

PERL

Bryan Watts
Kevin Wells

Jimmy Theis
Peter Klein

The History of Perl

- Developed in 1987 by Larry Wall
- Influenced by AWK, smalltalk 80, lisp, C/C++, sed, sh, pascal
- Practical Extraction Report Language
- “Swiss Army Chainsaw”
- “Duct Tape of the Internet”



The coolest man on the internet

Perl Overview

- ◉ Multi-Paradigm (Functional, Imperative, Object Oriented)
- ◉ Dynamically Typed

First Steps

- print "Hello World\n";
- print 1+1;
- \$ans = 7 * 6;
- print “Answer is \$ans\n”
- @large = grep \$_ > 10, (1,2,4,8,16,25);
- @more = map \$_ + 3, (1,2,4,8,16,25);

Variables

- ⦿ \$_ -> the result of the previous operation
- ⦿ @_ -> the array of arguments
- ⦿ %ENV -> hash of environment variables

- ⦿ \$ -> scalar
- ⦿ @ -> array/ list
- ⦿ % -> hash table
- ⦿ \ -> reference (POINTERS!)

Arrays

- @names = (“Bryan”, “Kevin”, “Jimmy”,
“Peter”);

```
foreach $person (@names) {  
    print "Hello $person\n"  
}
```
- @a = (1,2..5);
\$b = @a;
@c = (@a, (), (1,2,3), (\$a[\$#a-1]..\$b));

Hashes

- ⦿ %birthdays = {"Bryan", "11-30-89", "Kevin", "10-12-68"};
- ⦿ %ages = {Bryan => 20, Kevin => 42};
- ⦿ @list = %birthdays;
- ⦿ \$ages{Bryan}++;

Scoping

- ➊

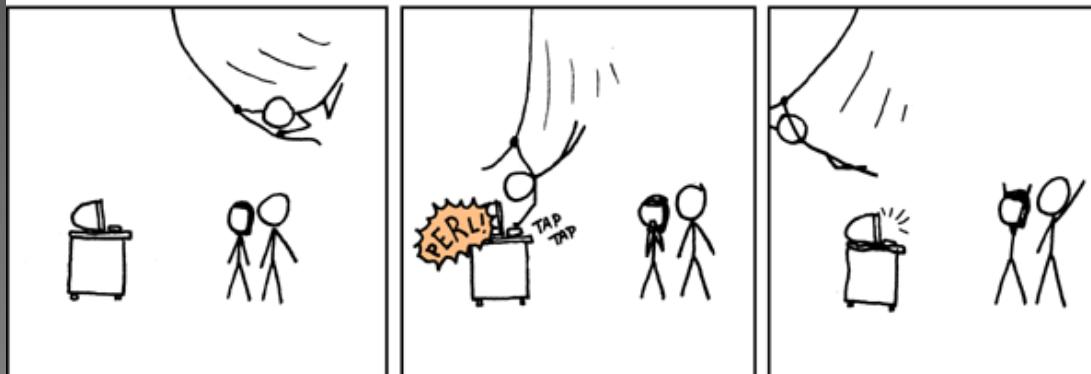
```
for($i = 0; $i < 10; $i++) {  
    print $i;  
}  
print $i; # last value of $i
```
- ➋

```
for(my $j = 0; $j < 10; $j++) {  
    print $j;  
}  
print $j; # ERROR (my is out of scope)
```

Subroutines

- ➊ sub add {
 my(\$a, \$b) = @_;
 \$a + \$b;
}
- ➋ print add(1, 2);

Cartoon of the day



Regex

- Regular Expressions
- Perl is reference implementation
- Powerful string manipulation across whole documents

Control Structures

- \$a != 0 ? \$a : \$b = 0
- return \$a if \$a != 0
- while(\$i < \$size) { ... }
- foreach \$member (@list) { ... }
- if (\$i >= 0) { ... } elsif (\$i == -1) { ... } else { ... }
- for (\$i = 0; \$i < @array; \$i++) { ... }

Perl CGI

- Perl module CGI `qw(:standard)`
 - › Helper functions: `h1(...)`, `start_form()`, `text_field(...)`, `hr(...)`, `submit(...)`
- CGI variables in environment (`%ENV`)
- `stdout` returned to client
- `stderr` to webserver log

PERL IS FUN

Perl Pros and Cons

- Pros

- › Concise
- › Dynamic Typing
- › Open Standard
- › Automatic type coercion
- › Variable substitution in string
- › Regex as syntax

- Cons

- › Butt ugly
- › Unsafe parameters
- › Horribly ugly
- › No R.E.P.L.
- › Weak debugger

Our Project

BlackPerl Movie Organizer

99 bottles of Perl