

# Perl

The *Friendly* Programming Language.

A short talk to introduce the Perl programming language and its *culture*.

Adam John Trickett

[www.iredale.net](http://www.iredale.net)

[adam.trickett@iredale.net](mailto:adam.trickett@iredale.net)

GPG Key: 0xAF0DB8C8



# History

- Perl is a programming language created by Larry Wall on 18 December 1987
- Perl 5 released on 17 October 1994
  - Perl 5.10.0 released 18 December 2007
  - 5.12 ... 5.14 ... 5.16 ... 5.18
  - Perl 5.20.0 released 27 May 2014
- Perl 6 before xmas 2015...



# Perl's Parents

- Perl is a language that unashamedly takes ideas from other languages, including its self...
- awk/nawk
- sed
- shell
- c
- Lisp
- Haskell & Ruby



# Compiled or Interpreted?

- Perl source code is compiled very quickly into a syntax tree
- The syntax tree is automatically optimised
- Finally the optimised syntax tree is executed
- Perl is compiled and interpreted just like other modern languages such as Java.



# **“hard tasks easy and seriously non-trivial tasks possible”**

- Perl is designed to make life easy for the programmer
- Autovivication of variables
- Dynamic typing
- Automatic memory management
- Hashes
- Procedural, object-oriented or both
- Many, many built-in convenience features



# Code



# Hello World

```
print 'Hello World'
```



# Hello World MK-2

```
#!/usr/bin/perl
```

```
use strict;
```

```
use warnings;
```

```
my $greeting = "Hello World\n";
```

```
print $greeting;
```

```
exit;
```





# Hello New World

```
#!/usr/bin/perl  
use 5.01;  
use strict;  
use warnings;  
  
my $greeting = 'Hello World';  
say $greeting;  
exit;
```



# Scalars

- Are not typed, can hold a number, string, or reference
- Perl will automatically cast between types for you

```
$var1 = 'this is a string';
```

```
$var2 = 23.5;
```

```
$var3 = $var1 . $var2;
```

```
say $var3
```

```
this is a string23.5
```



# Arrays & Lists

- Lists are just a list of scalars, lots of built in commands generate or process lists
- You can put a list into an array to store it

```
@array = (1,2,'three');
```

```
say $array[2];
```

```
three
```

```
$var4 = shift @array;
```

```
say $var4
```

```
1
```



# Hashes

- Hashes are special arrays that use keys rather than numbers to access them
- They are VERY fast but the order is not predictable
  - but there are fast sorting tools

```
%hash = ( 'k1' => 'v1', 'k2' => 'v2' );
```

```
say $hash{'k1'};
```

```
v1
```



# Pros & Cons



# Perl's Strengths

- Expressive language
  - **T**here **I**s **M**ore **T**han **O**ne **W**ay **T**o **D**o **I**t
- Popular
- Easy



# Perl's Weaknesses

- Expressive language
  - **T**here **I**s **M**ore **T**han **O**ne **W**ay **T**o **D**o **I**t
- Popular
- Easy



# Getting Help

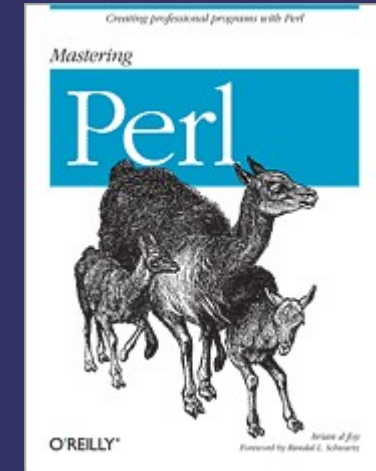
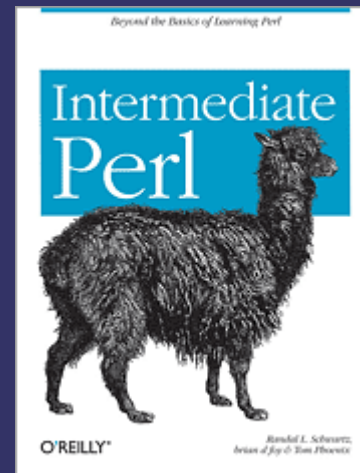
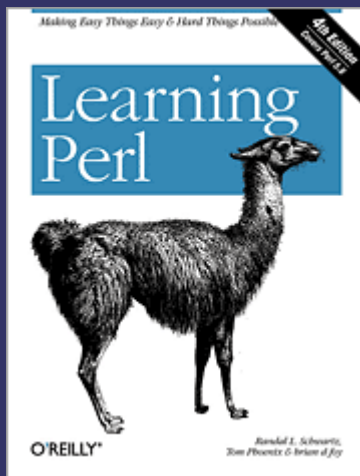




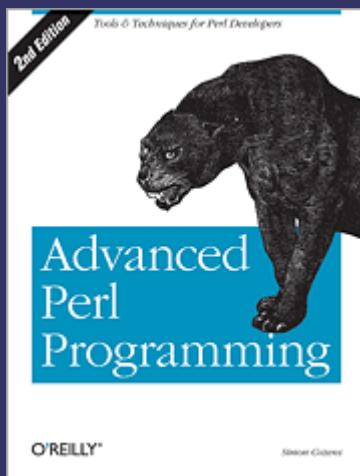
## Useful Resources

- Perl Training Australia:
  - <http://perltraining.com.au/notes.html>
- Ovid's CGI Course:
  - <http://ovid-cgi-course.perl-begin.org/cgi-course/>
- Learn.perl.org:
  - <http://learn.perl.org/>
- On-line Documentation:
  - <http://perldoc.perl.org/>
- Randal's Perl Columns:
  - <http://www.stonehenge.com/merlyn/columns.html>

# Some Books



<http://books.perl.org/>





# Comprehensive Perl Archive Network

- Over 25,000 modules
- A module for every imaginable task:
  - Web/CGI
  - Databases
  - SAP
  - MP3/Oggs
  - Time/Dates
  - Templating
  - and much much more!





