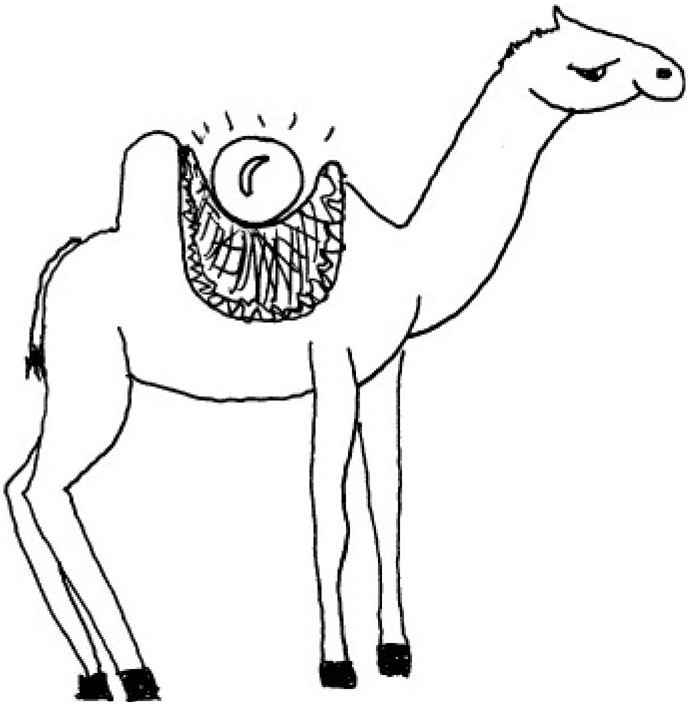
Chapter 1. The Practical Extraction and Report Language



1.1 What Is Perl?

> *"Laziness, impatience, and hubris. Great Perl programmers embrace those virtues."*
> —Larry Wall

Perl is an all-purpose, open-source (free software) interpreted language maintained and enhanced by a core development team called the Perl Porters. It is used primarily as a scripting language and runs on a number of platforms. Although initially designed for the UNIX operating system, Perl is renowned for its portability and now comes bundled with most operating systems, including RedHat Linux, Solaris, FreeBSD, Macintosh, and more. Due to its versatility, Perl is often referred to as the Swiss Army Knife of programming languages.

Larry Wall wrote the Perl language to manage log files and reports scattered over the network. According to *Wikipedia.org*, Perl was originally named "Pearl," but when Larry Wall realized that

PEARL was another programing language that had been around since 1977, he simply dropped the "a" and the name became "Perl." Perl was later dubbed the Practical Extraction and Report Language, and by some, it is referred to as the Pathologically Eclectic Rubbish Lister. Perl is really much more than a practical reporting language or eclectic rubbish lister, as you'll soon see. Perl makes programming easy, flexible, and fast. Those who use it, love it. And those who use it range from experienced programmers to novices with little computer background at all. The number of users continues to grow at a phenomenal rate.[1]

[1] Perl is spelled "Perl" when referring to the language, and "perl" when referring to the interpreter.

Perl's heritage is UNIX. Perl scripts are functionally similar to UNIX *awk*, *sed*, shell scripts, and *C* programs. Shell scripts consist primarily of UNIX commands; Perl scripts do not. Whereas *sed* and *awk* are used to edit and report on files, Perl does not require a file in order to function. Whereas *C* has none of the pattern matching and wildcard metacharacters of the shells, *sed,* and *awk*, Perl has an extended set of characters. Perl was originally written to manipulate text in files, extract data from files, and write reports, but through continued development, it can manipulate processes, perform networking tasks, process Web pages, talk to databases, and analyze scientific data. Perl is truly the Swiss Army Knife of programming languages; there is a tool for everyone.

The examples in this book were created on systems running Solaris, Linux, Macintosh, UNIX, and Win32.

Perl is often associated with a camel symbol, a trademark of O'Reilly Media, which published the first book on Perl called *Programming Perl* by Larry Wall and Randal Schwartz (also referred to as "the Camel Book").

1.2 What Is an Interpreted Language?

To write Perl programs, you need two things: a text editor and a perl interpreter, which you can download very quickly from any number of Web sites, including *perl.org*, *cpan.org*, and *activestate.com*. Unlike with compiled languages, such as *C++* and *Java*, you do not need to first compile your program into machine-readable code before executing it. The perl interpreter does it all; it handles the compilation, interpretation, and execution of your program. Advantages of using an interpreted language such as Perl is that it runs on almost every platform, is relatively easy to learn, and is very fast and flexible.

Languages such as Python, JavaScript, and Perl are interpreted languages that use an intermediate representation, which combines both compilation and interpretation. It compiles the user's code into an internal condensed format called bytecode, or threaded code, which is then executed by the interpreter. When you run Perl programs, you need to be aware of two phases: the compilation phase, and then the run phase where you will see the program results. If you have syntax errors, such as a misspelled keyword or a missing quote, the compiler will send an error. If you pass the compiler phase, you could have other problems when the program starts running. If you pass both of these phases, you will probably start working on formatting to make the output look nicer, improving the program to make it more efficient, and so forth.

The interpreter also provides a number of command-line switches (options) to control its behavior. There are switches to check syntax, send warnings, loop through files, execute statements, turn on the debugger, and so forth. You will learn about these options throughout the following chapters.

1.3 Who Uses Perl?

Because Perl has built-in functions to easily manipulate processes and files, and because Perl is portable (that is, it can run on a number of different platforms), it is especially popular with system administrators, who often oversee one or more systems of different types. The phenomenal growth of the World Wide Web greatly increased interest in Perl, which was the most popular language for writing CGI scripts to generate dynamic Web pages. Even today, with the advent of other languages focused on processing Web pages, such as Ruby, Node, and ASP.net, Perl continues its popularity with system and database administrators, scientists, geneticists, and anyone who has a need to collect data from files and manipulate it.

