

# LING/C SC/PSYC 438/538

Lecture 1

Prof. Sandiway Fong

# Contents

---



The Big Picture



Syllabus



Questions about the Syllabus



Homeworks 1 and 2 (*possibly the null homework*)



A note on Programming Languages



Some real intro next time!

# Computation and Language

- We assume human 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?
- What we study here has much to do with formal systems

## **Examples:**

- programming (languages)
- regular expressions (regex), finite state automata (FSA)
- context-free (CFG) and context-sensitive grammars
- parse trees, word dependencies, sets

# Computation and Language

- Why only us?
  - 50 billion species since life emerged on earth almost 4 billion years ago (Mayr, 1995)
  - Modern humans a recent arrival (300TYA) through a remarkable series of accidents
- Language and thought is a **species-specific** property (Chomsky)
  - closest relatives to modern humans have about the same auditory system
  - only a human infant reflexively develops complex systems for constructing and expressing thought
- Language shows peculiar properties
  - **not arbitrary** (there are rules: man-made or otherwise?)
  - **Research question:** why is it the way it is, and not some other way?

# 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 ([sandiway@arizona.edu](mailto:sandiway@arizona.edu), submit homeworks here)
- Homepage: sandiway.arizona.edu
- Dept. of Linguistics Office: Douglass 308

## **Hours:**

- make appointments by email or drop by my office

## **Meet:**

- Rm 411, Social Sciences: 3:30PM-4:45PM Tuesdays/Thursdays
- *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 programming languages **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

---

## Boilerplate

- <https://catalog.arizona.edu/syllabus-policies>
- *By placing **this URL** on your syllabus you no longer need to list the individual university policies.*
- Specific to this class
  - **Absence and Class Participation Policy**
  - I expect you to attend lectures (though attendance will not be taken).
  - 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.**

# 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
- Mac, PC (Windows 11) or Linux.

# Syllabus

---

## Assignments and Examinations: Schedule/Due Dates

- All homeworks will be introduced **and reviewed** in class.
- Homework submissions by email to me (sandikay@arizona.edu).
- Late homeworks will be not accepted since all homeworks will be solved in class.
- Homeworks are typically assigned in class on a **Thursday** and due **Sunday** midnight. (I will attempt to grade them **Monday**.)
- We discuss the homework in class the following **Tuesday**. (Some longer homeworks may have an extended due date.)
- Students can expect 10+ homeworks over the course.

# 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.

# 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, dependency trees, algorithms)
- – *and more ...*

# Syllabus

---

## Code of Academic Integrity

- You may discuss homework questions with anyone (or **anything**).
- 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 including **ChatGPT**, and credit 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.

# Syllabus

---

## **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.

## **Schedule changes**

- I will notify ahead of time if there are any changes

## **• Questions?**

## UofA website

- Download lecture slides, also links to Panopto recordings, from:
  - sandiway.arizona.edu > Classes > LING/C SC/PSYC 438/538
  - URL: <https://sandiway.arizona.edu/ling538-26>
  - available from just before class time
    - (afterwards, please look again for updates and corrections)
  - in .pptx (good for animations) and .pdf formats

# UofA website

Notice a difference between  
2023, 2025 & 2026?

The image shows a Google search for 'sandiway' with a callout bubble asking to notice a difference between the 2023, 2025, and 2026 versions of the profile. The search results show two versions of the profile for Sandiway Fong at the University of Arizona. The top version, dated May 23, 2025, describes her work as intersecting computer science and linguistics, with a focus on computational modeling of linguistic theory. The bottom version, dated Feb 2, 2023, describes her work as intersecting computer science and linguistics, with a focus on computational modeling of linguistic theory. The callout bubble points to the difference in the date and the description of her work.

Google sandiway

Maps Images Directions

About 458,000 results (0.38 seconds)

**Sandiway Fong**  
https://sandiway.arizona.edu

**Sandiway Fong: The University of Arizona**

Feb 2, 2023 — My work intersects computer science and linguistics. Principally, I'm interested in the computational modeling of linguistic theory. I work at ...

