

LING/C SC/PSYC 438/538

Lecture 1
Sandiway Fong

Contents

- Chomsky lecture announcement
- Syllabus
- Questions about the Syllabus
- Homeworks 1 and 2 (*possibly the null homework*)
- Some real intro next time!

Chomsky talk: free participation

- **Wednesday August 24**, 10:30am to 1pm: Noam Chomsky (UofA)
- Zoom link for all the lectures <https://arizona.zoom.us/j/83059854210>
- Auditing and participating with questions in the Q&A part of each lecture are open without any formality.

Title: Genuine Explanation and the Strong Minimalist Thesis

ABSTRACT:

- The goal of theoretical inquiry is explanation: Why this, and not that? In the study of language, search for explanatory theory proceeds at two levels: for individual languages (a generative grammar) and for the general faculty of language FL (UG), the latter apparently a true species property, common to humans and without significant analogue in the animal world. UG must meet several conditions: learnability, evolvability, coverage. These conditions appear to conflict, and are far more severe than had earlier been supposed. A solution to the conundrum would be satisfaction of SMT for UG combined with recourse to language-independent principles of computational efficiency, with diversity sequestered in components of language subject to simple algorithmic search. For the first time, hopes for such an outcome seem to be on the horizon, with significant implications if the hopes can be realized.

Chomsky talk: free participation

- Language can be modeled as a formal system

Research Question:

- What is the nature of this formal system for human language?
- How is it different from other (mathematical) formal systems?

- Much of what we study in this class is to do with basic formal systems

Examples:

- some programming
- regular expressions, finite state automata
- context-free grammars
- etc ...

Syllabus

Description of Course

- An introductory level course at the advanced level for computational linguistics. Required core course for the Master's Human Language Technology (HLT) program.

Course Pre-requisites

- 438: LING 388 or familiarity with one or more of the following: formal languages, syntax,
- data structures, or compilers.
- 538: no formal pre-requisites.

Syllabus

Instructor and Contact Information

- Instructor: Sandiway Fong
- Contact email: sandiway@arizona.edu (all homework to be submitted here)
- Homepage: sandiway.arizona.edu
- Dept. of Linguistics Office: Douglass 311

Hours:

- make appointments by email or try my office

Meet:

- Rm 351, Education Building: 3:30-4:45pm Mondays/Wednesday
- *another good time to ask questions is right after class!*

Syllabus

Course Format and Teaching Methods

- Lecture with slides. Panopto videos (when available) for lecture review.
- All homeworks will be introduced and **reviewed** in class.

Course Objectives

Topics covered include:

- Introductory programming relevant to computational linguistics in two or more programming languages. We will use Perl, Python and Prolog this semester.
- Introduction to a range of topics in computational linguistics, see detailed list of topics later below.

Syllabus

Course Learning Outcomes

After completing this course, students will:

1. Have acquired the ability to read and write programs in two or more programming languages.
 - Relates to Linguistics Department HLT program outcome #1.
2. Be familiar with basic concepts, techniques and applications in computational linguistics.
 - Relates to Linguistics Department HLT program outcome #2 and Linguistics Department Undergraduate program outcome #1.
3. **538-only:** be able to present and explain advanced concepts in computational linguistics. (See chapter presentation requirement.)
4. Be equipped to take more advanced classes in computational linguistics, e.g. 581 (Spring) or 439/539 (Statistical NLP).

Syllabus

Absence and Class Participation Policy

- I expect you to attend lectures (attendance will not be taken).
- The UA's policy concerning Class Attendance, Participation, and Administrative Drops is available at: <http://catalog.arizona.edu/policy/class-attendance-participation-and-administrative-drop>.
- Tell me ahead of time so we can make alternative arrangements in the case of missed homeworks. **No homework will be accepted late. Explained below.**
- Absences pre-approved by the UA Dean of Students (or Dean Designee) will be honored. See: <https://deanofstudents.arizona.edu/absences>.
- The UA policy regarding absences for any sincerely held religious belief, observance or practice will be accommodated where reasonable, <http://policy.arizona.edu/human-resources/religious-accommodation-policy>.

Syllabus

Required Text

- 438: None.
- 538 Presentations: *Speech and Language Processing*, Jurafsky & Martin, draft 3rd edition (PDF available).

Required or Special Materials

- All necessary software will be available online at no cost to the student.
- However, students are expected to either have a laptop/desktop capable of handling homework and classwork, or make use of UA lab computers (?)
- Mac, PC (Windows 10) or Linux.

Syllabus

Assignments and Examinations: Schedule/Due Dates

- All homeworks will be introduced and reviewed in class.
- Homework submissions by email to me only.
- Late homework will not be accepted since all homeworks will be solved/reviewed in class.
- **Quick homeworks** are normally due at midnight before the next class, and are generally assigned in class on a **Monday** and due **Tuesday** midnight (before **Wednesday** class).
- Homeworks not categorized as quick are normally assigned in class on a **Wednesday** and due the following **Sunday** midnight (before **Monday's** class). (As deemed appropriate by the instructor, some longer homeworks may have an extended due date.)
- Students can expect a total of 10-14 homeworks over the course.
- An in-class quick quiz or two maybe be scheduled. If so, students will be notified ahead of time during the prior class.

Syllabus

Final Examination or Project

- No examinations, e.g. mid-term or final, are scheduled for this course.

Grading Scale and Policies

- **438:**
 - 100% of the grade comes from the homework assignments.
- **538:**
 - 75% of the grade comes from the homework assignments (possibly a superset of the 438 assignments), 25% of the grade comes from a textbook chapter presentation.
 - Requests for incomplete (I) or withdrawal (W) must be made in accordance with University policies, which are available at <http://catalog.arizona.edu/policy/grades-and-grading-system#incomplete> and <http://catalog.arizona.edu/policy/grades-and-grading-system#Withdrawal> respectively.

Syllabus

Scheduled Topics/Activities

- Topics will be drawn from the following:
 - – Programming Languages: Perl and Python
 - – Regular Expressions (Theoretical and practical)
 - – Automata (Finite State) and Transducers (Finite State)
 - – Programming Language: Prolog (definite clause grammars)
 - – NLTK (Natural Language Toolkit)
 - – Part of Speech (POS) Tagging
 - – Stemming (Morphology)
 - – Edit Distance (Spelling)
 - – Grammars (Regular, Context-free)
 - – Parsing (Syntax trees, algorithms)
 - – *and more ...*

Syllabus

Code of Academic Integrity

- You may discuss homework questions with anyone.
- You may look things up on the web and use answers found therein; however, you must write it up yourself (in your own words/own code etc.).
- You must cite all (web) references and your classmates (in the case of shared discussion).
- Students are encouraged to share intellectual views and discuss freely the principles and applications of course materials.
- However, graded work/exercises must be the product of independent effort unless otherwise instructed.
- Students are expected to adhere to the UA Code of Academic Integrity as described in the UA General Catalog. See: <http://deanofstudents.arizona.edu/academic-integrity/students/academic-integrity>.