Anyone can use Perl, but it is easier to learn if you are already experienced in writing UNIX shell scripts or languages derived from *C*, such as *C++* and *Java*. For these people, the migration to Perl will be relatively easy. For those who have little programming experience, the learning curve might be a little steeper, but after learning Perl, there may be no reason to ever use anything else.

If you are familiar with UNIX utilities such as *awk*, *grep*, *sed*, and *tr*, you know that they don't share the same syntax; the options and arguments are handled differently, and the rules change from one utility to the other. If you are a shell programmer, you usually go through the grueling task of learning a variety of utilities, shell metacharacters, regular expression metacharacters, quotes, more quotes, and so forth. Also, shell programs are limited and slow. To perform more complex mathematical tasks and to handle interprocess communication and binary data, for example, you may have to turn to a higher-level language, such as *C*, *C++*, or *Java*. If you know *C*, you also know that searching for patterns in files and interfacing with the operating system to process files and execute commands are not always easy tasks.

Perl integrates the best features of shell programming, *C*, and the UNIX utilities *awk*, *grep*, *sed*, and *tr*. Because it is fast and not limited to chunks of data of a particular size, many system administrators and database administrators have switched from the traditional shell scripting to Perl. *C++* and *Java* programmers can enjoy the object-oriented features added in Perl 5, including the ability to create reusable, extensible modules. Now, with Perl you can generate Perl in other languages, and you can embed other languages in Perl. There is something for everyone who uses Perl, and for every task. As Larry Wall says, "There's more than one way to do it."[2]

[2] Larry Wall, "Diligence, Patience, and Humility," *http://www.oreilly.com/catalog/opensources/book/larry.html*.

You don't have to know everything about Perl to start writing scripts. You don't even have to be a programmer. This book will help you get a good jump-start, and you will quickly see some of its many capabilities and advantages. Then you can decide how far you want to go with Perl. If nothing else, Perl is fun!

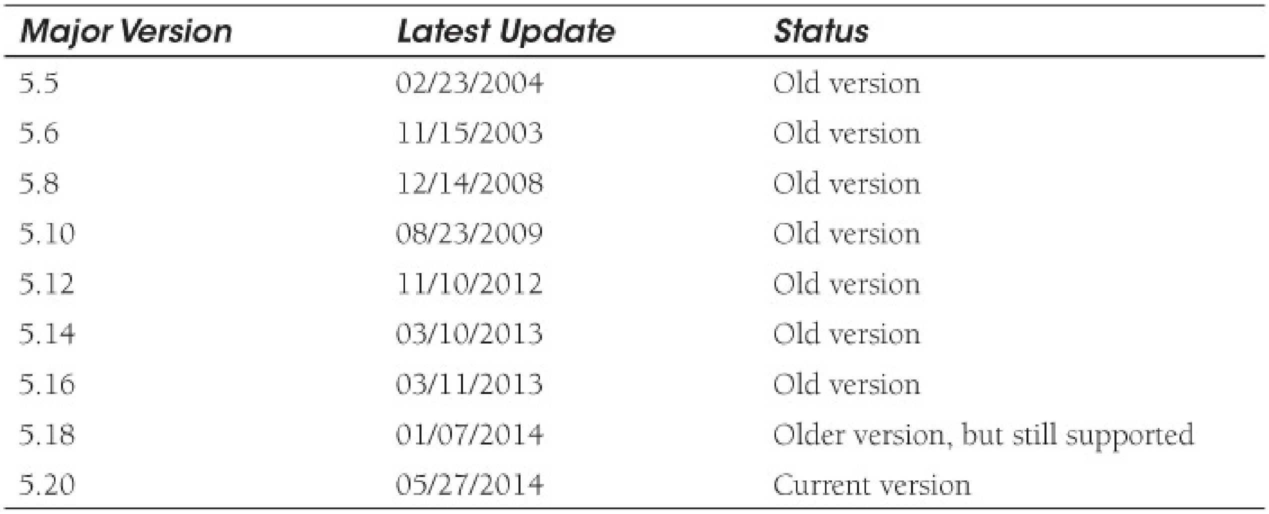
1.3.1 Which Perl?

Perl has been through a number of revisions. The last version of Perl 4 was Perl 4, patchlevel 36 (Perl 4.036), released in 1992, making it ancient. Perl 5.000 (also ancient), introduced in fall 1994, was a complete rewrite of the Perl source code that optimized the language and introduced objects and many other features. Despite these changes, Perl 5 remains highly compatible with the previous releases. As of this writing, the current stable version of Perl is 5.20, actively maintained by a large group of voluntary contributors listed at *www.ohloh.net/p/perl/contributors*. Perl 6 is the next generation of another Perl redesign and does not have an official release date. It has new features, but the basic components of the Perl language you learn here will be essentially the same.

From Wikipedia:

> Some observers credit the release of Perl 5.10 with the start of the Modern Perl movement. In particular, this phrase describes a style of development which embraces the use of the CPAN, takes advantage of recent developments in the language (see Table 1.1), and is rigorous about creating high-quality code.[3]

[3]. Wikipedia.org, "Perl," *http://en.wikipedia.org/wiki/Perl*.