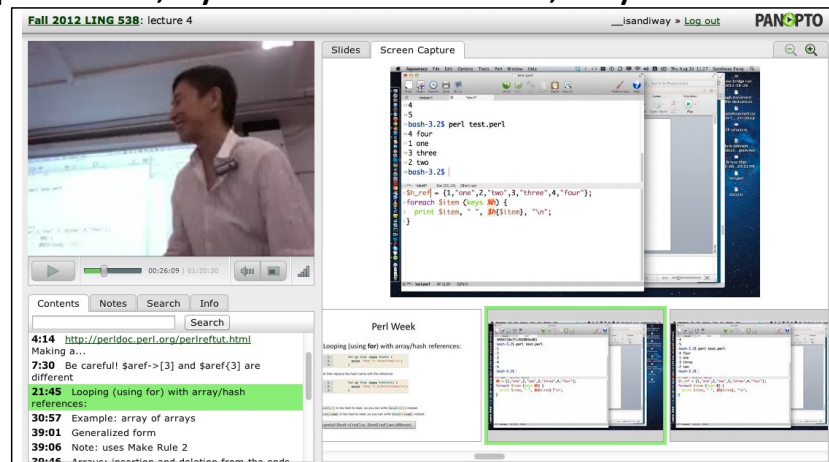
**Sandiway Fong**  
https://sandiway.arizona.edu

**Sandiway Fong: The University of Arizona**

May 23, 2025 — **Sandiway Fong**'s work intersects computer science and linguistics, focusing on computational modeling of linguistic theory. She directs the HLT ...

# Panopto

- Lectures will be recorded using the Panopto system
  - accessible via the course webpage and your browser
  - **sometimes crashes** (not guaranteed to be recorded)
  - (video, laptop screen, synchronized slides, keyword search)



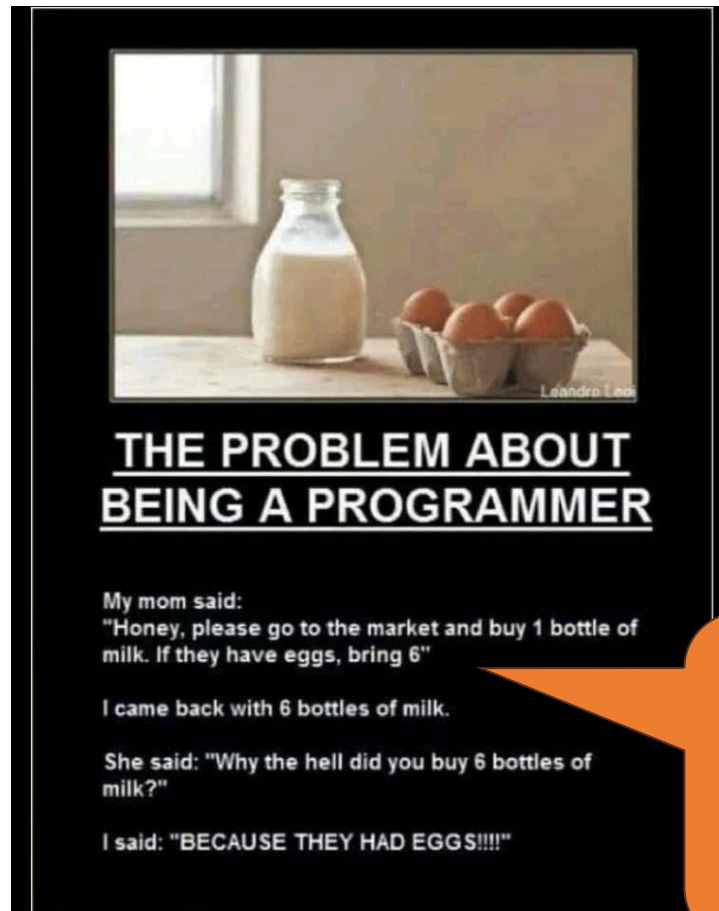
# Syllabus

---

- I'm gonna tentatively 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**

## Being Precise

- Always check to see if you typed something in exactly right.
- *saw this on my Facebook feed*
- Do you understand the joke?