# CPAN

- Global system with mirrors worldwide
- PAUSE
  - Perl Authors Upload SErver
- RT
  - Request Tracker
- CPAN testers & CPANTS
- Search/Meta Search
  - Two Web Based Search engines



# Killer Modules!

- CGI
- DBI
- Template
- CPAN
- Apache mod\_perl
- File::Find::Rule
- LWP/WWW::Mechanize
- MVC/ORM



# Some Community Sites

- PerlMonks:
  - <http://www.perlmonks.org/>
- Perl Blogs:
  - <http://blogs.perl.org/>
- Perl Foundation:
  - <http://www.perlfoundation.org/>
- Perl Mongers:
  - <http://www.pm.org/>



## More Sites

- Perl 5 Wiki
  - <https://www.socialtext.net/perl5/>
- {Perl School}
  - <http://perlschool.co.uk/>
- Enlightened Perl Organisation
  - <http://www.enlightendperl.org/>



# Meetings

- OSCON - Open Source Conference
  - Originally Perl only, now everything open-source
  - Expensive
  - <http://conferences.oreillynet.com/>
- YAPC - Yet Another Perl Conference
  - Still just Perl
  - Cheaper
  - <http://www.yapceurope.org/>





# Smaller Meetings

- PM Workshops
  - London.PM Workshop every winter
- PM TechMeets
  - London.PM and ThamesValley.PM
- Perl Birds of a Feather “BOF” meetings
  - LUGs
- Perl tracks/streams at other meetings
  - FOSDEM



## More Links...

- <http://www.cpan.org/>
- <http://search.cpan.org/>
- <http://metacpan.org/>
- <http://rt.cpan.org/>
- <http://pause.perl.org/>
- <http://jobs.perl.org/>
- <http://cpants.perl.org/>
- <http://perldoc.perl.org/>
- <http://www.perl.com/>



# Some Screen Shots of Sites



There's more than one way to do things

Search PerlMonks

# The Monastery Gates

- [| Login](#) | [| Create a new user](#) | [| The Monastery Gates](#) | [| Super Search](#)
- [| Seekers of Perl Wisdom](#) | [| Meditations](#) | [| PerlMonks Discussion](#)
- [| Obfuscation](#) | [| Reviews](#) | [| Cool Uses For Perl](#) | [| Perl News](#) | [| Q&A](#) | [| Tutorials](#)
- [| Poetry](#) | [| Recent Threads](#) | [| Newest Nodes](#) | [| Donate](#) | [| What's New](#)

(#131=superdoc: [print w/ replies, xml](#))

[Need Help??](#)

**Donations gladly accepted**

If you're new here please read [PerlMonks FAQ](#) and [Create a new user](#).

Want Mega XP? Prepare to have your hopes dashed, join in on the: [poll ideas quest 2013](#) (Don't worry; you've got plenty of time.)

## New Questions

[how do I force a specific session, not my own, to end?](#) by [ted.byers](#)  
2 direct replies — [Read more / Contribute](#) on Mar 08, 2013 at 14:38

I am using CGI::Session, and for the most part, all is well. However, I have one requirement that I have not found even mentioned in the documentation.

One of the session parameters that I set is user\_id (guaranteed to be unique). The use case I need to address involves an administrator who needs to disable a given user's account. However, there is the possibility of a problem since all users can change their own password. Since the obvious solution to disable an account is to change the password to something random, if the user has a current session, it is entirely possible that he might change his password to something else after the administrator has attempted to disable the account. On writing this, it occurs to me that I

**Log In**

Username:

Password:

remember me

[What's my password?](#)

[Create A New User](#)

**Chatterbox**

[NodeReaper](#) stumbles on his way to the attic

[How do I use this?](#) | [Other CB clients](#)

**Other Users**

Others making s'mores by the fire in the courtyard of the Monastery: (3)

[atcroft](#)

[stefby](#)

[ljinism](#)

# Perl.com

news and views of the Perl programming language

## Perl Today (February 2014)

By brian d foy on February 10, 2014 6:00 AM



*brian d foy is the author of [Mastering Perl](#), now available in its second edition, as well as several other Perl books. As the founder of Perl mongers, he's been active in the Perl community for almost 20 years.*

Perl's so big now that it's almost impossible to pay attention to everything going on. Instead of reading the thoughts of me, just one person, on the current state of Perl, why not gather several major players who can cover many areas that you may not have noticed? I've collected some voices from parts of the large and diverse Perl community to highlight a small fraction of everything going on, from community development, hard core coding, the job market, and various problem domains. Although not definitive, their combination makes a good summary.



*Randal Schwartz is the original author of [Programming perl](#) and [Learning Perl](#), along with numerous other Perl credits. He's the current host of [FLOSS Weekly](#)*

Perl started as a Unix shell replacement to help sysadmins get more productive with less risk and a flatter learning curve (you didn't have to learn the quirks of dozens of small utilities, just the quirks of One [Larry Wall](#)).

But then along came the web, and "scripts" could be used to provide interactive web pages, and Perl became the darling language of the web, first with "cgi-lib.pl", and then with [CGI.pm](#), and later with larger frameworks like [Mason](#) and [Catalyst](#).

Of course, Perl's scripting abilities also matured, and the ability to scale Perl programs with modern testing and OO frameworks (like [Moose](#)) empowered programmers to write 10-line scripts and 100,000-line applications using the same core language features: a distinctive advantage.

Perl 5 development stagnated a bit, triggering the whole "Perl 6" revival, and the relationship between Perl 5 and Perl 6 is still not understood by most people outside the direct Perl community (nor even by some who are "in" the community). But with the establishment of the quarterly point release, and the annual major releases, Perl 5 seems to have gotten firmly back in the saddle of modern development again. And while Perl 6 continues to be developed and redesigned, parts of it are already quite useful for early adopters.



*Karen Pauley was a founding member of the Belfast Perl Mongers, a volunteer with the YAPC Europe Foundation, and is the President of The Perl Foundation.*

For me, Perl is a reason for creating community systems to support Perl itself. [The Perl Foundation](#) doesn't write the Perl language, we don't influence its direction; instead we work with the Perl community to organize funding, volunteers, marketing, and legal counsel.

Visit the home of the Perl programming language: [Perl.org](#)

[Download](#)

[Documentation](#)

[Perl Bloggers](#)

[Foundation News](#)

Sponsored by

**Tom Christiansen  
Perl Training**

Recent Entries

[Perl Today \(February 2014\)](#)

[The Year in Perl 2013 Retrospective](#)

[Enable Backtraces with Carp::Always \(Modern Perl Tips\)](#)

[Lexing and Parsing Continued](#)

[Consuming RESTful Services with Perl](#)

[Designing Board Games With Perl](#)

[Newcomer Experience in the Perl Community Survey](#)

[An Overview of Lexing and Parsing](#)



## 25,000 extensions on CPAN

That's why we love Perl 5

▶ [Get started](#)

↓ [DOWNLOAD PERL](#)



Perl 5 is a highly capable, feature-rich programming language with over 27 years of development. [More about why we love Perl...](#)



### [Learning Perl 5](#)

With free online books, over 25,000 extension modules, and a large developer community, there are many ways to learn Perl 5.



### [The Perl Community](#)

Perl has an active world wide community with over 300 local groups, mailing lists and support/discussion websites.



### [Documentation](#)

Core documentation, FAQs and translations.