Major Version	Latest Update	Status
5.5	02/23/2004	Old version
5.6	11/15/2003	Old version
5.8	12/14/2008	Old version
5.10	08/23/2009	Old version
5.12	11/10/2012	Old version
5.14	03/10/2013	Old version
5.16	03/11/2013	Old version
5.18	01/07/2014	Older version, but still supported
5.20	05/27/2014	Current version

Table 1.1 Release Dates and Recent Developments

1.3.2 What Are Perl 6, Rakudo Perl, and Parrot?

> *"Perl 5 was my rewrite of Perl. I want Perl 6 to be the community's rewrite of Perl and of the community."*
>
> —Larry Wall, State of the Onion speech, TPC4

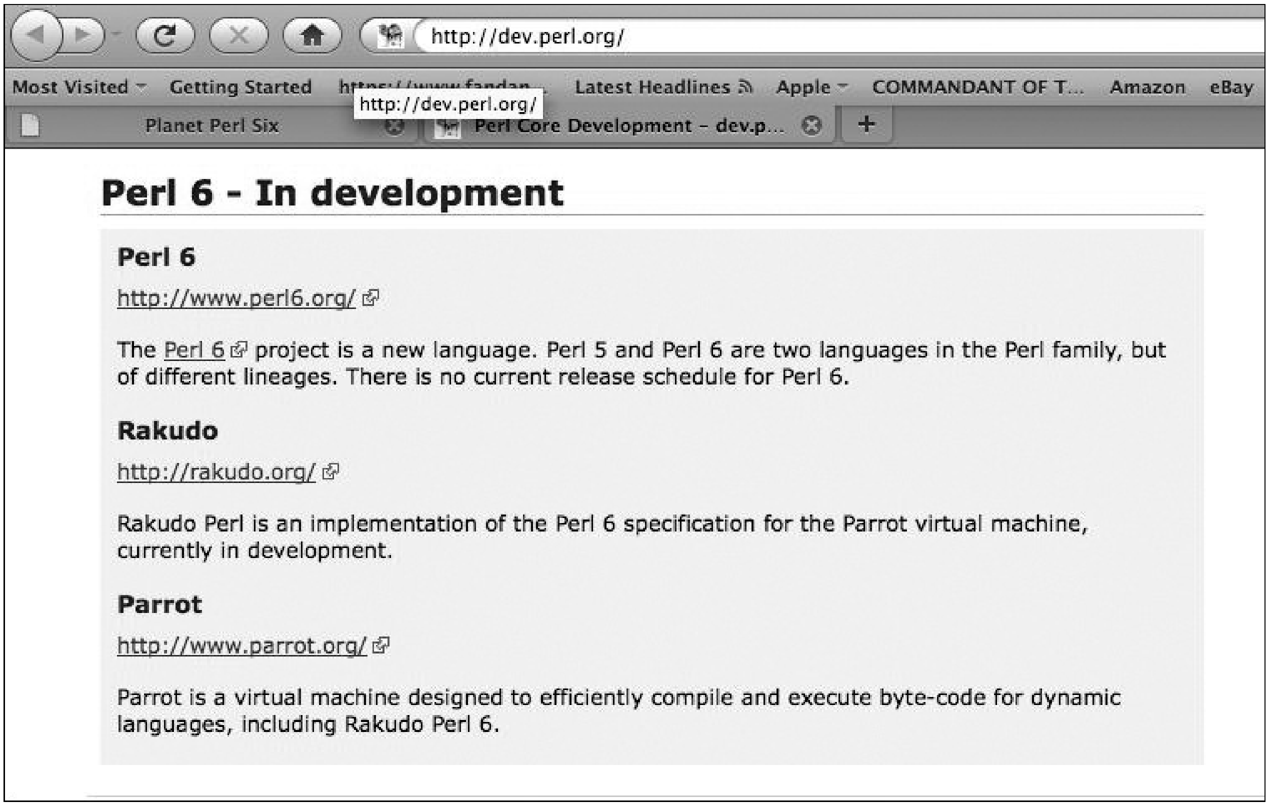
Perl 6 is essentially Perl 5 with many new features. Although they continue to develop in parallel, Perl 6 will not supersede Perl 5. The basic language syntax, features, and purpose will be the same. If you know Perl, you will still know Perl. If you learn Perl from this book, you will be prepared to jump into Perl 6 when it is released. Perl 6 has been described by Perl.org as learning Australian English if you speak American English, rather than trying to switch from English to Chinese.

Rakudo Star, a useful and usable distribution of Perl 6 that runs on the Parrot virtual machine, was recently released in October 2013. To find out more go to *http://rakudo.org*.

Parrot is a virtual machine designed to efficiently compile and execute bytecode for dynamic languages. Parrot currently hosts a variety of language implementations in various stages of completion, including Tcl, JavaScript, Ruby, Lua, Scheme, PHP, Python, Perl 6, APL, and a .NET bytecode translator.[4]

[4] Parrot Speaks Your Language, *http://parrot.org*.

To learn more about the latest Perl core development with Perl 6, Rakudo, and Parrot, go to *http://dev.perl.org* (see Figure 1.1).



Perl 6 - In development

Perl 6

http://www.perl6.org/

The Perl 6 project is a new language. Perl 5 and Perl 6 are two languages in the Perl family, but of different lineages. There is no current release schedule for Perl 6.

Rakudo

http://rakudo.org/

Rakudo Perl is an implementation of the Perl 6 specification for the Parrot virtual machine, currently in development.

Parrot

http://www.parrot.org/

Parrot is a virtual machine designed to efficiently compile and execute byte-code for dynamic languages, including Rakudo Perl 6.

Figure 1.1 The Perl 6 development Web site.

And for a biographical sketch of Larry Wall and the history of Perl, go to *http://www.softpanorama.org/People/Wall/index.shtml#Perl_history*.

1.4 Where to Get Perl

Perl downloading and instructions are available from a number of sources. You can check the following popular sites for a Perl distribution for your computer: *cpan.org*, *perl.org*, and *activestate.com*, and *strawberryperl.com*.