*if condition:  
do  
command*

# Homework: Reading

## Homework 1:

- Chapter 1 from JM (2<sup>nd</sup> ed)
  - **READ IT before next time!**
  - Try to understand the concepts
  - **Homework 3 next time**
  - available on course webpage as:

[sandiway.arizona.edu/ling538-26/1.pdf](http://sandiway.arizona.edu/ling538-26/1.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*

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**).

*Dialogue system*

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.

*Machine translation*

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:

*Question answering*

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

# Homework 2

Could be the *null* homework for many of you:

- Install **Perl**, and
- Install **Python** (version 3.X) (no longer available - **obsolete 2.7**)

# Homework 2: Install Perl

- Install Perl on your computer
  - pre-installed on macOS 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
  - <https://www.perl.org/get.html>

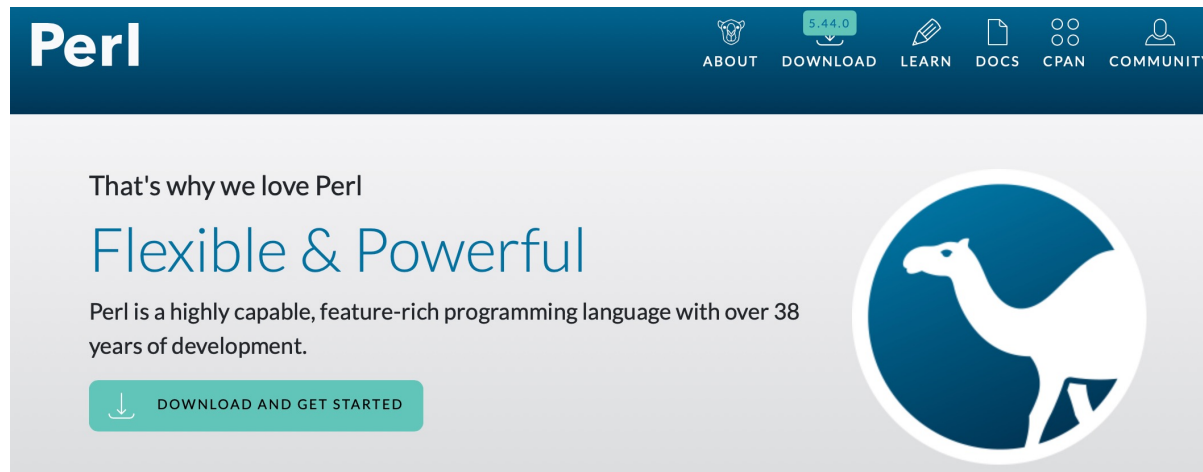
**Perl runs on over 100 platforms!**

We recommend that you always run the latest stable version, currently 5.38.0. 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<br>(may not be latest)                                                                 | Included<br>(may not be latest)                                                                 | Strawberry Perl<br>&<br>ActiveState Perl                                                          |
|  GET STARTED |  GET STARTED |  GET STARTED |

# Homework 2: Install Perl

- Pre-installed on macOS and Linux (**why?**), check your machine from the Terminal/command line
  - on Windows PCs, if you don't already have it, it's freely available from perl.org



# Homework 2: Install Perl

## Windows

Windows does not have Perl installed by default.

### Binaries

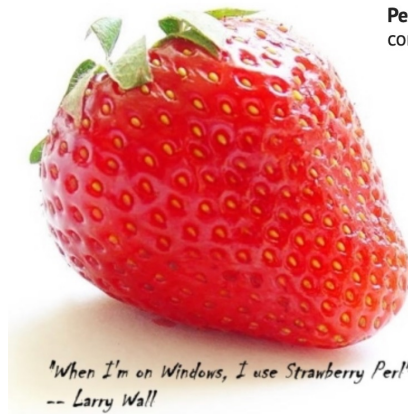
[Strawberry Perl](#): A 100% Open Source Perl for Windows that is exactly the same as Perl everywhere else; this includes using modules from [CPAN](#), without the need for binary packages. Help is available from other Windows Perl developers on the #win32 irc channel on irc.perl.org (see [website](#) for access through a browser).