Syllabus

UA Nondiscrimination and Anti-harassment Policy

- The University is committed to creating and maintaining an environment free of discrimination; see <http://policy.arizona.edu/human-resources/nondiscrimination-and-anti-harassment-policy>.

Subject to Change Statement

- Information contained in the course syllabus, other than the grade and absence policy, may be subject to change with advance notice, as deemed appropriate by the instructor.

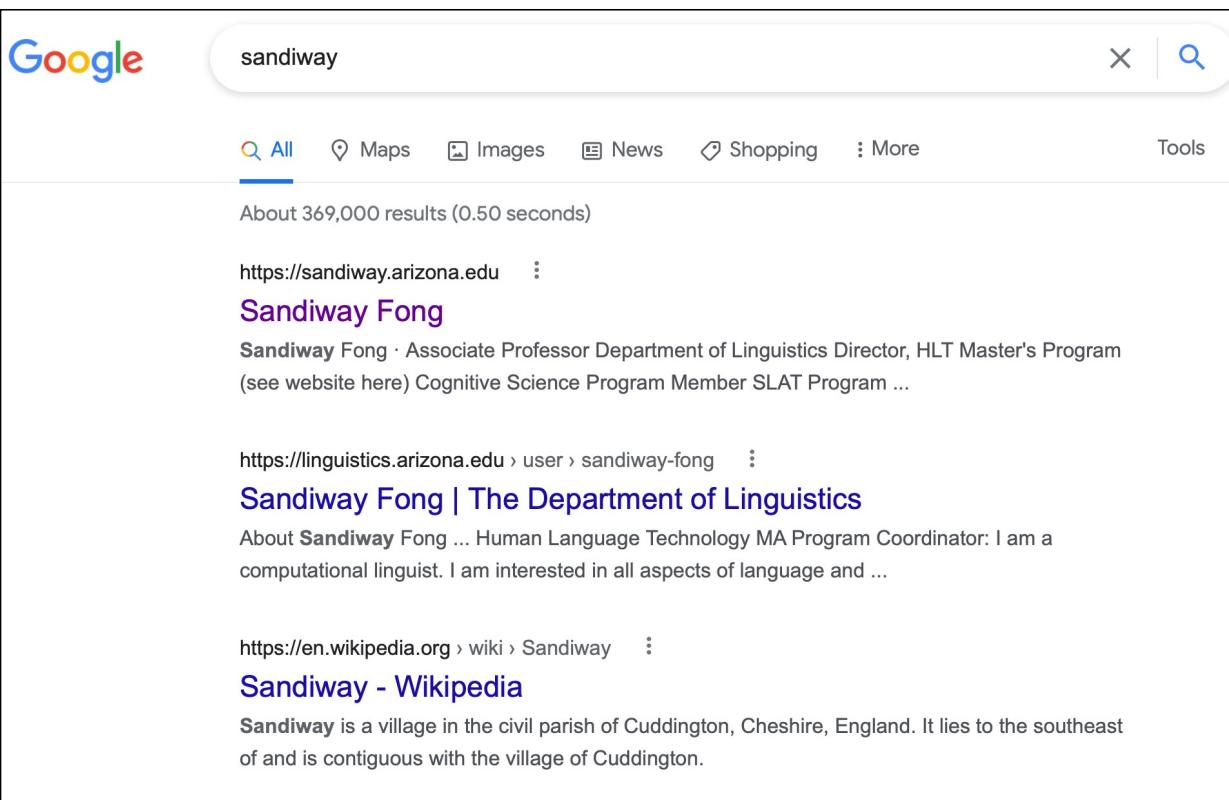
Syllabus

- Questions?

Course website

- Download lecture slides from my homepage
 - <http://sandaway.arizona.edu/ling538-22>
 - available from just before class time
 - (afterwards, please look again for updates and corrections)
 - in .pptx (good for animations) and .pdf formats

Course website



A screenshot of a Google search results page. The search query "sandiway" is entered in the search bar. The results are filtered under the "All" tab. The first result is a link to "https://sandiway.arizona.edu" with the title "Sandiway Fong". Below it is a snippet of text about Sandiway Fong's academic roles. The second result is a link to "https://linguistics.arizona.edu › user › sandiway-fong" with the title "Sandiway Fong | The Department of Linguistics". A snippet of text follows. The third result is a link to "https://en.wikipedia.org › wiki › Sandiway" with the title "Sandiway - Wikipedia". A snippet of text follows. A blue arrow points from a box labeled "1st hit" to the first search result.

1st hit →

Google sandiway

All Maps Images News Shopping More Tools

About 369,000 results (0.50 seconds)

<https://sandiway.arizona.edu> ::
Sandiway Fong
Sandiway Fong · Associate Professor Department of Linguistics Director, HLT Master's Program
(see website here) Cognitive Science Program Member SLAT Program ...

<https://linguistics.arizona.edu › user › sandiway-fong> ::
Sandiway Fong | The Department of Linguistics
About Sandiway Fong ... Human Language Technology MA Program Coordinator: I am a computational linguist. I am interested in all aspects of language and ...

<https://en.wikipedia.org › wiki › Sandiway> ::
Sandiway - Wikipedia
Sandiway is a village in the civil parish of Cuddington, Cheshire, England. It lies to the southeast of and is contiguous with the village of Cuddington.