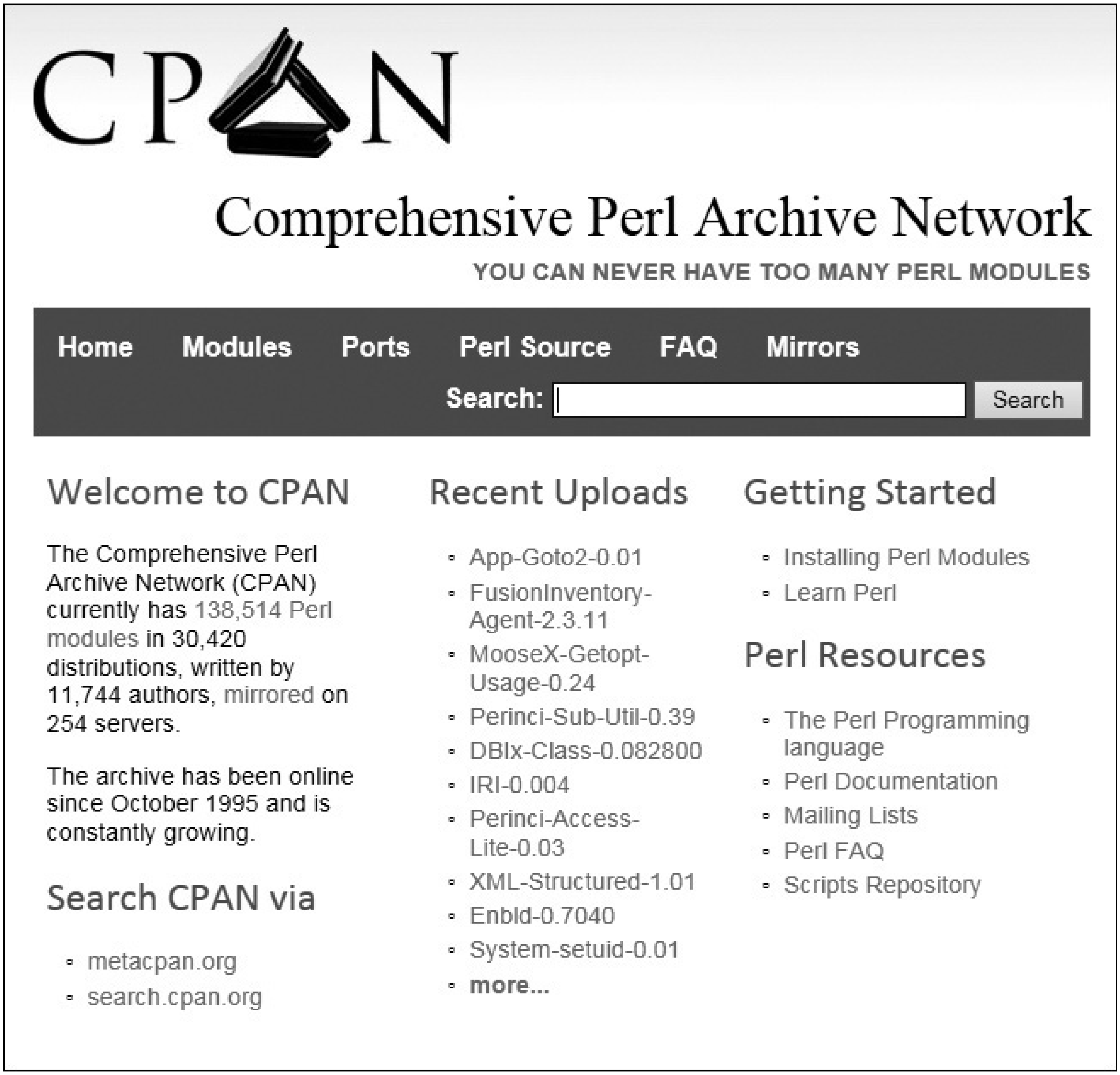
What Is Strawberry Perl?

Strawberry Perl is Perl for the Microsoft Windows platform. While most other distributions rely on the user having software development tools already set up to install certain Perl components, Strawberry Perl ships with the most commonly used tools preconfigured and packaged, including Perl binaries, the gcc compiler, all external libraries, all bundled database clients, and more. The latest version, as of this writing, is Strawberry Perl 5.20.1.1, for both 32- and 64-bit versions of Windows. (You can download it at *http://strawberryperl.com*). Larry Wall says, "When I'm on Windows, I use Strawberry Perl."

1.4.1 CPAN (cpan.org)

The primary source for Perl distribution is CPAN, which is available at *www.cpan.org* (see Figure 1.2). CPAN, the "gateway to all things Perl," stands for the Comprehensive Perl Archive Network, a Web site that houses all the free Perl material you will ever need, including documentation, FAQs, modules and scripts, binary distributions and source code, and announcements. CPAN is mirrored all over the world, and you can find the nearest mirror at

- *www.perl.com/CPAN*
- *www.cpan.org*


CPAN
Comprehensive Perl Archive Network
YOU CAN NEVER HAVE TOO MANY PERL MODULES
Home
Modules
Ports
Perl Source
FAQ
Mirrors
Search:
Search

Welcome to CPAN

The Comprehensive Perl Archive Network (CPAN) currently has 138,514 Perl modules in 30,420 distributions, written by 11,744 authors, mirrored on 254 servers.

The archive has been online since October 1995 and is constantly growing.

Search CPAN via

- metacpan.org
- search.cpan.org

Recent Uploads

- App-Goto2-0.01
- FusionInventory-Agent-2.3.11
- MooseX-Getopt-Usage-0.24
- Perinci-Sub-Util-0.39
- DBIx-Class-0.082800
- IRI-0.004
- Perinci-Access-Lite-0.03
- XML-Structured-1.01
- Enbld-0.7040
- System-setuid-0.01
- more...