### [Contribute to Perl](#)

Perl is being [actively developed](#). There are many ways to [get involved](#).



### [CPAN](#)

The Comprehensive Perl Archive Network (CPAN) has over 25,000 open source distributions available for download.



### [Events and Conferences](#)

Conferences, social and technical events around the world offer lots of networking and learning opportunities.



### [Perl 6](#)

Perl 6 is a sister language, part of the Perl family. Perl 6 is not production ready yet, you can get involved with its [development](#) though.



### [The Perl Foundation](#)

The Perl Foundation is dedicated to the advancement of the Perl programming language through open discussion, collaboration, design, and code.

### Current Perl version

[5.20.1 - download now](#)

### Find out more

[Learn](#)

[Documentation](#)

[Community](#)

[Events](#)

### Tip

#### Web frameworks

[Catalyst](#) is a Model View Controller (MVC) Web application framework

### Sponsor

[digital craftsmen](#)



bespoke hosting services

Home

About

History

Resources

People

Businesses

Wiki

Join



# London Perl Mongers

## London Perl Mongers

### Forthcoming meetings

Today ◀ ▶ Friday, 13 February ▾

Week Month **Agenda**

#### Wednesday, 4 March

13:00 [Dim-Sum](#)

#### Saturday, 21 March

12:00 [Perl Programming Hackday](#)

#### Friday, 10 April

[NLPW - Dutch Perl Workshop](#)

*Displaying events until 15/4. [Look for more](#)*

Events shown in time zone: London

Google Calendar

### Who are the London Perl Mongers?

We are a group of [people](#) dedicated to the encouragement of all things to do with the [Perl](#) programming language in London, U.K.. This involves helping each other, [discussing](#) topics, sharing information and the occasional drink.

We are part of the world-wide [Perl Mongers](#) group and play host to (as well as visit) others from round the world.

### Where Are the London Perl Mongers?



in All





in All CPAN Search

### XML-RSS-Tools-0.34

**This Release** XML-RSS-Tools-0.34 [Download] [Browse] 27 May 2014

**Other Releases** XML-RSS-Tools-0.31 -- 30 Jun 2008 [Goto]

**Links** [ Discussion Forum ] [ View/Report Bugs (0) ] [ Dependencies ] [ Other Tools ]

**Rating** ☆☆☆☆ (0 Reviews) [ Rate this distribution ]

**License** open\_source

**Special Files** [Build.PL](#) [LICENCE](#) [MANIFEST](#) [README](#)  
[Changes](#) [Makefile.PL](#) [META.json](#) [SIGNATURE](#)



**Modules**

[XML::RSS::Tools](#) A tool-kit providing a wrapper around a RSS client, a RSS parser, and a XSLT engine. 0.34

- Documentation**
- [Example-1](#) Very simple example
  - [Example-2](#) A longer example
  - [Example-3](#) Another simple example
  - [Example-4](#) Using a single XSLT stylesheet
  - [Example-5](#) A complete web server based RSS-client
  - [Introduction](#) A Simple Introduction To RSS

97576 Uploads, 1307 Distributions  
1233 Modules, 12041 Uploaders

Hosted by [craftsmen](#)  
digital craftsmen

XML-RSS-Tools-0.34 ... x

https://metacpan.org/release/XML-RSS-Tools

Ceausescu

Local Remote News Computing Personal Various Stuff Planets Web Admin Basingstoke Weather G TV Guide

Home Recent FAQ Feedback News Lab

Sign in

meta::cpan



7 results (0.004 seconds)

Adam John Trickett / XML-RSS-Tools-0.34

MAY 27, 2014

- Browse (raw)
- Changes
- Issues (0)
- Testers (228 / 6 / 3)
- Kwalitee
- License: open\_source

ACTIVITY

24 month

TOOLS

- MOBI | EPUB
- Download (38.79Kb)
- MetaCPAN Explorer
- ★ Rate this distribution
- Subscribe to distribution




PERMALINKS

- This version
- Latest version
- S.C.O
- This version
- Latest version

#### Documentation

- Example-1
- Example-2 - A Longer example
- Example-3
- Example-4
- Example-5 - A complete web server based RSS-client
- Introduction - A Simple Introduction To RSS

#### Modules

**XML::RSS::Tools** - A tool-kit providing a wrapper around a HTTP client, a RSS parser, and a XSLT engine.

#### Examples

- docs/example-1.pod
- docs/example-2.pod
- docs/example-3.pod
- docs/example-4.pod
- docs/example-5.pod
- examples/example-1.pl [pod]
- examples/example-2.pl [pod]
- examples/example-3.pl [pod]
- examples/example-4.pl [pod]
- examples/example-4.xsl
- examples/example-5.pl [pod]

#### Other files

- Build.PL
- Changes
- MANIFEST
- META.json
- META.yml
- Makefile.PL
- README



ATRICKETT

Adam John Trickett  
and 1 contributors

show them

#### DEPENDENCIES

LWP  
URI  
XML::LibXML  
XML::LibXSLT  
XML::RSS

and possibly others

- CPAN Testers List
- Reverse dependencies
- Dependency graph

## Perl 5 Wiki

▼ hide

### Registration and Support

804 Revisions

 [Watch](#)

 [Tools](#) ▶

#### Tags

[description](#) × [global](#) ×  
[hyperlinks](#) × [introduction](#) ×  
[main](#) × [navigation](#) ×  
[Recent Changes](#) × [style](#) ×

#### Incoming Links

There are no pages that link to this page yet.

#### Attachments

#### Contents

- [Registration and Support](#)
  - [Perl 5 Wiki Resources](#)
    - [Open Data](#)
    - [General](#)
    - [CPAN \(Comprehensive Perl Archive Network\)](#)
    - [News & Discussions](#)
    - [Perl Culture](#)
    - [Who Uses Perl?](#)
    - [Perl for your career](#)
    - [How to Find More Information](#)
    - [Introductory/General](#)
    - [Intermediate](#)
  - [About This Wiki](#)
    - [What Goes On This Wiki?](#)
    - [What Does Not Go On This Wiki?](#)



## GFU Cyrus AG is sponsoring the QA Hackathon

By **tinita** on February 13, 2015 12:12 AM

We're happy to announce that [GFU Cyrus AG](#) will sponsor the [Perl QA Hackathon 2015](#).

Originally founded as GFU Cyrus + Rölke mbH in 1980, GFU Cyrus AG is now one of the best-known IT training companies in Germany.

Several thousand IT specialists from all over the Federal Republic of Germany visit seminars at GFU every year. The program includes more than 1000 different seminars.