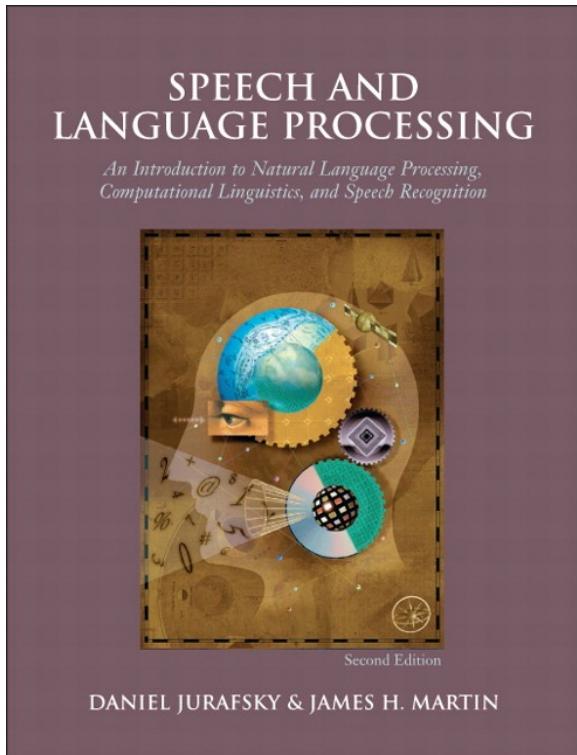
Panopto

- Lectures will be recorded using the panopto system
 - accessible via the course webpage and your browser
 - **sometimes crashes**
 - (video, laptop screen, synchronized slides, keyword search)

The screenshot shows the Panopto interface for a lecture titled "Fall 2012 LING 538: lecture 4". The top right shows the user "isandiway" and a "Log out" button. The interface has tabs for "Slides" and "Screen Capture". The "Screen Capture" tab is active, displaying a video of a person speaking and a terminal window showing Perl code. Below the video is a control bar with a play button, a progress bar from 00:26:09 to 01:20:30, and other controls. A "Contents" sidebar on the left lists lecture topics and links. The main content area shows a "Perl Week" section with three windows: a terminal, a code editor, and another terminal window. The code editor window contains the following Perl script:

```
#!/usr/bin/perl
# perl test.pl
my @array = ("one", "two", "three", "four");
foreach $item (@array) {
    print $item, "\n";
}
```

Textbook (J&M)



2008 (2nd edition)

1. Introduction
- 1.1. Knowledge in Speech and Language Processing
- 1.2. Ambiguity
- 1.3. Models and Algorithms
- 1.4. Language, Thought, and Understanding
- 1.5. The State of the Art
- 1.6. Some Brief History
 - 1.6.1. Foundational Insights: 1940s and 1950s
 - 1.6.2. The Two Camps: 1957–1970
 - 1.6.3. Four Paradigms: 1970–1983
 - 1.6.4. Empiricism and Finite-State Models Redux: 1983–1993
 - 1.6.5. The Field Comes Together: 1994–1999
 - 1.6.6. The Rise of Machine Learning: 2000–2008
 - 1.6.7. On Multiple Discoveries
 - 1.6.8. A Final Brief Note on Psychology
- 1.7. Summary
- Bibliographical and Historical Notes

Syllabus

- I'm gonna assume you don't know how to program at all (*yet*)
 - we're going to use **Perl** and **Python**
 - good to learn both ...
 - good to be polyvalent, you'll also get some **Prolog**

Homework: Reading

Homework 1:

- Chapter 1 from JM (2nd ed)
 - **READ IT before next time!**
 - **in-class Quick Quiz on Thursday**
 - available on course webpage as:
<https://sandiway.arizona.edu/ling538-22/1.pdf>

Chapter 1 Introduction

*Dave Bowman: Open the pod bay doors, HAL.
HAL: I'm sorry Dave, I'm afraid I can't do that.*
Stanley Kubrick and Arthur C. Clarke,
screenplay of 2001: A Space Odyssey

The idea of giving computers the ability to process human language is as old as the idea of computers themselves. This book is about the implementation and implications of that exciting idea. We introduce a vibrant interdisciplinary field with many names corresponding to its many facets, names like **speech and language processing**, **human language technology**, **natural language processing**, **computational linguistics**, and **speech recognition and synthesis**. The goal of this new field is to get computers to perform useful tasks involving human language, tasks like enabling human-machine communication, improving human-human communication, or simply doing useful processing of text or speech.

Conversational agent

Dialogue system

Machine translation

Question answering

One example of a useful such task is a **conversational agent**. The HAL 9000 computer in Stanley Kubrick's film *2001: A Space Odyssey* is one of the most recognizable characters in 20th century cinema. HAL is an artificial agent capable of such advanced language behavior as speaking and understanding English, and at a crucial moment in the plot, even reading lips. It is now clear that HAL's creator, Arthur C. Clarke, was a little optimistic in predicting when an artificial agent such as HAL would be available. But just how far off was he? What would it take to create at least the language-related parts of HAL? We call programs like HAL, that converse with humans in natural language **conversational agents** or **dialogue systems**. In this text we study the various components that make up modern conversational agents, including language input (**automatic speech recognition** and **natural language understanding**) and language output (dialogue and response planning and **speech synthesis**).

Let's turn to another useful language-related task, that of making available to non-English-speaking readers the vast amount of scientific information on the Web in English. Or translating for English speakers the hundreds of millions of Web pages written in other languages like Chinese. The goal of **machine translation** is to automatically translate a document from one language to another. We introduce the algorithms and mathematical tools needed to understand how modern machine translation works. Machine translation is far from a solved problem; we cover the algorithms currently used in the field, as well as important component tasks.

Many other language processing tasks are also related to the Web. Another such task is **Web-based question answering**. This is a generalization of simple Web search, where instead of just typing keywords, a user might ask complete questions, ranging from easy to hard, like the following:

- What does "divergent" mean?
- What year was Abraham Lincoln born?
- How many states were in the United States that year?

Homework 2

- Install Perl and Python (version 3, **not** 2.7)
 - could be the null homework ...

Homework: Install Perl

- Install Perl on your laptop
 - should be pre-installed on macs and Linux, check your machine from the Terminal/command line
 - on Windows PCs, if you don't already have it, it's freely available here
 - www.perl.org
 - <http://www.activestate.com/> (*don't pay, get the free version*) or

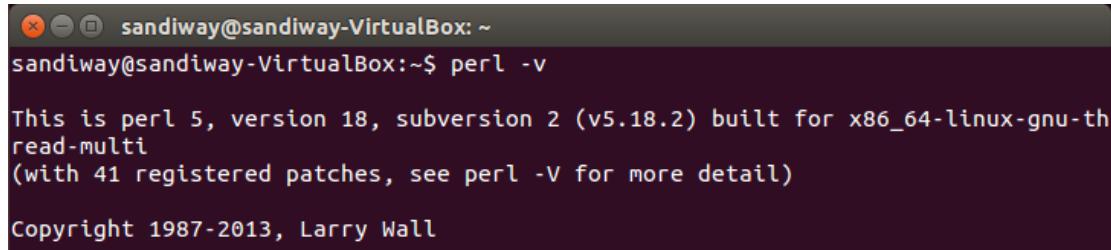
Perl runs on over 100 platforms!