Getting Started

- Installing Perl Modules
- Learn Perl

Perl Resources

- The Perl Programming language
- Perl Documentation
- Mailing Lists
- Perl FAQ
- Scripts Repository

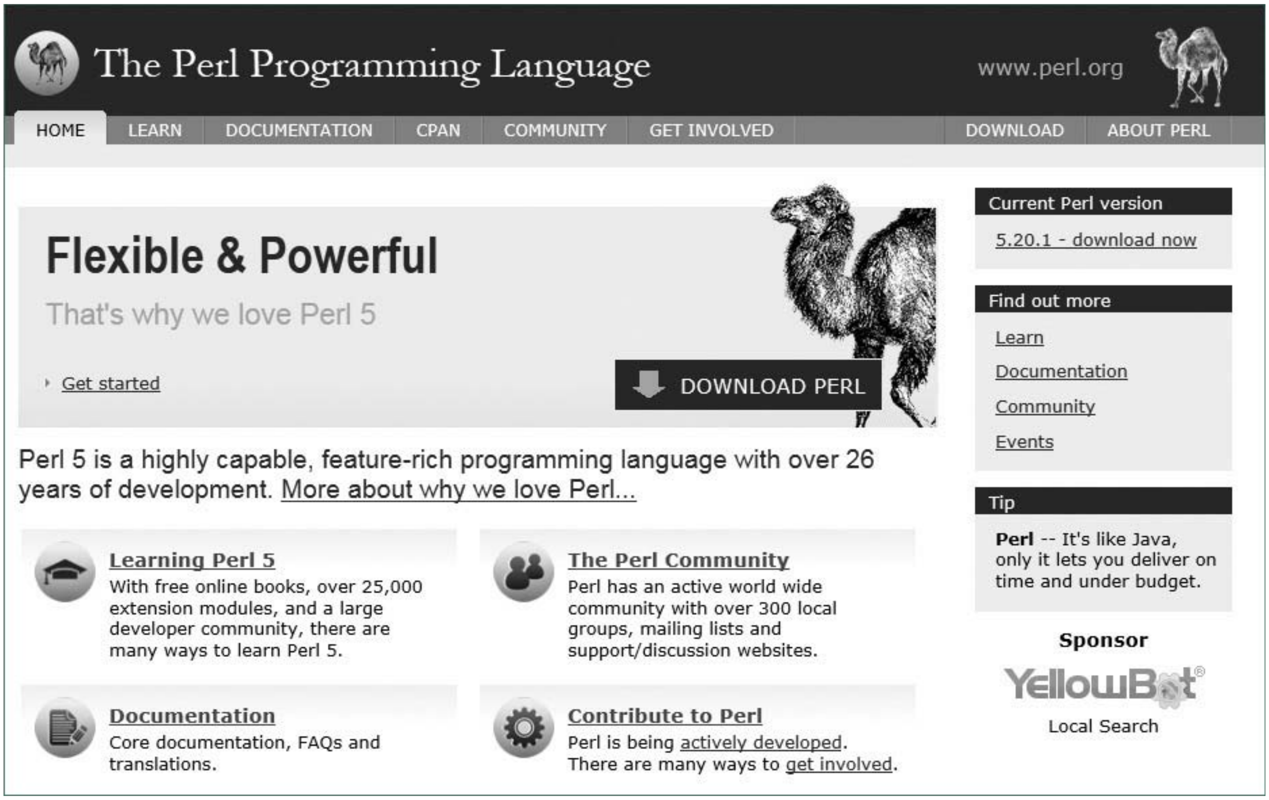
Figure 1.2 The CPAN Web site. Click on the Ports tab to find your platform.

CPAN is the place you will go to if you want to find modules to help you with your work. The CPAN search engine will let you find modules under a large number of categories. Modules are discussed in Chapter 13, "Modularize It, Package It, and Send It to the Library!"

Go to *www.cpan.org/ports* to find out more about what's available for your platform, of which Perl supports more than 100.

1.4.2 Downloads and Other Resources for Perl (perl.org)

The official Perl home page, run by O'Reilly Media, Inc. is *www.perl.com*, but it seems that everything you will need is found at *www.perl.org* (see Figure 1.3).



Flexible & Powerful

That's why we love Perl 5

Perl 5 is a highly capable, feature-rich programming language with over 26 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.

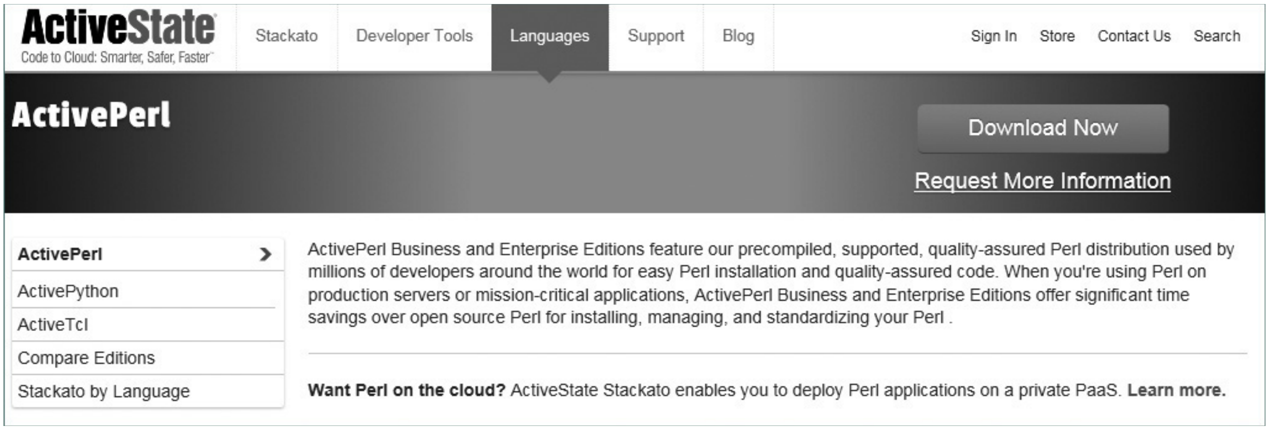
Contribute to Perl
Perl is being actively developed. There are many ways to get involved.