0 comments

[Continue reading »](#)

## STRATO is sponsoring the QA Hackathon

By **Neil Bowers** on February 10, 2015 9:30 PM

We're happy to announce that [STRATO](#) has decided to sponsor this year's [QA Hackathon](#).

If you live in certain parts of Europe then you will probably have heard of STRATO, but others might not.

STRATO is an ISO 27001-certified hosting provider with its headquarters in Germany. STRATO offers domains,

email and homepage packages, online storage, web shops and servers through to high-end solutions.



0 comments

[Continue reading »](#)

### About blogs.perl.org

blogs.perl.org is a common blogging platform for the Perl community. Written in Perl and offering the modern features you've come to expect in blog platforms, the site is run by Dave Cross and Aaron Crane, with a design donated by Six Apart, Ltd.

### Search blogs.perl.org



# catalyst

Web Framework

## Perl MVC framework

## Turbocharge your web development!

Catalyst is the most popular Perl MVC framework and makes creating web applications fun, rewarding and quick.



**Get started now!**

Open Source, GPL/Artistic

[About](#)
[Sites using Catalyst](#)
[Documentation](#)
[Wiki](#)
[Download](#)
[Support](#)
[How-Tos](#)

## What is Catalyst?

Catalyst is an open-source Perl MVC web framework that encourages rapid development and clean design without getting in your way by forcing rules.

We tend to keep things small and simple. This gives us robustness and scalability - your Catalyst-based app also inherits those. No complicated object hierarchies.

Why reinvent the wheel? When you write your app with Catalyst, you don't have to care much about session handling or authorization. You just use it. CPAN has a vast amount of power, which we bring to you. If you don't like something in default setup, why not replace it? There Is More Than One Way To Do It. You can choose from a wealth of available models and views, drop in a bunch of plugins and create the setup that fits you best.

Many high-profile sites like the **BBC iPlayer**, **123people**, **magazines.com** and **urbia.de** are already using Catalyst. Check

## Why Catalyst?

### Versatile Models

Use a powerful database ORM like `DBIx::Class` for your model, plain old `DBI` or even `LDAP` - Catalyst supports a wide range of models.

### Flexible Views

Use whatever templating language you like best: `Template::Toolkit`, `Mason 2`, `HTML::Template` and many more are fully supported.

### Integrated Development server

Catalyst comes with its own lightweight test server for development. It automatically restarts when your sources have changed, so you get instant results!

### Works with all major webservers

Catalyst 5.9 has native `PSGI/Plack` support which makes deployment for most webservers a no-brainer. Plus, there are loads of engines

## How do I get started?

### 1. Quick Install Guide

For more details and other platforms read the extended [Installing Catalyst guide](#).

### 2. Tutorial

Learn how to build your first complete Catalyst application with all the nuts'n'bolts of modern web applications.

## Get the book

### The Definitive Guide to Catalyst



This book by Kieren Diment, Matt Trout and other core Catalyst developers, published by **Apress** is now available for purchase from **Amazon.com**, or get the eBook (PDF) from **Apress**.

• The easiest way to write web applications in Perl •

[Quick Start](#) [Docs](#) [About](#) [IRC](#)



[Contribute](#) [Dancers](#) [Slides](#)

Fork me on GitHub

## What is Dancer?

Dancer is a simple but powerful web application framework for Perl.

## Key features

- **Dead Simple** - Intuitive, minimalist and very expressive syntax.
- **Flexible** - PSGI support, plugins and modular design allow for strong scalability.
- **Few dependencies** - Dancer depends on as few CPAN modules as possible making it easy to install.

## Prepare your moves... and dance!

Dancer has been designed as a Perl web framework which is easy and intuitive to get to grips with - this is how easy it can be:

```
#!/usr/bin/env perl
use Dancer;

get '/' => sub {
    "Hello World!"
};

dance;
```


```
$ perl bin/app.pl &
...
$ curl http://localhost:3000/
Hello world!
```

Download 0.158

Getting started with Dancer is easy, but it provides [stable releases](#) if you need to write your web applications with ease! You can also follow Dancer on [GitHub](#) and on [Twitter](#)



1



# Perl

The *Friendly* Programming Language.

A short talk to introduce the Perl programming language and its *culture*.

Adam John Trickett


`www.iredale.net`  
`adam.trickett@iredale.net`  
GPG Key: `0xAF0DB8C8`

Version 1.4.1 © Adam Trickett February 2015 Distributed under a creative commons Attribution-NonCommercial-ShareAlike licence.

Hi!

This is a little talk about Perl. It's not really a talk about the language much, but about the culture around it and where to begin if you want to avoid some of the pitfalls.





2

## History

- Perl is a programming language created by Larry Wall on 18 December 1987
- Perl 5 released on 17 October 1994
  - Perl 5.10.0 released 18 December 2007
  - 5.12 ... 5.14 ... 5.16 ... 5.18
  - Perl 5.20.0 released 27 May 2014
- Perl 6 before xmas 2015...

Version 1.4.1 © Adam Trickett February 2015

Distributed under a creative commons Attribution-NonCommercial-ShareAlike licence.

Perl is old but still very much alive.

Current Perl started with version 5, but modern Perl really started with the more recent 5.10 in 2007.

Current versions of Perl 5 are released in a time based manner and an enormous under the covers clean up has taken place over the years, prolonging the useful life of the core engine

Perl 6 was first talked about in 2000, but has taken on a life of its own, and the general joke is that it would be released for xmas – though which one was never specified. This year it was announced that a beta would be released this year and “1.0” release would arrive by xmas 2015... Perl 6 is a different language and I'm not going to talk about it.



## Perl's Parents

- Perl is a language that unashamedly takes ideas from other languages, including its self...
- awk/nawk
- sed
- shell
- c
- Lisp
- Haskell & Ruby

Perl doesn't care where good ideas come from, so it sucks in ideas from all over the place.

It is really a high level language at heart, so it pulls in a lot from shell, with the text processing abilities from sed and awk.

To make it more powerful than these more basic scripting languages it pulled in ideas from c and Lisp, and more recently from Haskell and Ruby, Perl 5 even includes ideas from Perl 6...



## Compiled or Interpreted?

- Perl source code is compiled very quickly into a syntax tree
- The syntax tree is automatically optimised
- Finally the optimised syntax tree is executed
  
- Perl is compiled and interpreted just like other modern languages such as Java.

Perl is designed to get results as quickly as possible, so it trades pure run time speed for ease of use. It's not an interpreted language like old style BASIC nor is it a pure compiled language.

Llike Java it's converted and optimised into an intermediate language and that is the one that's run through the core engine.