We recommend that you always run the latest stable version, currently 5.28.0. If you're running a version older than 5.8.3, you may find that the latest version of CPAN modules will not work.

Unix/Linux	macOS	Windows
 Included (may not be latest)	 Included (may not be latest)	 Windows Strawberry Perl & ActiveState Perl
GET STARTED	GET STARTED	GET STARTED

Homework 2: Install Perl

- Ubuntu (Terminal):



```
sandiway@sandiway-VirtualBox: ~
sandiway@sandiway-VirtualBox:~$ perl -v

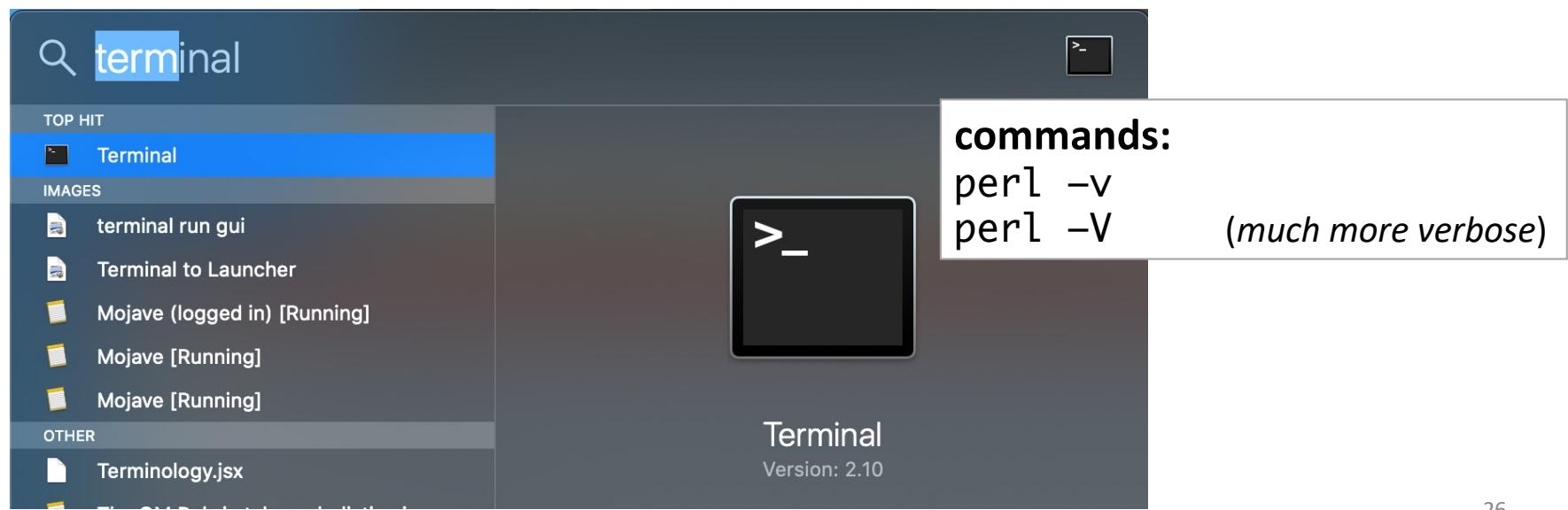
This is perl 5, version 18, subversion 2 (v5.18.2) built for x86_64-linux-gnu-thread-multi
(with 41 registered patches, see perl -V for more detail)

Copyright 1987-2013, Larry Wall
```

commands:
perl -v
which perl

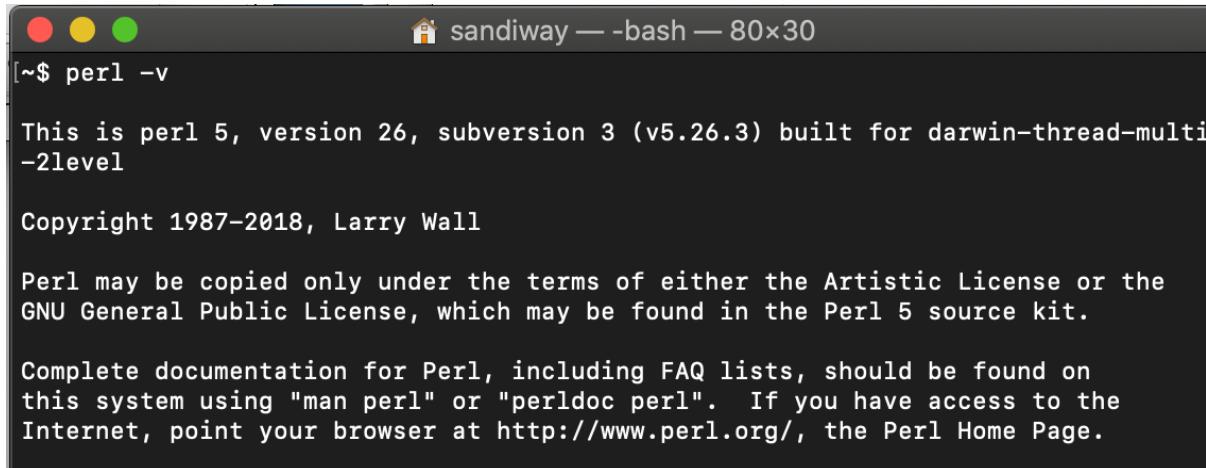
Homework 2: Install Perl

- Mac (Terminal): (*complete path specified here*)
 - /usr/bin/perl

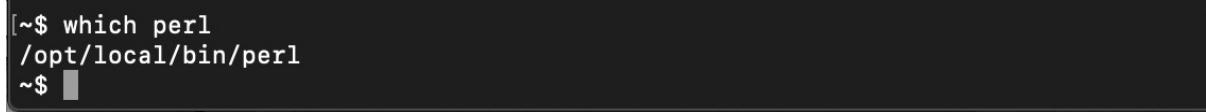


Homework 2: Install Perl

- Actually, on my Mac laptop, typing perl at the Terminal gives ...



A screenshot of a Mac OS X terminal window titled "sandway — -bash — 80x30". The window shows the output of the command "perl -v". The output text is:
This is perl 5, version 26, subversion 3 (v5.26.3) built for darwin-thread-multi-2level
Copyright 1987-2018, Larry Wall
Perl may be copied only under the terms of either the Artistic License or the
GNU General Public License, which may be found in the Perl 5 source kit.
Complete documentation for Perl, including FAQ lists, should be found on
this system using "man perl" or "perldoc perl". If you have access to the
Internet, point your browser at <http://www.perl.org/>, the Perl Home Page.

