

CS306: Introduction to Perl

Introduction to the Class

U. of Alabama at Birmingham
Dept. of Computer & Information Sciences

Slide 1

Intro: What is Perl?

Introduction
History
Philosophy
Hello World

Slide 2

What is Perl?

- A multi-purpose, cross-platform interpreted scripting language that fits somewhere between shell scripting and languages like C or Java. *
- A freely available open-source project
- “Practical Extraction and Report Language” or “Pathologically Eclectic Rubbish Lister”

* This, like a lot of Perl things, isn't strictly true, but if I explained all of Perl's exceptions, this course would be Introduction to Perl's Exceptions instead.

Slide 3

What do you use Perl for?

- A question with a shorter answer might be “When can't you use Perl?”
- Perl is good at system administration tasks, text parsing, database access, networking, graphical programming, web scripting, quick prototyping
- Used by sysadmins, web scripters, bioinformaticists, physicists, mathematicians, etc...

Slide 4

Perl's Strengths

- Everything from quick and dirty scripts to massive programs with multiple programmers working for many months or years
- It has been said that Perl is optimized for “problems which are about 90% text and 10% everything else”
- Really a jack of all trades, and master of some

Slide 5

History of Perl

- Created by Larry Wall. Perl 1.0 in 1987.
- He started by gluing together UNIX utilities that he found useful
- Many people over the years have contributed to the growth of Perl through modules
- Current version is 5.8.6

Slide 6

Philosophy of Perl

- TIMTOWTDI – There Is More Than One Way To Do It
- Perl generally does not have One Right And True Way to do something. This is a double-edged sword.
 - It's very flexible, and...
 - ...you have to be careful not to abuse that fact.

Slide 7

Philosophy of Perl 2

- Perl will not stop you if you want to:
 - Write your entire program on one line
 - Use your integer like a string
 - Use a string like an integer
 - Assign a value to the 17th element of an array that only has 3 elements at the moment
- Perl will not hold your hand, but it's so easy to use, you won't need it anyway

Slide 8

Philosophy of Perl 3

- Perl loves shortcuts
 - Perl has shortcuts for **everything**. You'll learn to love this eventually, but it will be maddening at first. This is because Larry had a choice between...
- Easy to learn vs. easy to use
 - and he chose easy to use

Slide 9

Philosophy of Perl 4

- Perl will often guess (usually correctly) at what you were trying to do
 - For example, if Perl sees you try to assign an array to a scalar, it will guess that what you really wanted to know was the length of the array
- No hard limits
 - Perl tries not to impose arbitrary limits like variable name length or size of a string

Slide 10

Perl at CIS and at Home

- In CIS, Perl is available on the blazer machines. You will need a CIS UNIX account to use these.
- Perl, being free and open-source, is available for you to install at home or on your laptop
- Knoppix - linux with a safety net
- Feel free to develop on Windows, but test on Linux before sending it to me. (Just develop on Linux, it's easier)

Slide 11

How to get Perl

- For UNIX-like OS (Linux, OS X, etc...) - You already have Perl. Popular UNIX-like OS flavors have been shipping with Perl for many years.
- For Windows, ActivePerl – www.activestate.com/activeperl
- Can also build from source – get the source from the CPAN (CPAN.org) or perl.com

Slide 12

What is the CPAN?

- Comprehensive Perl Archive Network at cpan.org
- One-stop shopping for Perl source code, documentation, and extensions
- Most of the class will focus on Perl's core functionality, but towards the end of the semester we will explore CPAN a bit more

Slide 13

A first program - Hello World

```
#!/usr/bin/perl  
# A first Perl program  
print "Hello World!\n";
```

Slide 14

A first program - Hello World

<code>#!/usr/bin/perl</code>	sh-bang or <i>shebang</i> line
<code># A first Perl program</code>	A comment
<code>print "Hello World!\n";</code>	A typical Perl statement

Slide 15

Run The Program

- `# chmod a+x helloworld`
- `# ./helloworld`
Hello World!
#

Slide 16

Run The Program

- # chmod a+x helloworld Make the program executable
- # ./helloworld Run the program
Hello World!
#
- Alternate way to run the program...
 - # perl nameofprogram.pl (in this usage, the shebang line is ignored)

Slide 17

Lifecycle of a Perl Program

- When we typed ./helloworld a lot happened
 - The shell read the shebang line and went out and loaded the perl binary (if not loaded already)
 - The perl binary in turn parsed through the text of the program and used its internal compiler to produce *bytecode*
 - The bytecode was executed, producing the output on the terminal, and the program exited

Slide 18

Welcome to Perl

- Use the discussion board liberally. I check it multiple times daily, and encourage students to discuss perl amongst themselves

Slide 19