DOWNLOAD STRAWBERRY PERL

- You could install **Ubuntu** on Windows 11 using Microsoft's WSL (*Windows Subsystem for Linux*)
  - In PowerShell or Terminal
    - `wsl --install`
  - *then you'd have Perl in addition to many other things ...*

# Homework 2: Install Perl

strawberryperl.com



## The Perl for MS Windows, free of charge!

Perl is a programming language suitable for writing simple scripts as well as complex applications — see <https://www.perl.org>.

**Strawberry Perl** is a perl environment for MS Windows containing all you need to run and develop perl applications. It is designed to be as close as possible to perl environment on UNIX systems.

It includes perl binaries, compiler (gcc) + related tools, all the external libraries (crypto, math, graphics, xml...), all the bundled database clients and all you expect from Strawberry Perl.

### Latest Release: 5.42.2.1 (2026-04-07)

**5.42.2.1 MSI** (200.1 MB)

Sha256: 1b473a709e4d69128f52b30426a423f44d181cfc6460e88b4f519f534cc6583

**5.42.2.1 Portable zip** (290.2 MB)

Sha256: 32d83be90cf04b807cfb9477482bc36302cdee6f5b04cf57e81adebcb8f07898

**5.42.2.1 PDL zip** (310.8 MB)

Sha256: 5969e7dbd20eff6d1bcde5d8b2b182158686c596ef8fa45a65ac5d2e59c6d6e2

### More downloads (all releases):

**ZIP, Portable, special editions**

You can find here release notes and other details.

# Homework 2: Install Perl

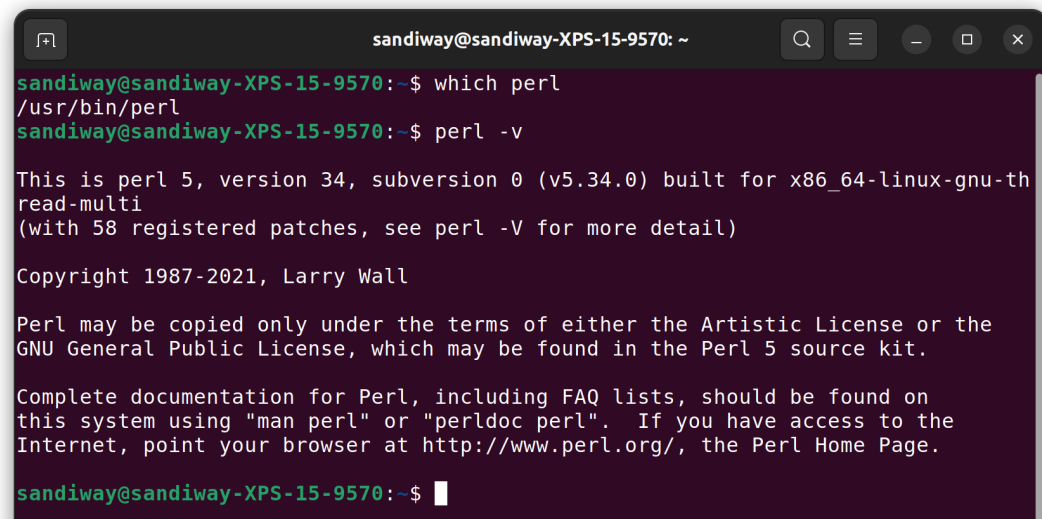
- How to check?
  - `which perl`
  - `perl -v`
- Ubuntu (Terminal):

Mac OS X

Binaries

✓ Already Installed

Mac OS X already has Perl installed.