## **“hard tasks easy and seriously non-trivial tasks possible”**

- Perl is designed to make life easy for the programmer
- Autovivification of variables
- Dynamic typing
- Automatic memory management
- Hashes
- Procedural, object-oriented or both
- Many, many built-in convenience features

Perl trades CPU cycles and RAM for ease of use and getting the job done. It is not designed for highly restricted spaces or roles.

Generally speaking programmer time is expensive and computers are not, so Perl is optimised for easy prototyping, debugging and getting an answer, not using the least resources possible.



# Code



Here is hello world, save it as a text file and send it to the Perl interpreter. Should work on any Unix, Linux, Mac OSX and even Windows box with the right bits added.

It's not much use on it's own.



## Hello World MK-2

```
#!/usr/bin/perl

use strict;
use warnings;

my $greeting = "Hello World\n";
print $greeting;
exit;
```

The first line is for Unix like systems, Perl doesn't actually need it.

The “use” command tells Perl to load something already written into this program. This pair are a good idea but not required on the very simple programs you can write.

We then create a scalar variable and assign something to it. We then print it out and finally we end.



## Hello New World

```
#!/usr/bin/perl  
use 5.01;  
use strict;  
use warnings;  
  
my $greeting = 'Hello World';  
say $greeting;  
exit;
```

This version uses the new features that came out with 5.10. If you try to run it on an older version it will complain that it's too old rather than crashing in a heap. The say command is shorter than print and automatically adds an end of line for you.

There are lots of other things in 5.10 but you need to tell Perl to use them or you won't have access to them.





## Scalars

- Are not typed, can hold a number, string, or reference
- Perl will automatically cast between types for you

```
$var1 = 'this is a string';  
$var2 = 23.5;  
$var3 = $var1 . $var2;  
say $var3  
this is a string23.5
```

Perl's most basic variable type is a “scalar”, which holds a single thing, it could be a number, a character, a string of characters, a reference to something else, or a file handle.

You can name then most things, but they always begin with a dollar, a modified S for scalar.



## Arrays & Lists

- Lists are just a list of scalars, lots of built in commands generate or process lists
- You can put a list into an array to store it

```
@array = (1,2,'three');  
say $array[2];  
three  
$var4 = shift @array;  
say $var4  
1
```

Perl's next type is a list, which is just that, it's a collection of things, they don't have to be the same kind of things, so each value in the list could be as with a scalar one of many things. Lots of functions work on lists, return lists or expect lists.

An array is a named thing that holds a list, and it is named with an at symbol, @ for array.

To access an individual element of an array you use an index number, but now the symbol changes to a dollar as you are accessing a single value.

@foo is an array

\$foo[3] is the scalar at index 3 (starting from 0)



## Hashes

- Hashes are special arrays that use keys rather than numbers to access them
- They are VERY fast but the order is not predictable
  - but there are fast sorting tools

```
%hash = ( 'k1' => 'v1', 'k2' => 'v2' );  
say $hash{'k1'};  
v1
```

Hashes are a fancy kind of array, where you don't access the value with a number but you use a named key. Perl has very clever hashing software which means you can have an hash of millions of things but still get to the one you one almost instantly. All that matters is that your keys are unique. The order in the hash is not preserved, so there are fast sorting tools allowing you to sort by key or value ascending or descending. The % is used to name a hash, but as with an array, the \$ is used when you refer to a single element.

Hashes are used for lots of things because they allow you to create complex multidimensional objects.



# Pros & Cons



## Perl's Strengths

- Expressive language
  - **T**here **I**s **M**ore **T**han **O**ne **W**ay **T**o **D**o **I**t
- Popular
- Easy

Perl is a highly expressive language, which makes it popular with creative people. People revel in the ability to solve problems in many different ways and styles.

Many Perl hackers are proud of “TIMTOWTDI”, it is something engrained into the culture.

In the early days of the WWW, Perl was extremely popular, even today lots of things are written in Perl.

Perl is also very easy to learn like most of the other high level scripting languages.



15

## Perl's Weaknesses

- Expressive language
  - **T**here Is **M**ore **T**han **O**ne **W**ay **T**o **D**o It
- Popular
- Easy

Version 1.4.1 © Adam Trickett February 2015 Distributed under a creative commons Attribution-NonCommercial-ShareAlike licence.

Sadly because Perl is expressive there are lots of ways to solve a given problem, often most of the solutions are inefficient or insecure.

Too much choice can put people off, new users in particular can feel intimidated

Being popular can create a situation where lots of unqualified but keen people can dominate the blogsphere comment space, often repeating the same bad/wrong advice

Having a reputation for being easy can attract people who aren't prepared to work.

Perl has suffered from a lot of “Cargo Cult” programming that is bad and sometimes downright dangerous, and needs to be avoided.



# Getting Help



## Useful Resources

- Perl Training Australia:
  - <http://perltraining.com.au/notes.html>
- Ovid's CGI Course:
  - <http://ovid-cgi-course.perl-begin.org/cgi-course/>
- Learn.perl.org:
  - <http://learn.perl.org/>
- On-line Documentation:
  - <http://perldoc.perl.org/>
- Randal's Perl Columns:
  - <http://www.stonehenge.com/merlyn/columns.html>

There are lots of resources on the web. Some of them are more trustworthy than some of the others. The sample here are some from people who are respected in the community, work in the community and write books for the community.



18



## Some Books



<http://books.perl.org/>




Version 1.4.1 © Adam Trickett February 2015  
Distributed under a creative commons Attribution-NonCommercial-ShareAlike licence.

There use to be lots of very bad Perl books, they are less common now, but I've listed the canonical set here that are recommended by most reputable instructors and often.


On the perl.org web site there are some additional books that are also available free to download of the internet

19



## Comprehensive Perl Archive Network

- Over 25,000 modules
- A module for every imaginable task:
  - Web/CGI
  - Databases
  - SAP
  - MP3/Oggs
  - Time/Dates
  - Templating
  - and much much more!



Version 1.4.1 © Adam Trickett February 2015 Distributed under a creative commons Attribution-NonCommercial-ShareAlike licence.

One of Perl's great strength is that many coders give away bits of their programs, normally in the form of modules. They come in different flavours, simple procedural code, object oriented code and some with complicated bits done in other languages such as c.

Generally speaking you want to avoid re-inventing the wheel if you can and use an existing module.

Other languages may have archives, but Perl's is the definitive one that they are based on.



## CPAN

- Global system with mirrors worldwide
- PAUSE
  - Perl Authors Upload SErver
- RT
  - Request Tracker
- CPAN testers & CPANTS
- Search/Meta Search
  - Two Web Based Search engines

CPAN is more than just a collection of reusable code, it has a load server, is connected to bug & request tracking, has testing, quality assurance, dependency checking and search capability. On its own CPAN would be pretty dull, it's the rich greater-CPAN than gives it its strength.



