ⌃ ⌄ ⌅ ⌆ "[^"]\*" 3 of 25 matches

↑  
newlines can now appear between quotes



6824  
6825     "This, madam, is a f  
6826     been concerned toget  
6827     false, you will, I h  
6828     Wickham. I know not  
6829     had imposed on you;  
6830     at. Ignorant as you  
6831     detection could not  
6832     your inclination.  
6833  
6834     "You may possibly wo  
6835     T was not then master

.\* Aa "" ⌂ ⌃ ⌄ ⌅ ⌆ "[^"]\*" 991 of 1776 matches

But in Pride & Prejudice, multiline quotations are split. They start with a quotation mark, have some text with NO quotation mark, then a quotation mark.

So we want to count these as separate quotations.

# Split multiline quotes

A special sequence indicating a paragraph break in these files:  
a carriage return \r, followed by a newline \n,  
and then a blank line ( another \r\n )

```
"[^"]*"\\r\\n\\r\\n| "[^"]*"
```

Multiline quotes consist of one or more paragraphs that begin with  
one quotation mark and have no closing quotation mark

...followed, finally, by a paragraph that has quotation marks at the beginning and the end.

| (The pipe symbol) combines two regex expressions – it's a logical “OR”

*We want to have both the split multiline and short quotations.*

1339  
7340 "Yes," thought Elizabeth, "\_tha  
7341 and completely do for us at onc  
7342 campful of soldiers, to us, who  
7343 regiment of militia, and the m  
7344

7345 "Now I have got some news for  
7346 table. "What do you think? It  
7347 a certain person we all like!"  
7348

7349 Jane and Elizabeth looked at ea  
7350 not stay. Lydia laughed, and sa  
7351

7352 "Aye, that is just like your fa  
7353 waiter must not hear, as if he  
7354 things said than I am going to  
7355 he is gone. I never saw such a  
7356 my news; it is about dear Wick  
7357 There is no danger of Wickham's  
7358 is gone down to her uncle at L  
7359

That matter of Troy and Achilles' wrath,  
wanderings,  
Placard "Removed" and "To Let" on the rock  
Repeat at Jerusalem, place the notice high  
Mount Moriah,

16917 "Going Somewhere"

16918

16919

16920

16921

16922

16923

16924

16925

16926

16927

16928

16929

16930

My science-friend, my noble  
(Now buried in an English g  
sake,) Ended our talk--"The sum, c

learning, intuitions de

"Of all Geologies--Historie  
Metaphysics all,

"Is, that we all are onward

"Life, life an endless marc  
duly over,)

"The world, the race, the s

"All bound as is befitting

16931

16932

16933

16934

16935

16936

16937

16938

16939

16940

16941

16942

16943

16944

16945

16946

16947

16948

16949

16950

16951

16952

16953

16954

16955

16956

16957

16958

16959

16960

16961

16962

16963

16964

16965

16966

16967

16968

16969

16970

16971

16972

16973

16974

16975

16976

16977

16978

16979

16980

16981

16982

16983

16984

16985

16986

16987

16988

16989

16990

16991

16992

16993

16994

16995

16996

16997

16998

16999

17000

17001

17002

17003

17004

17005

17006

17007

17008

17009

17010

17011

17012

17013

17014

17015

17016

17017

17018

17019

17020

17021

17022

17023

17024

17025

17026

17027

17028

17029

17030

17031

17032

17033

17034

17035

17036

17037

17038

17039

17040

17041

17042

17043

17044

17045

17046

17047

17048

17049

17050

17051

17052

17053

17054

17055

17056

17057

17058

17059

17060

17061

17062

17063

17064

17065

17066

17067

17068

17069

17070

17071

17072

17073

17074

17075

17076

17077

17078

17079

17080

17081

17082

17083

17084

17085

17086

17087

17088

17089

17090

17091

17092

17093

17094

17095

17096

17097

17098

17099

17100

17101

17102

17103

17104

17105

17106

17107

17108

17109

17110

17111

17112

17113

17114

17115

17116

17117

17118

17119

17120

17121

17122

17123

17124

17125

17126

17127

17128

17129

17130

17131

17132

17133

17134

17135

17136

17137

17138

17139

17140

17141

17142

17143

17144

17145

17146

17147

17148

17149

17150

17151

17152

17153

17154

17155

17156

17157

17158

17159

17160

17161

17162

17163

17164

17165

17166

17167

17168

17169

17170

17171

17172

17173

17174

17175

17176

17177

# Leonard Richardson

Swapping the dialogue between  
two different texts

## ALICE'S ADVENTURES IN THE WHALE

Leonard Richardson

*Alice's Adventures in the Whale* is a generated novel created by Leonard Richardson. His script ([accessible here](#) for the code-savvy and adventurous) replaces all dialogue in one novel with dialogue from another — in this case, *Alice's Adventures in Wonderland* and *Moby-Dick; or, The Whale*. We present this excerpt for your enjoyment.