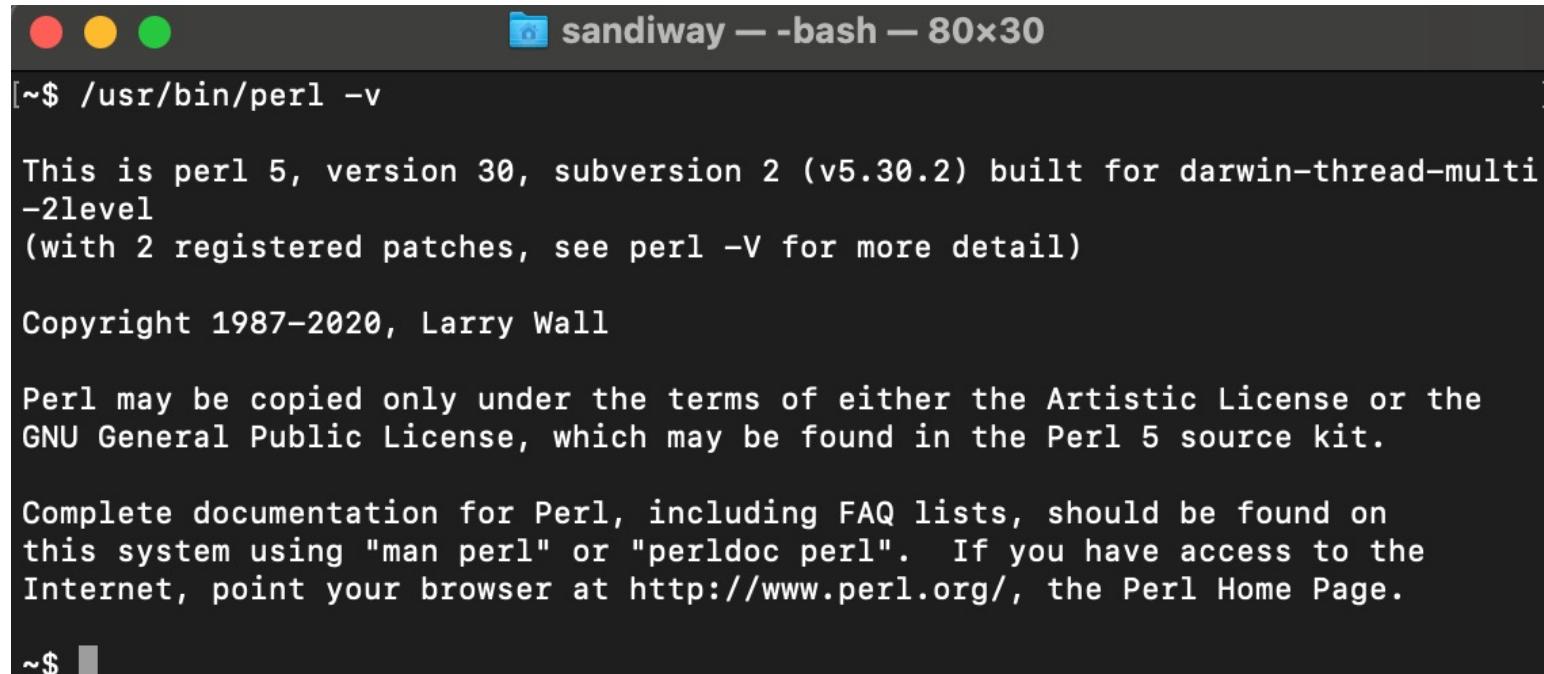


A screenshot of a Mac OS X terminal window showing the output of the command "which perl". The output text is:
/opt/local/bin/perl
~\$ █



Homework 2: Install Perl

- On my Mac laptop, typing `/usr/bin/perl -v` at the Terminal gives ...



```
[~$ /usr/bin/perl -v

This is perl 5, version 30, subversion 2 (v5.30.2) built for darwin-thread-multi-2level
(with 2 registered patches, see perl -V for more detail)

Copyright 1987-2020, Larry Wall

Perl may be copied only under the terms of either the Artistic License or the
GNU General Public License, which may be found in the Perl 5 source kit.

Complete documentation for Perl, including FAQ lists, should be found on
this system using "man perl" or "perldoc perl". If you have access to the
Internet, point your browser at http://www.perl.org/, the Perl Home Page.

~$ ]
```

Homework 2: Install Perl

<https://www.perl.org/learn.html>

The screenshot shows the Perl website homepage. At the top, there's a dark blue header with the Perl logo, version 5.32.0, and links for ABOUT, DOWNLOAD, LEARN, DOCS, CPAN, and COMMUNITY. Below the header, a large image of a camel is on the right. In the center, the text "That's why we love Perl" and "25,000 extensions on CPAN" is displayed. A call-to-action button labeled "DOWNLOAD AND GET STARTED" is present. On the left, a sidebar titled "Get Started" lists five items: learn.perl.org, A brief introduction, Free online Perl books, Join your local community, and Books and More. A blue arrow points from a text box on the far left to the "Get Started" list.

start here
next time

Get Started

- [learn.perl.org](#)
- [A brief introduction](#)
- [Free online Perl books](#)
- [Join your local community](#)
- [Books and More](#)

Perl 5.32.0

ABOUT DOWNLOAD LEARN DOCS CPAN COMMUNITY

That's why we love Perl

25,000 extensions on CPAN

Perl is a highly capable, feature-rich programming language with over 30 years of development.

