Interpolation

Variable Interpolation,
Backslash Interpolation

Interpolation

- Sometimes called “substitution”
  - – In Perl, “Substitution” means something else
- Interpolation = replacing symbol/variable with its meaning within a string
- Two kinds of interpolation – variable and backslash
- Done *only* in double-quoted strings, not single-quoted strings.

Backslash interpolation

- aka: character interpolation, character escapes, escape sequences.
- When any of these sequences are found inside a double – quoted string, they’re interpolated
- Listed on page 61 of Prog. Perl.
- Most common: “\n”, “\t”

Translation Escapes

- pg 61, table 2-2 of Prog.Perl
- \u – next character is uppercase
- \l – next character is lowercase
- \U – all characters until \E are uppercase
- \L – all characters until \E are lowercase
- \Q – all characters until \E are backslashed
- \E – end \U, \L, or \Q

Variable Interpolation

- variables found within “ ” are interpolated.
- ‘ ’ strings are NOT searched for interpolation
- $foo = “hello”;
- $bar = “$foo world”
  - $bar gets value: “hello world”
- $bar2 = ‘$foo world’
  - $bar2 gets value: ‘$foo world’

What can interpolate?

- Scalars, arrays, slices of arrays, slices of hash
  - NOT entire hashes
- Arrays (and slices) will print out each member of array separated by a space:
  - ex: `@array = (1, 3, 5, 7)`
  - `print “The first 4 odd numbers are @array\n”;`
  - outputs: The first 4 odd numbers are 1 3 5 7
- Change separation sequence via `$”` variable