Perl -- It's like Java, only it lets you deliver on time and under budget.

Figure 1.3 The Perl.org Web site.

1.4.3 ActivePerl (activestate.com)

If you want to install Perl quickly and easily, ActivePerl is a complete, self-installing distribution of Perl based on the standard Perl sources for Windows, Mac OS X, Linux, Solaris, AIX, and HP-UX. It is distributed online at the ActiveState site (*www.activestate.com*). The complete ActivePerl package contains the binary of the core Perl distribution, complete online documentation, and all the essential tools for Perl development, including PPM, a handy perl package manager. This is available at *www.activestate.com/activeperl* (see Figure 1.4).



ActivePerl Business and Enterprise Editions feature our precompiled, supported, quality-assured Perl distribution used by millions of developers around the world for easy Perl installation and quality-assured code. When you're using Perl on production servers or mission-critical applications, ActivePerl Business and Enterprise Editions offer significant time savings over open source Perl for installing, managing, and standardizing your Perl .

Want Perl on the cloud? ActiveState Stackato enables you to deploy Perl applications on a private PaaS. **Learn more.**

Figure 1.4 The ActiveState Web site, where you can download ActivePerl.

1.4.4 What Version Do I Have?

To obtain your Perl version, date the binary version was built, patches, and some copyright information, type the line shown in Example 1.1 (the dollar sign is the shell prompt).

EXAMPLE 1.1

Click here to view code image

```
1 $ perl -v

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

Copyright 1987-2013, 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.
```

Explanation

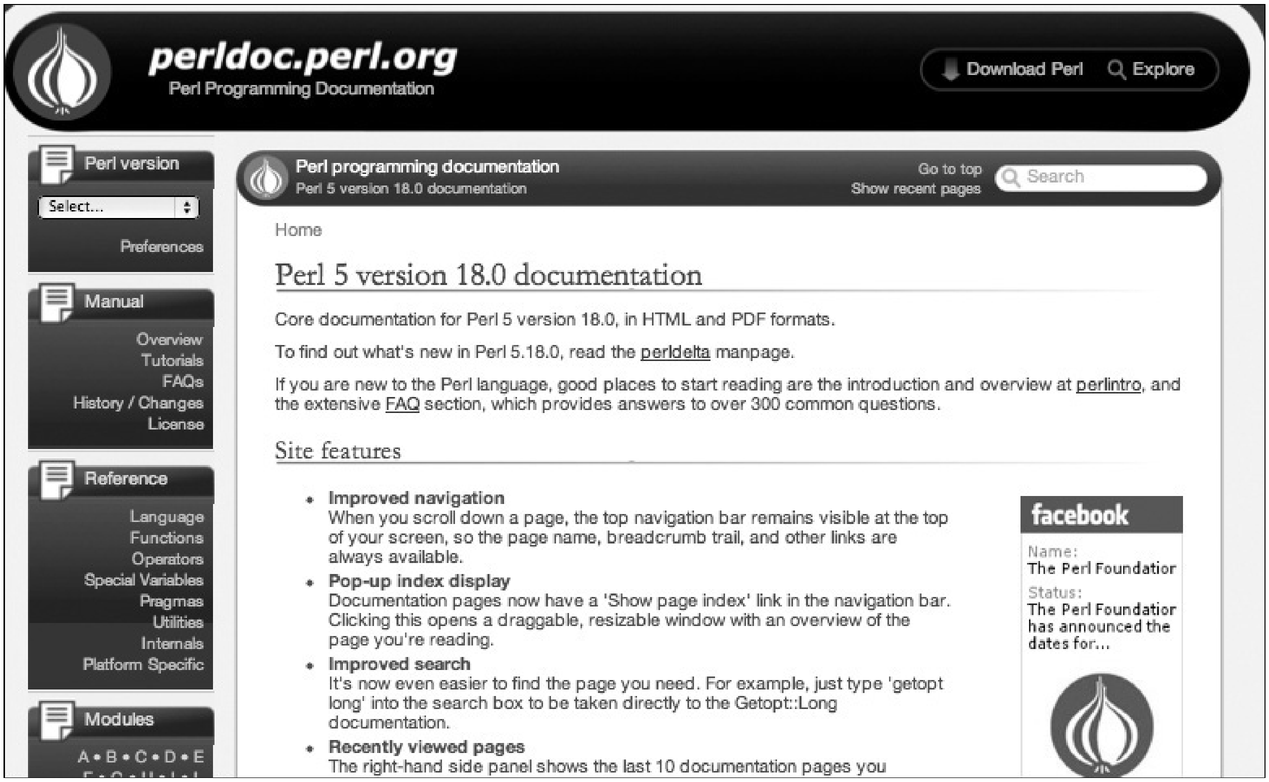
1. This version of Perl is 5.16. for the Mac.
2. Larry Wall, the author of Perl, owns the copyright.
3. Perl may be copied under the terms specified by the Artistic License or GNU. Perl is distributed under GNU, the Free Software Foundation, meaning that Perl is free.

1.5 Perl Documentation

Today, you can find answers to any Perl questions simply by using your favorite search engine or going to the Perl.org Web site. Most Perl distributions also come with full documentation in both HTML and PDF formats.

1.5.1 Where to Find the Most Complete Documentation from Perl

For the most complete documentation, type the Perl function you are looking for in your search engine or just go directly to *perldoc.perl.org* (see Figure 1.5) for all the complete documentation for any version of Perl.



Perl 5 version 18.0 documentation

Core documentation for Perl 5 version 18.0, in HTML and PDF formats.

To find out what's new in Perl 5.18.0, read the perldelta manpage.

If you are new to the Perl language, good places to start reading are the introduction and overview at perlintro, and the extensive FAQ section, which provides answers to over 300 common questions.