A terminal window with a dark purple background and white text. The window title is 'sandiway@sandiway-XPS-15-9570: ~'. The user enters the command 'which perl' and the output is '/usr/bin/perl'. Then the user enters 'perl -v' and the output shows Perl 5.34.0 details, including build information, patches, and copyright. The prompt returns to '\$' at the end.

```
sandiway@sandiway-XPS-15-9570:~$ which perl
/usr/bin/perl
sandiway@sandiway-XPS-15-9570:~$ perl -v

This is perl 5, version 34, subversion 0 (v5.34.0) built for x86_64-linux-gnu-th
read-multi
(with 58 registered patches, see perl -V for more detail)

Copyright 1987-2021, 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.

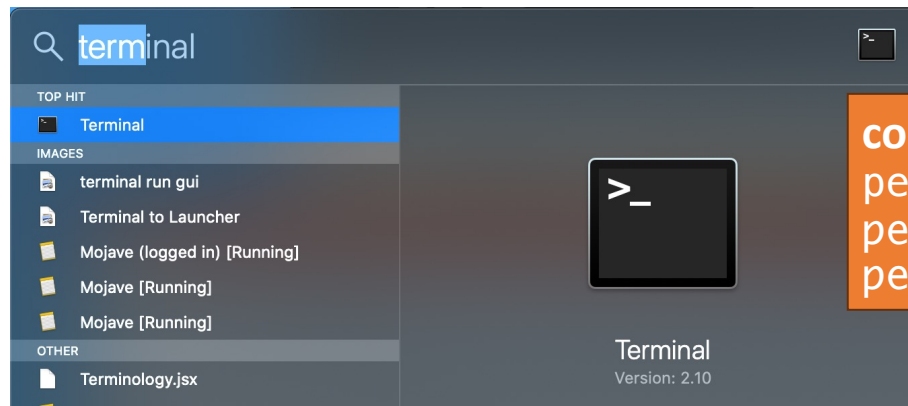
sandiway@sandiway-XPS-15-9570:~$
```

# Homework 2: Install Perl

- macOS (Terminal): *(complete path specified here)*

**PATH.**

- `/usr/bin/perl`
- `/opt/homebrew/bin/perl` (Homebrew package manager *brew.sh*)
- `/opt/local/bin/perl` (MacPorts package manager)



## commands:

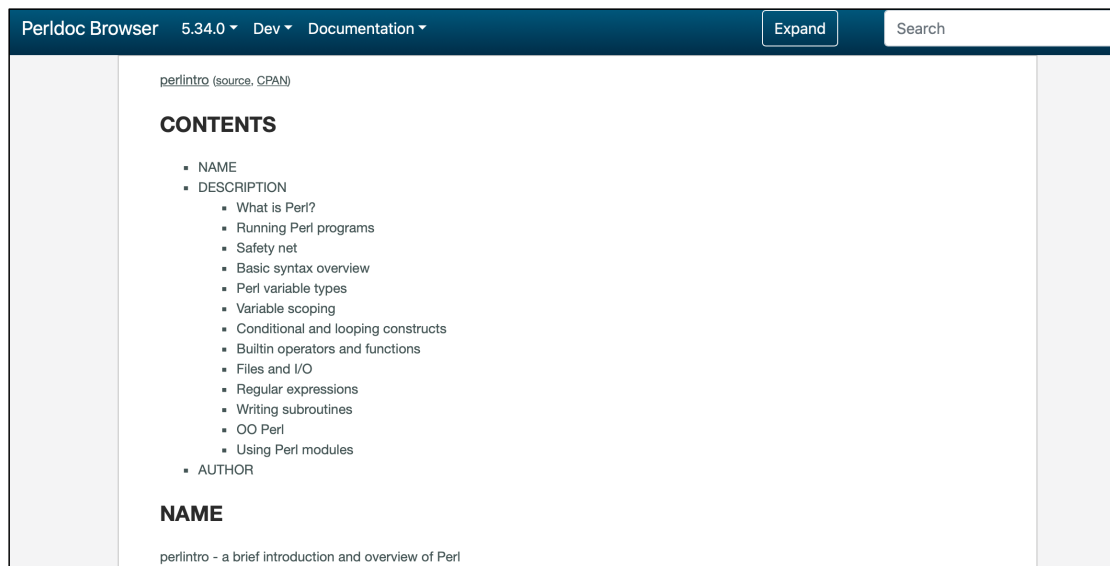
`perl` *(hangs; type Control-D)*

`perl -v`

`perl -V` *(much more verbose)*

# Learning Perl

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



The screenshot shows the Perl documentation browser interface. At the top, there is a dark blue header bar with the text "Perldoc Browser 5.34.0 Dev Documentation" and a search bar. Below the header, the page title is "perlintro (source, CPAN)". The main content area is divided into two columns. The left column contains a table of contents with sections: NAME, DESCRIPTION, and AUTHOR. The DESCRIPTION section is expanded, showing a list of topics: 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, and Using Perl modules. The right column contains the "NAME" section, which starts with the text "perlintro - a brief introduction and overview of Perl".

