The Pathologically Eclectic Rubbish Lister

“Languages were first invented by humans, for the benefit of humans. In the annals of computer science, this fact has occasionally been forgotten. Since Perl was designed (loosely speaking) by an occasional linguist, it was designed to work smoothly in the same ways that natural language works smoothly.”
- Larry Wall, creator of Perl

“At every level, Perl is about helping you to get from here to there with minimum fuss and maximum enjoyment. That’s why so many Perl programmers go around with a silly grin on their face.”
- Larry Wall, Programming Perl, 2nd Edition

“Just remember that Perl is trying to understand what you’re saying, like any good listener. Perl works pretty hard to try to keep up its end of the bargain. Just say what you mean, and Perl will usually ‘get it’.”
- Larry Wall

“Perl is conducive to dealing with chunks the way a human thinks of it instead of bending over backwards to make it easy on the computer. We [should] optimize for human time above computer time, although we try to optimize computer time when we get a chance. The person has to come first.”
- Tom Christiansen

“Larry Wall likes to say that Perl makes easy things easy and hard things possible. Professional programming languages tend to make all things equally difficult.”
- Tim O’Reilly

“There was a huge ecological niche between the C language and Unix shells. C was good for manipulating complex things -- you can call it ‘manipulexity.’ And the shells were good at whipping up things -- what I call ‘whipupitude.’ But there was this big blank area where neither C nor shell were good, and that’s where I aimed Perl.”
- Larry Wall, Salon Magazine

“The fact is, your brains are built to do Perl programming. You have a deep desire to turn the complex into the simple, and Perl is just another tool to help you do that.”
- Larry Wall, 1998 State of the Onion

“It’s big and slow, but it can go a long way without water.”
- Larry Wall on Perl, explaining why he asked for a camel as the animal on the cover of Programming Perl

“In a literary light, if UNIX is the Great Novel, Perl is the Cliffs Notes.”
- Thomas Scoville, “The Elements of Style: UNIX as Literature”

“Perl isn’t so much a Unix command as a way of life.”
- Mike Epstein
What is Perl?

Perl is technically one programming language among many, but to many people, Perl is the ultimate programming language.

Perl would not be a suitable language to teach in CS50 – while C is simple and usually has one best way to accomplish a programming task, Perl’s philosophy is “TMTOWTDI” – There’s More Than One Way To Do It. Perl is not particularly simple.

Perl would not be a good language with which to write an operating system, or a commercial word processor. But if you have small tasks that you need to get done, Perl is often by far the easiest way to accomplish it. Any Perl program is almost certainly going to run slower than its C counterpart. But when you factor in the extra hours that writing in C may require, the nanoseconds you save with a compiled program don’t seem so great anymore.

Ultimately, Perl is about getting tasks done quickly. Larry Wall, the creator of Perl, says that there are three virtues of a great programmer: “laziness, impatience, and hubris.” If you want the most efficient possible program, Perl is not for you. But if you just want to get a program working and get some sleep, then time and time again you will be very glad you know Perl.

This seminar will teach you enough Perl to write most simple programs. You won’t know everything about Perl – nobody except Larry Wall knows everything. There is no ANSI standard on Perl. However, you don’t need to know everything about Perl to program effectively. As you’ll hear time and time again, There’s More Than One Way To Do It.

What is CGI?

CGI is not a programming language. CGI stands for Common Gateway Interface. It is a very simple way in which a program, written in any language, can interact with users via the World Wide Web. It is a specification governing how programs can generate Web pages and can receive input via Web-based forms. Without CGI, Web pages would be restricted to static files on the Web server that are displayed to the user. CGI opens up huge possibilities for making Web pages interactive.

Because of Perl’s amazingly powerful text processing features and because it is so easy to write Perl programs, Perl has become the clear language of choice for writing CGI scripts. This use may have been the strongest factor in making Perl such a fixture of the Unix world today.

In this seminar, you will learn the details of the CGI specification and learn how to write programs that process input from Web forms, generate Web pages, and interact with the user.
The Perl Calendar

It is easy to explain Perl, but without actual experience programming in Perl, this knowledge would be useless. After all, Perl is a language for getting tasks done, not for sitting in a classroom analyzing programming algorithms. Therefore, we have decided to teach Perl in the context of a project. During the course of the seminar series, we will all work on this project together, teaching you the Perl you need and then working together to make the project come to life.

In the first half of the seminar, while you learn the Perl programming language, you will build a set of Perl scripts which keep track of your calendar in a file in your account. Then in the second half, while learning CGI, you will expand these scripts to use the Web for data input and output.

