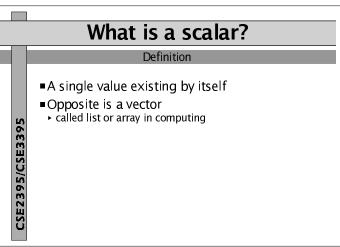


2

To be covered today ■Scalar values ▶ numbers ▶ strings ■Scalar variables ■Scalar operators ■Console input/output printing to the screen reading from the keyboard ■ Interpolating into strings

CSE2395/CSE3395



3 4

Scalars in Perl Perl considers these to be scalars numbers ▶ double-precision floating-point (C double) ► e.g., 87, -1.06E20, 0 ■ strings CSE2395/CSE3395 ▶ basic data type not an array of characters ▶ of arbitrary length ▶ no pointers needed ► e.g., "frotz", 'nitfol', "Hello world!\n" ■ references ► covered in Topic 11 Llama2 pages 31-35 5

Scalar variables How to store a scalar value Scalar values are stored in scalar variables ■Variables are global by default ■ All Perl scalar variables begin with \$ CSE2395/CSE3395 character ▶ \$apples ▶ \$text2 ▶ \$_ ■Perl doesn't care if a scalar variable contains a number or a string • numbers and strings are converted back and forth as ▶ \$a = 87 and \$a = "87" are almost identical in Perl 6

String literals

A fixed string in your Perl script

- ■Single-quote delimited
- ▶ all characters within string are literal, except
- \' (becomes ')
- \\ (becomes \)
- does not interpolate variables
- 'hello' (hello)
- '\$35.40' (\$35.40)
- 'it\'s' (it's)
- '\n' (backslash followed by n)



CSE2395/CSE3395

CSE2395/CSE3395

CSE2395/CSE3395

Llama2 pages 33-35; Camel3 pages 60-64; Camel2 pages 39-44

String literals

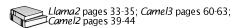
A fixed string in your Perl scrip

- Double-quote delimited
- ▶ usual C backslash-rules (e.g., \n) apply
- ► "hello" (hello)

CSE2395/CSE3395

CSE2395/CSE3395

- "it's \\$35.40" (it's \$35.40)
- "\n" (newline character)
- interpolates scalar and array variables
- "hello \$name" (becomes hello Tiger if \$name contains Tiger)
- Other string literal representations exist
 - to be introduced as they are met

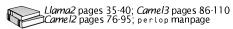


8

Scalar operators

Things to do with numbers

- Most familiar numeric operators available
- +, -, ++, %, etc.
- = (assignment)
- / (floating-point division)
- ► <, <=, >, >=, ==, != (numeric comparison)
- ** (exponentiation)
- New string operators
 - (join two strings)
 - "cat" . "fish" (produces string "catfish")
- ► lt, le, gt, ge, eq, ne (string comparison)



String operators The need for gt as well as >

- Perl converts between numbers and strings as needed
- ■Numerically, is 7 greater than 30?
- ▶ Use > for numeric comparisons
- ► Values on both sides of > are converted to numbers
- 7 > 30 is false (zero)
- Alphabetically, is "7" greater than "30"?

 "7" comes after "3" in ASCII
- ▶ Use gt for string comparisons
- Values on both sides of gt are converted to strings
- ▶ 7 gt 30 is true (non-zero)
- Need to use the correct comparison operator

10

String interpolation

9

Substituting a variable's value into a string

- ■In C, use printf or sprintf function to insert variable's value into a string
 - ▶ printf ("The sum is %d\n", total);
- In Perl, place variable name inside doublequoted string
- ▶ print "The sum is \$total\n";
- ■Can also use string concatenation operator

11

▶ print "The sum is " . \$total . "\n";