DOWNLOAD AND GET STARTED

Learning Community Docs

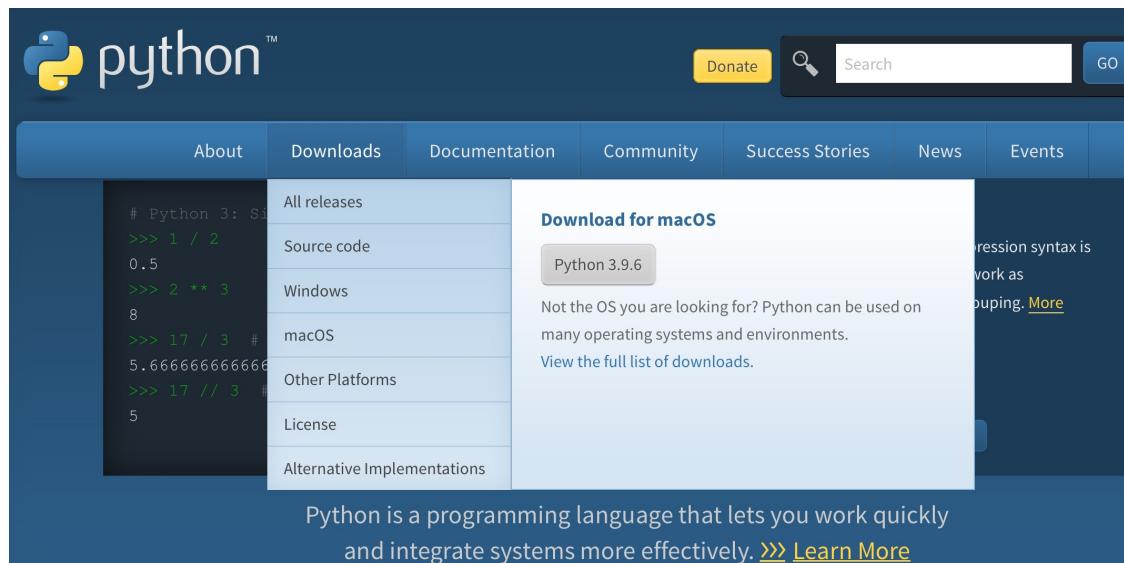
Learning Perl

- Learn Perl
 - <https://perldoc.perl.org/perlintro.html>

The screenshot shows a web-based Perldoc Browser interface. At the top, there's a dark header bar with the text "Perldoc Browser 5.34.0" and dropdown menus for "Dev" and "Documentation". To the right of the header are "Expand" and "Search" buttons. The main content area has a light gray background. On the left, there's a sidebar or navigation panel. The main content starts with the title "perlintro (source, CPAN)". Below the title is a section titled "CONTENTS" containing a bulleted list of topics: NAME, DESCRIPTION (with sub-points: What is Perl?, Running Perl programs, Safety net, Basic syntax overview, Perl variable types, Variable scoping, Conditional and looping constructs, Builtin operators and functions, Files and I/O, Regular expressions, Writing subroutines, OO Perl, Using Perl modules), and AUTHOR. Further down, there's a section titled "NAME" with the subtext "perlintro - a brief introduction and overview of Perl".

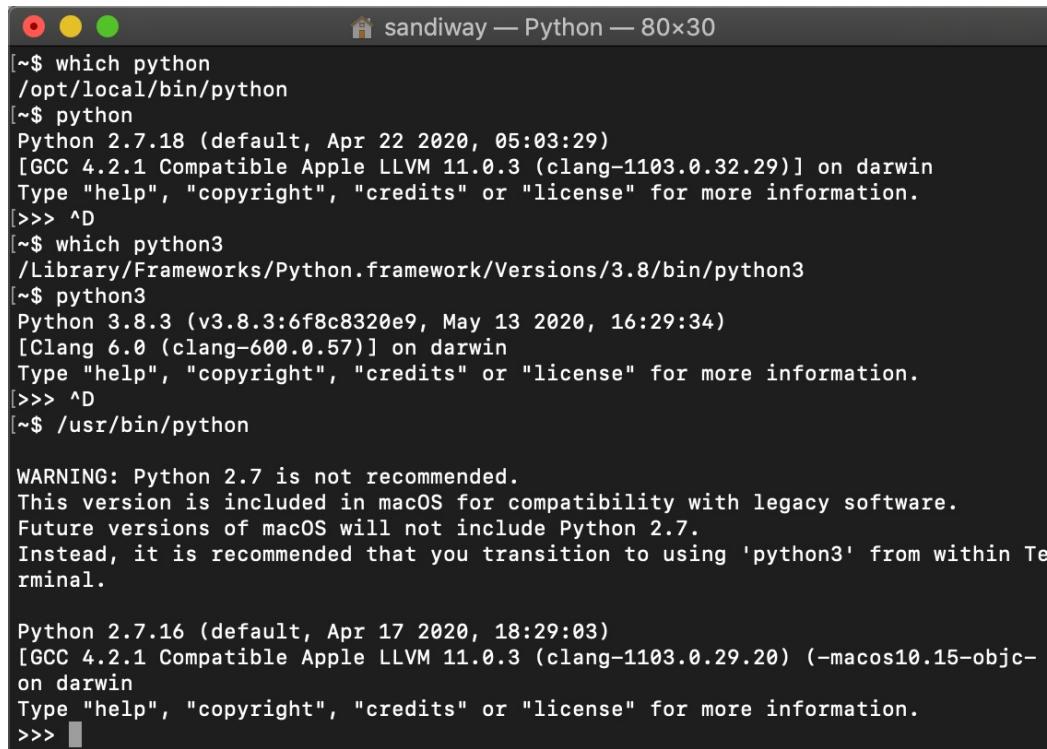
Homework 2: Install Python

- www.python.org
- **Note:** 3.x is **not** quite backwards compatible with Python 2.7!



Homework 2: Install Python

- On my Mac laptop, I have different versions:



```
[~$ which python
/opt/local/bin/python
[~$ python
Python 2.7.18 (default, Apr 22 2020, 05:03:29)
[GCC 4.2.1 Compatible Apple LLVM 11.0.3 (clang-1103.0.32.29)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
[>>> ^D
[~$ which python3
/Library/Frameworks/Python.framework/Versions/3.8/bin/python3
[~$ python3
Python 3.8.3 (v3.8.3:6f8c8320e9, May 13 2020, 16:29:34)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
[>>> ^D
[~$ /usr/bin/python

WARNING: Python 2.7 is not recommended.
This version is included in macOS for compatibility with legacy software.
Future versions of macOS will not include Python 2.7.
Instead, it is recommended that you transition to using 'python3' from within Terminal.

Python 2.7.16 (default, Apr 17 2020, 18:29:03)
[GCC 4.2.1 Compatible Apple LLVM 11.0.3 (clang-1103.0.29.20) (-macos10.15-objc-
on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> ]
```



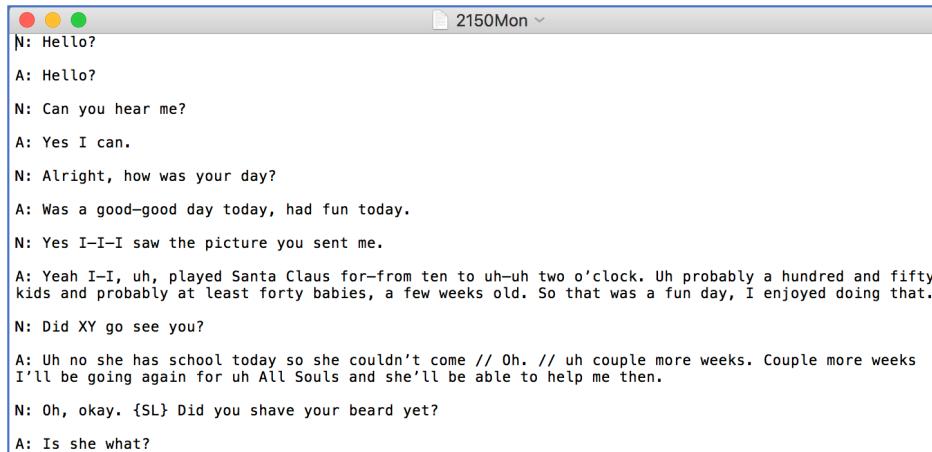
Which one is easier?