Perldoc Browser 5.34.0 Dev Documentation

Expand Search

perlintro (source, CPAN)

### CONTENTS

- NAME
- DESCRIPTION
  - 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
- AUTHOR

### NAME

perlintro - a brief introduction and overview of Perl



## Homework 2: Install Python

- [www.python.org](https://www.python.org)
- **Note:** 3.x is not backwards compatible with Python 2.7!



## Which language is easier?

---

*A subjective question ...*

- All good programmers know more than one programming language, always an advantage to be versatile.
- In NLP, Python is overwhelmingly popular, but we will do both Perl and Python. **AND** for writing grammars, Prolog.

# What is Python itself written in?

## The Core Implementation: CPython

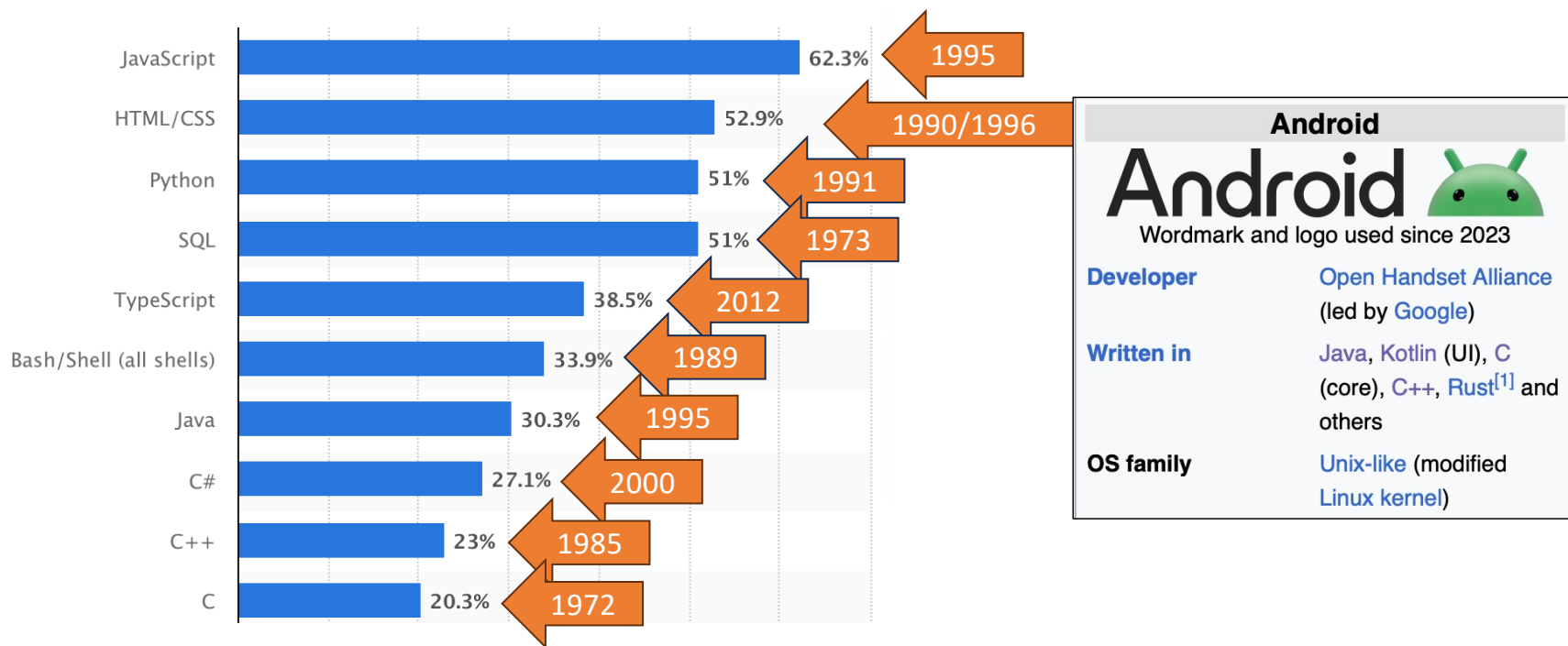
When you download Python from the official Python Website, you are installing