Site features

- **Improved navigation**
  When you scroll down a page, the top navigation bar remains visible at the top of your screen, so the page name, breadcrumb trail, and other links are always available.
- **Pop-up index display**
  Documentation pages now have a 'Show page index' link in the navigation bar. Clicking this opens a draggable, resizable window with an overview of the page you're reading.
- **Improved search**
  It's now even easier to find the page you need. For example, just type 'getopt long' into the search box to be taken directly to the Getopt::Long documentation.
- **Recently viewed pages**
  The right-hand side panel shows the last 10 documentation pages you

Name:
The Perl Foundatior
Status:
The Perl Foundatior has announced the dates for...

Figure 1.5 Documentation at perldoc.perl.org.

1.5.2 Perl *man* Pages

The standard Perl distribution comes with complete online documentation, called *man* pages, which provide help for all the standard utilities. (The name derives from the UNIX *man* [manual] pages.) Perl has divided its *man* pages into categories. If you type the following at your command-line prompt:

```
man perl
```

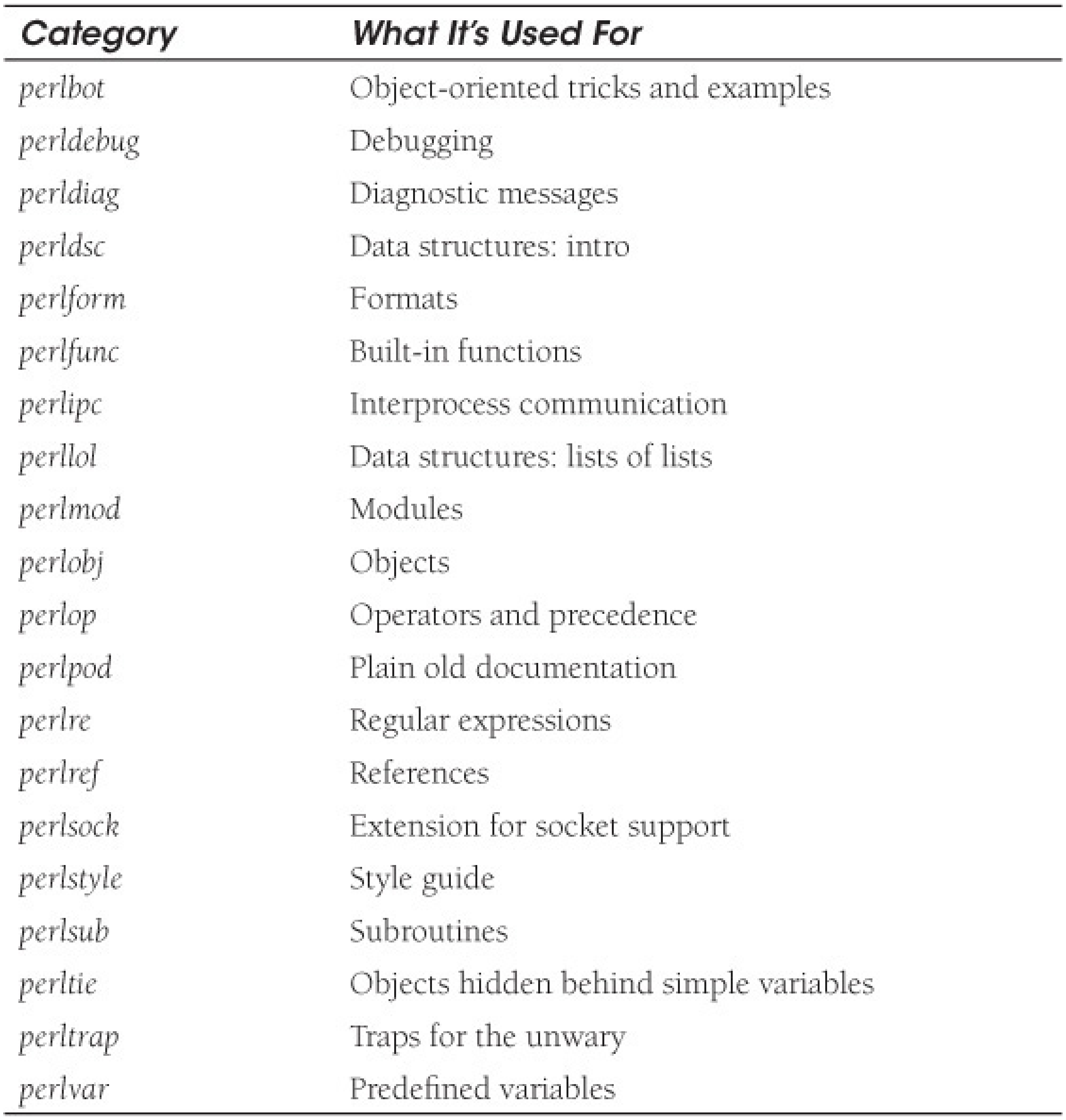
you will get a list of all the sections by category. So, if you want help on how to use Perl’s regular expressions, you would type:

```
man perlre
```

and if you want help on subroutines, you would type:

```
man perlsub
```

The Perl categories are listed in Table 1.2, with the following sections available only in the online reference manual.



Category	What It's Used For
perlbot	Object-oriented tricks and examples
perldebug	Debugging
perldiag	Diagnostic messages
perldsc	Data structures: intro
perlform	Formats
perlfunc	Built-in functions
perlipc	Interprocess communication
perllol	Data structures: lists of lists
perlmod	Modules
perlobj	Objects
perlop	Operators and precedence
perlpod	Plain old documentation
perlre	Regular expressions
perlref	References
perlsock	Extension for socket support
perlstyle	Style guide
perlsub	Subroutines
perltie	Objects hidden behind simple variables
perltrap	Traps for the unwary
perlvar	Predefined variables

Table 1.2 Perl Categories

If you are trying to find out how a particular library module works, you can use the *perldoc* command to get the documentation. (This command will give you documentation for the version of Perl you are currently using, whereas the *man* pages refer to the system Perl.) For example, if you want to know about the *Moose* module, type at the command line:

```
perldoc Moose
```

and the documentation for the *Moose.pm* module will be displayed. If you type:

```
perldoc Carp
```

the documentation for the *Carp.pm* module will be displayed.

