

perl

Submitted by: Kyle Ross
Prog 2200: Gap Analysis

Summary of Research

History

What is Perl?

Variables

Fig 1 - Declaring Variables in Perl

Data Structures

Subroutines

Object Oriented Programming

Fig 2 - Calling a Constructor in Perl

Database Connectivity

Learning Narratives

External References

Summary of Research

History

Perl was first Written by Larry Wall in 1987 to solve his problems at Systems Development Corporation.

Written to assist with text parsing, report creation and UNIX administration.

Perl has gone through many changes and branches of evolution, and has a massive hacker culture & fanbase

What is Perl?

Perl is a general purpose scripting language, it's High level, Dynamically typed

Interpreted, not Compiled, and generates no binary files

Perl borrows concepts from Shell Scripting (bash), awk, sed, and Regular Expressions (grep)

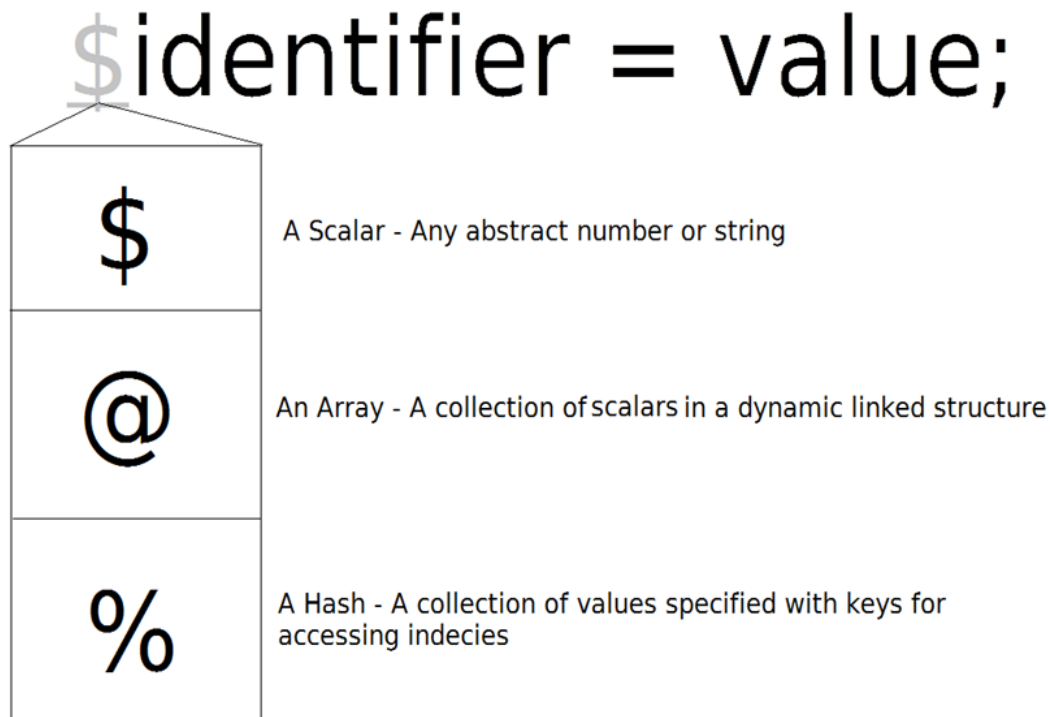
Variables

In Perl there are a large separation between Identifiers and Data Structures.

Variables are prefixed (sigil) to indicate their Data Structure Type

Fig 1 - Declaring Variables in Perl

Declaring variables in Perl



Data Structures

Scalar values are the most basic data structure. They represent any arbitrary number or String.

Arrays are like Linked Lists. They have a variety of methods that are meant to accept arrays for working with Array Elements.

Push (Add), Pop, Shift and Unshift (Push)

Hashes are Key-Value Associative Array structures

References are like pointers. They are a scalar variable that references a more complex type. They can be dereferenced into the original type by specifying the type Data Type Sigil and some curly braces. `@{$scalar_reference}` will create an array from a scalar reference for example.

Subroutines

When defining a subroutine in perl, you do not specify what can be passed into the function at runtime.

This is because, in Perl, when you call a function, it groups together your arguments into an array and gives that array context within the newly executing subroutine.

This is the first major use of Context in perl. The magic variable `_` is available at runtime to all perl programs depending on the current context.

This magic variable is usually accessed as an array, which represents a data structure of what arguments were passed into a subroutine.

Object Oriented Programming

What we would think of as a 'Class' is known as a Package in perl. All programs belong to package 'Main' and thus why the 'my' keyword must be used when declaring a variable in strict mode.

Any subroutine belonging to a Package may be designated the Constructor for the package. The constructor is not special from any other method, but the language understands that sub `Package()` is the same as `Package->sub`

Fig 2 - Calling a Constructor in Perl

- `Person->new()`
- `new Person()`
- `new(Person, ...)`

In either case the name of the Package is 'Unshifted' onto the array at call time and internally dealt with (via bless) within the Package constructor.

Packages must manage their own internal reference to a Data Structure, and return the appropriate symbol to the calling environment. Because data structures are public, public/private Object Oriented Encapsulation is tricky, although possible.

Database Connectivity

Perl supports connecting to all standard Data Providers through its DBI common interface. Perl contains all the usual constructs for a programming languages like executing or preparing queries, collecting a result into a variable, looping through the variable and printing the results.

Learning Narratives

I like learning new programming languages. I picked perl because I'd like to touch up my Unix skills and learn some more tricks of the trade when administering development environments. I think it's important to be able to set up a development environment in a Unix or Linux environment, and having the goto toolbox of Unix, perl, under my belt should hopefully make it easier.

External References

Arrays and Scalars

http://www.tutorialspoint.com/perl/perl_arrays.htm

Mysql DB access

<http://www.cyberciti.biz/faq/how-to-access-mysql-database-using-perl/>

Dereferencing arrays

<http://www.thegeekstuff.com/2010/06/perl-array-reference-examples/>

Special, context based variables in perl

<http://www.perl.com/pub/2004/06/18/variables.html>

Object Oriented Programming

http://www.tutorialspoint.com/perl/perl_oo_perl.htm