**CPython.**  Python Like You Mean It +2

- **The Interpreter & Compiler:** The core runtime engine—including the compiler, the parser, and the main evaluation loop—is written entirely in **C**. 
- **The Standard Library:** The built-in toolkit that comes with your download is a hybrid. Performance-heavy tasks like cryptography, file math, and math calculations are written in **C**. The remaining standard modules are written directly in **Python**.  Python documentation +2

- **The Bytecode Evaluation Loop ( `ceval.c` ):** The heart of the CPython interpreter (the main execution loop that processes bytecodes) is roughly **3,600 to 4,000 lines of C**.  Hacker News +2
- **The Parser, AST, and Compiler:** The subsystem that converts Python source code into an Abstract Syntax Tree (AST) and compiles it into bytecode spans roughly **50,000 to 100,000 lines of C**. 

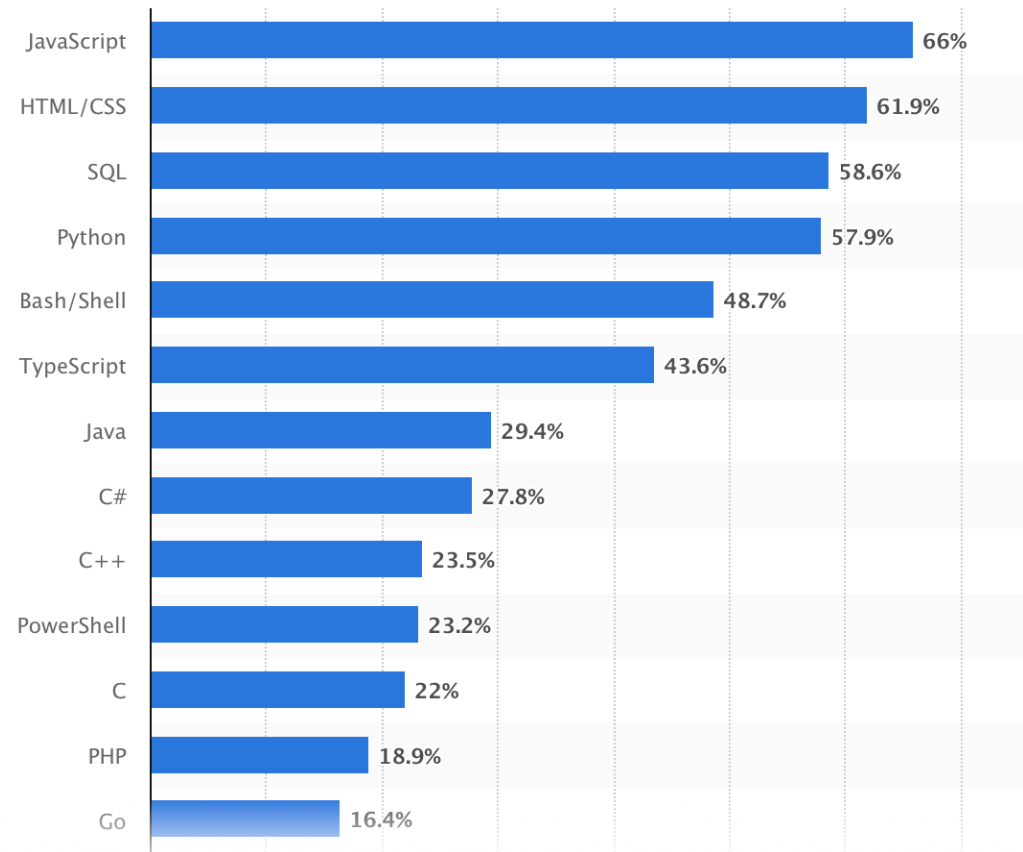
# Development work (2024 survey)



[www.statista.com/statistics/793628/worldwide-developer-survey-most-used-languages/](https://www.statista.com/statistics/793628/worldwide-developer-survey-most-used-languages/)

# Development work (2025 survey)

---



# How many programmers, how many lines?

- [medium.com/modern-stack/how-much-computer-code-has-been-written-c8c03100f459](https://medium.com/modern-stack/how-much-computer-code-has-been-written-c8c03100f459)

Back to the question of how many people are out there writing code. We have arrived at the educated guess that there are about 6 million people in the world writing code right now. The real number is probably much higher than this, but this number works for our calculation.