A subjective question ...

- All good programmers know more than one programming language
- In NLP, Python is *de facto* overwhelmingly popular we will do both Perl and Python, see pros and cons here:
<https://www.tecmint.com/python-vs-perl-debate-what-should-i-learn-python-or-perl/>

Simplest level of programming

- how do I automate repetitive tasks (**scripting**) on my laptop?
 - learn bash shell scripting...
- Search: how do I look for patterns (**regex**) in a text corpus?
 - `grep -Eo ':\s+Yea[^h]' 2150Mon.txt`
 - `perl -nle 'print $1 if /(.):\s+Yep/' 2149Mon.txt`



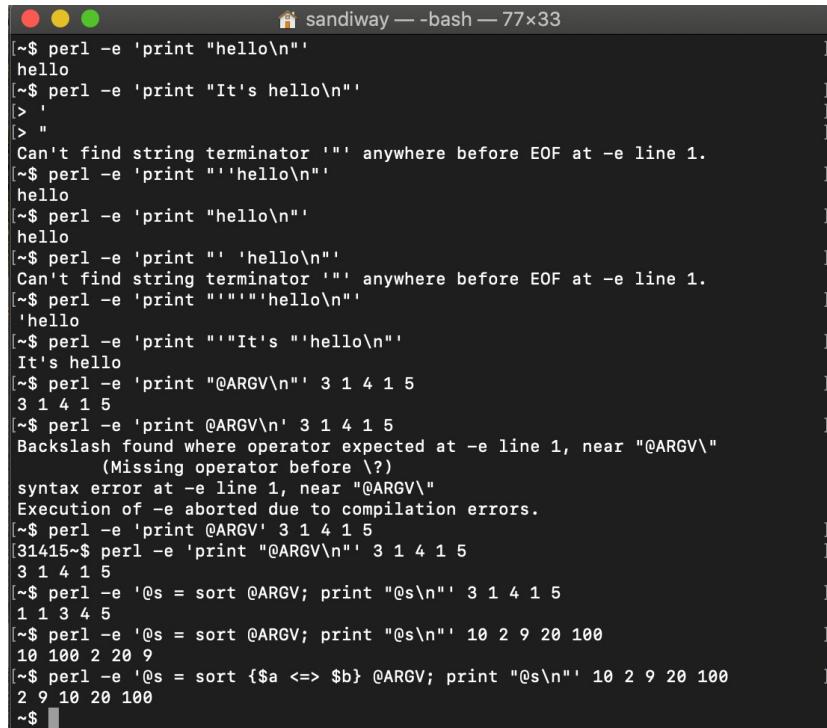
```
N: Hello?
A: Hello?
N: Can you hear me?
A: Yes I can.
N: Alright, how was your day?
A: Was a good-good day today, had fun today.
N: Yes I-I-I saw the picture you sent me.
A: Yeah I-I, uh, played Santa Claus for-from ten to uh-uh two o'clock. Uh probably a hundred and fifty kids and probably at least forty babies, a few weeks old. So that was a fun day, I enjoyed doing that.
N: Did XY go see you?
A: Uh no she has school today so she couldn't come // Oh. // uh couple more weeks. Couple more weeks I'll be going again for uh All Souls and she'll be able to help me then.
N: Oh, okay. {SL} Did you shave your beard yet?
A: Is she what?
```

Between the quotes we have a program!

command line options: -nle

-n	loop over lines of the input file
-l	print newlines
-e	execute code from command line

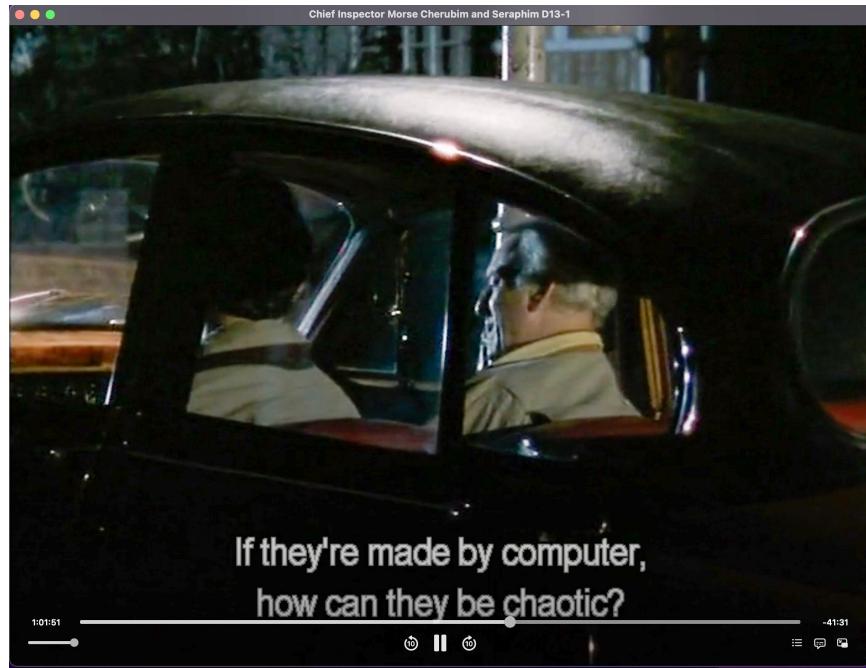
Programming Pitfalls

A terminal window titled "sandiway — bash — 77x33" showing a series of Perl command-line executions. The user types various Perl commands involving print statements and sorting arrays, which result in syntax errors due to missing quotes or operators. The terminal shows the error messages and the resulting output of the valid parts of the code.