### CHAPTER V. “Advice from a Caterpillar”

The Caterpillar and Alice looked at each other for some time in silence: at last the Caterpillar took the hookah out of its mouth, and addressed her in a languid, sleepy voice. “I say, pull like god-dam,” said the Caterpillar.

This was not an encouraging opening for a conversation. Alice replied, rather shyly, “There she slides, now! Hurrah for the white-ash breeze! Down with the Yarman! Sail over him!”

The Reef -- Minecraft books  
linked inside map objects



<https://www.youtube.com/watch?v=gpkdtv68DYU>

<https://github.com/leonardr/In-Dialogue/>

<http://wagsrevue.com/20/fiction/richardson1>

<https://nanogenmo.github.io/>

# Essential Concepts

- Regular expressions added to your tools

.	Any character except newline.
\.	A period (and so on for \*, \(. , \\, etc.)
^	The start of the string.
\$	The end of the string.
\d,\w,\s	A digit, word character [A-Za-z0-9_], or whitespace.
\D,\W,\S	Anything except a digit, word character, or whitespace.
[abc]	Character a, b, or c.
[a-z]	a through z.
[^abc]	Any character except a, b, or c.
aa bb	Either aa or bb.
?	Zero or one of the preceding element.
*	Zero or more of the preceding element.
+	One or more of the preceding element.
{n}	Exactly n of the preceding element.
{n,}	n or more of the preceding element.
{m,n}	Between m and n of the preceding element.
??,*?,+?,{n}?, etc.	Same as above, but as few as possible.
(expr)	Capture expr for use with \1, etc.
(?:expr)	Non-capturing group.
(?=expr)	Followed by expr.
(?!expr)	Not followed by expr.



RegExExpress  
a cool ASCII tutorial game  
<https://pyweek.org/e/RegExExpress/>

Stack Overflow  
<http://stackoverflow.com/>

Google  
<http://google.com>

Trevor Payne's Regex Tutorial  
on youtube

<http://regexpal.com>

# Regex, the 2nd coming

When you read a line of code like this:

```
name.match(/^[\da-z]+\d$/);
```

You probably actually see this:

```
name.match(justabunchofrandomjunkallsmashedtogether);
```

```
name.match(/  
    ^      # from the beginning  
    [      # a set of chars containing  
        \d  # any digit  
        a  # the letter a  
        -  # through  
        z  # the letter z  
    ]      # end class  
    +      # one or more  
    \d  # any digit  
    $      # to the end  
/);
```

"Match from the start a set of one or more digits or a-z followed by any digit to the end of the string." You could even get more succinct and say, "Match completely one more digits or lowercase letters followed by a digit."

# Homework



- 1** Analyze and compare a fiction book with a book of poetry in any way you see fit, using your existing knowledge of regular expressions and expanding that knowledge if you like. You can try to lexically determine the length of sentences (e.g. references to "Mr." Bennet may make that difficult), for instance. Or you can look for certain words, or sets of words, and compare what percentage of the text they take up in each case. Some of the simplest and most commonly overlooked words (articles or pronouns, for instance) may actually be very interesting to consider; avoid assembling large sets of nouns, verbs, or adjectives. You may use code (but its not required) to calculate these metrics.. Upload a file (could be a infographic or powerpoint slide) demonstrating your findings, including the regex you used to Canvas.
  
- 2** Watch a tutorial video on python.  
<https://www.youtube.com/watch?v=ZdDOauFIDkw>  
Follow along and extend the online tutorial to create custom expressions to grab the title, headers, and physical addresses (e.g. 120 Boylston St.) from a web page of your choosing\*.  
Upload your ipython notebook from the video, along with useful regex.  
\*Feel free to choose a plain web page.

# Next time, on Wednesday

- Quiz

Be prepared to:

- iterate through a string or list
- read and write a mall function
- arithmetic computation
- follow conditional logic
- manipulate and analyze strings
- explain what some code does
- There will be no Regex on the test.

- no school monday

# Summary of today

- Technical practice
  - Textual analysis of large files using regex
  - Grabbing source text off the web
  - Typing in your first regex expressions
- Other python tutorials that are good

       Trevor’s “Let’s Learn Python” tutorials are also good:

<https://www.youtube.com/playlist?list=PL82YdDfxhWsDJTq5f0Ae7M7yGcA26wevJ>

Lynda.com Programming fundamentals in the real world

<https://www.lynda.com/Python-tutorials/Programming-Fundamentals-Real-World/418249-2.html>