## Killer Modules!

- CGI
- DBI
- Template
- CPAN
- Apache mod\_perl
- File::Find::Rule
- LWP/WWW::Mechanize
- MVC/ORM

Some modules are very powerful and used by lots of people and lots of other modules. These are some.

I've left CGI.pm on the list, even though it's deprecated and you shouldn't use it in new code, but it is very popular and still widely used.

In the Model-View-Controller space, there are several options, of which Catalyst is very powerful and commonly used.

If you are interested in the Internet of Things, then look at Dancer2 which is a very snazzy all in one web server, database and programming framework.



## Some Community Sites

- PerlMonks:
  - <http://www.perlmonks.org/>
- Perl Blogs:
  - <http://blogs.perl.org/>
- Perl Foundation:
  - <http://www.perlfoundation.org/>
- Perl Mongers:
  - <http://www.pm.org/>

Like all things it's best to find your community, as asking the right people will get you the right answers faster than thrashing about.

Some of these groups hold meetings, like the mongers and are a good opportunity to meet and greet.



## More Sites

- Perl 5 Wiki
  - <https://www.socialtext.net/perl5/>
- {Perl School}
  - <http://perlschool.co.uk/>
- Enlightened Perl Organisation
  - <http://www.enlightendperl.org/>



## Meetings

- OSCON - Open Source Conference
  - Originally Perl only, now everything open-source
  - Expensive
  - <http://conferences.oreillynet.com/>
- YAPC - Yet Another Perl Conference
  - Still just Perl
  - Cheaper
  - <http://www.yapceurope.org/>

There are big mega meetings, OSCON in the US is the most famous, it started life as a Perl meeting but now does everything else as well.

YAPC is the main Perl only meeting, there are large continental scale ones each year, plus smaller regional ones as well. A lot cheaper than OSCON, but still costs you something.



## Smaller Meetings

- PM Workshops
  - London.PM Workshop every winter
- PM TechMeets
  - London.PM and ThamesValley.PM
- Perl Birds of a Feather “BOF” meetings
  - LUGs
- Perl tracks/streams at other meetings
  - FOSDEM

The various PM groups run workshops and smaller meetings, London has had one in Winter for over a decade now, they are free and well worth attending if you can.

Most PM groups run social and technical meetings as well.

Perl also comes as BOF at other groups and track/streams at other meetings, e.g. FOSDEM





## More Links...

- <http://www.cpan.org/>
- <http://search.cpan.org/>
- <http://metacpan.org/>
- <http://rt.cpan.org/>
- <http://pause.perl.org/>
- <http://jobs.perl.org/>
- <http://cpants.perl.org/>
- <http://perldoc.perl.org/>
- <http://www.perl.com/>

Copies of the links I may have already mentioned.



# Some Screen Shots of Sites

Some visual stuff.



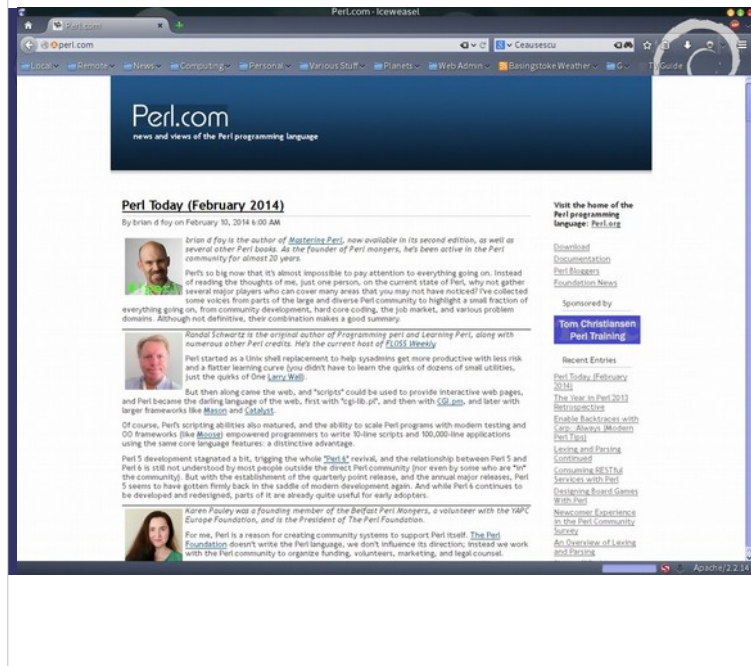
Perlmonks is the main social Q&A site. It's boisterous but there are good people there and if you are careful and ask your question you'll get some good answers.

DO NOT ask for homework solutions or show zero input or you'll get shot down.

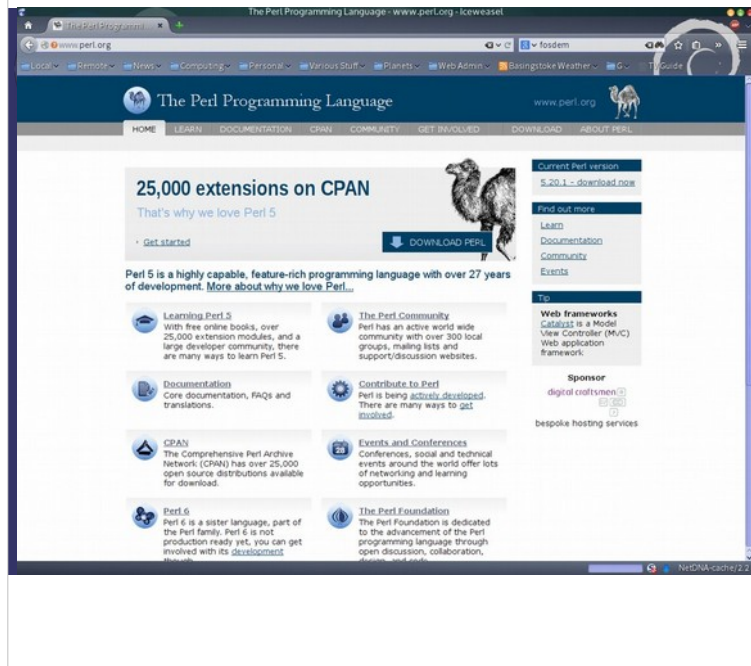
It's best to join and watch for a few days before asking a question and looking foolish.

There are no wrong questions, but many of them have already been answered, so it's worth using the search features first.