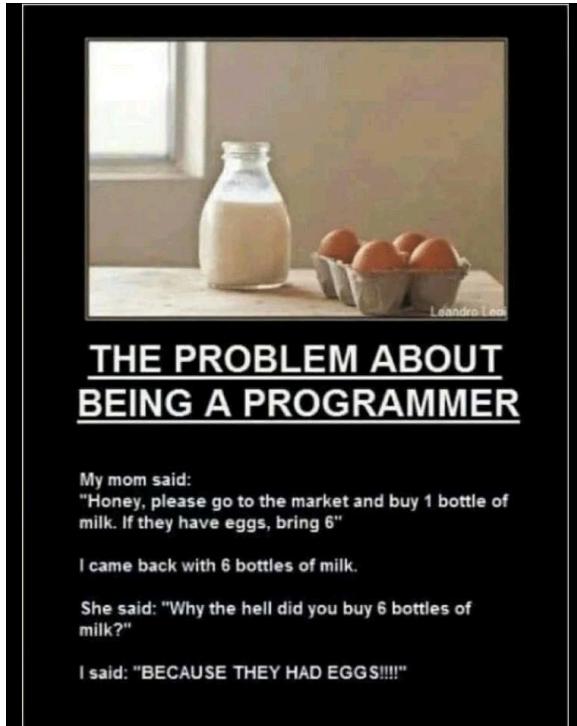
```
[~$ perl -e 'print "hello\n"'
hello
[~$ perl -e 'print "It's hello\n"
[> '
[> "
Can't find string terminator '"' anywhere before EOF at -e line 1.
[~$ perl -e 'print "'hello\n'"
hello
[~$ perl -e 'print "hello\n"
hello
[~$ perl -e 'print "' 'hello\n"
Can't find string terminator '"' anywhere before EOF at -e line 1.
[~$ perl -e 'print """"hello\n"
'hello
[~$ perl -e 'print """It's ""hello\n"
It's hello
[~$ perl -e 'print "@ARGV\n" 3 1 4 1 5
3 1 4 1 5
[~$ perl -e 'print @ARGV\n' 3 1 4 1 5
Backslash found where operator expected at -e line 1, near "@ARGV\""
          (Missing operator before \?)
syntax error at -e line 1, near "@ARGV\""
Execution of -e aborted due to compilation errors.
[~$ perl -e 'print @ARGV' 3 1 4 1 5
[31415~$ perl -e 'print "@ARGV\n"' 3 1 4 1 5
3 1 4 1 5
[~$ perl -e '@s = sort @ARGV; print "@s\n"' 3 1 4 1 5
1 1 3 4 5
[~$ perl -e '@s = sort @ARGV; print "@s\n"' 10 2 9 20 100
10 100 2 20 9
[~$ perl -e '@s = sort {$_ <=> $b} @ARGV; print "@s\n"' 10 2 9 20 100
2 9 10 20 100
~$ ]
```

- Can be maddening!

Chief Inspector Morse



Be Precise



THE PROBLEM ABOUT BEING A PROGRAMMER

My mom said:

"Honey, please go to the market and buy 1 bottle of milk. If they have eggs, bring 6"

I came back with 6 bottles of milk.

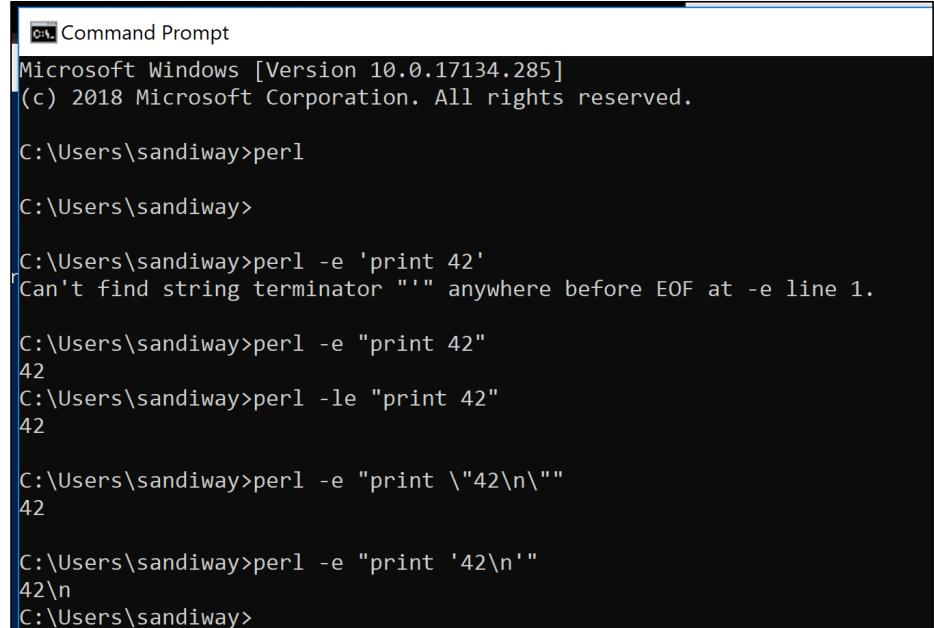
She said: "Why the hell did you buy 6 bottles of milk?"

I said: "BECAUSE THEY HAD EGGS!!!!"

- Saw this on my Facebook feed the other day
- Do you understand the joke?

Windows 10: command prompt

- Windows 10 command prompt
 - single quotes not recognized as quotes
 - double quotes can be used
- inside the (initial) double quotes, everything will be passed to Perl:
 - you can use single quotes
 - you can escape double quotes, i.e. \" to get a double quote.



```
C:\> Command Prompt
Microsoft Windows [Version 10.0.17134.285]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\ sandiway> perl

C:\Users\ sandiway>

C:\Users\ sandiway> perl -e 'print 42'
Can't find string terminator "" anywhere before EOF at -e line 1.

C:\Users\ sandiway> perl -e "print 42"
42
C:\Users\ sandiway> perl -le "print 42"
42

C:\Users\ sandiway> perl -e "print \"42\n\""
42

C:\Users\ sandiway> perl -e "print '42\n'"
42\n
C:\Users\ sandiway>
```

Windows 10: PowerShell

- You can also use PowerShell in Windows 10:
 - both single and double quotes can be used as quoting characters in the shell.
 - Inside the initial quotes, you can escape any quotes that need to be passed to Perl.

```
Windows PowerShell
PS C:\Users\sandiway> perl -e "print 42"
42
PS C:\Users\sandiway> perl -e 'print 42'
42
PS C:\Users\sandiway> perl -e 'print \'Hello, world\''
Hello, world
PS C:\Users\sandiway> perl -e 'print "Hello, world"'
No comma allowed after filehandle at -e line 1.
PS C:\Users\sandiway>
```