Llama2 pages 44-45; Camel3 pages 62-63; Camel2 pages 40-41

String interpolation

Pitfalls and notes

- ■Single-quoted strings do not interpolate
- print 'the sum is \$total\n' prints "the sum is
 \$total\n"
- If variable name is ambiguous, use braces
 - ▶ want to print "Today is the 6th" when \$day is 6
 - ▶ print "Today is the \$dayth\n"; - wrong, inserts value of \$dayth (possibly undefined)
 - ▶ print "Today is the \${day}th\n";
- right, uses value of \$day
- Braces can be used this way in any situation used for accessing complex nested data structures

12

CSE2395/CSE3395



Communicating with Per

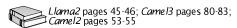
- Output to screen with
 - ▶ print function
 - ▶ printf function
 - ▶ print "testing\n";
- ■Input from keyboard with
- ► <STDIN> operator

CSE2395/CSE3395

CSE2395/CSE3395

CSE2395/CSE3395

- reads from standard input up to (and including) the first newline character
- \$ \$line = <STDIN>;
- ▶ chomp function can be used to remove newline



13

Example

Calculating the circumference of a circle

```
#!/usr/bin/perl -w
    # Set $pi.
    $pi = 3.1415926535898;
CSE2395/CSE3395
    # Read a number from the user.
    print "Please enter radius: ";
    $radius = <STDIN>;
    # Remove the newline from $radius.
    chomp $radius;
    # Calculate the circumference.
    $around = $radius * 2 * $pi;
    # Print the result.
    print "The circumference is $around\n";
```

14

Example

Calculating the maximum of two numbers

#!/usr/bin/perl -w # Read two numbers.
print "Enter a number: "; \$num1 = <STDIN>; chomp \$num1; print "Enter another number: \$num2 = <STDIN>; chomp \$num2; # Calculate the bigger value.
if (\$num1 > \$num2) { \$max = \$num1; } else { \$max = \$num2; } # Print result
print "Maximum is \$max\n";

undef

The undefined value

- Scalar variables used without being defined contain undef
- ■Often used as NULL is in C.
- undef returned by some functions and operators on out-of-range input
 - ► <std in> operator on end of file
 - ▶ use defined function to test an expression
- undef is converted to
 - ▶ empty string ("") in string context
 - ▶ zero in numeric context



CSE2395/CSE3395

Llama2 page 46; Camel3 pages 818-819; Camel2 page 235

16

15

Some scalar functions

Useful tools for working on scalars

- chomp \$string
- removes a newline from the end of \$string
- ■length \$string
- ▶ returns length of \$string in characters
- ■uc|lc|ucfirst|lcfirst \$string
 - returns a version of \$string entirely in uppercase/lowercase (or with just first letter changed)
- ■rand \$number
 - ▶ returns pseudorandom number from 0 to \$number

Camel3 chapter 29, pages 677-830; *Camel2* chapter 3, ⇒pages 141-242; perlfunc manpage

Covered today

- ■Scalar values
- numbers
- strings
- ■Scalar variables
- ▶ always start with \$ character
- ■Scalar operators
- ■Console input/output
- printing to the screen
- reading from the keyboard
- Interpolating into strings

CSE2395/CSE3395

17 18

Going further

More things related to today's topic

- References
- the "other kind" of scalar value
 Topic 11
- ■Unicode
- support for national character sets
 Camel3 pages 401-410
- ■Operator overloading
- providing new behaviour for builtin operators
 Camel2 pages 463-469; Camel3 pages 347-362

Next time To be covered in Topic 3 ■ Lists ■ Arrays variables that contain lists CSE2395/CSE3395 ■ List and array functions sorting lists
 adding and removing array elements ■Context scalar versus list **Reading:** Learning Perl 2nd edition chapter 3 Programming Perl 3rd edition pages 69-76 Programming Perl 2nd edition pages 45-49

19

20

CSE2395/CSE3395 lecture notes copyright © 2000-2001 Deborah Pickett.
Reproduction of this presentation for nonprofit study use is permitted. All other reproduction, including for other educational courses, must be authorized in writing by the author. writing by the author.

CSE2395/CSE3395