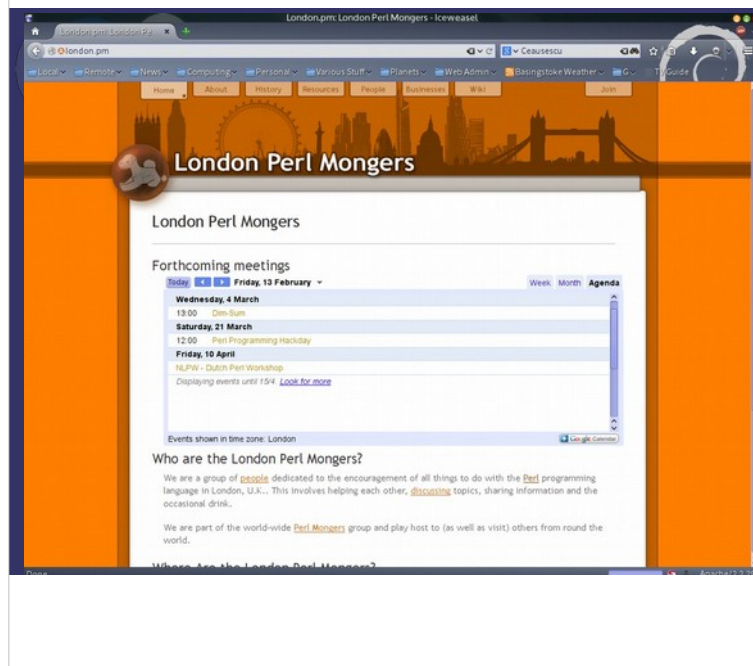
Site: perlmonks.org



Perl.com is a commercial site that promotes Perl. There is nothing wrong with it, but it's not the real home page.

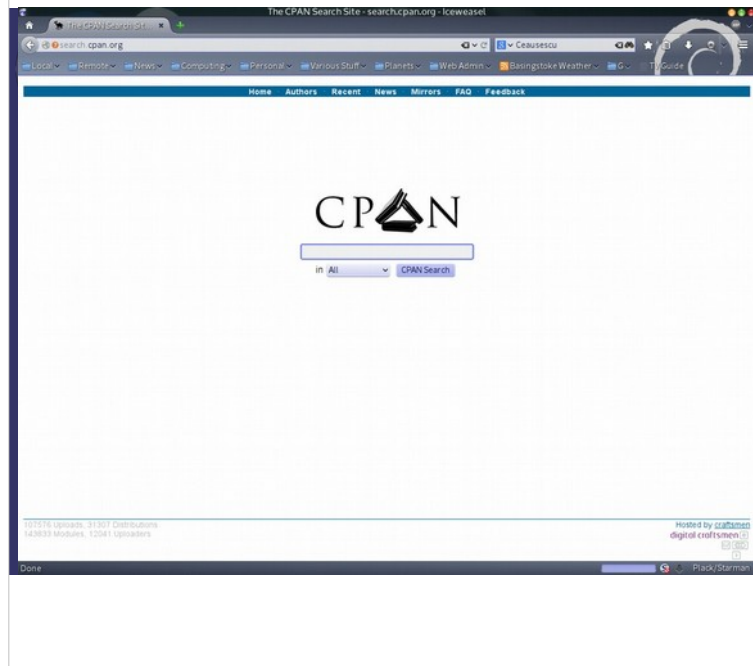


Perl.org is the real community home, it has more links and a lot of other services hang of the domain.



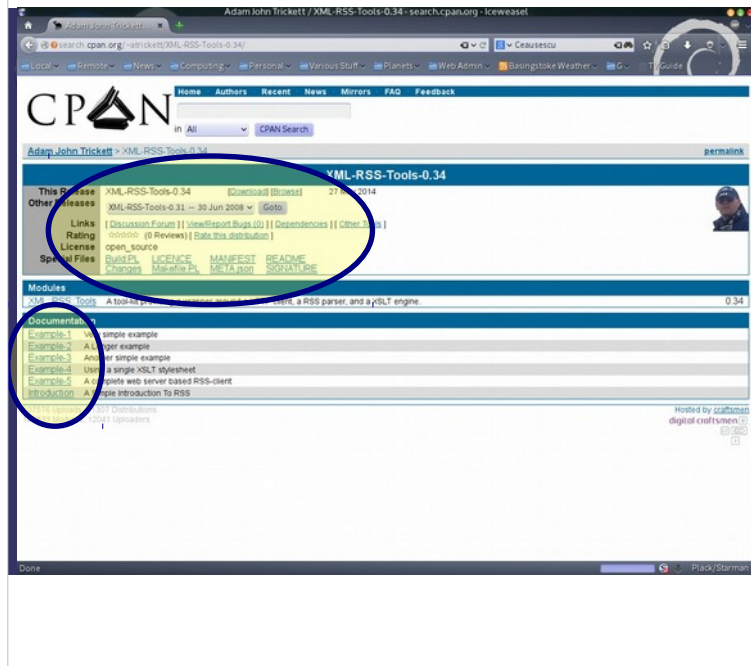
London.pm a near by group that is large and noisy, good people.

There is also the the Thames Valley Perl Mongers and the Southampton Perl Mongers



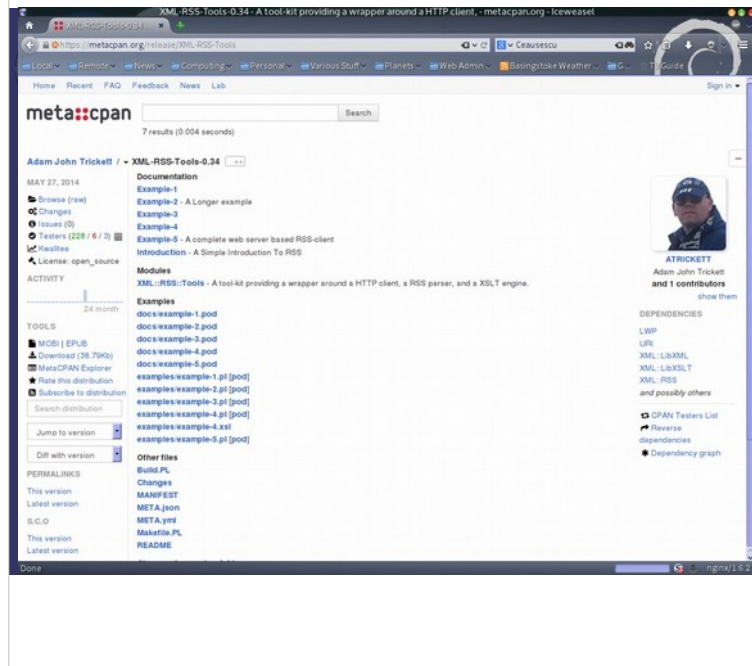
CPAN, Google style...