**2,781,000,000,000**

Roughly 2.8 Trillion Lines of Code have been written in the past 20 years.

That is more than 5X the estimated number of stars in the Milky Way!

# AI and Coding (2025)

---

[www.businessinsider.com/anthropic-ceo-ai-90-percent-code-3-to-6-months-2025-3](https://www.businessinsider.com/anthropic-ceo-ai-90-percent-code-3-to-6-months-2025-3)

**Anthropic's CEO says that in 3 to 6 months, AI will be writing 90% of the code software developers were in charge of**

Summarize

Kwan Wei Kevin Tan



"And then in twelve months, we may be in a world where AI is writing essentially all of the code," Anthropic CEO Dario Amodei said at a Council on Foreign Relations event on Monday. Halil Sagirkaya/Anadolu via

# AI and Coding

Home > News > AI

## AI Now Writes Over 25% of Code at Google

Google CEO Sundar Pichai says the company is using AI to write code, w engineers. But will that be worth it in the long run?



By [Kate Irwin](#)

October 30, 2024



[www.pcmag.com/news/ai-now-writes-over-25-percent-of-code-at-google](https://www.pcmag.com/news/ai-now-writes-over-25-percent-of-code-at-google)

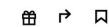
## Google says 75% of the company's new code is AI-generated

By [Hugh Langley](#) [+ Follow](#)



Google CEO Sundar Pichai and DeepMind CEO Demis Hassabis. Raj K Raj/Hindustan Times via Getty Images

Apr 22, 2026, 6:17 AM MT



AD

SPONSORED BY Fourth

The restaurant tech fix that saves time, money and headaches

Less-experienced developers: AI functions as both assistant and tutor.

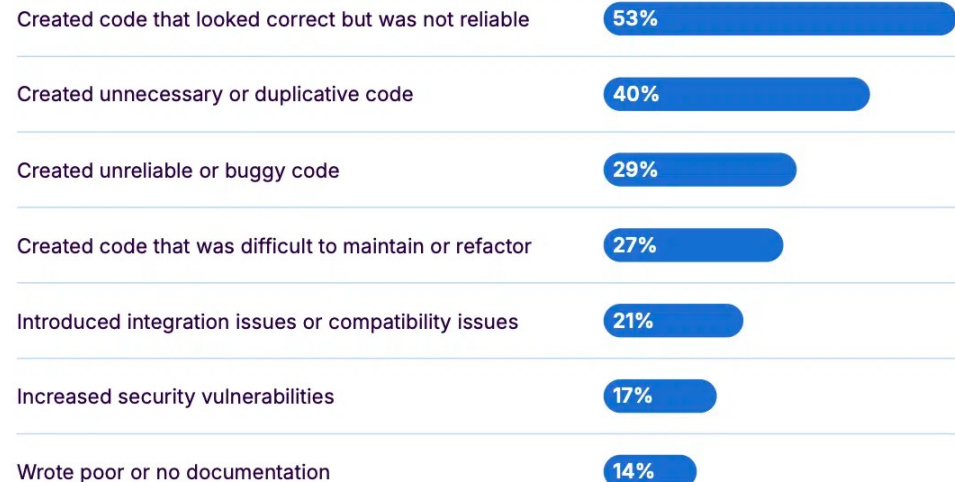
# AI and Coding

<https://shiftmag.dev/state-of-code-2025-7978/>

## 88% of developers report at least one negative impact of AI on technical debt

*In which of the following ways, if any, has AI-generated or assisted code impacted your team / company's technical debt?*

**88%**  
respondents  
citing at least  
one issue



*n*=1,136

# Vibe Coding

---

<https://shiftmag.dev/the-illusion-of-vibe-coding-5297/>

6.06.2025. [Artificial Intelligence](#) [Frantisek Lucivjansky](#)

## The Illusion of Vibe Coding: There Are No Shortcuts to Mastery

In the age of AI-assisted development, it's tempting to believe we're entering a golden era where human effort in programming becomes obsolete.

 [Frantisek Lucivjansky](#)