My Harvard

Many big software and/or Internet companies, such as Microsoft, Netscape, Yahoo, Amazon.com, and C|Net have recently been strongly pushing portal sites. A portal site is a single place that you want to make your home page and that provides one-stop access to all of the vital information you want on a regular basis. Commercial portals have news headlines, weather, sports scores, and useful links.

The Harvard Computer Society is creating such portal site for Harvard students, which is tentatively named My Harvard. The portal will contain a variety of very useful information, such as:

- Upcoming events from the Harvard Online Calendar System
- Current headlines from the Harvard Crimson
- The next dining hall menu
- The next shuttle(s) to where you live
- Boston/Cambridge weather

Besides having all of this amazingly useful information, the portal will be fully customizable. Any user can set his/her own preferences regarding what information should appear. Then, when he/she views the portal, whether from a dorm room computer or a kiosk, the portal will use the saved preferences when generating the page.

Lee Feigenbaum and David Mitby are leading the My Harvard project. After the normal Perl instruction from 3-4 p.m., they will be working on the project from 4-5 p.m. using the Perl knowledge that you have learned that day. You may not think you know enough Perl right now to do the project, but that is why you are taking the seminar. As you learn more Perl, you will easily be able to participate in every way.
Syllabus

Week 1 (October 25): Introduction to Perl.
- The philosophies underlying Perl.
- Scalars, quoting, and variable interpolation.
- Basic I/O and regular expressions.
- First stage of your Perl calendar: displaying events from a file.

Week 2 (November 1): Reading, writing, and processing text.
- Input and output with filehandles.
- Arrays and split().
- Basic control structures.
- Second stage your Perl calendar: getting events from the command line.

Week 3 (November 8): Data manipulation.
- Hashes.
- Functions and custom sorting.
- Formatted output.
- Third stage of your Perl calendar: sorted and formatted output.

- The basics of CGI: programs that output a Web page.
- Running scripts from the Web.
- A review of HTML.
- Fourth stage of your Perl calendar: displaying your calendar on the Web.

Week 5 (December 6): Making the Web interactive.
- Processing Web forms.
- CGI security.
- Data persistence with CGI.
- Fifth stage of your Perl calendar: creating forms to enter events.

Week 6 (December 13): A taste of things to come.
- The Perl manpages.
- CPAN: Modules and packages.
- References and advanced data structures.
- Perl command-line operation.
- Wrapping up your Perl calendar.
Prerequisites

This seminar is designed for people who already have some knowledge of programming. If you already know C or are taking CS50, then you definitely know enough to take this seminar. If you are unsure, please contact perl@hcs.harvard.edu.

When, Where, and How

The seminars will be held on Sunday afternoons in the PC Classroom (Science Center 120) from 3 to 4 p.m. each week starting October 25 and continuing through December 13. We will not have a seminar November 22, for the Harvard-Yale game, or November 29, for Thanksgiving Vacation. Therefore, we will meet a total of six times.

Immediately following the seminars, from 4-5 p.m., Lee Feigenbaum and David Mitby will work on the My Harvard project with everyone who is interested.

We have a mailing list for the seminar: perl-list@hcs.harvard.edu. You will be automatically added if you sign up at the first seminar meeting; if you are not on the list, send e-mail to majordomo@hcs.harvard.edu with the line "subscribe perl-list" in the message body.

If you have any questions, please send e-mail to perl@hcs.harvard.edu.

Reading List

- **Perl Language Reference Guide**, by David Alpert, published by FAS Computer Services. You should get a free copy of this packet along with the one you are now reading. This packet has an overview of the Perl language, which you may find useful later when programming in Perl.

- **CGI Reference Guide**, by David Alpert, published by FAS Computer Services. You will get a free copy of this packet once we begin the CGI portion of this seminar. The packet has a lot of information about CGI which you may find useful for creating CGI programs.

- **Programming Perl, 2nd ed. (Camel)**, by Larry Wall, Tom Christiansen, and Randal L. Schwartz, published by O'Reilly and Associates, Inc. This is the Bible of Perl. You should absolutely definitely own this book.

- **Learning Perl, 2nd ed. (Llama)**, by Randal L. Schwartz and Tom Christiansen, published by O'Reilly and Associates, Inc. This is the best way to teach yourself Perl. You don't need this book if you go to this seminar series, but it can be extremely helpful. I particularly recommend it if you don't have very much background in programming.

- **The Perl Cookbook**, by Tom Christiansen and Nathan Torkington, published by O'Reilly and Associates, Inc. This contains a lot of examples of Perl code to do a lot of really useful things. If you are trying to figure out how to do something in Perl, it is probably in here.