Site: search.cpan.org



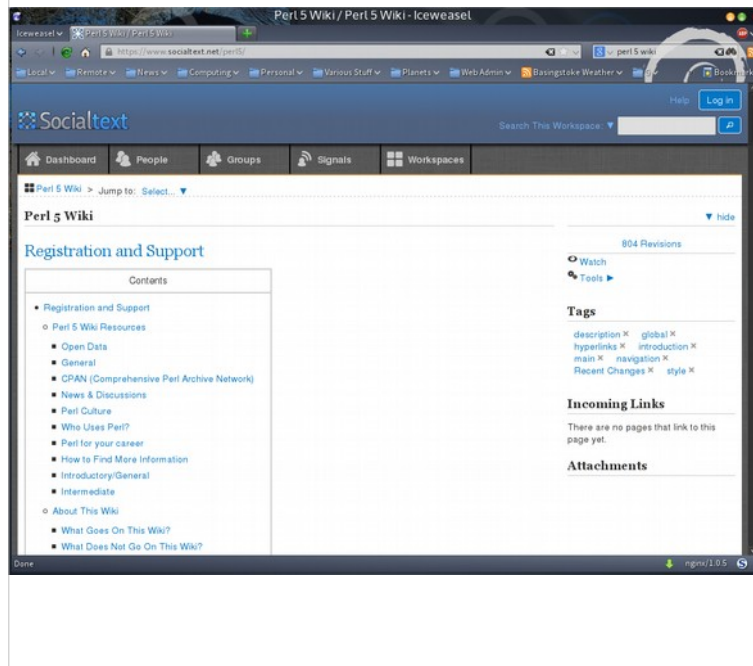
My stuff on CPAN





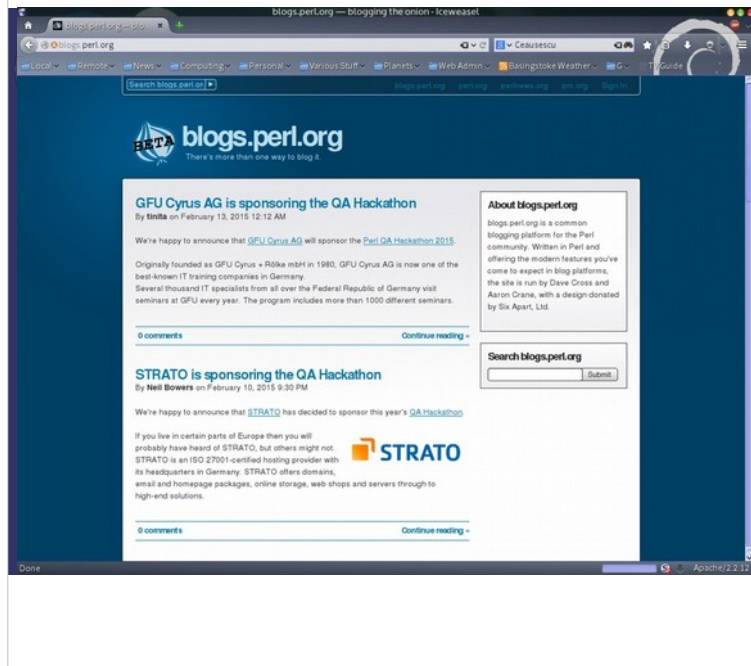
Meta::cpan, and alternate search engine.

Site: metacpan.org



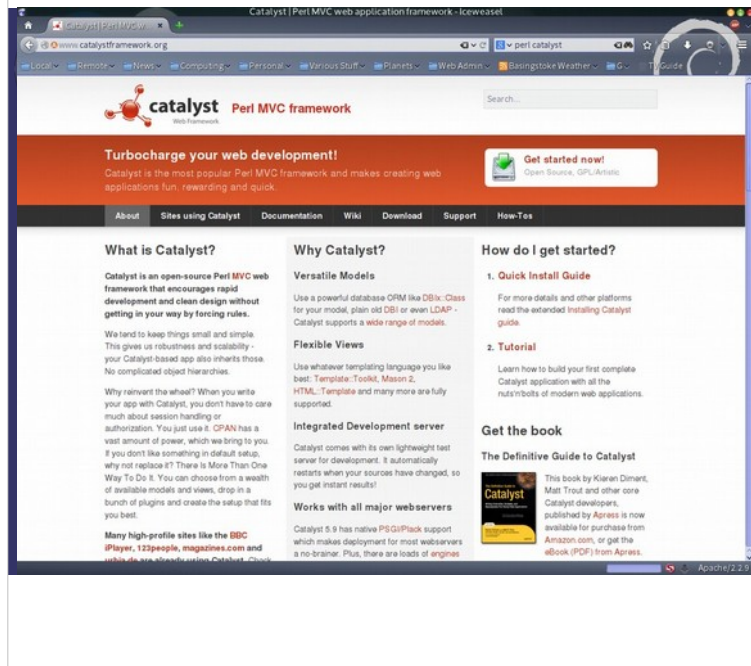
Perl5 wiki

Site: [www.socialtext.net/perl5](http://www.socialtext.net/perl5)



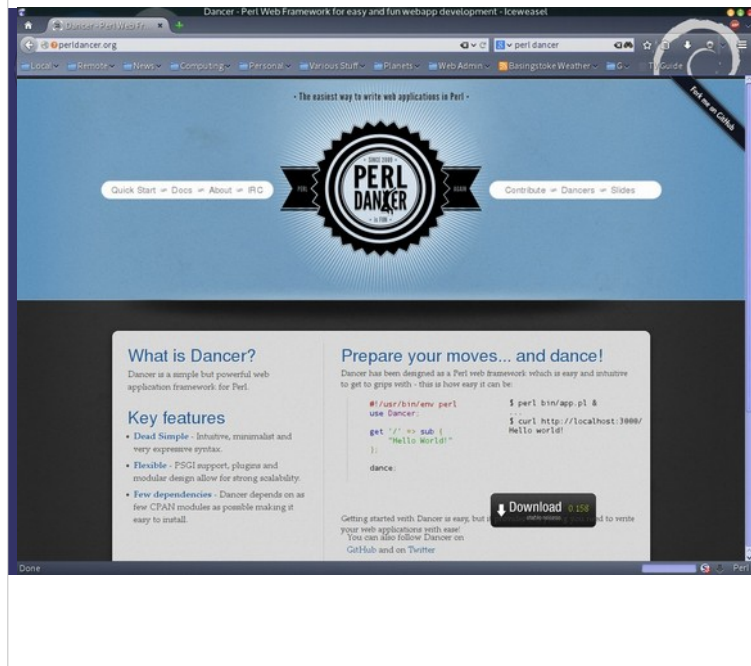
Blogs for perl

Sites: blogs.perl.org



Catalyst

Site catalystframework.org



Dancer

Site: perldancer.org



# Q&A