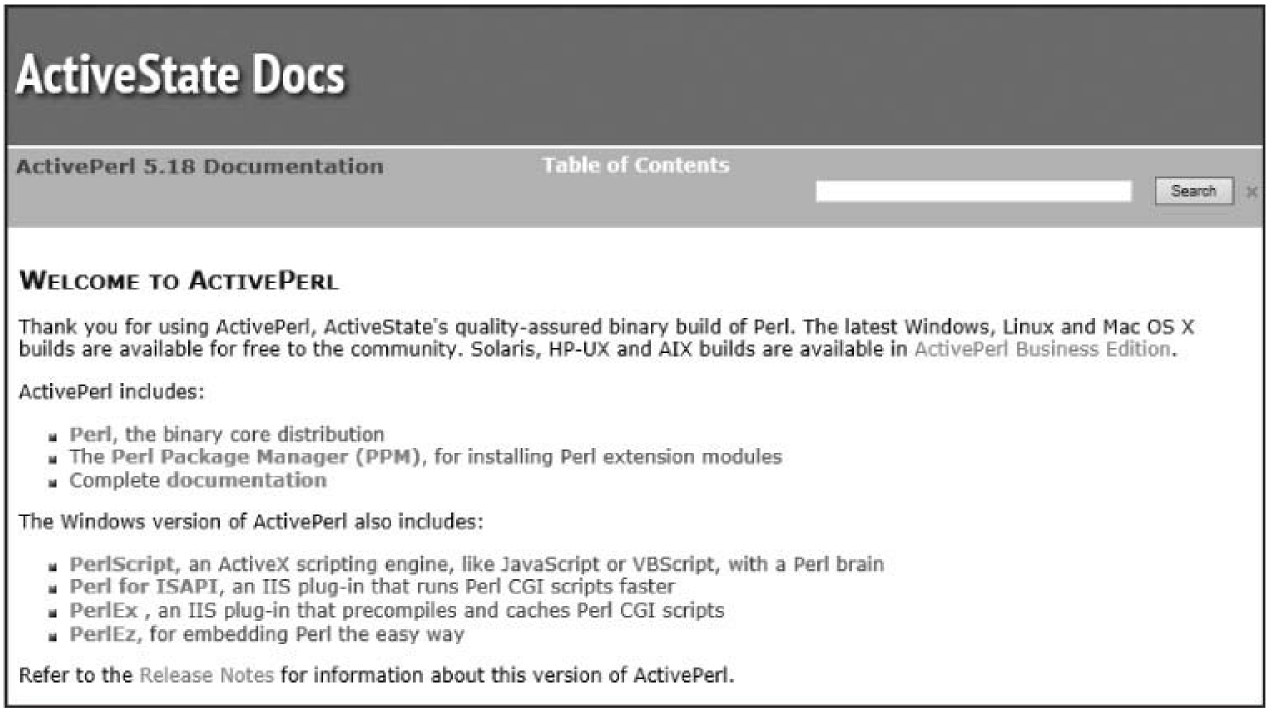
To get documentation on a specific Perl function, type *perldoc -f* and the name of the function. For example, to find out about the *localtime* function, you would execute the following command at your command-line prompt (you may have to set your UNIX/DOS path to execute this program directly):

Click here to view code image

```
perldoc -f localtime
localtime EXPR
localtime
        Converts a time as returned by the time function to a 9-element
        list with the time analyzed for the local time zone. Typically
        used as follows:
           #   0    1    2     3     4    5     6     7     8
            ($sec,$min,$hour,$mday,$mon,$year,$wday,$yday,$isdst) =
                                                       localtime(time);
<continues>
```

1.5.3 Online Documentation

ActivePerl provides excellent documentation (from *ActiveState.com*) when you download Perl from its site. As shown in Figure 1.6, there are links to everything you need to know about Perl.


ActiveState Docs

ActivePerl 5.18 Documentation

Table of Contents

WELCOME TO ACTIVEPERL

Thank you for using ActivePerl, ActiveState's quality-assured binary build of Perl. The latest Windows, Linux and Mac OS X builds are available for free to the community. Solaris, HP-UX and AIX builds are available in ActivePerl Business Edition.

ActivePerl includes:

- Perl, the binary core distribution
- The Perl Package Manager (PPM), for installing Perl extension modules
- Complete documentation

The Windows version of ActivePerl also includes:

- PerlScript, an ActiveX scripting engine, like JavaScript or VBScript, with a Perl brain
- Perl for ISAPI, an IIS plug-in that runs Perl CGI scripts faster
- PerlEx , an IIS plug-in that precompiles and caches Perl CGI scripts
- PerlEz, for embedding Perl the easy way

Refer to the Release Notes for information about this version of ActivePerl.

Figure 1.6 Perl documentation from ActiveState.

1.6 What You Should Know

1. Who wrote Perl?
2. What does Perl stand for?
3. What is the meaning of “open source”?
4. What is the current release?
5. What is Perl used for?
6. What is an interpreted language?
7. Where can you get Perl?
8. What is Strawberry Perl?
9. What is ActivePerl?

10. What is CPAN?
11. Where do you get documentation?
12. How would you find documentation for a specific Perl function?

1.7 What’s Next?

In the next chapter, you will learn how to create basic Perl scripts and execute them. You will learn what goes in a Perl script, and about Perl syntax, statements, and comments. You will learn how to check for syntax errors and how to execute Perl at the command line with a number of Perl options.