

CS306: Introduction to Perl

Introduction to the Class

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

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.

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...

Perl's Strengths

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

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 5.8.8

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.

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, or vice versa
 - Assign a value to the 17th element of an array that only has 3 elements at the moment
- Perl will not hold your hand

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

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 (a variable that holds only one piece of data), 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

A first program - Hello World

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

A first program - Hello World

```
#!/usr/bin/perl -w
```

sh-bang or *shebang* line

```
# A first Perl program
```

A comment

```
print "Hello World!\n";
```

A typical Perl statement

Run The Program

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

Run The Program

- `# chmod a+x helloworld` Make the program executable
- `# ./helloworld`
Hello World!
Run the program
- Alternate way to run the program...
 - `# perl nameofprogram.pl` (in this usage, the shebang line is not used to locate the perl interpreter)